

Integration Objects'

OPC DA Access for Modbus Devices

OPC Server for Modbus
Version 2.0Rev.1

QUICK USER GUIDE

Integration Objects' OPC Server for Modbus Quick User's Guide Version 2.0Rev.1
Published July 2021
Copyright © 2015-2021 Integration Objects

ABOUT THIS USER GUIDE

This document is a quick step by step guide on how to install and use the OPC Server for Modbus.

INSTALLATION PRE-REQUISITES

In order to properly run the OPC Server for Modbus, install these software components on the target system:

- The OPC core components 3.00, which consist 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 4.5 or higher.

INSTALLING THE OPC SERVER FOR MODBUS

To install the OPC Server for Modbus, right-click on the installation executable and select “Run as administrator” from the displayed menu. The prompted wizard will then take you through the different installation steps.

To start the OPC Server for Modbus, you need to follow the below steps:

1. Go to **Start => Programs => Integration Objects => OPC Server for Modbus => OPC Server for Modbus**
2. Start the OPC Server for Modbus.

ADDING A NEW DEVICE

To add a new device, right click on the Address Space node, select **New Device** as illustrated below.

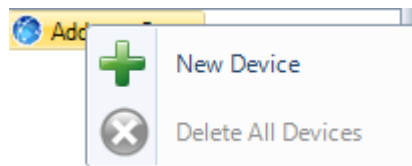


Figure 1: Add a New Device

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

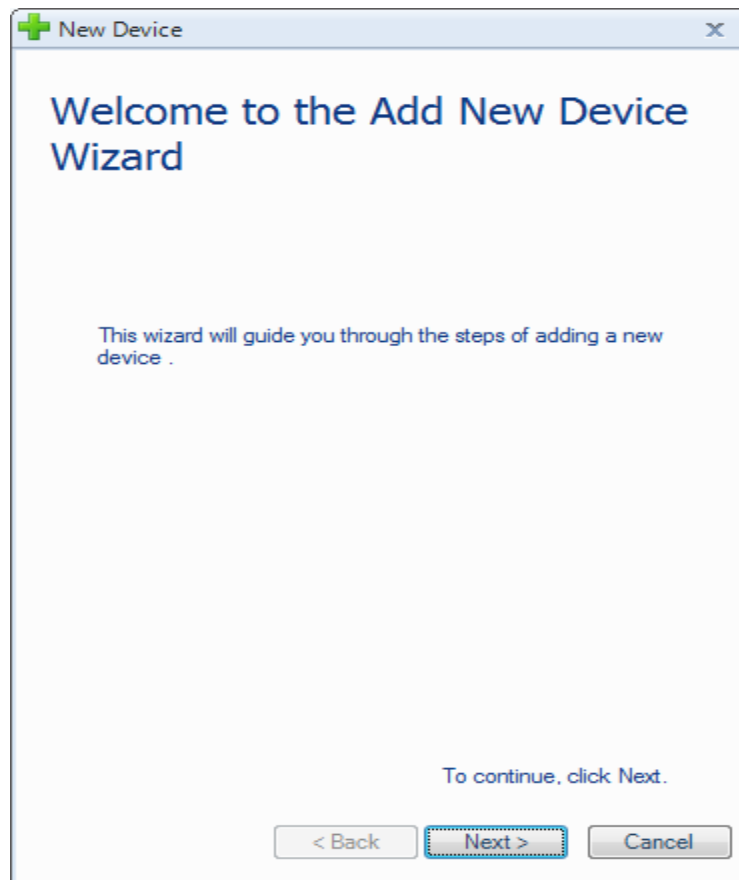


Figure 2: Add New Device Wizard

- Step 1: Connection Parameters

Click the **Next** button. The following window will be displayed:

