Integration Objects'
OPC DA Access for Modbus Devices

OPC Server for Modbus
Version 2.0 Rev.2

USER GUIDE

OPC Compatibility
OPC Data Access 3.00
OPC Data Access 2.05a

OPC Server for Modbus User's Guide Version 2.0 Rev .2
July 2021

Table of Contents

Table of Figures

TABLE OF TABLES

PREFACE

About this User Guide

This guide:

- Describes the main features of Integration Objects' OPC Server for Modbus.
- Lists the system requirements for installing and running the OPC Server for Modbus.
- Explains how to use and run this OPC server.
- Describes how to connect to the server using an OPC DA client.

Target Audience

This document is intended for users that are looking for applications providing standard OPC DA (Data Access) connectivity to Modbus devices. Knowledge of the basics of OPC DA specification is assumed. It is also expected that you have some prior knowledge of the Modbus protocol.

Document Conventions

Convention	Description
Bold	Click/selection action required
	Information to be noted
Blue bold italics	Reference to other sections, or to other Integration Objects user guides

Customer Support Services

Phone	Email
Americas: +1 713 609 9208 **Europe-Africa-Middle East** +216 71 195 360	Support: customerservice@integrationobjects.com Sales: sales@integrationobjects.com Online: www.integrationobjects.com

INTRODUCTION

1. Overview

Integration Objects' OPC Server for Modbus is an OPC Server software designed to provide an OPC DA standard interface to Modbus compliant devices such as PLC, RTU and DCS. It establishes connection with one or more Modbus devices via TCP/IP or Serial protocol in order to collect data in real-time.

2. System Architecture

This OPC Server reads and updates data from/to devices such as RTU, PLC, and DCS. It can be accessed locally or remotely via DCOM by any OPC DA compliant client.

The following figure illustrates the client/server architecture and demonstrates the interaction between the OPC DA clients, the OPC Server for Modbus and the various Modbus-communicating devices.

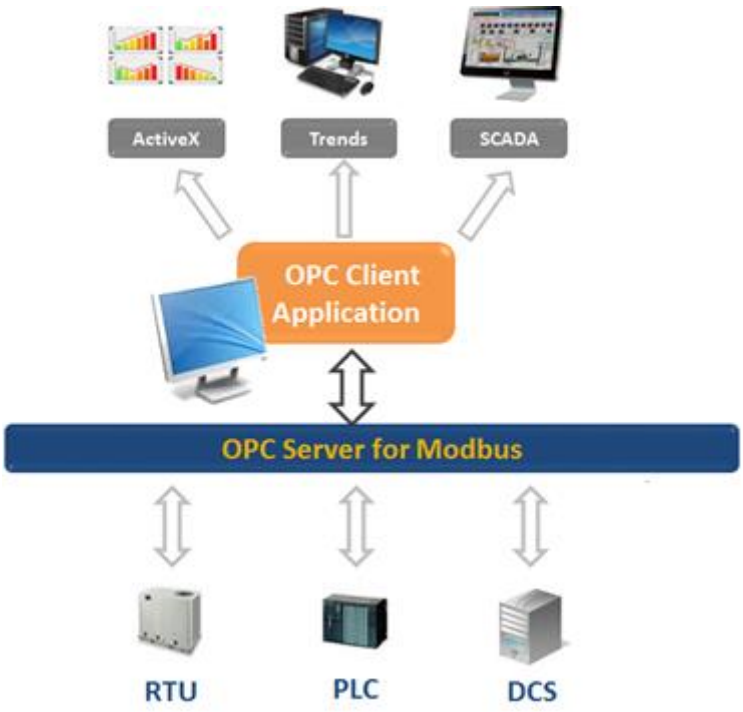


Figure 1: System Architecture

This OPC server can collect data from Modbus devices via the Modbus TCP or Serial protocol and expose those data to OPC DA Clients.

3. Features

This section details the features offered by this OPC server including OPC DA server configuration.

- Intuitive user interface for Modbus tags configuration
  The OPC Server for Modbus offers an intuitive graphical user interface to configure the tags. This configuration will be saved in a CSV file and will be loaded at the next OPC server startup. After startup, the server loads the configuration file to create tags and build its address space. The server will then map the Modbus addresses with the OPC tags.
- Tags configuration per blocks
- Support of multiple OPC DA client connections
- Support of multiple Modbus TCP devices connections
- Support of multiple Modbus Serial devices connections
- Automatic reconnection to the configured Modbus TCP devices after network glitches
- Windows service capability allowing automatic restart after the restart of the host machine
- Traceability of events using log files and viewer
- Comprehensive OPC data access capabilities: This OPC server allows OPC DA Clients to retrieve in real-time the preconfigured Modbus slave devices addresses values.
  The following are the current supported OPC DA interfaces:

Object	Interface	Supported
OPC DA Server	IUnknown	Yes
	IOPCCommon	Yes
	IOPCServer	Yes
	IConnectionPointContainer	Yes
	IOPCBrowseServerAddressSpace (Optional)	Yes
	IOPCItemProperties	Yes
OPC DA Group	IUnknown	Yes
	IOPCItemMgt	Yes
	IOPCGroupStateMgt	Yes
	IOPCPublicGroupStateMgt (Optional)	Yes
	IOPCSyncIO	Yes
	IOPCAsyncIO	Yes

Table 1: Supported OPC DA Interfaces

4. OPC Compatibility

Integration Objects' OPC Server for Modbus implements OPC Data Access specification version 2.05 and 3.0.

5. Operating Systems Compatibility

This application was successfully installed and executed under the following operating systems:

- Windows Seven
- Windows Server 2008
- Windows 8
- Windows Server 2012
- Windows 10
- Windows Server 2016
- Windows Server 2019

6. System Requirements

The following are the minimum requirements to run the OPC Server for Modbus:

- Processor: 1 GHz (higher recommended)
- RAM: 512 MB (higher recommended)
- Disk Space: 100 MB hard disk space for full installation Required OPC DLLs (described in more details in the next chapter)
- An OPC DA client compliant with OPC DA 2.05 standard
- Modbus Slave Simulators or Devices
- .NET Framework version 4.5 or higher
  Click here if you need to download a client application or here for free OPC test clients download.

GETTING STARTED

1. Pre-Installation Considerations

In order to properly run the OPC Server for Modbus, the following software components need to be installed on the target system:

- The OPC core components 3.00 which consists of all shared OPC modules including the DCOM proxy/stub libraries, the OPC Server Enumerator, .NET wrappers, etc. You can deploy the OPC core components during the installation by checking the option "Install OPC Core Components" or after installation by using the setup available in the installation folder of the OPC Server for Modbus.
- .NET framework version 4.5 or higher.

Make sure there is no firewall or antivirus blocking the application.

2. Installing OPC Server

To install the OPC Server for Modbus:

1. **Double-click** on the Integration Objects' OPC Server for Modbus installation package.

The installation welcome dialog box will appear.

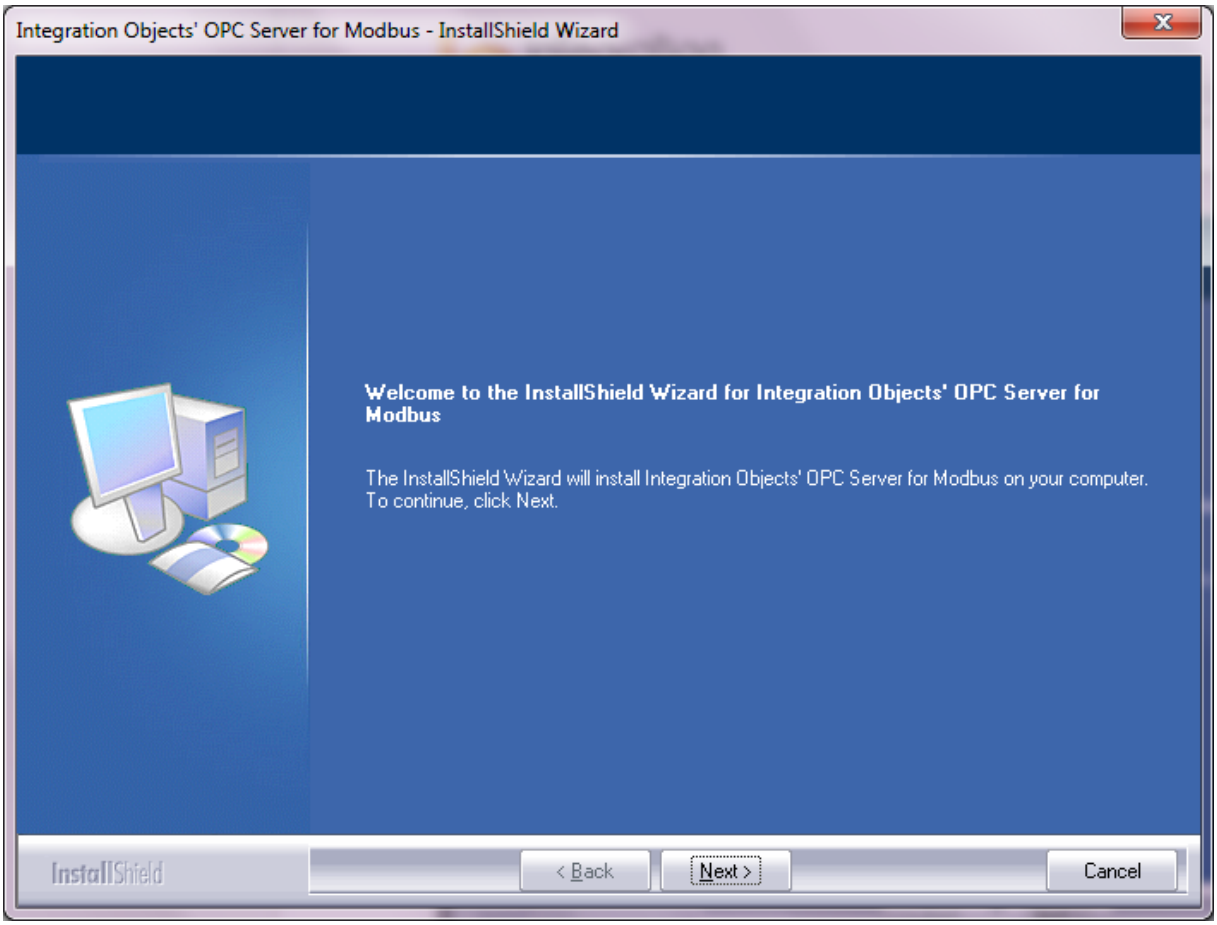


Figure 2: Installation Welcome Dialog Box

2. **Click** the Next button. The license agreement will be displayed

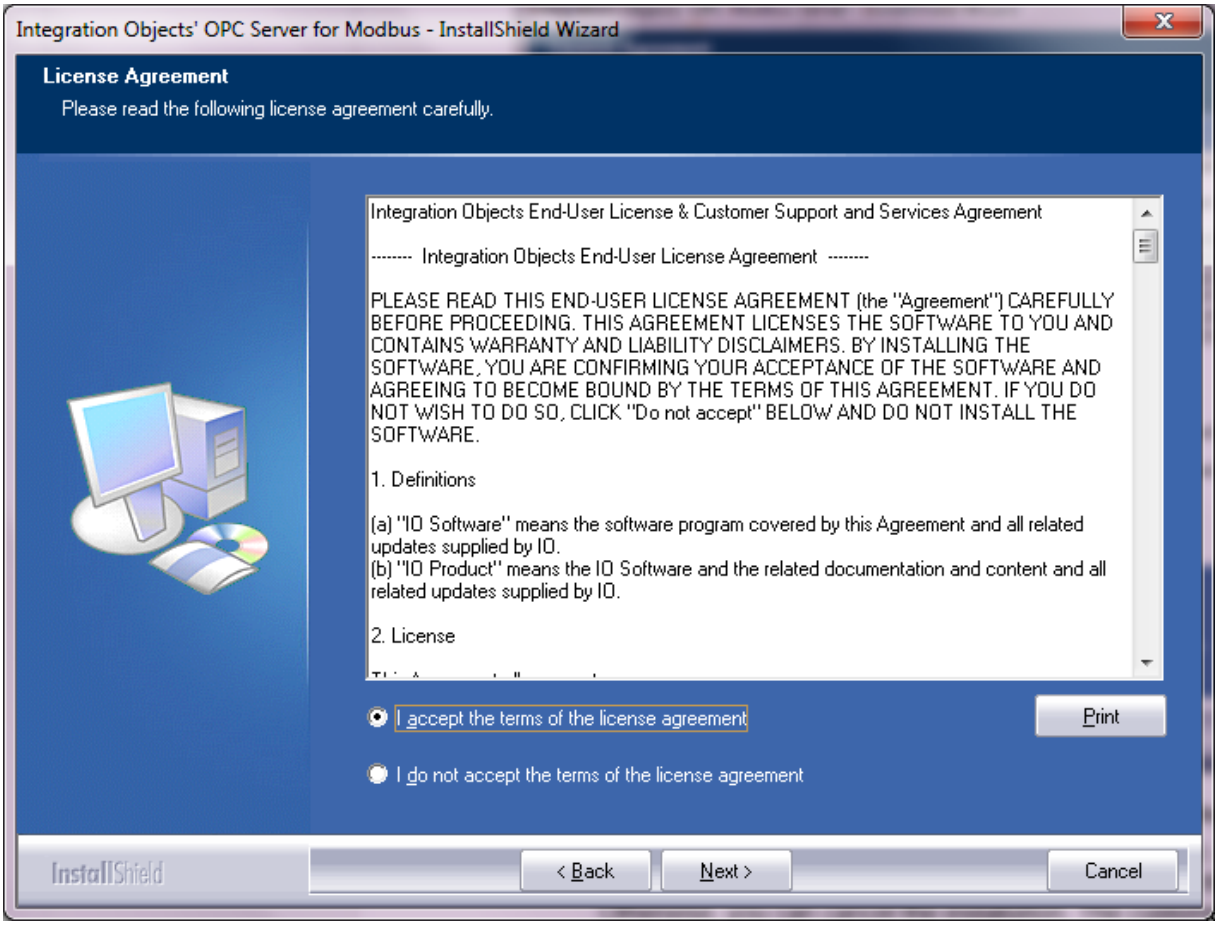


Figure 3: License Agreement Dialog Box

3. After reading the license agreement, **select** the first option and **click** the Next button. By proceeding, you are accepting all of the license agreement terms. Otherwise, you can cancel the installation. The customer information dialog box will then appear.

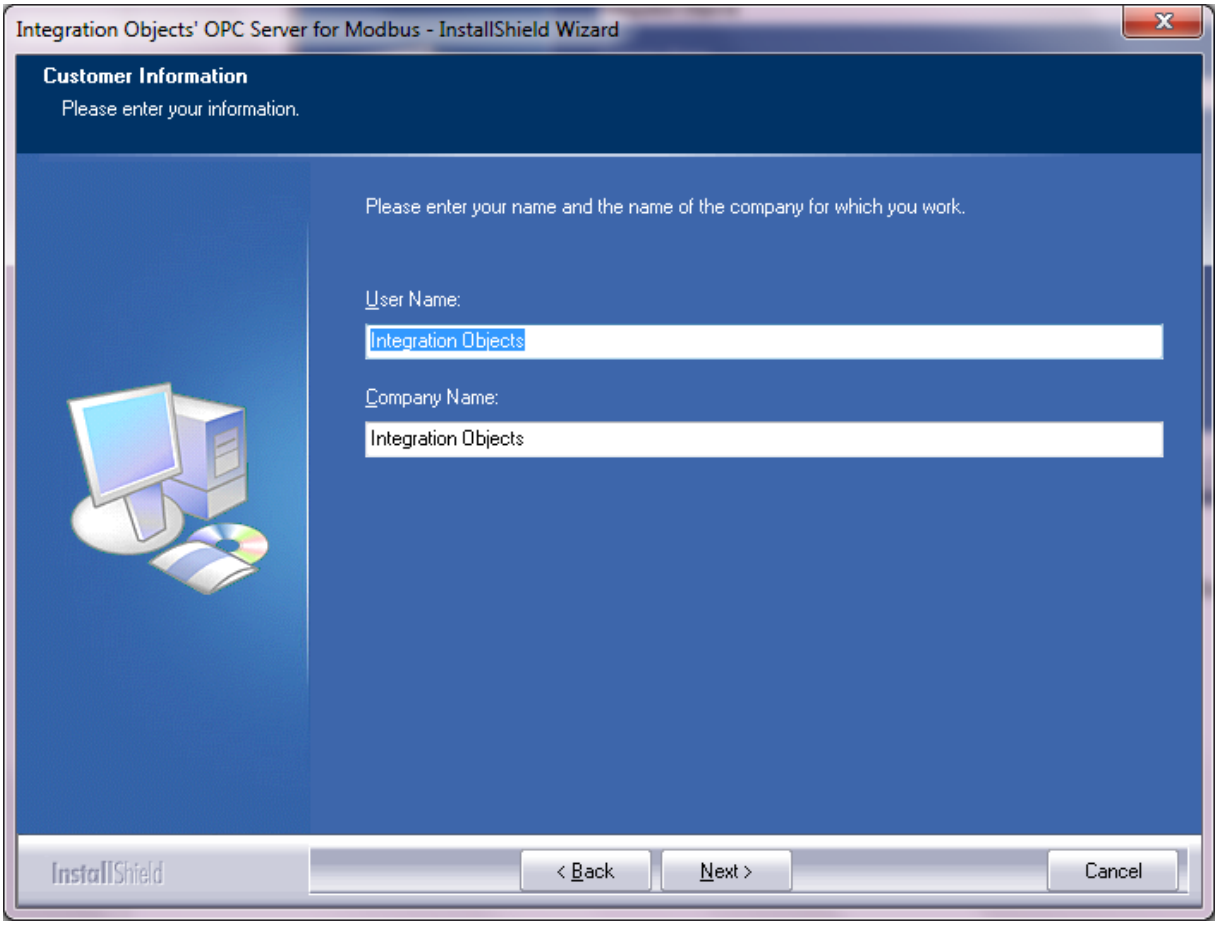


Figure 4: Customer Information Dialog Box

4. **Enter** the user name and the company name and then **click** the Next button. The dialog box for selecting the destination folder will be displayed.