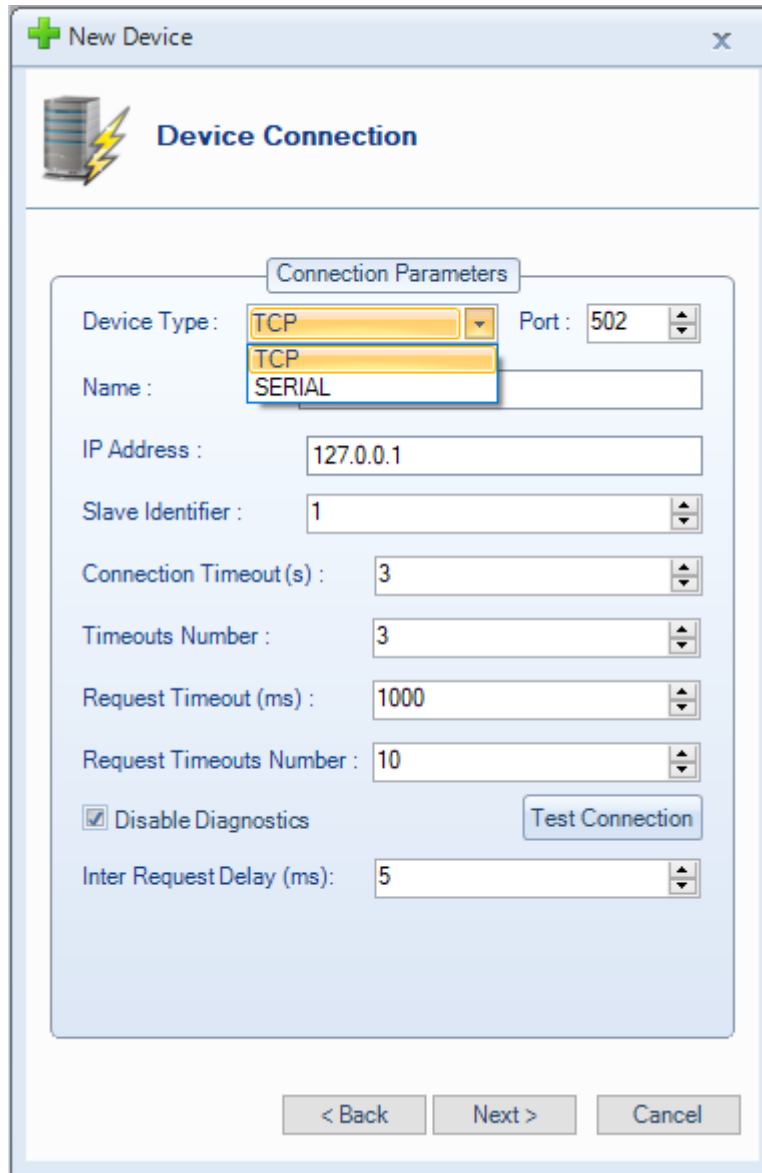


Figure 3: Select Device Connection Configuration

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.

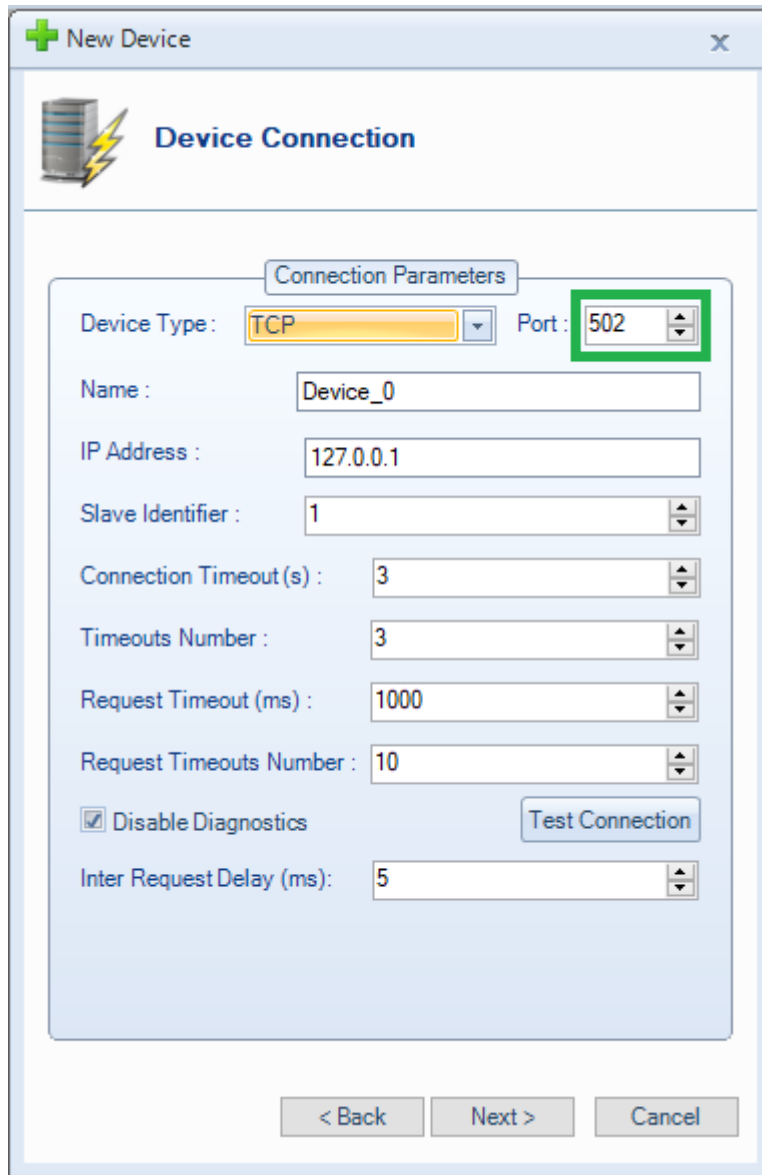


Figure 4: Select the TCP/IP Listening Modbus Port

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

The screenshot shows a 'New Device' dialog box with a 'Device Connection' tab. The 'Device Type' is set to 'SERIAL'. The 'COM Port' is set to 'COM1'. The 'Port Settings' button is highlighted with a green box. Other settings include: Name: Device_2, Slave Identifier: 1, Connection Timeout (s): 3, Timeouts Number: 3, Request Timeout (ms): 1000, Request Timeouts Number: 10, Inter Request Delay (ms): 5, Response Delay (ms): 0, and Transmission Mode: RTU. A 'Test Connection' button is also visible.

Figure 5: Select the Serial COM Modbus Port

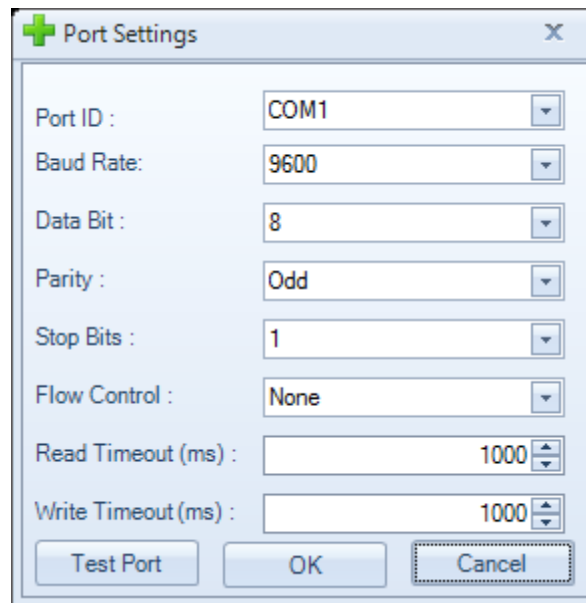


Figure 6: Select the COM Port Settings

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 1: COM Port Settings Parameters

After choosing the device type, enter the connection parameters.