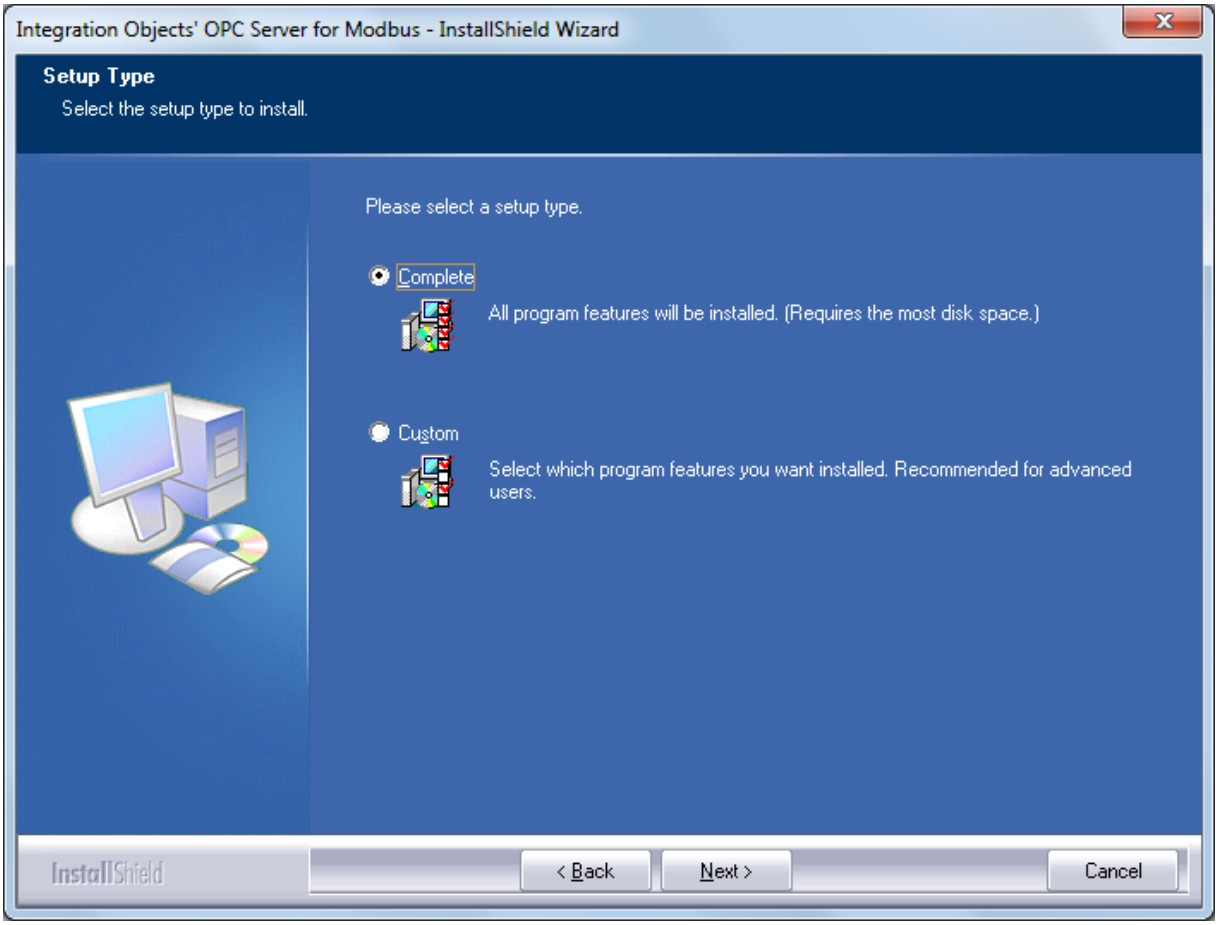


Figure 5: Setup Type Dialog

5. If you choose the **Complete** setup type, all features will be installed.
   If you choose **Custom** setup type, the following dialog will be displayed and you will need to check the specific features that you want to install:

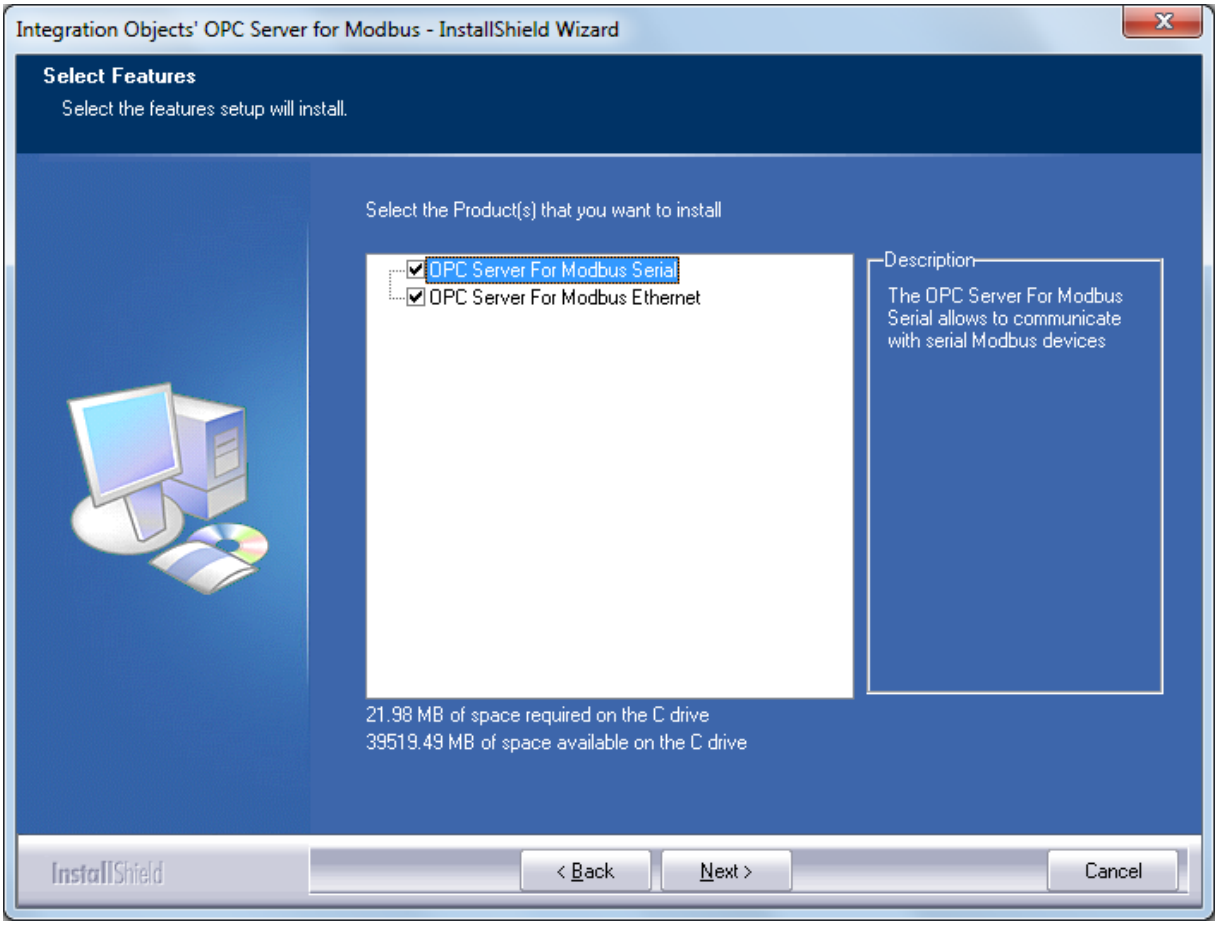


Figure 6: Features Dialog

6. After selecting the features to be installed, the dialog box for choosing the destination folder will be displayed.

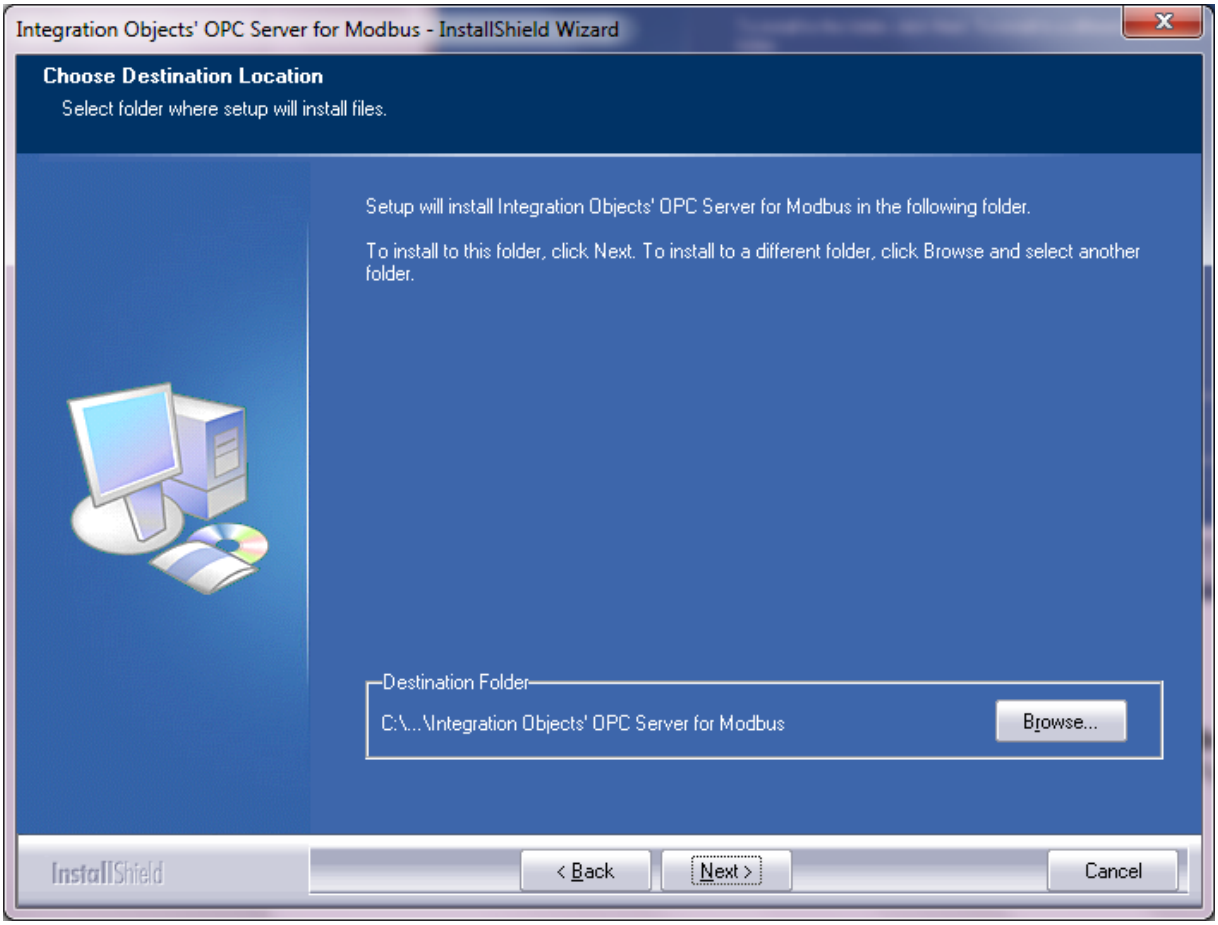


Figure 7: Choose Destination Folder Dialog Box

7. **Click** the Next button to continue the installation, or the Browse button to choose a different destination folder. The installation dialog box will then appear.

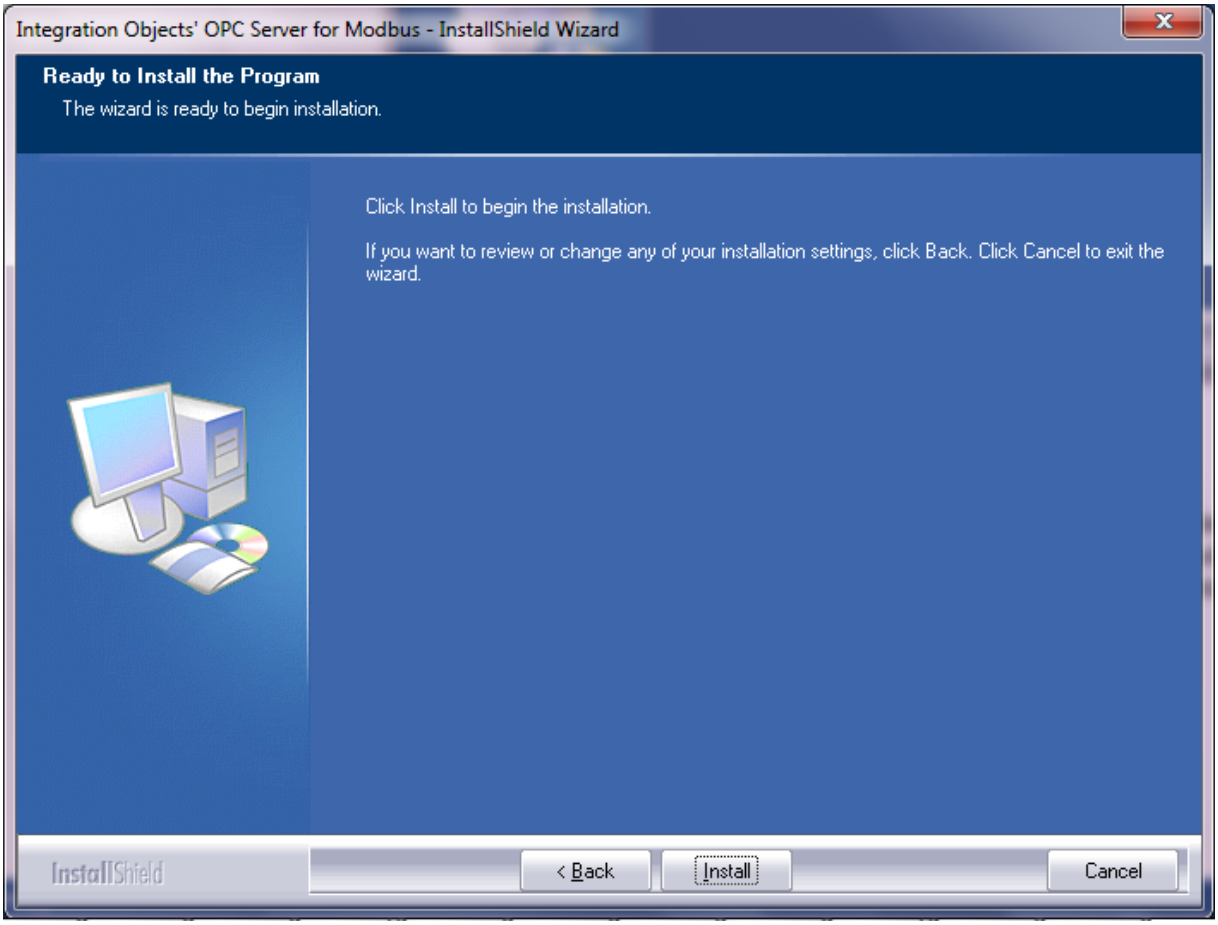


Figure 8: Installation Dialog Box

8. **Click** the Install button to start installation.
   The setup will then:
   - Copy the necessary files to the selected target folder,
   - Create shortcut icons to launch the OPC Server for Modbus and license authorization program from the start menu and the desktop,
   - Make an un-installation entry in the Programs and Features in the Control Panel.

9. If the OPC Core Components are not installed in your machine, you can select **Install OPC Core Components** option as shown in the figure below.

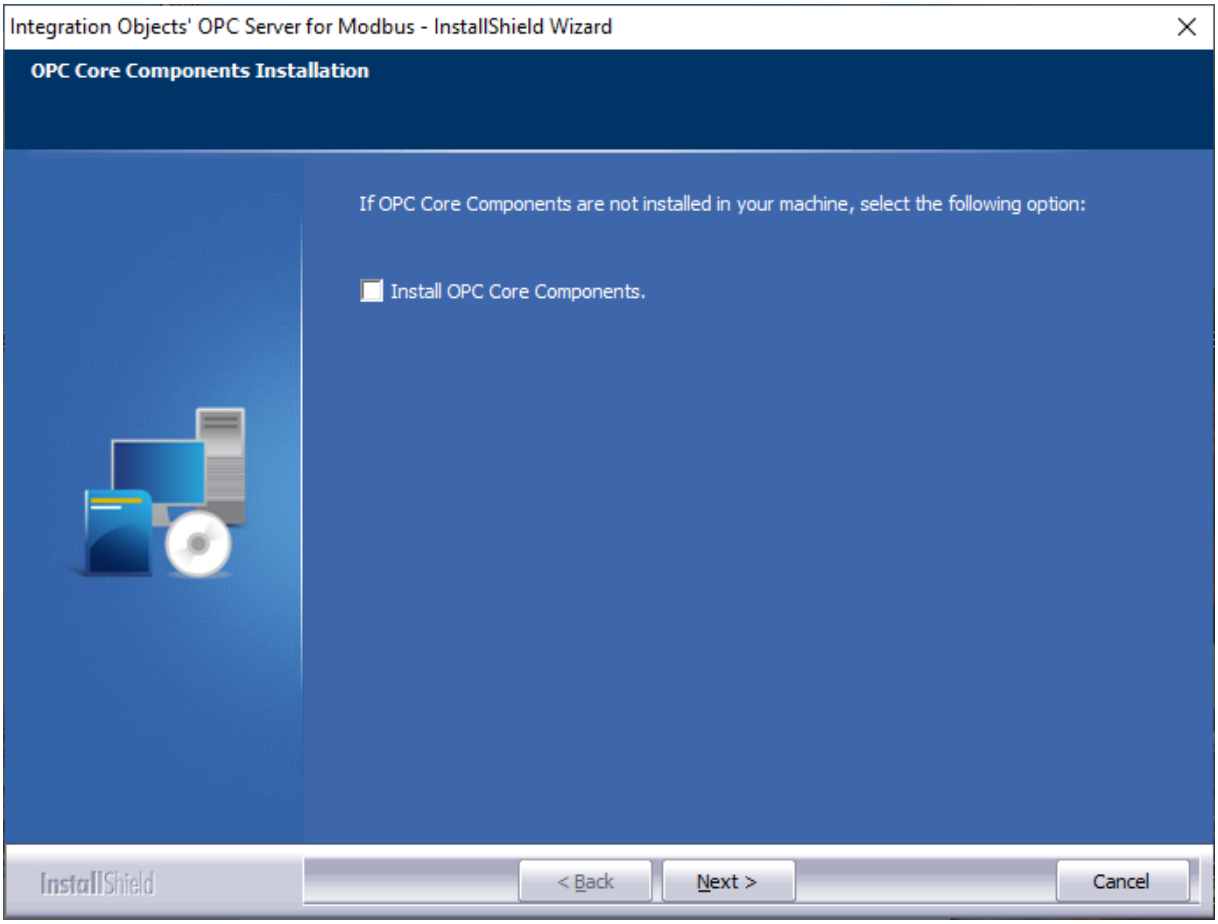


Figure 9: Install OPC Core Components Dialog Box

10. **Click** the finish button.

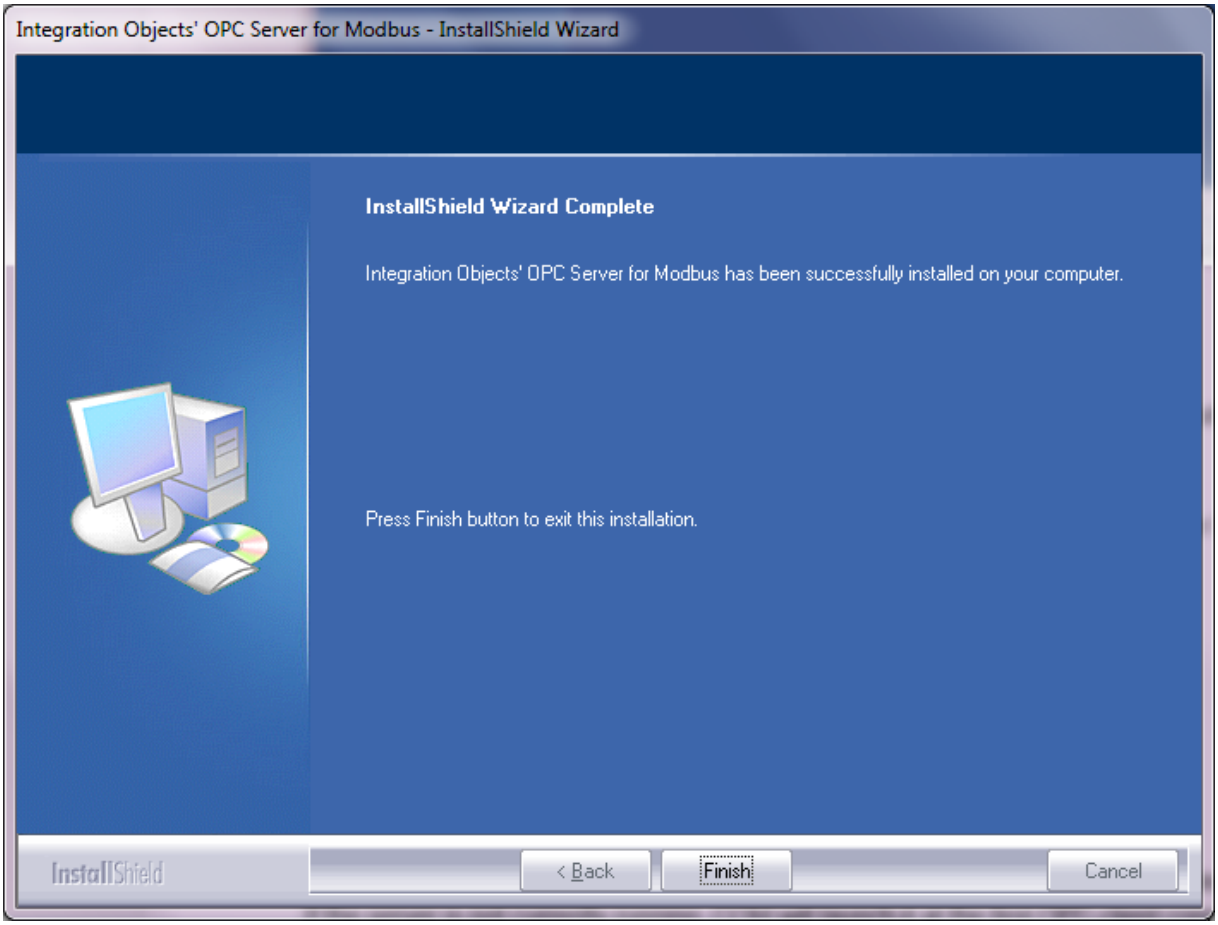


Figure 10: Installation Completed Dialog Box

3. Starting-up

The OPC Server for Modbus service is started automatically with the host machine restart. It can be started and stopped manually from the Windows services manager or from the OPC Server for Modbus user interface. This user interface can be launched from the start menu shortcut.