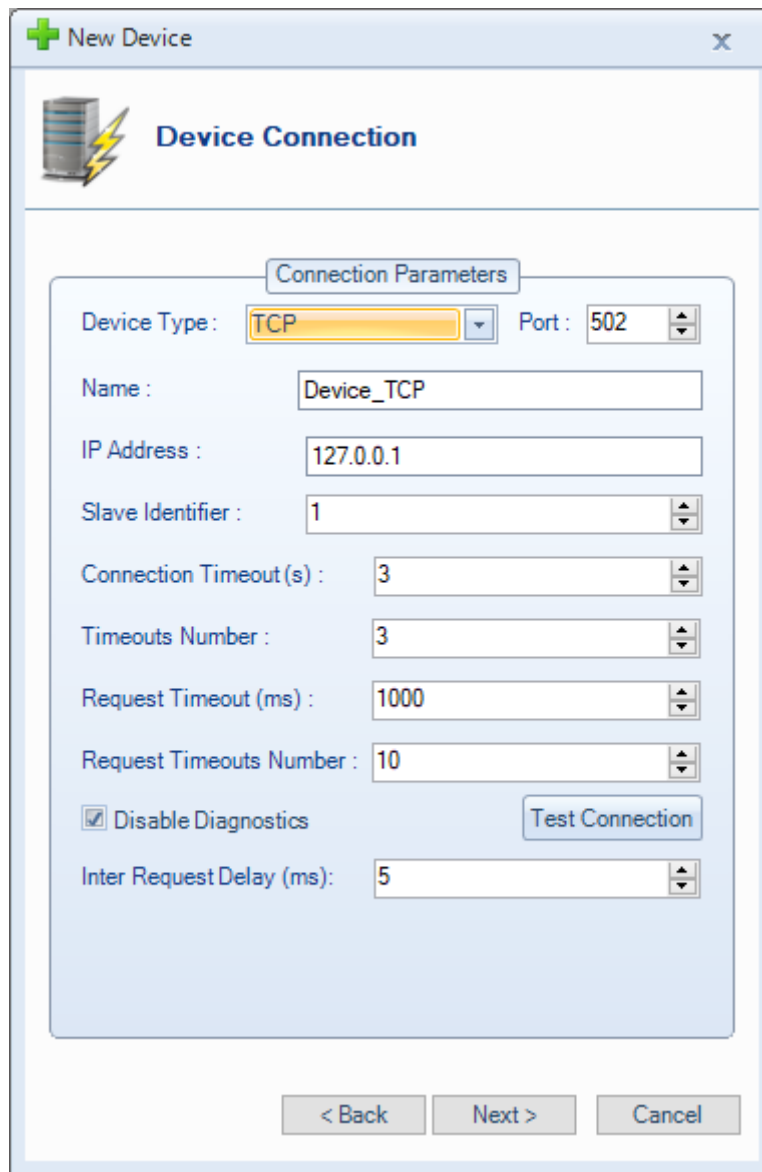


Figure 7: Select the TCP/IP Device Connection Parameters

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	<ul style="list-style-type: none"> • 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 2: TCP/IP Device Connection Parameters