To do so, **click on** Start ➔ Programs ➔ Integration Objects ➔ OPC Server for Modbus ➔ OPC Server for Modbus

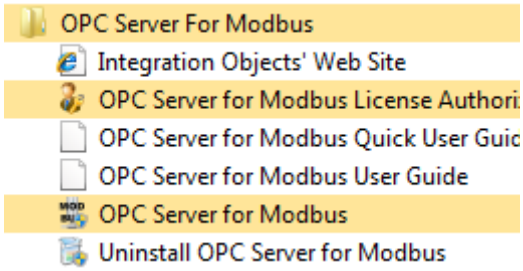


Figure 11: Launch the OPC Server for Modbus User Interface

The server can also be activated dynamically when an OPC DA client is connecting to it.

If the server is not currently running, COM will automatically launch it at the first OPC client connection.

4. Server Registration

In compliance with the OPC and COM specifications, the OPC Server for Modbus creates the following registry entries under HKEY_CLASSES_ROOT when installed on the target system. These entries will be removed when the server is uninstalled.

Registry Entry	Description
IntegrationObjects.OPC.Modbus.1	Integration Objects' OPC Server for Modbus; http://www.integrationobjects.com
IntegrationObjects.OPC.Modbus.1\CLSID	{***CLSID***} = { 66A0F806-7490-46CB-A3D5-40EC8C9F80DF }
CLSID\{***CLSID***}	Integration Objects' OPC Server for Modbus; http://www.integrationobjects.com
CLSID\{***CLSID***}\AppID	{***CLSID***}
CLSID\{***CLSID***}\ProgID	IntegrationObjects.OPC.Modbus.1

Table 2: OPC Server for Modbus Registry Entries

5. Removing the OPC Server

You can remove the server from your machine by clicking the **Uninstaller** shortcut from the start menu.

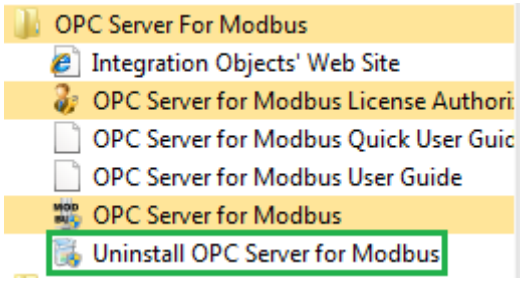


Figure 12: Start Menu – Uninstaller Shortcut

This OPC Server can also be removed manually as follows:

1. **Click** on the Start Menu
2. **Select** the Settings
3. **Click** on the Control panel
4. In the Add/Remove programs dialog box, **select** "Integration Objects' OPC Server for Modbus"
5. **Click** the Change/Remove button and then OK

If you are using Windows 10, Windows Server 2012 or Windows Server 2016 operating systems, the uninstaller needs to be run from the start menu as illustrated below.

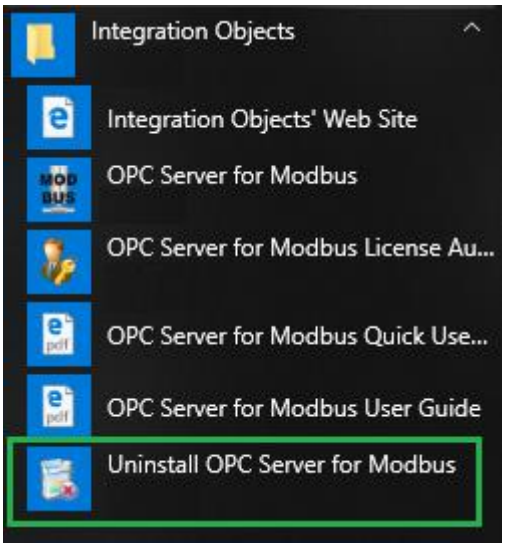


Figure 13: Windows 10 Startup Menu - Uninstall Shortcut

USING OPC SERVER FOR MODBUS

In this section, you will find an overview of the OPC Server for Modbus user interface as well as the steps required to configure and use the application.

1. User Interface Overview

Users can configure the OPC Server for Modbus with an intuitive graphical user Interface.

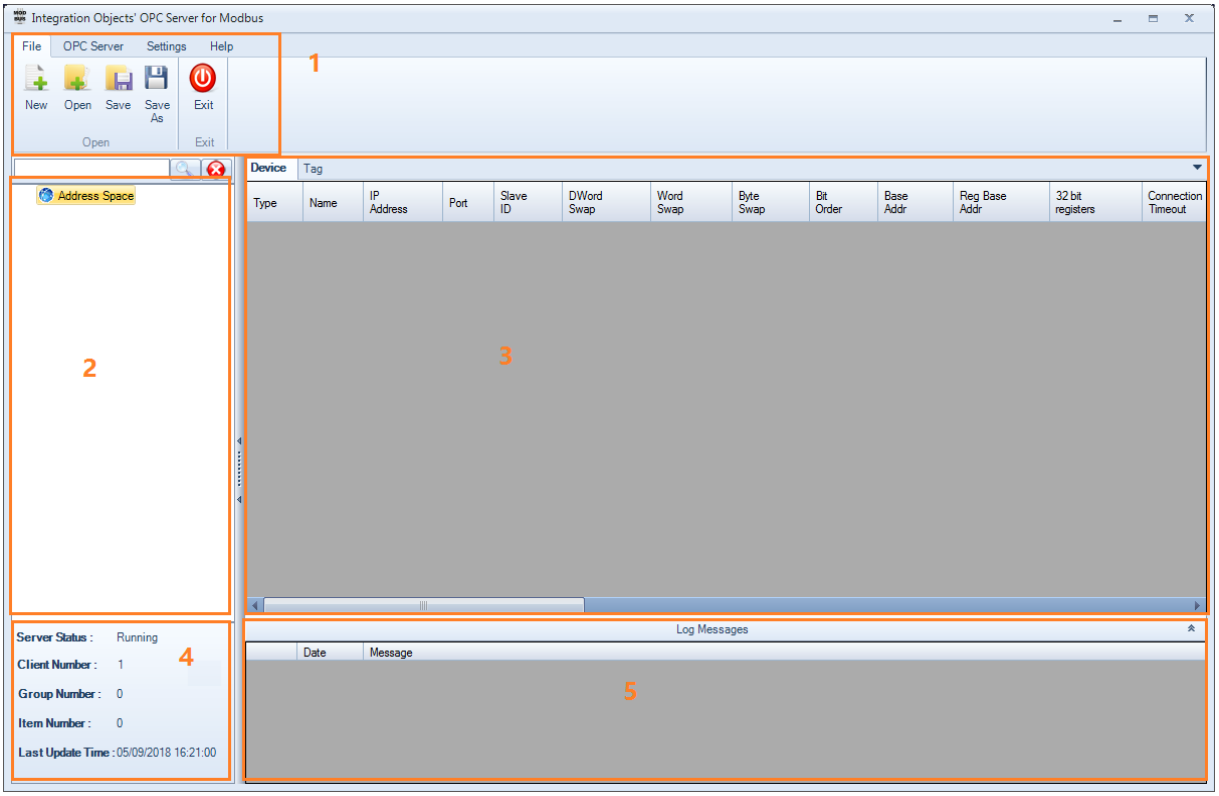


Figure 14: OPC Server for Modbus – Main Interface

The main user interface includes four main sections:

- **Menu bar (1):** contains the file menu, the server menu, the settings menu and the help menu.
- **Server address space tree view (2):** allows to configure the OPC Server address space.
- **Tag & Device properties display (3):** The Tag Tab displays the tags' properties selected from the server address space tree view and the device tab displays the selected device properties.

- **Server status summary (4):** displays server status (running or suspended), the number of connected clients, the number of created groups, the number of created items and the last updated time.
- **Log messages (5):** displays the log messages including the Modbus requests and responses exception messages.

2. File Menu

Figure 15: File Menu

Using the File menu, you can**:**

- Create new configuration by clicking on **New**,
- Open an existing configuration by clicking on **Open** and selecting the appropriate CSV configuration file,
- Save your current configuration by clicking on **Save** or **Save As**,
- Close the application by clicking the **Exit** button.

3. OPC Server Menu

The OPC Server is registered automatically during the installation. The end user can also use the OPC Server menu in the user interface to manually register and unregister the server.

In the same OPC Server menu, you can start and stop the OPC Server for Modbus service by using the right buttons.

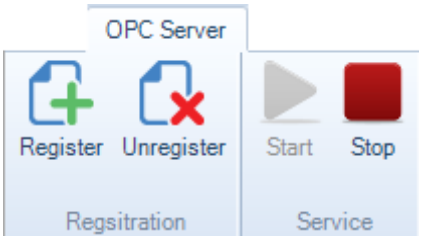


Figure 16: OPC Server Menu

4. Settings Menu

Using the Settings menu, you can:

- **Define** the default configuration that will be loaded automatically when you restart the server.
- **Remove** the default configuration,
- Set up the configuration parameters through the displayed window when you click on the "Configuration" button.
- Select the style of the graphical user interface, which is set by default to "Office2007Blue".
- Configure the server setting when clicking the **Configure** button

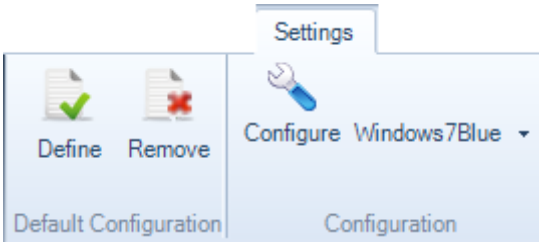


Figure 17: Settings Menu

When the user clicks the **Configure** button, the window below will be prompted:

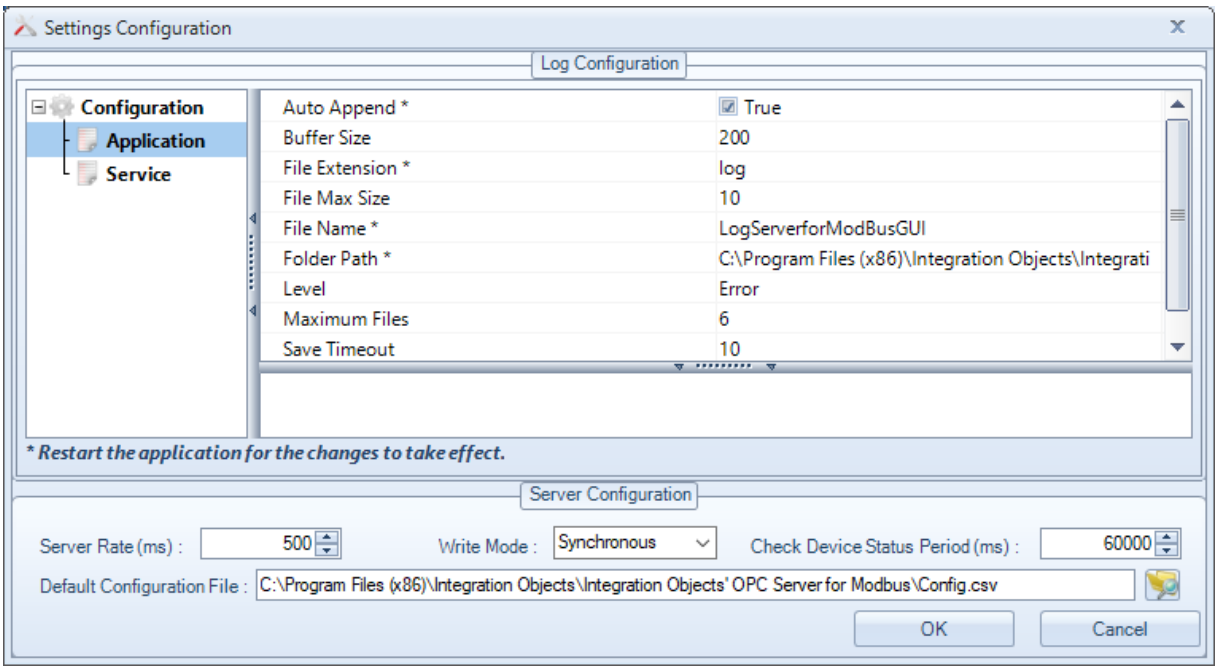


Figure 18: Settings Configuration Windows

In this window, you can configure the following parameters:

Parameter	Description	Default Value
Server Rate	The frequency at which the server handles the asynchronous reads/updates	500 ms (milliseconds)
Write Mode	• Synchronous writes perform a write operation on the Modbus slave and wait for it to complete. • Asynchronous writes perform a write on the Modbus slave but do not wait for the write to complete.	Synchronous
Check device status period	The frequency at which the server will check the device connection status	60 000 ms
Default configuration file	The full path of the server CSV configuration file	Empty
Log Configuration for both Application & Service		
Auto append	Set to true to continue writing log messages in the existed log file or to false to create a new file.	True
Buffer size	The maximum number of messages to be stored in the runtime memory before launching writes action in the hard disk. It must be greater than 100.	200
File extension	The log file extension	log
File max size	The maximum size of the log file (in Mb)	10 Mb
File name	The log file name	• LogServerforModBus GUI: log file of the configuration user interface • LogServerforModBus Service: log file of the service
Folder path	The application folder path	Installation Folder
Level	The type of log messages to be logged. The value can be	Error

	Control, Error, Warning, Inform, and Debug.	
Maximum files	Maximum number of files	5
Auto save timeout	Time to wait to read all messages from the buffer	10

Table 3: Settings

5. Server Configuration

The Server address space can be configured from the tree view at the left side from the main interface.

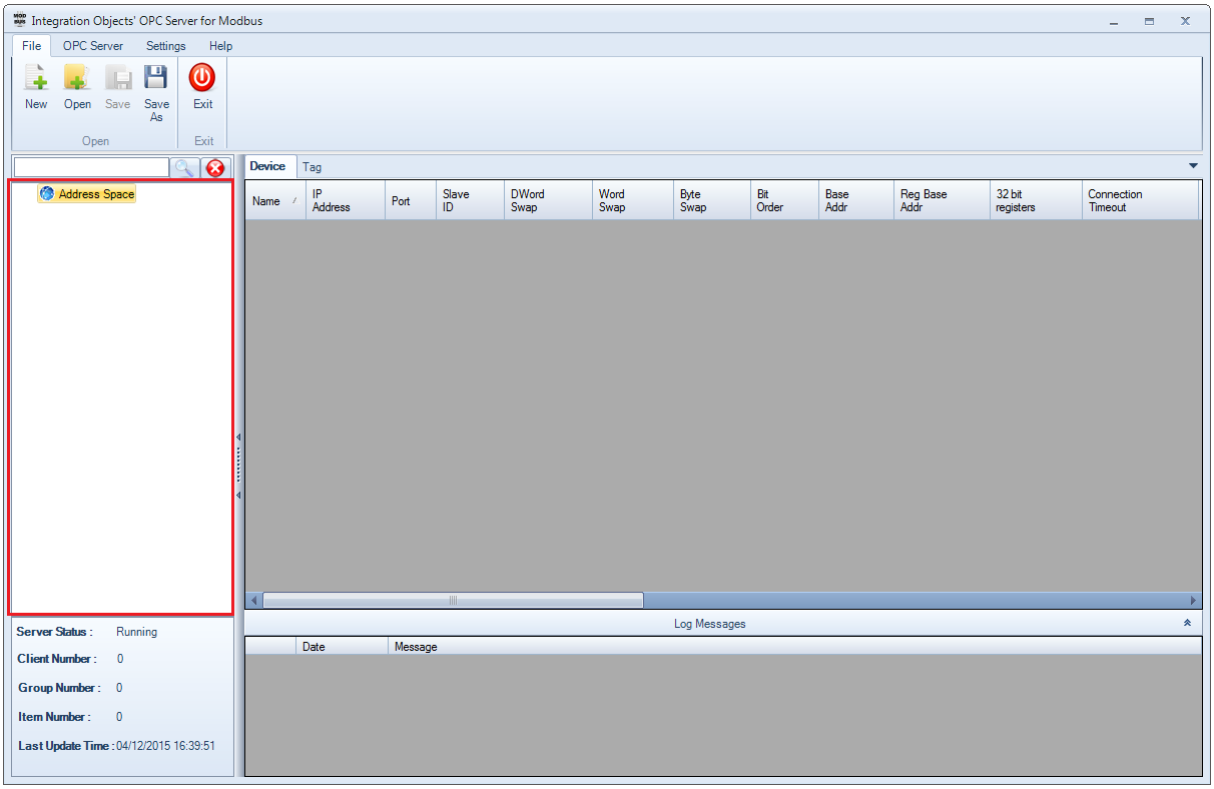


Figure 19: OPC Server for Modbus Address Space Tree View

The sections below describe how to add, edit and delete Modbus devices and tags.

5.1. Add Device

Right click on the Address Space node, select **New Device** as illustrated below.

Figure 20: Add a New Device

Then, the following dialog screen will appear as shown below:

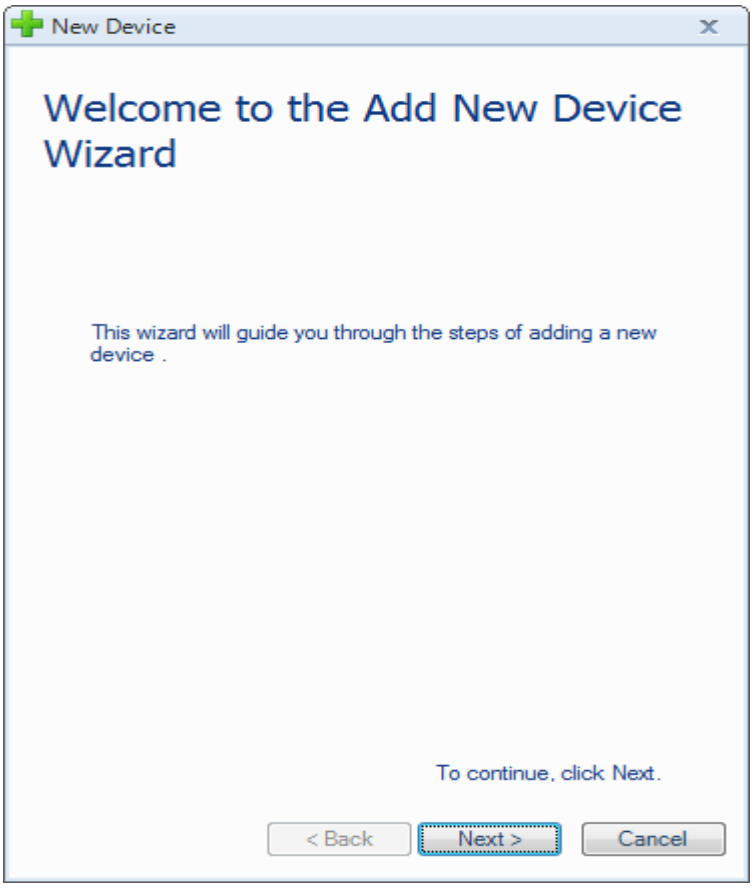


Figure 21: Add New Device Wizard

- Step 1: Connection Parameters
  Click the **Next** button. The following window will be displayed:

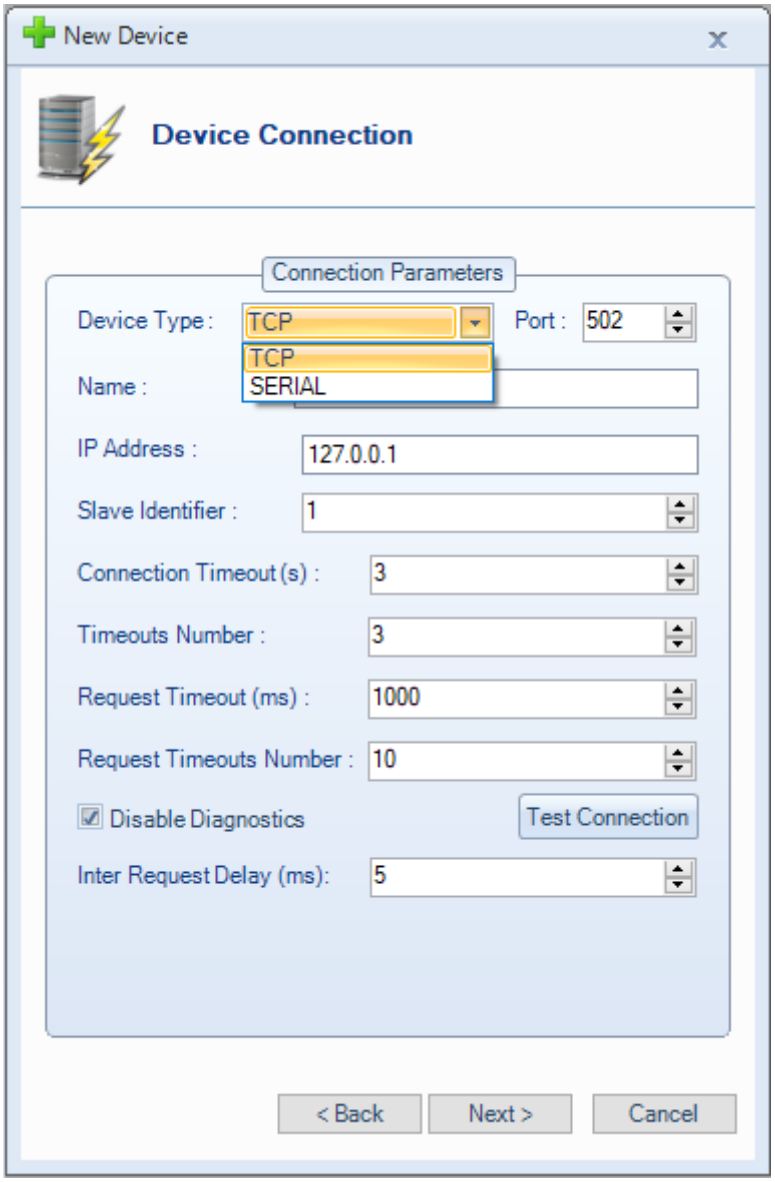


Figure 22: Select Device Connection Configuration

Depending on the selected features during the installation, you can configure the communication with the following Modbus devices types:

- TCP/IP Modbus Device
- Serial Modbus Device

You will then configure the port with reference to the selected type.