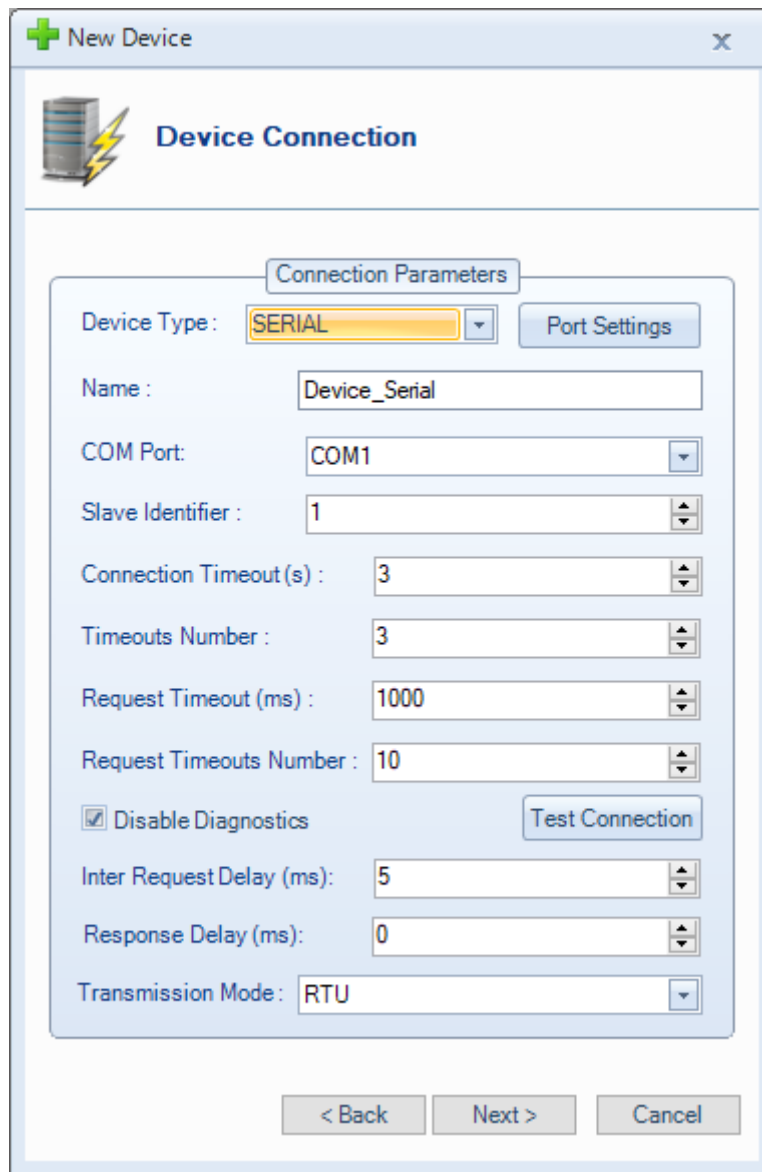


Figure 8: Select the Serial Device Connection Parameters

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	<ul style="list-style-type: none"> • 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 3: 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:

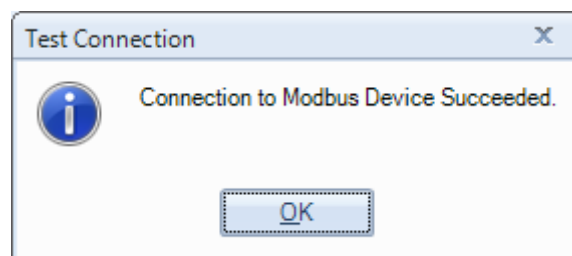


Figure 9: Test Connection with Modbus Device

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.

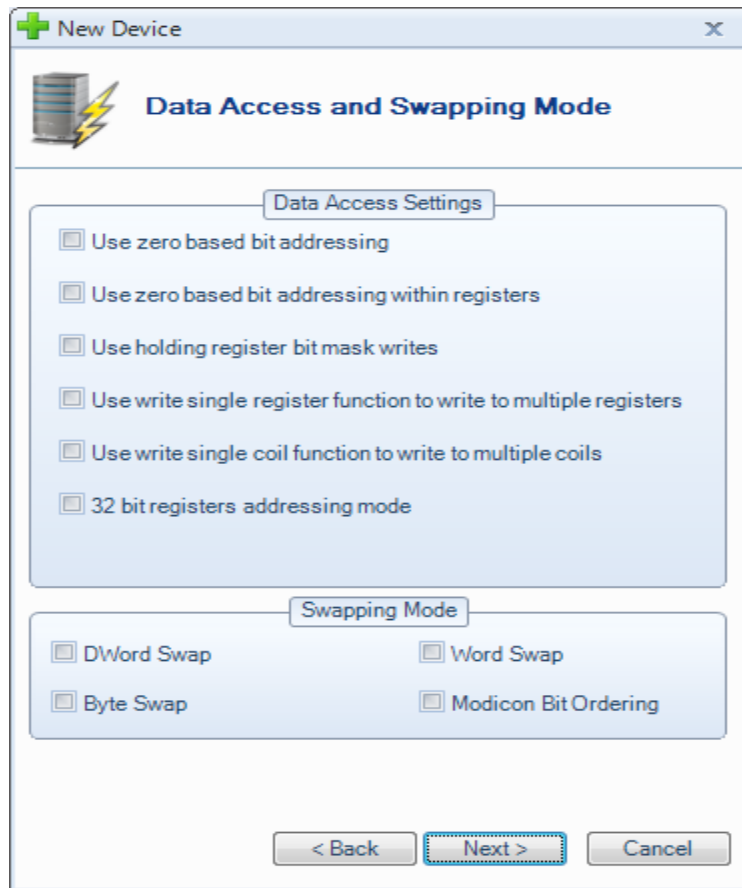


Figure 10: Select Device Data Access & Swapping Mode Parameters

The data access parameters are described in the below table:

Parameter	Description
Use zero based bit addressing	<ul style="list-style-type: none"> 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

	<p>included in the Modbus frame request will remain the same.</p>
Use zero based bit addressing within registers	<p>This option is used with bits within registers referenced as Boolean</p> <ul style="list-style-type: none"> • 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	<p>This option is used to write in a bit within register using the boolean datatype with holding registers.</p> <ul style="list-style-type: none"> • 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	<p>This option is related to 32 bits and 64 bits OPC tags.</p> <ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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.
32 bit Registers Addressing Mode	<p>This check box determines how floating points and long integer values are handled:</p> <ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • When unchecked, it means that the asynchronous communication will be used read/write data from/to Modbus device

	<ul style="list-style-type: none"> When checked, it means that the synchronous communication will be used read/write data from/to Modbus device
--	--

Table 4: Device Data Access Parameters

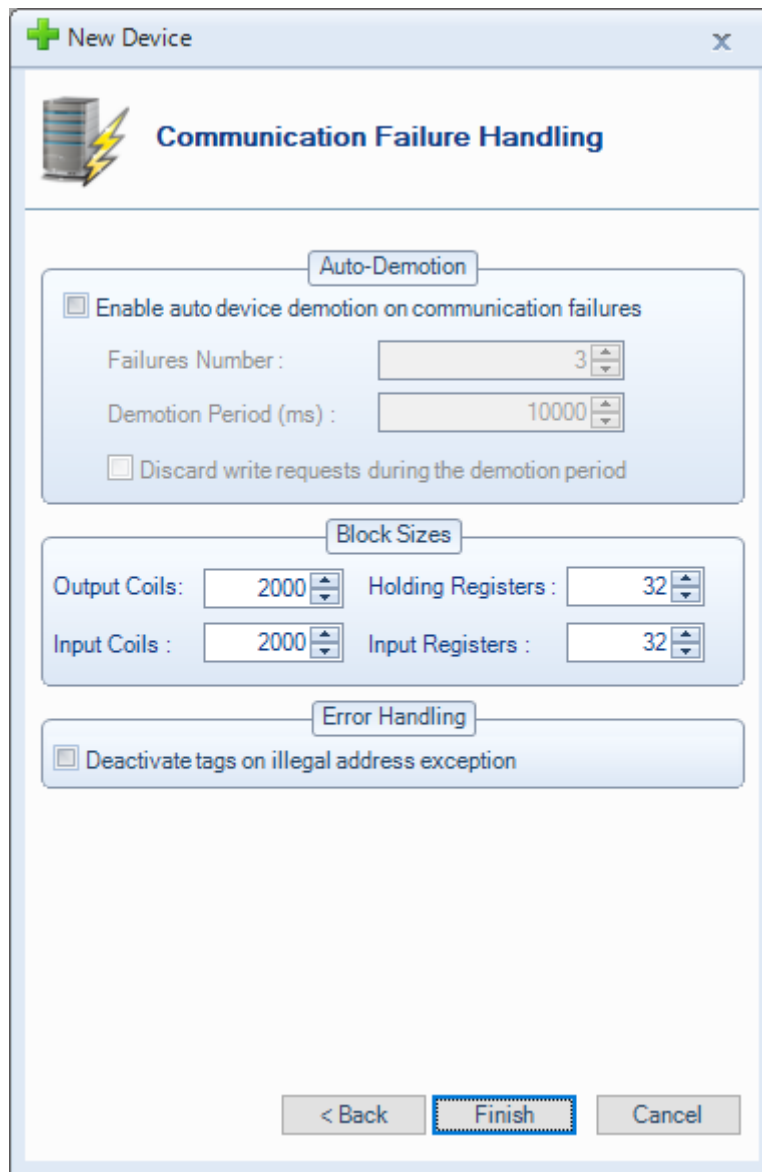
You can also select the swapping mode with reference to the below description:

Parameter	Description
DWord Swap	<ul style="list-style-type: none"> False (unchecked): Higher dword is sent/received first for double values. True (checked): Lower dword is sent/received first for double values.
Word Swap	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 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 5: 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.