- For the Modbus TCP/IP protocol, enter the listening TCP port reserved for the Modbus device communications. The default value is 502.

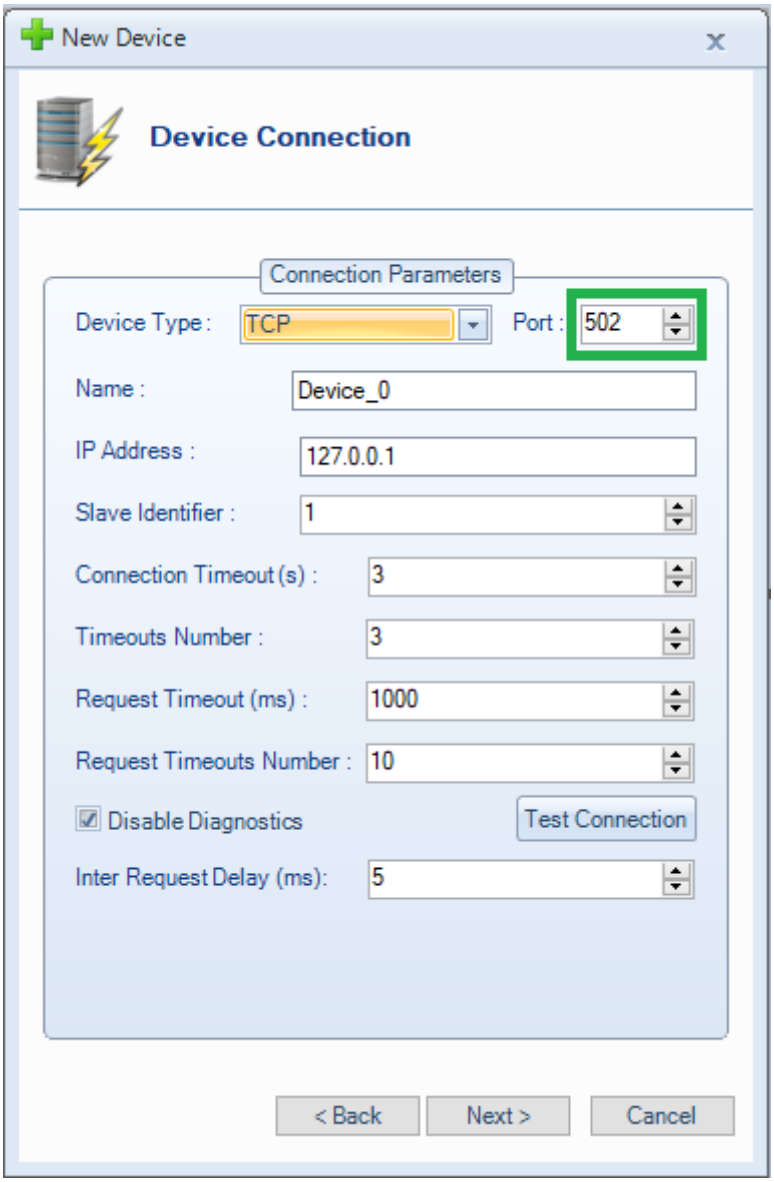


Figure 23: Select the TCP/IP Listening Modbus Port

For the Modbus serial protocol, Click on the **Port Settings** button as shown below:

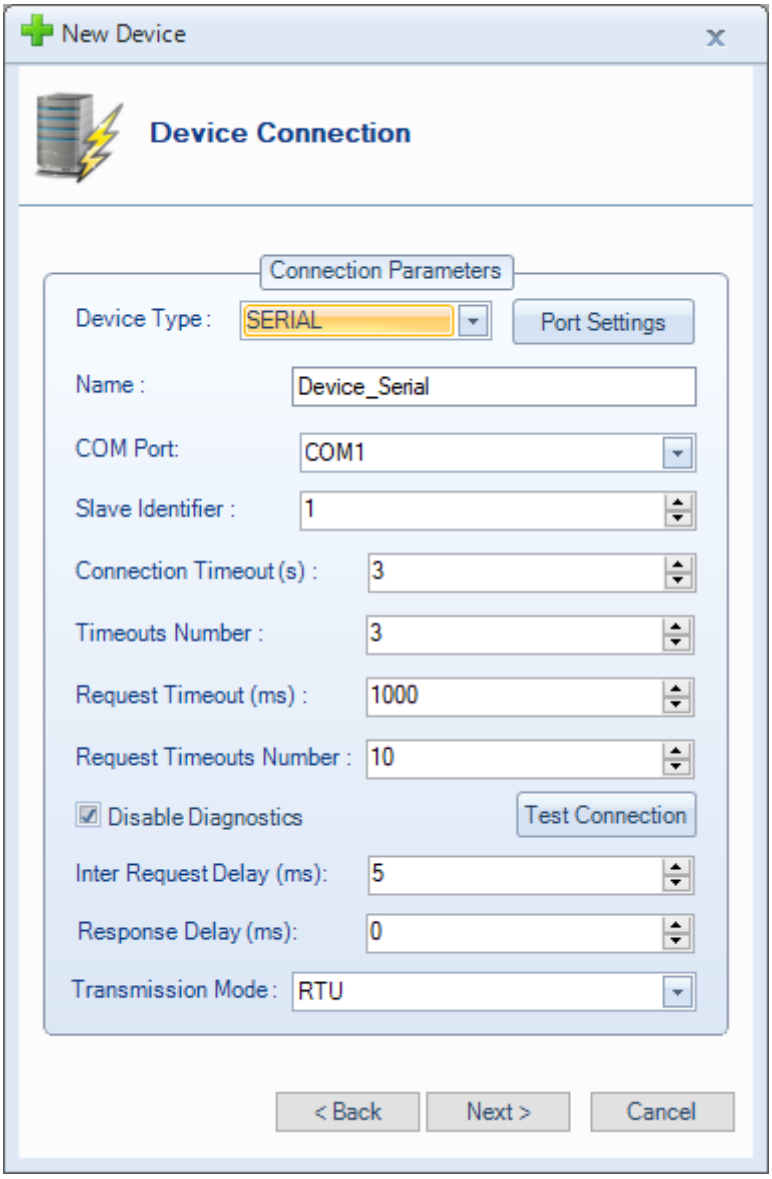


Figure 24: Select the Serial COM Modbus Port

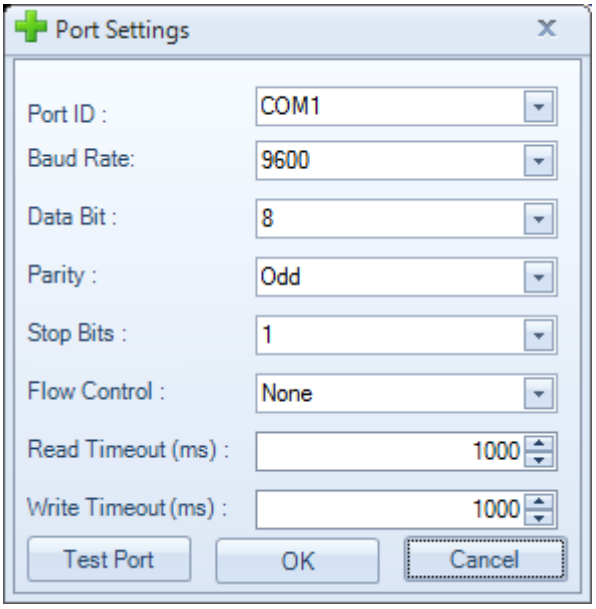


Figure 25: Select the COM Port Settings

The table below summarizes the parameters to configure the serial COM port:

Parameter	Description
Port ID	The port ID
Baud Rate	The baud rate to be used to configure the com port
Data Bits	The number of data bits per data word
Parity	The type of parity for the data
Stop Bits	The number of stop bits per data word
Flow Control	Defines how the RTS and DTR control lines are used
Read Timeout	The read timeout
Write Timeout	The write timeout

Table 4: COM Port Settings Parameters

After choosing the device type, enter the connection parameters.

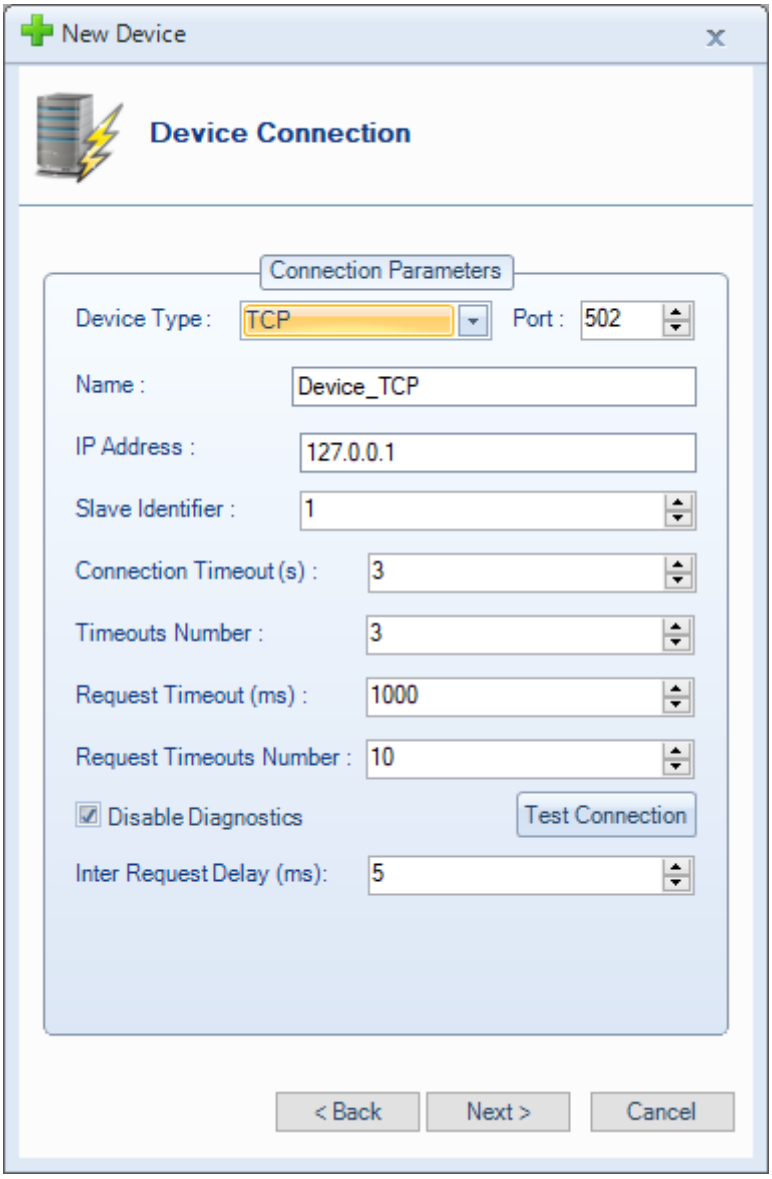


Figure 26: Select the TCP/IP Device Connection Parameters

Parameter	Description
Name	The device name
IP Address	The Modbus device IP address
Slave Identifier	The Identifier of the slave device
Connection Timeout	The waiting period for an unresponsive server
Timeout Number	The allowed timeouts number when the server does not respond
Request Timeout	The amount of seconds that the OPC Server will wait before setting the OPC Tag quality to bad

Request Timeouts Number	The allowed request timeouts number before starting the reconnection procedure to the device
Disable Diagnostics	• When unchecked, it means that the diagnostics function will be used to check the device communication status. • When checked, it means that the diagnostics function will not be used to check the device communication status
Inter Request Delay	Specifies the amount of time between two read requests

Table 5: TCP/IP Device Connection Parameters

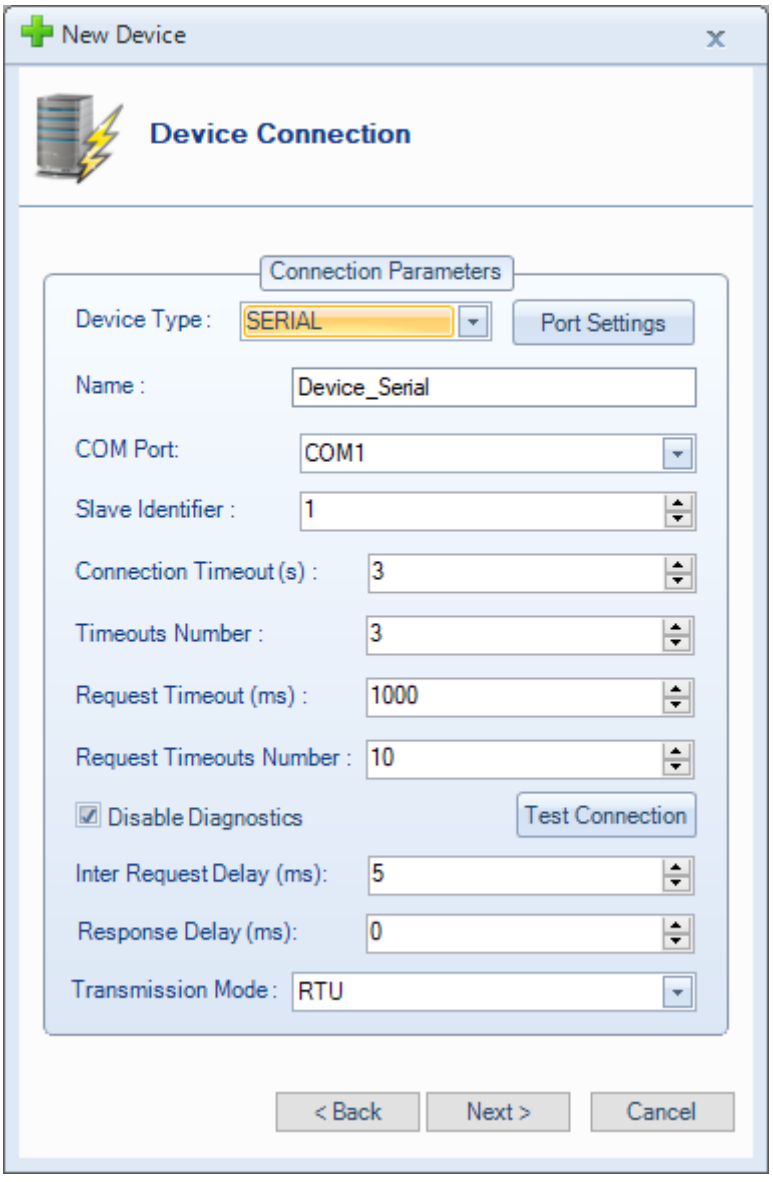


Figure 27: Select the Serial Device Connection Parameters

Parameter	Description
Name	The device name
COM Port	The serial communication port
Slave Identifier	The Identifier of the slave device
Connection Timeout	The waiting period for an unresponsive server
Timeout Number	The allowed timeouts number when the server does not respond
Request Timeout	The amount of seconds that the OPC Server will wait before setting the OPC Tag quality to bad
Request Timeouts Number	The allowed request timeouts number before starting the reconnection procedure to the device
Transmission Mode	The transmission mode. It can be either the RTU mode or the ASCII mode
Disable Diagnostics	• When unchecked, it means that the diagnostics function will be used to check the device communication status. • When checked, it means that the diagnostics function will not be used to check the device communication status
Inter Request Delay	Specifies the amount of time between two read requests
Response Delay	Specifies the amount of time to wait a response from the target device. This delay may be useful in case of devices with slow performance

Table 6: Serial Device Connection Parameters

You can test the availability of the Modbus Device by clicking on the **Test Connection** button. If the Modbus device is available, a message box will be displayed:

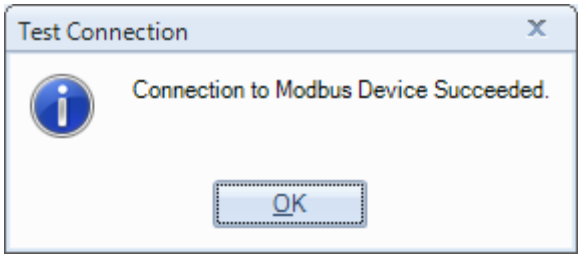


Figure 28: Test Connection with Modbus Device

Step 2: Data Access and Swapping Mode Configuration

Once the device connection parameters are configured, click the **Next** button and the following window will be displayed.

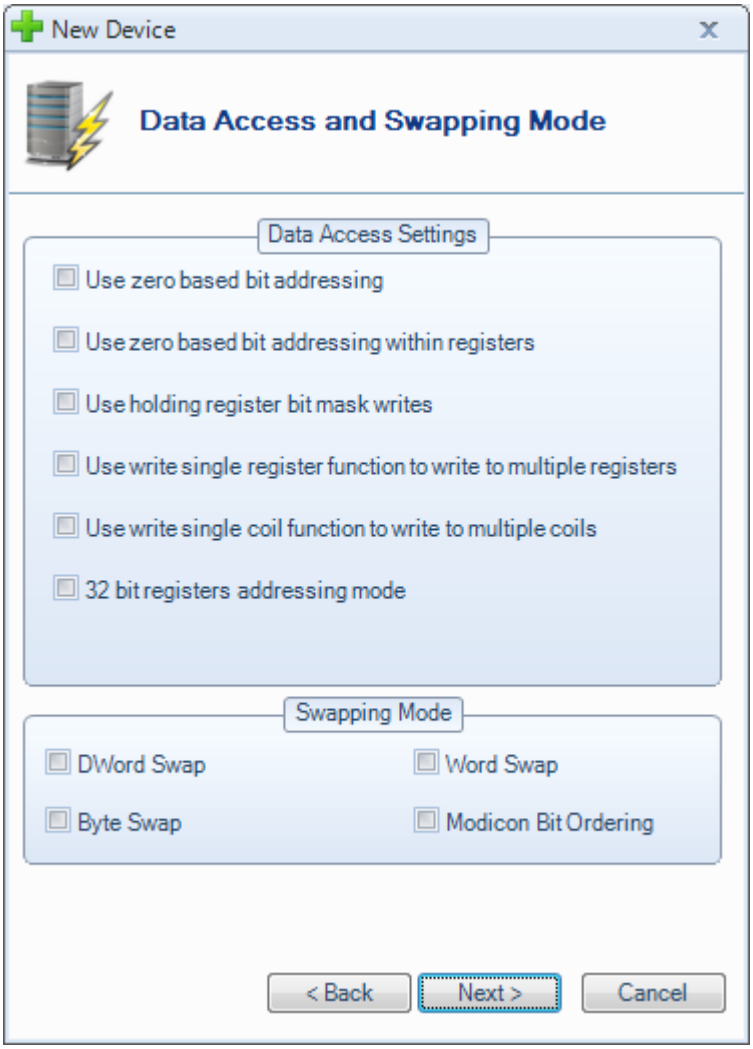


Figure 29: Select Device Data Access & Swapping Mode Parameters

The data access parameters are described in the below table:

Parameter	Description
Use zero based bit addressing	• False (unchecked): The Modbus device address numbering starts at 1 which makes the starting address sent in the Modbus frames request will have one subtracted. • True (checked): The Modbus device address numbering starts at 0 and the starting address included in the Modbus frame request will remain the same.

Use zero based bit addressing within registers	This option is used with bits within registers referenced as Boolean • False (unchecked): The first bit within register begins at one • True (checked): The first bit within register begins at zero
Use holding register bit mask writes	This option is used to write in a bit within register using the boolean datatype with holding registers. • False (unchecked): The OPC Server will use a read /Write operation to update the bit of interest. • True (checked): The OPC Server will use function 22 to update the bit of interest.
Use write single register function to write to multiple registers	This option is related to 32 bits and 64 bits OPC tags. • False (unchecked): The OPC Server will use the function 16 to write to multiple registers. • True (checked): The OPC Server will use function 06 to write into multiple registers.
Use write single coil function to write to multiple coils	• False (unchecked): The OPC Server will use the function 15 to write to multiple coils. • True (checked): The OPC Server will use function 05 to write into multiple coils.
32 bit Registers Addressing Mode	This check box determines how floating points and long integer values are handled: • When unchecked, it means that the Modbus device allocates two 16 bit registers for containing a floating point or long integer value. • When checked, it means that the device allocates one 32 bit register for the value.
Enable synchronous communication	• When unchecked, it means that the asynchronous communication will be used read/write data from/to Modbus device • When checked, it means that the synchronous communication will be used read/write data from/to Modbus device

Table 7: Device Data Access Parameters

You can also select the swapping mode with reference to the below description:

Parameter	Description
DWord Swap	• False (unchecked): Higher dword is sent/received first for double values. • True (checked): Lower dword is sent/received first for double values.
Word Swap	• False (unchecked): Higher word is sent/received first for long, unsigned long or float values. • True (checked): Lower word is sent/received first for long, unsigned long or float values.
Byte Swap	• False (unchecked): Higher byte is sent/received first for integer, unsigned integer, long, unsigned long or float values. • True (checked): Lower byte is sent/received first for integer, unsigned integer, long, unsigned long or float values.
Modicon Bit Ordering	• False (unchecked): Higher bit is sent/received first for Boolean, integer, unsigned integer, long, unsigned long or float values. • True (checked): Lower bit is sent/received first for Boolean, integer, unsigned integer, long, unsigned long or float values.

Table 8: Device Swapping Mode Parameters

Step 3: Auto-demotion, Block Sizes and Error Handling

After configuring the device data access and swapping mode, click the **Next** button and the following window will be displayed.

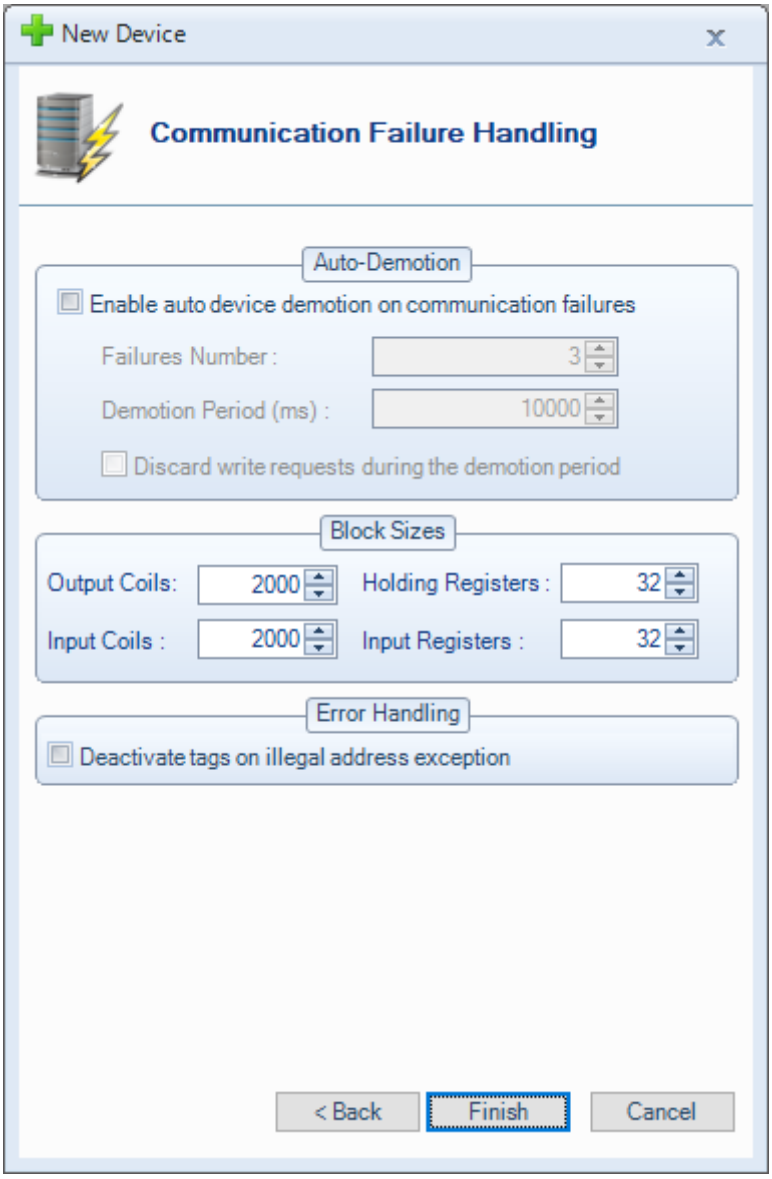


Figure 30: Select Communication Failure Handling Parameters

In addition to the connection parameters, the communication failure is al handled using the Auto-Demotion, the block sizes and the error handling parameters.
Below are the descriptions of each option:

Parameter	Description
Enable auto device demotion on communication failures	When checked, This option allows to demote a device for a specific period of time when communication failures reaches an already configured limited number.
Failures Number	The number of successive failures before demoting the device

Demotion Period	During this period ,no read request will be sent to the device
Discard write requests during the demotion period	When checked, no write request will be sent to the device during the demotion period

Table 9: Auto-Demotion Parameters

Parameter	Description
Output Coils	Specifies the number of output coils in the Modbus frame requests
Input Coils	Specifies the number of input coils in the Modbus frame requests
Input Registers	Specifies the number of output registers in the Modbus frame requests
Holding Registers	Specifies the number of input registers in the Modbus frame requests

Table 10: Device Block Sizes Parameters

After updating the block size properties, you need to restart the service so the changes take effects.

Parameter	Description
Deactivate tags on illegal address exception	• False (unchecked): When illegal data address error is occurred with a block addresses. The OPC Server remains sending read requests to the Modbus device • True (checked): When illegal data address error is occurred with a block addresses. The OPC Server stops sending read requests to the Modbus device

Table 11: Device Error Handling Parameters

Step 4: Click the **Finish** button and the device will be then added to tree view.

5.2. Edit Device

You can edit the device configuration parameters by right clicking on its node and selecting the **Edit** option from the displayed menu.

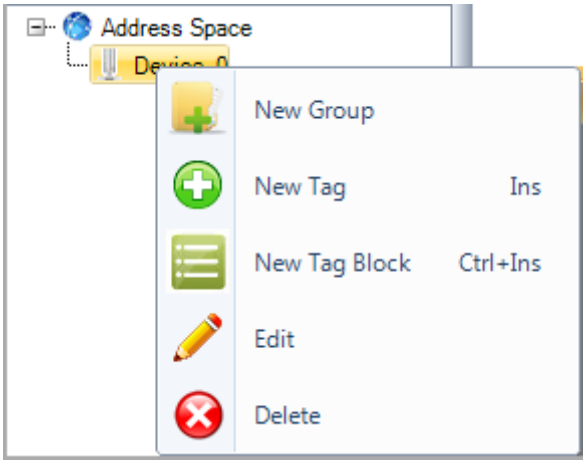


Figure 31: Edit Device

Then, the Edit Device dialog box will be displayed as illustrated below:

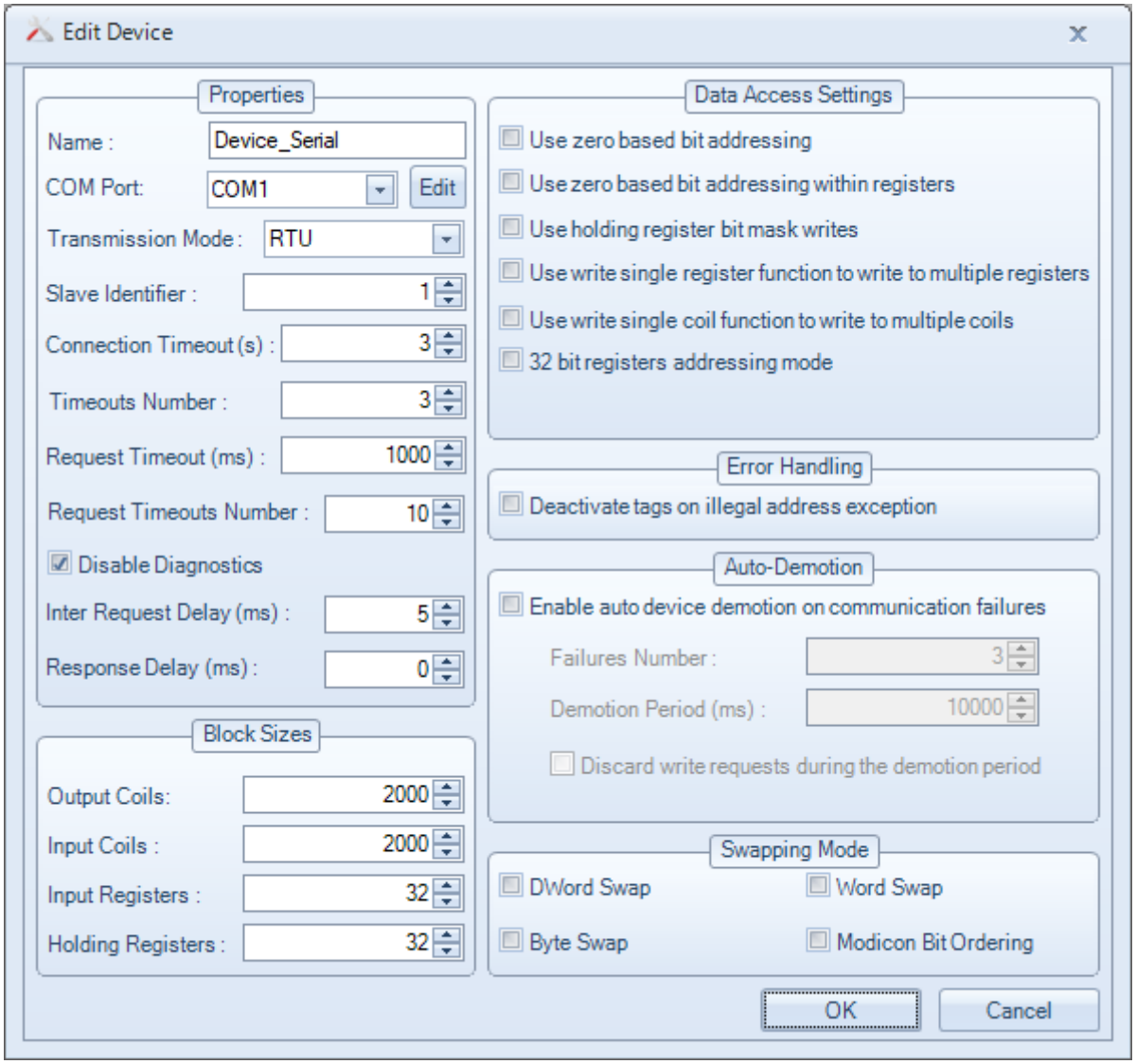


Figure 32: Edit Serial Device Dialog Box

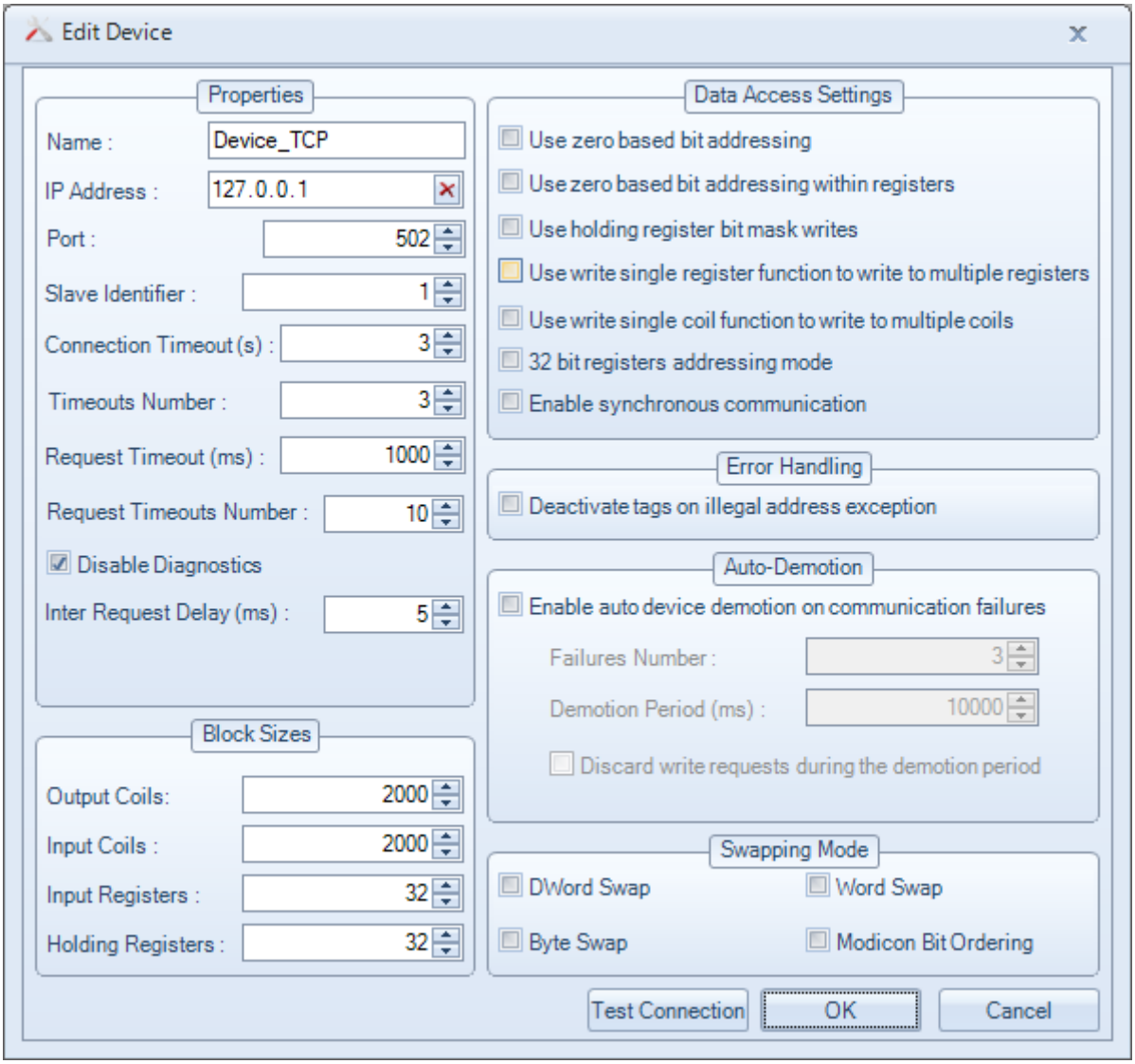


Figure 33: Edit TCP/IP Device Dialog Box

5.3. DELETE DEVICE

You can remove the selected device by right clicking on its node and selecting on the **Delete** option from the displayed menu.

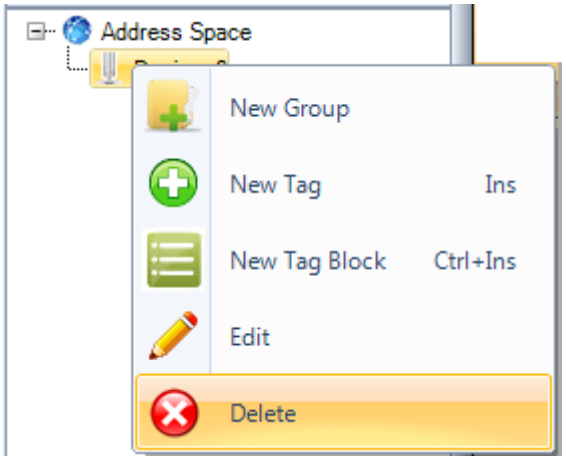


Figure 34: Delete Device

You can also delete multiple devices simultaneously by selecting the devices to be deleted or clicking the **Delete All Devices** option to remove all the devices. To select multiple devices in the address space tree, use the Shift or Ctrl keys.

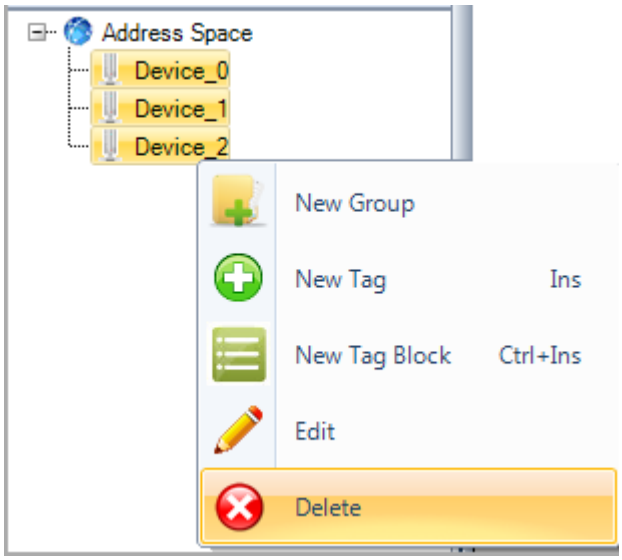


Figure 35: Delete Multiple Devices

Figure 36: Delete All Devices

5.4. ADD GROUP

You can add groups under the device node in order to build hierarchal organization of the server address space. A group belongs to a device and contains one or more OPC tags. It can also contain other sub-groups.

Right click on the device node and then select the **New Group** option from the displayed menu.

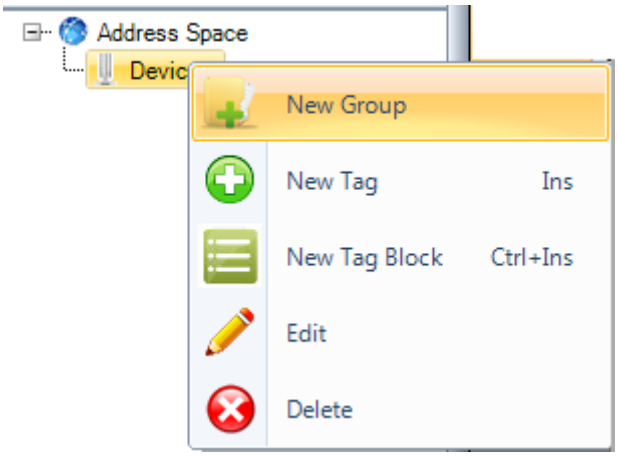


Figure 37: New Group

Then, the New Group dialog box will be displayed as illustrated below. You can have to specify the group name.

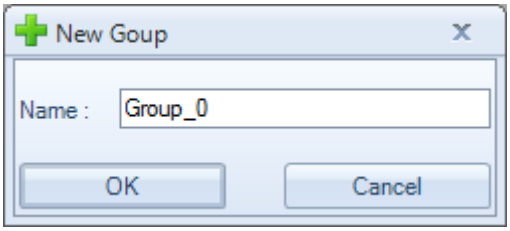


Figure 38: New Group Dialog Box

5.5. Edit & Remove Group

You can edit the group name by right clicking on the group node and then select the **Edit** option as shown in the following figure.

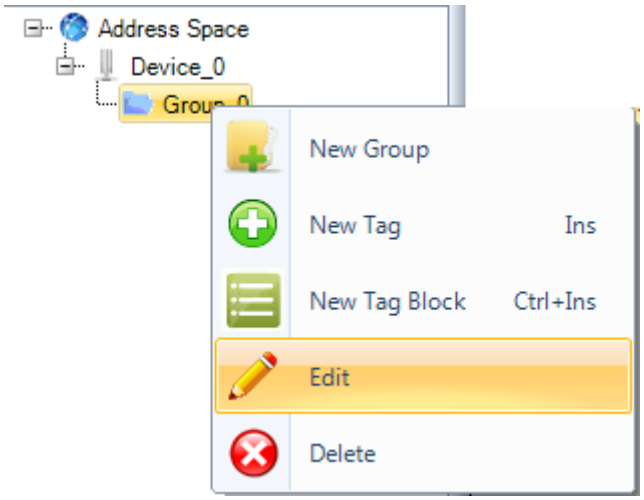


Figure 39: Edit Group

The Edit Group dialog box will be then displayed as below:

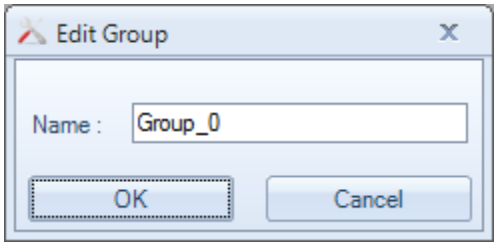


Figure 40: Edit Group Dialog Box

You can also remove the selected group by right clicking on its node and then selecting on the **Delete** option from the displayed menu. The Group node and all its tags and subgroups will be removed from the address space.

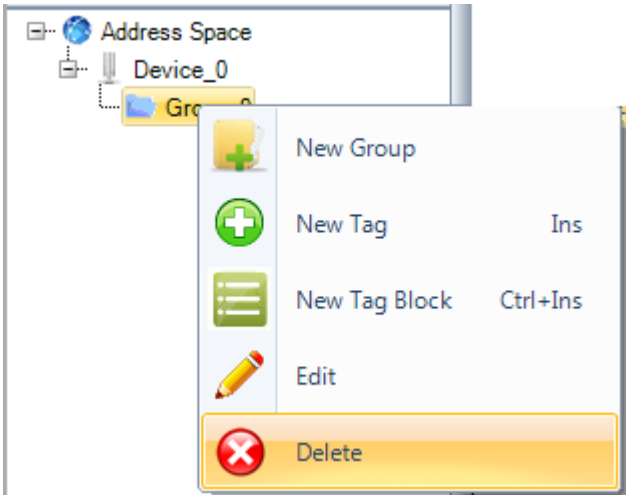


Figure 41: Delete Group

The end user can delete multiple groups simultaneously by selecting the groups using the Shift or Ctrl keys and clicking the **Delete** option available in the right click menu.

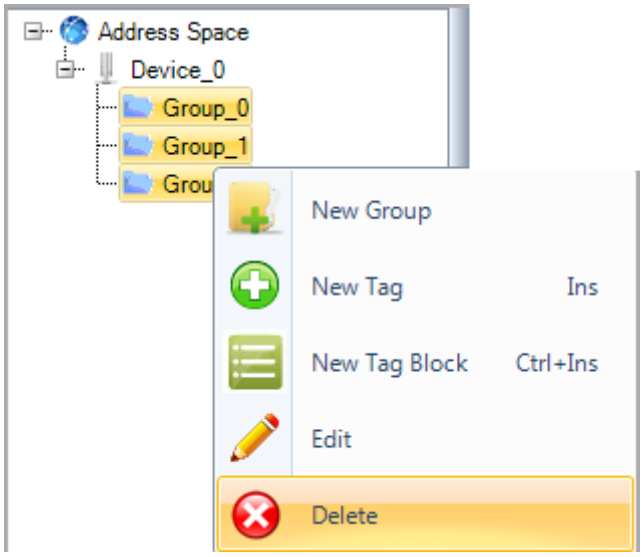


Figure 42: Delete Multiple Groups

5.6. ADD TAG

A tag can be added directly to a device (address space flat organization) or to a group. The figure below shows how to add a new tag to a device or to a group of tags.

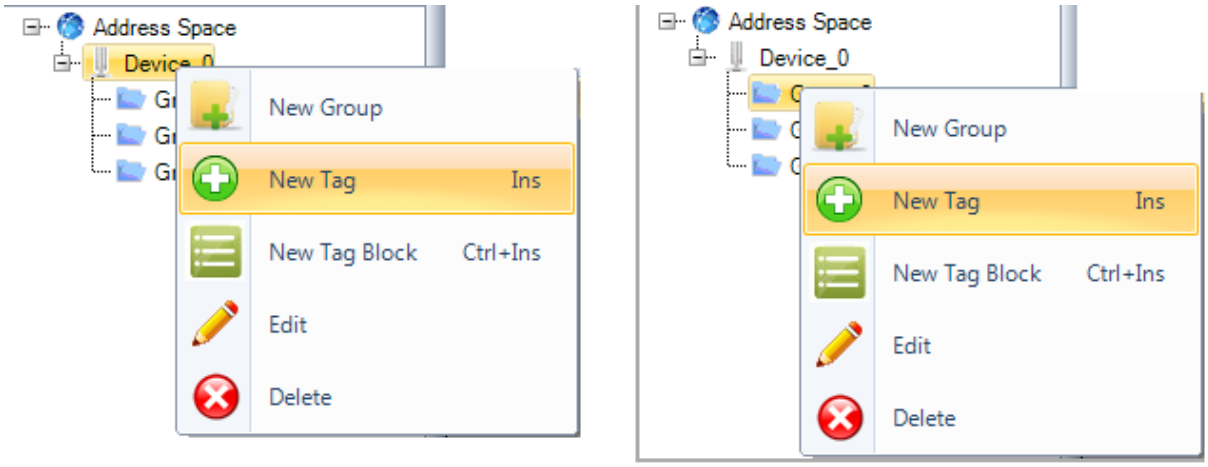


Figure 43: New Tag

Once the **New Tag** option is selected, a dialog box will be displayed allowing the user to configure the tag parameters.

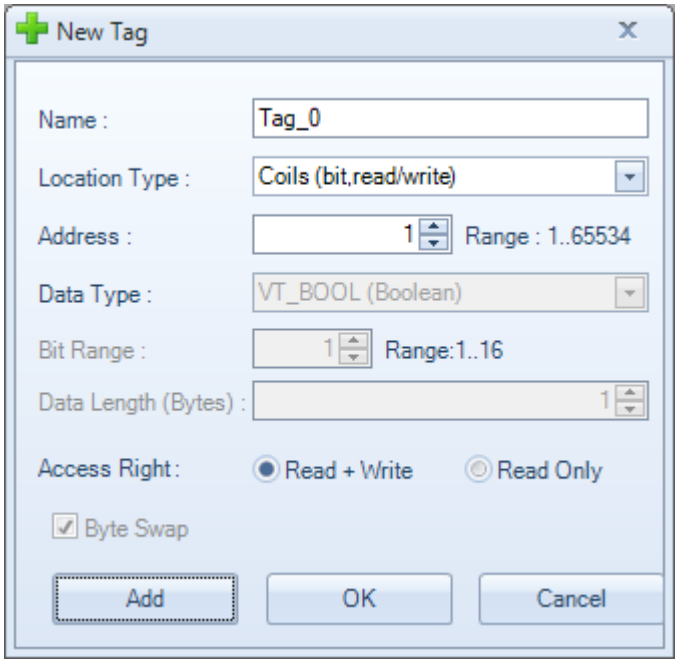


Figure 44: New Tag Dialog Box

The table below summarizes the parameters to configure your tags:

Parameter	Description
Name	The tag name
Location Type	The Modbus address type, which can be: • Coils • Discrete Inputs • Holding Registers • Input Registers ***Refer to the Location Type table for more details.***
Address	The address from where the data will be extracted or modified. The value should be between 1 and 65534.
Bit Range	This is enabled with Boolean tags whose address type is Holding Register or Input Register. It denotes the range of a bit in one register.
Data Type	The type of data, which can be: • VT_I2: Variable type is 2-byte signed integer • VT_I4: Variable type is 4-byte signed integer • VT_R4: Variable type is 4-byte real • VT_R8: Variable type is 8-byte double • VT_UI2: Variable type is an unsigned integer • VT_UI4: Variable type is an unsigned long • VT_BSTR: Variable type is binary string • VT_BOOL: Variable type is Boolean

Data Length	The bytes number of the selected data type
Access Right	The data access right, which can be read only or read/write
Byte Swap	• False (unchecked): Higher byte is sent/received first for string values. • True (checked): Lower byte is sent/received first for string values.

Table 12: Tag Parameters

Name	Object Type	Access Right	Description
Coils	Single bit	Read-Write	This type of data can be altered by an application program.
Discrete Inputs	Single bit	Read-Only	This type of data can be provided by an I/O system.
Holding Registers	16-bit word	Read-Write	This type of data can be altered by an application program.
Input Registers	16-bit word	Read-Only	This type of data can be provided by an I/O system

Table 13: Location Type

5.7. ADD TAG PER BLOCK

You can add simultaneously multiple tags having the same location type and access right but different addresses. Right click on the group or device node and then select the **Add Tag Block** option.

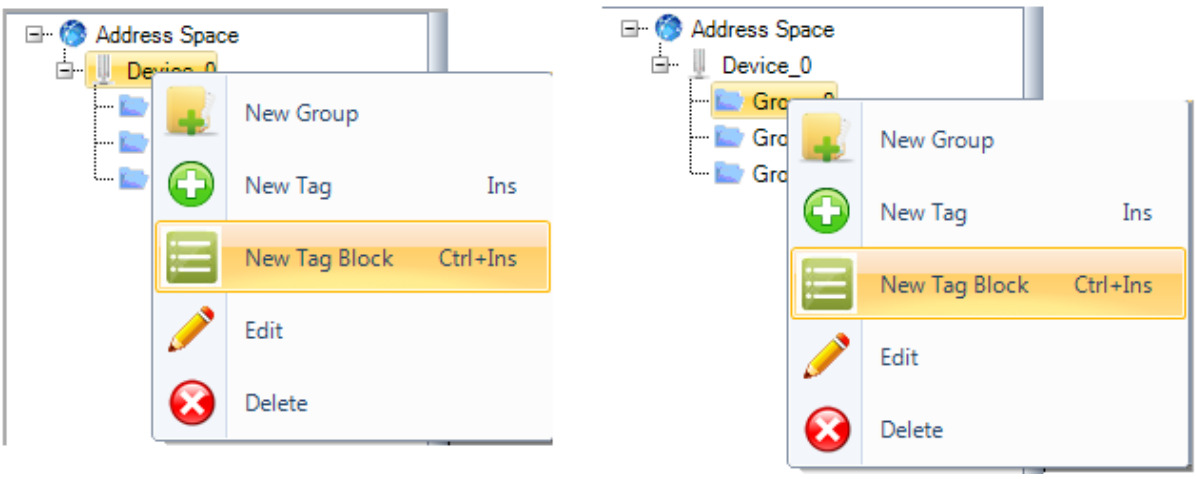


Figure 45: New Tag Block

The New Tag Block dialog box will be displayed as following:

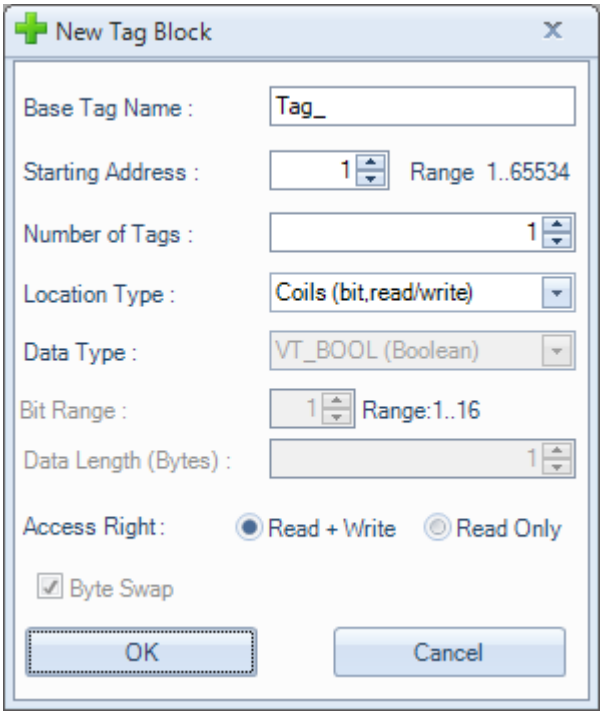


Figure 46: New Tag Block Dialog Box

Parameter	Description
Base Tag Name	The pattern used to generate the name of the tags
Location Type	The Modbus address type, which can be : • Coils • Discrete Inputs • Holding Registers • Input Registers ***Refer to the Location Type table for more details.***
Starting Address	The address of the first tag. Its value should be between 1 and 65534.
Number of Tags	The number of the tags to be added
Data Type	The type of data, which can be: • VT_I2: Variable type is 2-byte signed integer • VT_I4: Variable type is 4-byte signed integer • VT_R4: Variable type is 4-byte real • VT_UI2: Variable type is an unsigned integer • VT_UI4: Variable type is an unsigned long • VT_BSTR: Variable type is binary string • VT_BOOL: Variable type is Boolean
Data Length	The bytes number of the selected data type

Access Right	The data access right, which can be read only or read/write
Byte Swap	• False (unchecked): Higher byte is sent/received first for string values. • True (checked): Lower byte is sent/received first for string values

Table 14: Tag Block Properties

5.8. EDIT TAG

You can update a tag configuration by clicking on the **Edit** option in the right click menu.

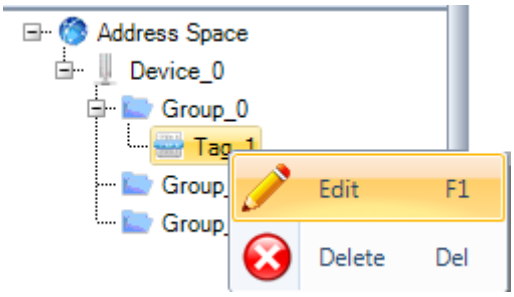


Figure 47: Edit Tag

The Edit Tag Dialog Box will be displayed as illustrated below:

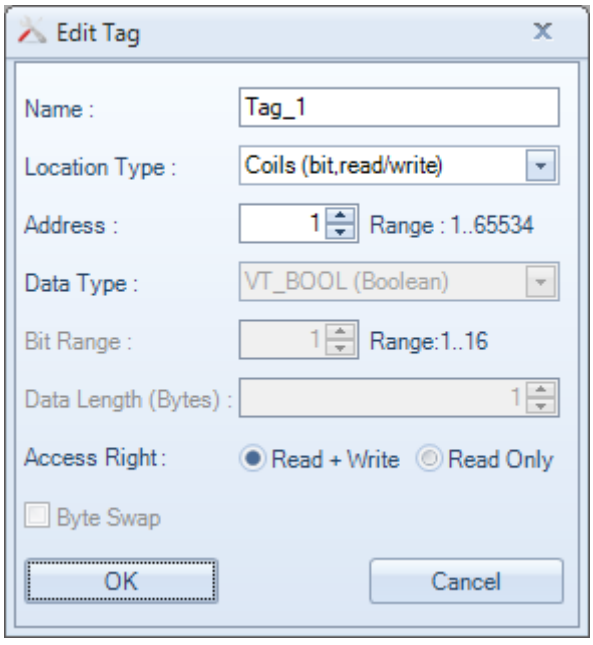


Figure 48: Edit Tag Dialog Box

5.9. DELETE TAG

You can delete a tag from the address space by clicking the **Delete** option in the right click menu.

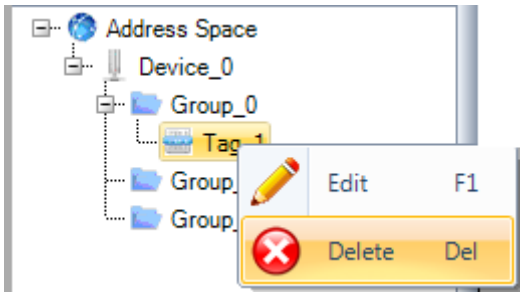


Figure 49: Delete Tag

You can delete more than one tag simultaneously by selecting the tags to be removed using the Shift or Ctrl keys and then selecting the **Delete** option in the right click menu.

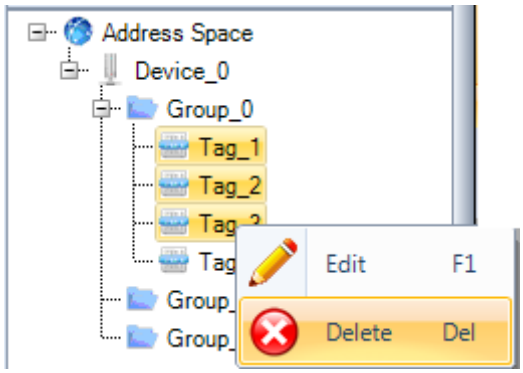


Figure 50: Delete Multiple Tags

5.10. VIEW TAGS PROPERTIES

Click on the group, device or tag node in the address space tree view and the tags properties will be displayed in the grid view on the right. In the figure below, we can see that only the tags under the selected groups are displayed in the grid view.

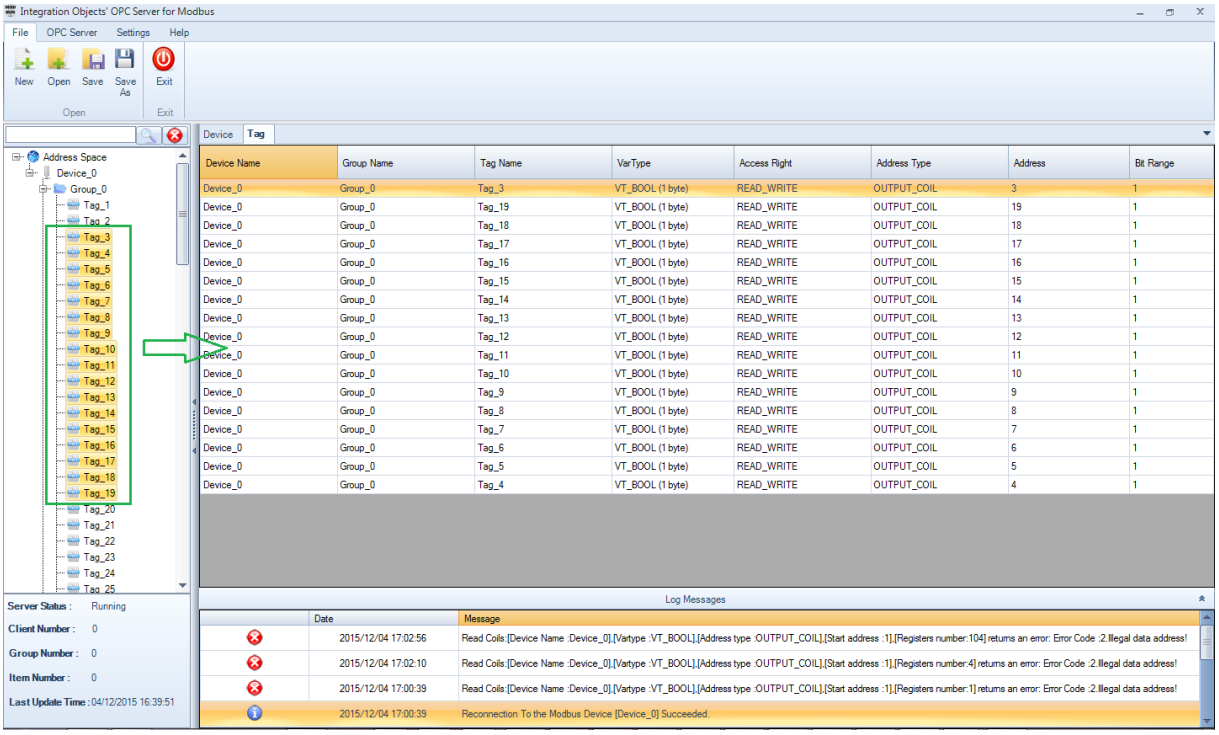


Figure 51: Display Tags of Selected Groups

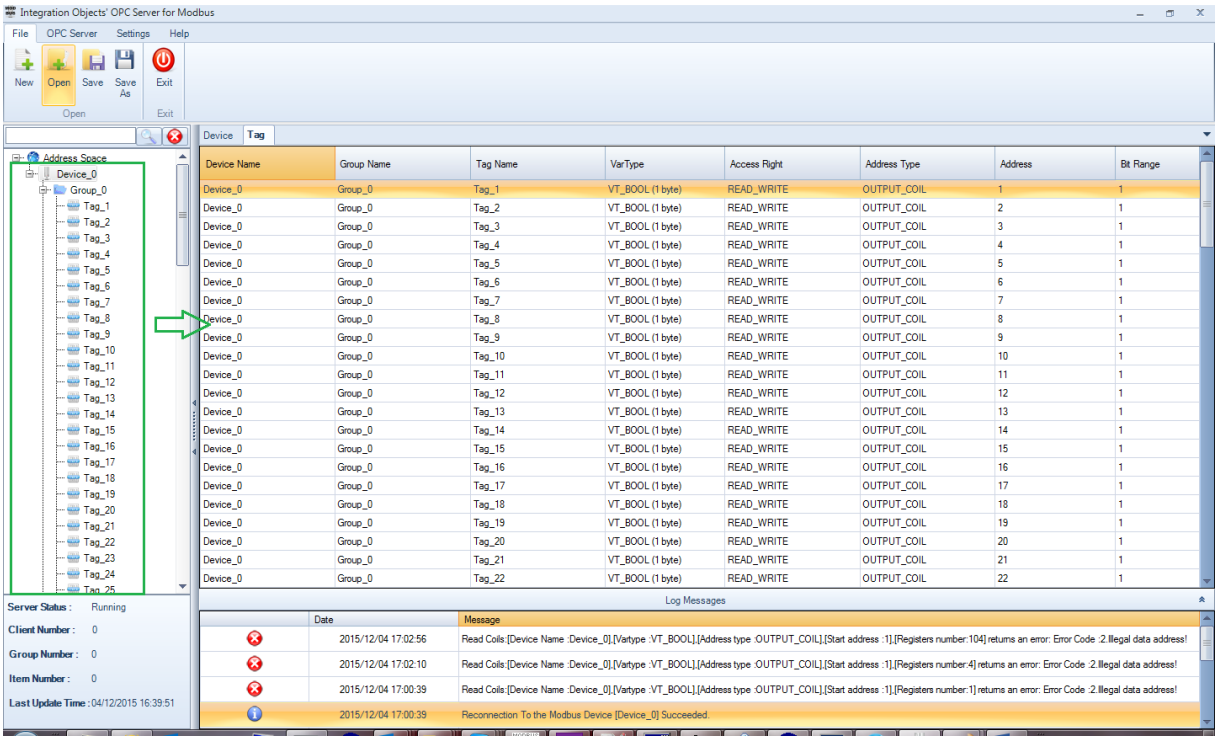


Figure 52: Display Tags of a Selected Device

5.11. SAVE CONFIGURATION

Save your configuration using the **Save As** or **Save** button available in the file menu. The Save File As dialog box is then displayed to choose the path where the configuration will be saved.

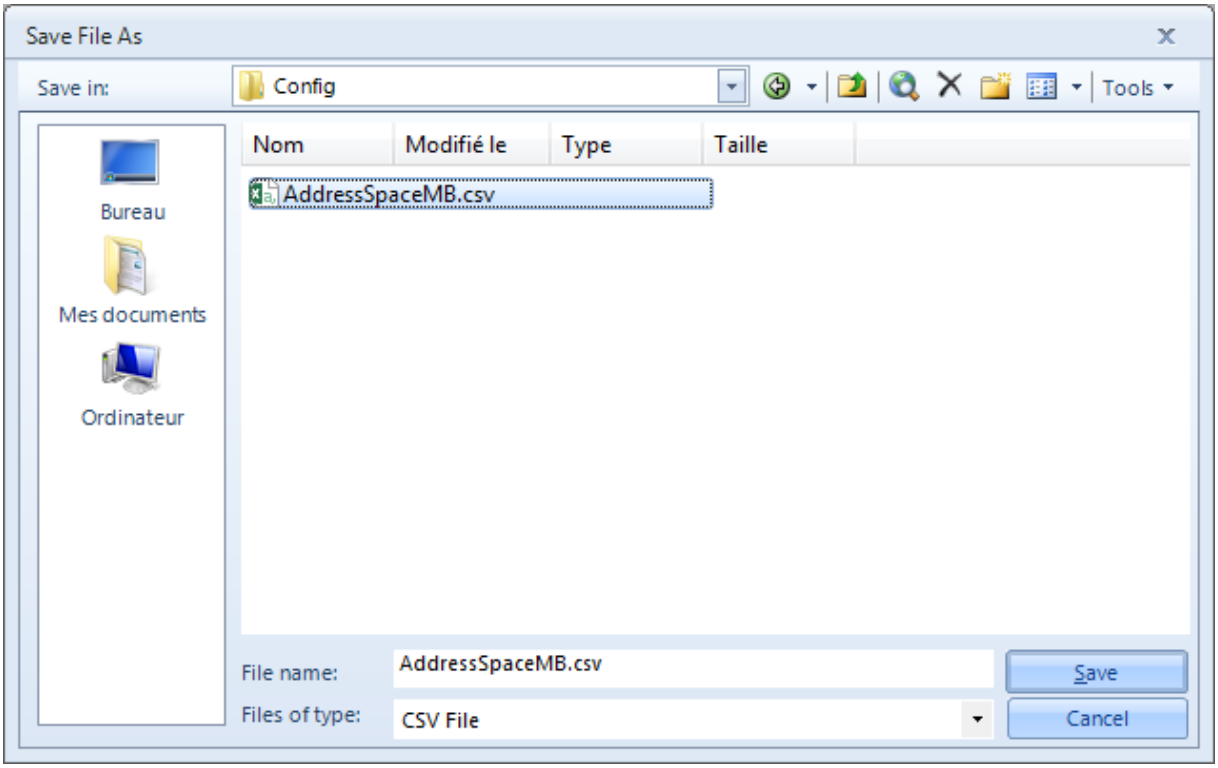


Figure 53: Save File As Dialog Box

The configuration will be then saved in CSV file format. The configuration is divided into three sections;

- COM port
- Devices
- Tags

The table below describes the fields of the COM port configuration section.

	Field	Description
1	**Port ID**	The port ID
2	**Baud Rate**	The baud rate to be used to configure the com port
3	**Data Bits**	The number of data bits per data word
4	**Parity**	The type of parity for the data
5	**Stop Bits**	The number of stop bits per data word
6	**Flow Control**	Defines how the RTS and DTR control lines are used
7	**Read Timeout**	The read timeout
8	**Write Timeout**	The write timeout

Table 15: COM Port Configuration Section Format

The table below describes the fields of the devices configuration section.

	Field	Description
1	**Type**	The device type (SERIAL/TCP)
2	**Serial Port**	The serial com port ID
3	**Device Name**	The device name
4	**IP Address**	The Modbus device IP address
5	**Slave ID**	The Identifier of the slave device
6	**Port**	The listening TCP port reserved for the Modbus device communications. The default value is 502.
7	**Connection Timeout**	The waiting period for an unresponsive server
8	**Timeouts Nbr**	The allowed timeouts number when the server does not respond
9	**Request Timeout**	The amount of seconds that the OPC Server will wait when using the diagnostics function to check the Modbus Device availability.
10	**In Coils Blck Size**	Specifies the number of input coils in the Modbus frame requests
11	**Out Coils Blck Size**	Specifies the number of output coils in the Modbus frame requests
12	**In Reg Blck Size**	Specifies the number of input registers in the Modbus frame requests
13	**H Reg Blck Size**	Specifies the number of output registers in the Modbus frame requests
14	**DWord Swap**	• False (unchecked): Higher dword is sent/received first for double values. • True (checked): Lower dword is sent/received first for double values.
15	**Word Swap**	• False (unchecked): Higher word is sent/received first for long, unsigned long or float values. True (checked): Lower word is sent/received first for long, unsigned long or float values.

16	**Byte Swap**	• False (unchecked): Higher byte is sent/received first for integer, unsigned integer, long, unsigned long or float values. • True (checked): Lower byte is sent/received first for integer, unsigned integer, long, unsigned long or float values.
17	**Bit Order**	• False (unchecked): Higher bit is sent/received first for Boolean, integer, unsigned integer, long, unsigned long or float values. • True (checked): Lower bit is sent/received first for Boolean, integer, unsigned integer, long, unsigned long or float values.
18	**Base Address**	• False (unchecked):The Modbus device address numbering starts at 1 which makes the starting address sent in the Modbus frames request will have one subtracted. • True (checked): The Modbus device address numbering starts at 0 and the starting address included in the Modbus frame request will remain the same.
19	**Reg Base Address**	This option is used with bits within registers referenced as Boolean • False (unchecked): The first bit within register begins at one • True (checked): The first bit within register begins at zero
20	**H Reg Bit Mask**	This option is used to write in a bit within register using the Boolean data type with holding registers. • False (unchecked): The OPC Server will use a read /Write operation to update the bit of interest. • True (checked): The OPC Server will use function 22 to update the bit of interest.
21	**Only Single Reg Write**	This option is related to 32 bits and 64 bits OPC tags. • False (unchecked): The OPC Server will use the function 16 to write to multiple registers. • True (checked): The OPC Server will use function 06 to write into multiple registers.
22	**Only Single Coil Write**	• False (unchecked): The OPC Server will use the function 15 to write to multiple coils. • True (checked): The OPC Server will use function 05 to write into multiple registers.
23	**Double Register**	This check box determines how floating points and long integer values are handled:

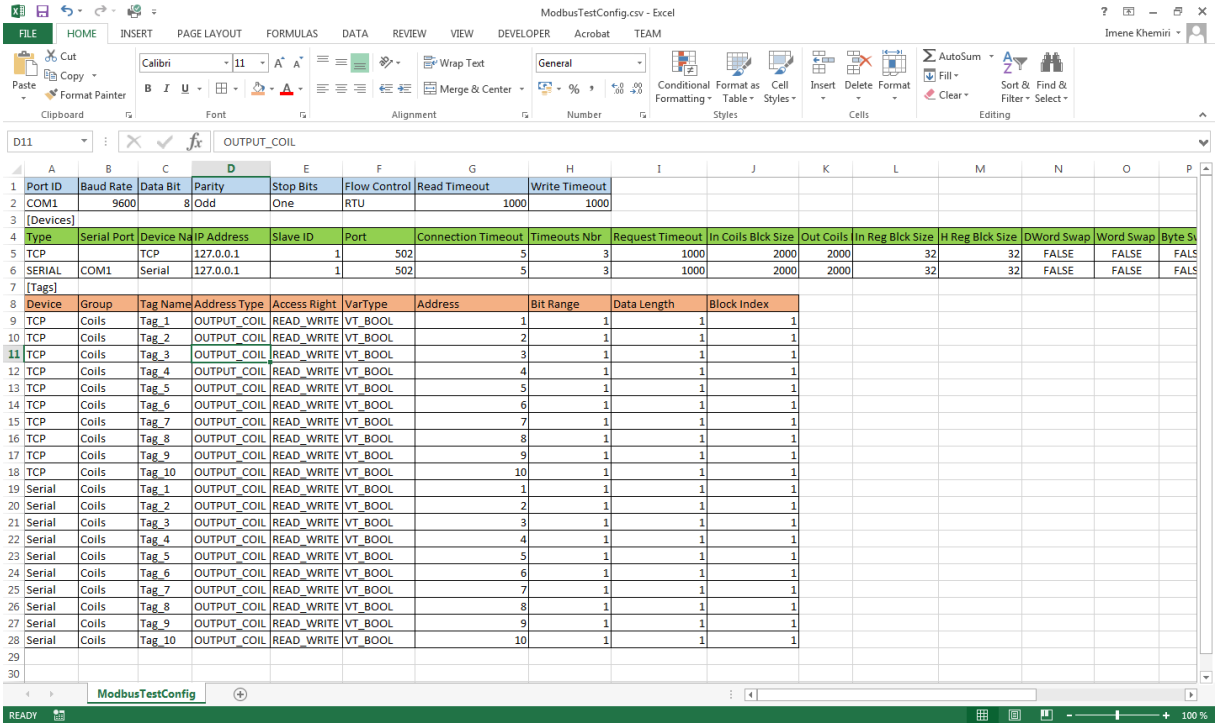
		• When unchecked, it means that the Modbus device allocates two 16 bit registers for containing a floating point or long integer value. • When checked, it means that the device allocates one 32 bit register for the value.
24	**Auto Demotion**	When checked, This option allows to demote a device for a specific period of time when communication failures reaches an already configured limited number.
25	**Failures Nbr**	The number of successive failures before demoting the device
26	**Demotion Period**	During this period ,no read request will be sent to the device
27	**Discard Write**	When checked, no write request will be sent to the device during the demotion period
28	**Deactivate Tags**	• False (unchecked): When illegal data address error is occurred with a block addresses. The OPC Server remains sending read requests to the Modbus device • True (checked): When illegal data address error is occurred with a block addresses. The OPC Server stops sending read requests to the Modbus device
29	**Tr Mode**	The transmission mode (RTU/ASCII)
30	**Synchronous Enabled**	• When unchecked, it means that the asynchronous communication will be used read/write data from/to Modbus device • When checked, it means that the synchronous communication will be used read/write data from/to Modbus device
31	**Diagnostic Enabled**	• When unchecked, it means that the diagnostics function will be used to check the device communication status. • When checked, it means that the diagnostics function will not be used to check the device communication status
32	**Request Timeout Nbr**	The allowed request timeouts number before starting the reconnection procedure to the device

Table 16: Devices Configuration Section Format

The table below describes the fields of the tags configuration section.

	Field	Description
1	**Device**	The device name
2	**Group**	The group name
3	**Tag Name**	The tag name
4	**AddressType**	The Modbus address type, which can be: • Discrete Inputs • Coils • Input Registers • Holding Registers ***Refer to the Location Type table for more details.***
5	**Access Right**	The access right, which can be read only or read/write
6	**VarType**	• The type of data, which can be: • VT_I2 • VT_I4 • VT_R4 • VT_UI2 • VT_UI4 • VT_BSTR • VT_BOOL
7	**Address**	The address of the configured tag. The value should be between 1 and 65534.
8	**Bit Range**	It denotes the range of a bit in one register.
9	**Data Length**	The Bytes number of the corresponding Var Type
10	**Block ID**	It denotes the block index to which the tag is belonging. A block represents a group of adjacent addresses sent and received in one message.

Table 17: Tags Configuration Section Format



ModbusTestConfig.csv - Excel

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Port ID	Baud Rate	Data Bit	Parity	Stop Bits	Flow Control	Read Timeout	Write Timeout								
2	COM1	9600	8	Odd	One	RTU	1000	1000								
3	[Devices]															
4	Type	Serial Port	Device Na	IP Address	Slave ID	Port	Connection Timeout	Timeouts Nbr	Request Timeout	In Coils Blck Size	Out Coils	In Reg Blck Size	H Reg Blck Size	DWord Swap	Word Swap	Byte Sv
5	TCP		TCP	127.0.0.1	1	502	5	3	1000	2000	2000	32	32	FALSE	FALSE	FALS
6	SERIAL	COM1	Serial	127.0.0.1	1	502	5	3	1000	2000	2000	32	32	FALSE	FALSE	FALS
7	[Tags]															
8	Device	Group	Tag Name	Address Type	Access Right	VarType	Address	Bit Range	Data Length	Block Index						
9	TCP	Coils	Tag_1	OUTPUT_COIL	READ_WRITE	VT_BOOL	1	1	1	1						
10	TCP	Coils	Tag_2	OUTPUT_COIL	READ_WRITE	VT_BOOL	2	1	1	1						
11	TCP	Coils	Tag_3	OUTPUT_COIL	READ_WRITE	VT_BOOL	3	1	1	1						
12	TCP	Coils	Tag_4	OUTPUT_COIL	READ_WRITE	VT_BOOL	4	1	1	1						
13	TCP	Coils	Tag_5	OUTPUT_COIL	READ_WRITE	VT_BOOL	5	1	1	1						
14	TCP	Coils	Tag_6	OUTPUT_COIL	READ_WRITE	VT_BOOL	6	1	1	1						
15	TCP	Coils	Tag_7	OUTPUT_COIL	READ_WRITE	VT_BOOL	7	1	1	1						
16	TCP	Coils	Tag_8	OUTPUT_COIL	READ_WRITE	VT_BOOL	8	1	1	1						
17	TCP	Coils	Tag_9	OUTPUT_COIL	READ_WRITE	VT_BOOL	9	1	1	1						
18	TCP	Coils	Tag_10	OUTPUT_COIL	READ_WRITE	VT_BOOL	10	1	1	1						
19	Serial	Coils	Tag_1	OUTPUT_COIL	READ_WRITE	VT_BOOL	1	1	1	1						
20	Serial	Coils	Tag_2	OUTPUT_COIL	READ_WRITE	VT_BOOL	2	1	1	1						
21	Serial	Coils	Tag_3	OUTPUT_COIL	READ_WRITE	VT_BOOL	3	1	1	1						
22	Serial	Coils	Tag_4	OUTPUT_COIL	READ_WRITE	VT_BOOL	4	1	1	1						
23	Serial	Coils	Tag_5	OUTPUT_COIL	READ_WRITE	VT_BOOL	5	1	1	1						
24	Serial	Coils	Tag_6	OUTPUT_COIL	READ_WRITE	VT_BOOL	6	1	1	1						
25	Serial	Coils	Tag_7	OUTPUT_COIL	READ_WRITE	VT_BOOL	7	1	1	1						
26	Serial	Coils	Tag_8	OUTPUT_COIL	READ_WRITE	VT_BOOL	8	1	1	1						
27	Serial	Coils	Tag_9	OUTPUT_COIL	READ_WRITE	VT_BOOL	9	1	1	1						
28	Serial	Coils	Tag_10	OUTPUT_COIL	READ_WRITE	VT_BOOL	10	1	1	1						
29																
30																

Figure 54: Configuration File Example

After saving your configuration, you need to set the default configuration, which will be loaded automatically at the OPC Server for Modbus Service start-up. To define a default configuration, click the Define button available in the Settings Menu.

6. Errors Management

The OPC Server for Modbus handles both OPC and Modbus protocols related errors. All errors are tracked in the service log file and in the log messages view in the graphical user interface, as illustrated below:

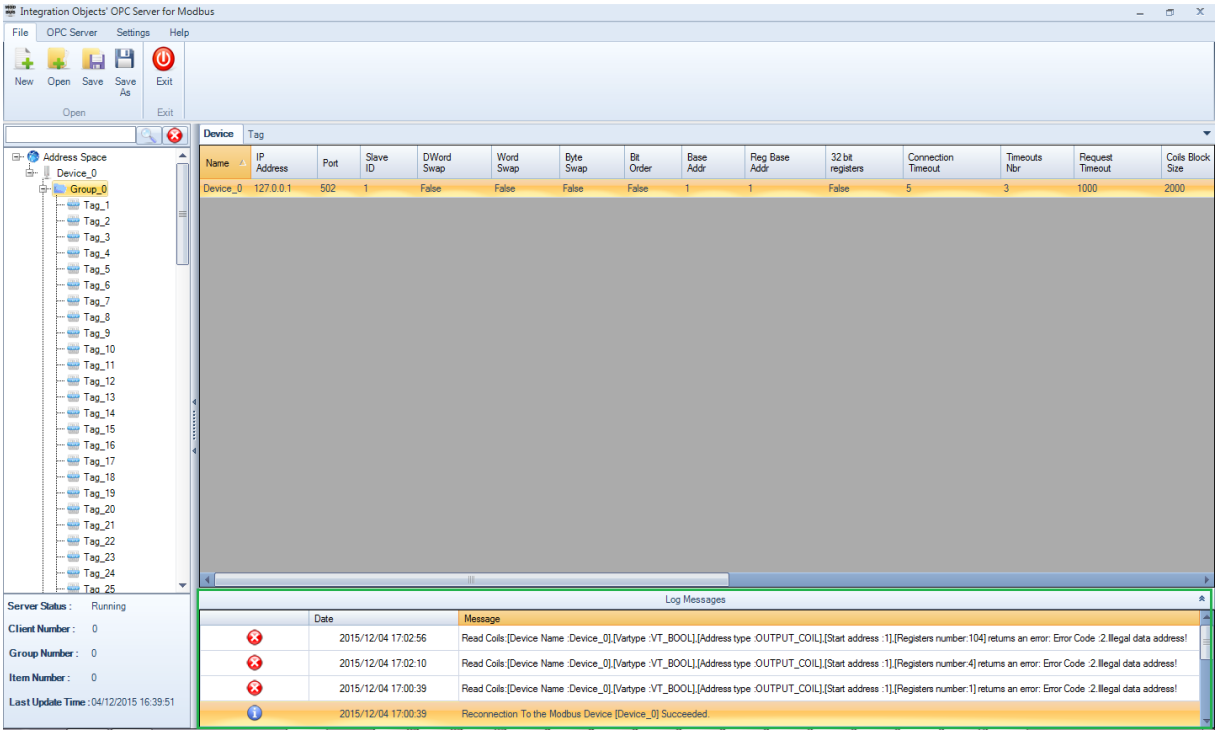


Figure 55: OPC Server for Modbus Log Messages

When sending the Modbus request to slave devices via the OPC Server for Modbus exception responses may happen and error codes will be sent in the response message and will be recorded in the software log file. These errors are described in the table below.

Code Dec/Hex	Name	Meaning
01/0x01	Illegal Function	The function code received in the query is not an allowable action for the server (or slave). This may be because the function code is only applicable to newer devices, and was not implemented in the unit selected. It could also indicate that the server (or slave) is in the wrong state to process a request of this type, for example because it is not configured and is being asked to return register values.
02/0x02	Illegal Data Address	The data address received in the query is not an allowable address for the server (or slave). More specifically, the combination of reference number and transfer length is invalid. For a controller with 100 registers a request of offset 96 and a length of 5 will generate exception 02.
03/0x03	Illegal Data Value	The value contained in the query data field is not an allowable value for the server (or slave). This indicates a fault in the structure of the remainder of a complex request, such as that the implied length being incorrect. It specifically does NOT

		mean that a data item, submitted for storage in a register, has a value outside the expectation of the application program, since the MODBUS protocol is unaware of the significance of any particular value of any particular register.
04/0x04	Failure In Associated Device	An unrecoverable error occurred while the server (or slave) was attempting to perform the requested action.
05/0x05	Acknowledge	Specialized in conjunction with programming commands. The server (or slave) has accepted the request and is processing it, but long duration of time will be required to do so. This response is returned to prevent a timeout error from occurring in the client (or master). The client (or master) can next issue a poll program complete message to determine if processing is completed.
06/0x06	Busy, Rejected Message	Specialized use in conjunction with programming commands. The server (or slave) is engaged in processing a long-duration program command. The client (or master) should retransmit the message later when the server (or slave) is free.
10/0x0A	Gateway Path Unavailable	Specialized use in conjunction with gateways. It indicates that the gateway was unable to allocate an internal communication path from the input port to the out port for processing the request.
11/0x0B	Gateway Target Device Failed to Respond	Specialized use in conjunction with gateways. It indicates that no response was obtained from the target device. Usually means that the device is not present on the network.

Table 18: Modbus Response Exceptions

7. Connection to the OPC Server

Once the OPC Server is configured, the next step will be the connection to the OPC Server and reading/writing data. To do so, you only need to:

1. Launch your OPC DA Client as administrator.
2. Enter the IP address of the machine where the OPC Server is installed
3. Connect to the OPC Server with the following progID: "IntegrationObjects.OPC.ModBus.1"
4. Add a group and select the items to be read

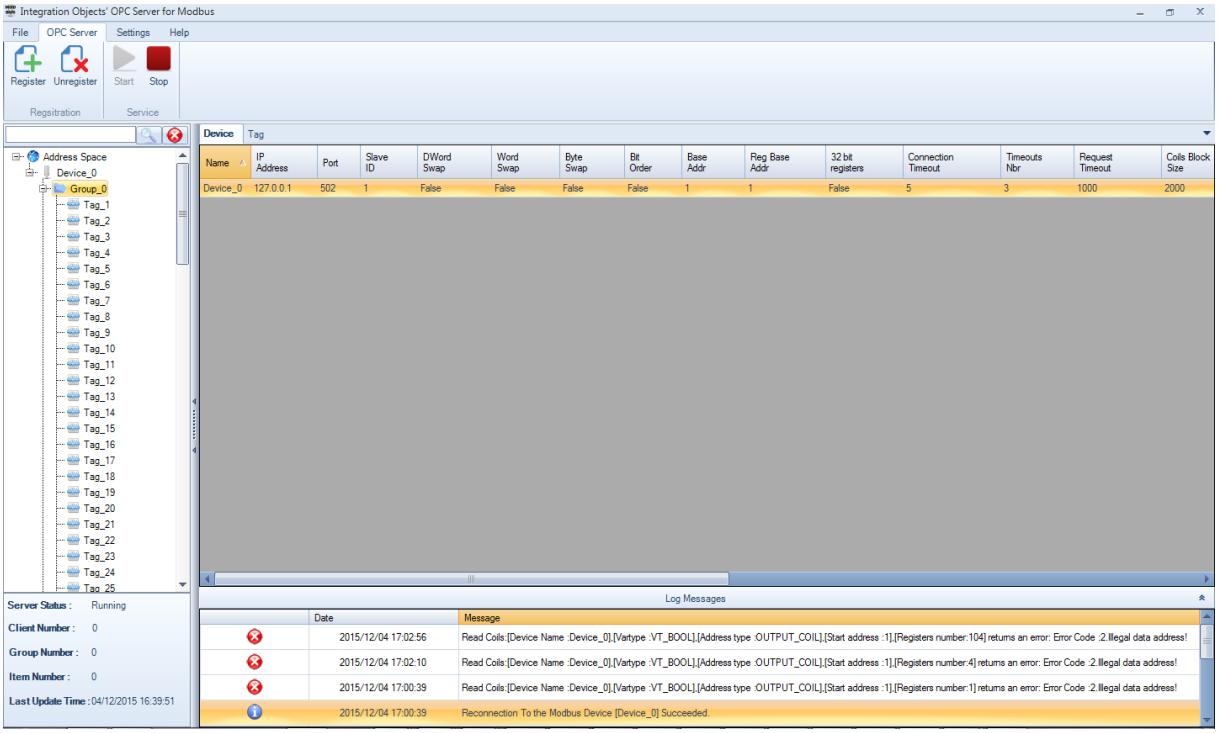


Figure 56: OPC DA Client Connected to the OPC Server for Modbus

In case the local connection to the OPC Server for Modbus failed due to an access deny you need to follow the steps below:

1. Open the windows service manager
2. Select the Integration Objects' OPC Server for Modbus Service.
3. Right click and select the Log on tab.
4. Check the "This account" radio button.
5. Enter your administrator account credentials as shown in the following figure:

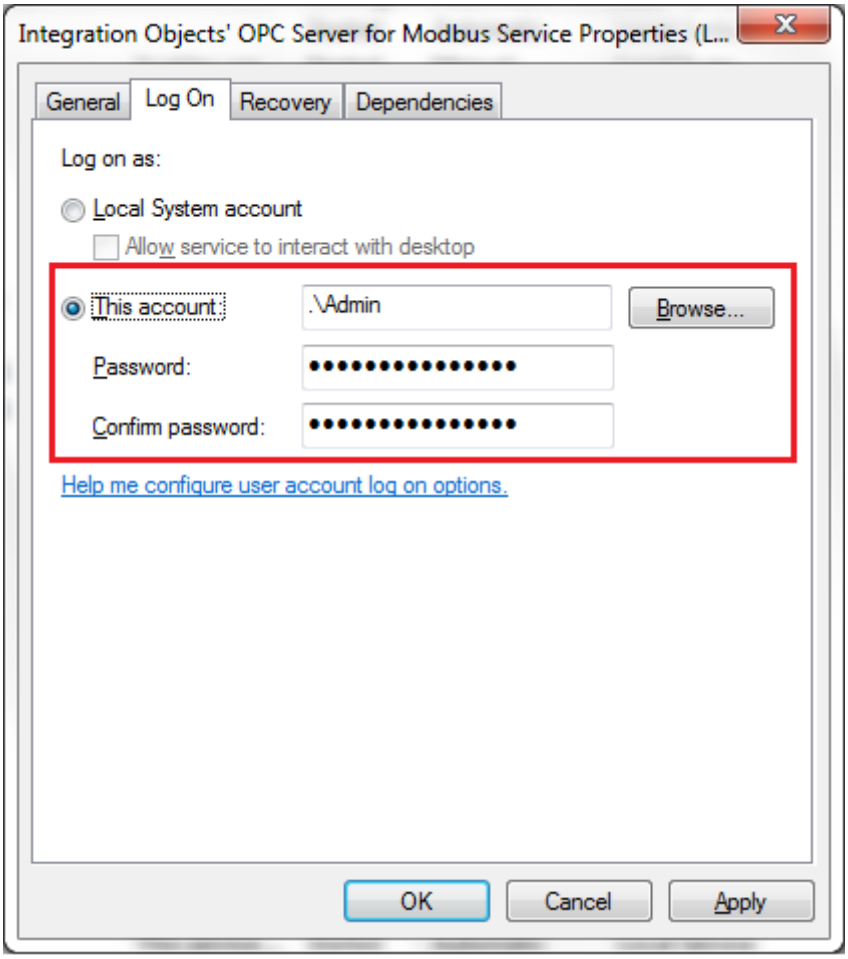


Figure 57: OPC Server for Modbus Service Administrator LogON

6. Click the OK button.

TROUBLESHOOTING

Case 1: Cannot launch the OPC Server for Modbus

If you are using an evaluation license, you should first check the license validity by launching the License Authorization tool. You can start it directly from the startup menu:

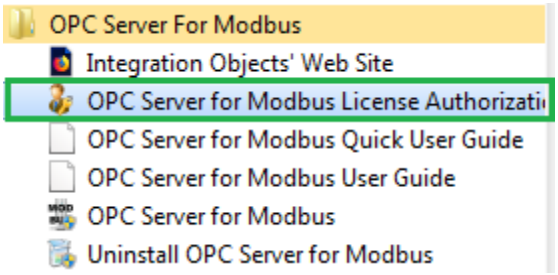


Figure 58: Open License Authorization Tool

If the License Authorization tool shows that the demo has expired and you want to activate it using your full activation license, you should in this case follow the steps below:

1. Run the License Authorization tool using an administrator account

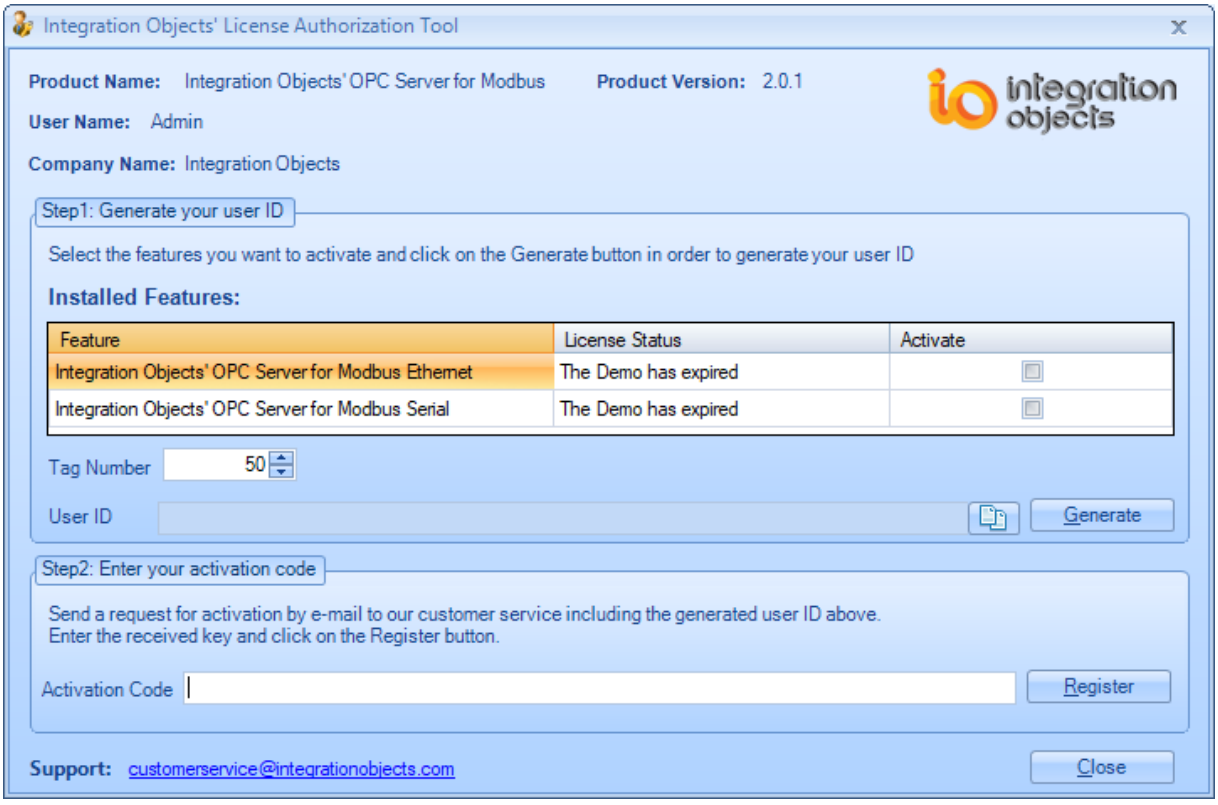


Figure 59: Demo License Expired

2. Choose the features you want to activate

3. Click the **Generate** button

4. Copy and send the User ID to the sales team {sales@integrationobjects.com} so they can generate the dedicated activation code.
5. Enter the given **Activation Code**

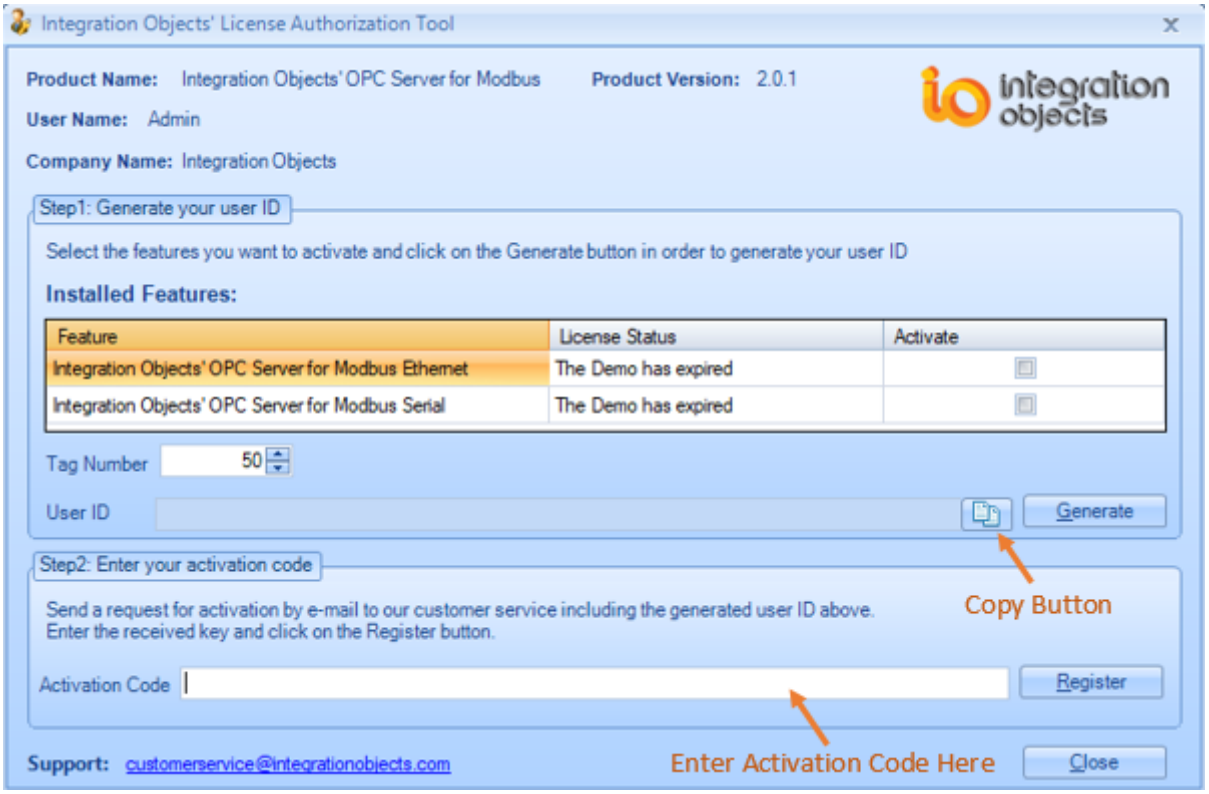


Figure 60: Activate License

6. Click the **Register** button

Case 2: Cannot start the OPC Server for Modbus Service?

In case the local connection to the OPC Server for Modbus failed due to an access deny you need to follow the steps below:

1. Open the windows service manager
2. Select the Integration Objects' OPC Server for Modbus Service.
3. Right click and select the **Log On** tab.
4. Check the "This account" radio button.
5. Enter your administrator account credentials as shown in the following figure:

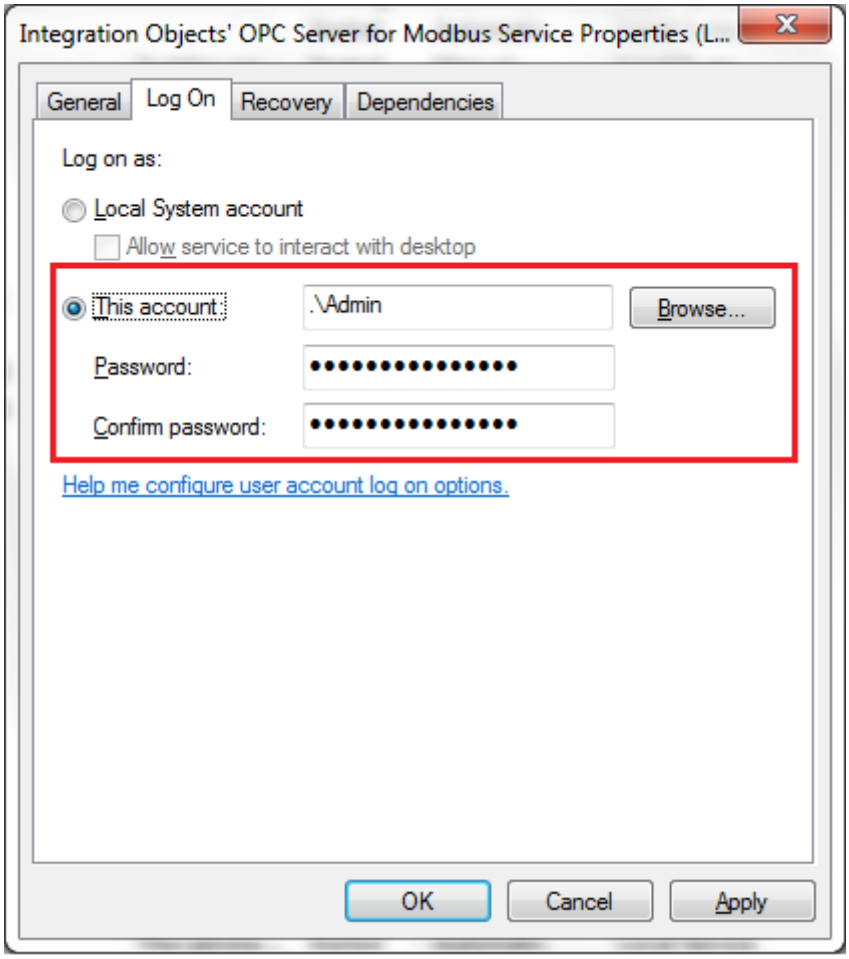


Figure 61: OPC Server for Modbus Service Properties

6. Click the **OK** button.

Case 3: The OPC Server ProgID cannot be displayed by the OPC Client.

In case the OPC Client is not able to display the server ProgID then you need to check the following items:

- Make sure that the OPC Core Components are installed in your machine
- Make sure that a user with local administrative rights installs the OPC server so that it is able to create the necessary entries in the registry.
- Ensure that the OPCEnum is registered as a service, DCOM permissions are configured properly, and the OPCEnum service is running.
- Make sure that the client is running as a user that has read access to the registry

Case 4: Why does the connection to the TCP IP Modbus device fail?

In case the connection to the TCP\IP Modbus device fails, you need to check the following items:

- The port is not locked by another Modbus master
- The IP address and the ID of the Modbus slave are correct

Case 5: Why does the connection to the serial Modbus device fail?

In case the connection to the serial Modbus device fails, you need to check that The COM port is not locked by another Modbus master.

Case 6: Why do the read/write from/to the serial Modbus device timeout?

In case the reads and writes time out, you need to check your configuration and make sure it matches the configuration of your Modbus device for the following parameters:

- Device ID
- Device Transmission Mode
- COM Port Baud Rate
- COM Port Data Bit
- COM Port Parity
- COM Port Stop Bits
- COM Port Flow Control

Case 7: Why do I get wrong values from the Modbus device?

In case the reads and writes values are wrong, you need to check your configuration and make sure that the data access settings and the swapping mode match the configuration of your Modbus device.

For additional information on this guide, questions or problems to report, please contact:

Offices
- Americas: +1 713 609 9208
- Europe-Africa-Middle East: +216 71 195 360

Email
- Support Services: customerservice@integrationobjects.com
- Sales: sales@integrationobjects.com

To find out how you can benefit from other Integration Objects products and custom-designed solutions, please visit our website: www.integrationobjects.com