Communication Failure Handling

Auto-Demotion

Enable auto device demotion on communication failures

Failures Number : 3

Demotion Period (ms) : 10000

Discard write requests during the demotion period

Block Sizes

Output Coils : 2000 Holding Registers : 32

Input Coils : 2000 Input Registers : 32

Error Handling

Deactivate tags on illegal address exception

< Back Finish Cancel

Figure 11: Select Communication Failure Handling Parameters

In addition to the connection parameters, the communication failure is also 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 6: 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 7: 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	<ul style="list-style-type: none"> 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 8: Device Error Handling Parameters

Step 4: Finish Adding New Device

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

ADDING A NEW 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.

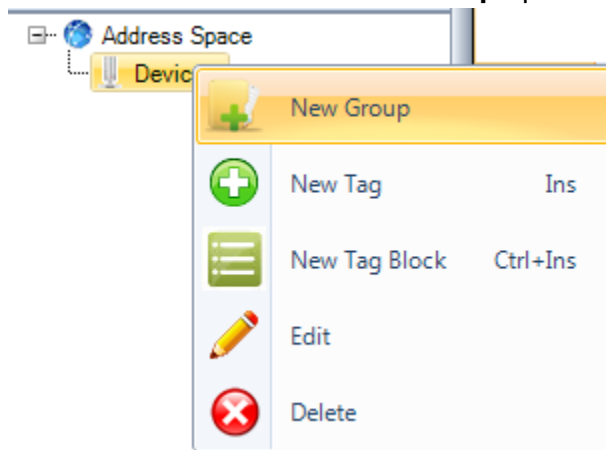


Figure 12: New Group

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

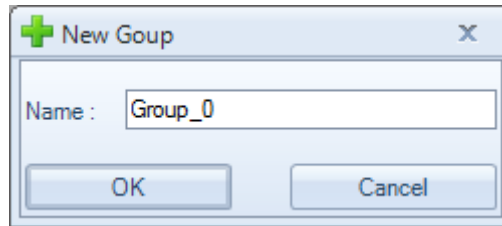


Figure 13: New Group Dialog Box

ADDING 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.

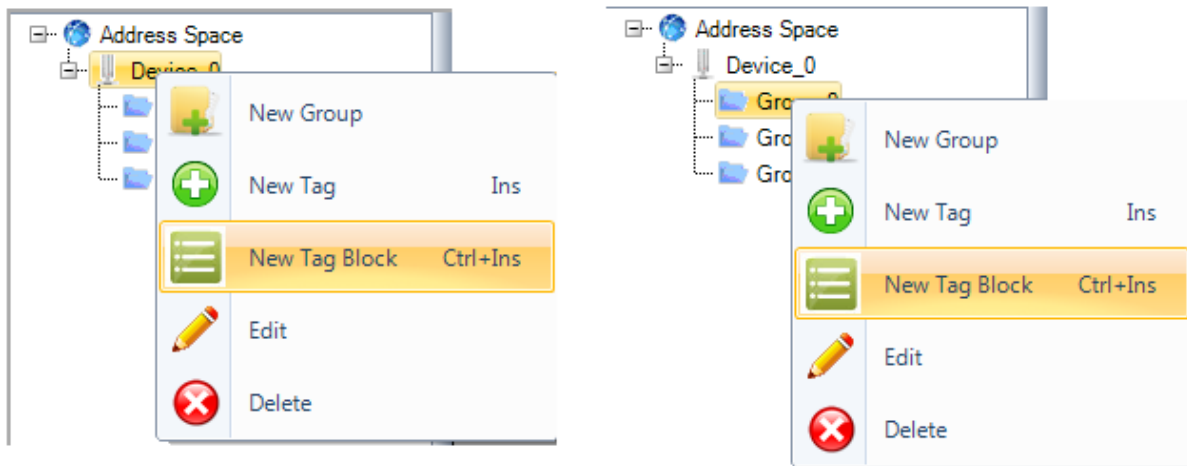


Figure 14: New Tag Block

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

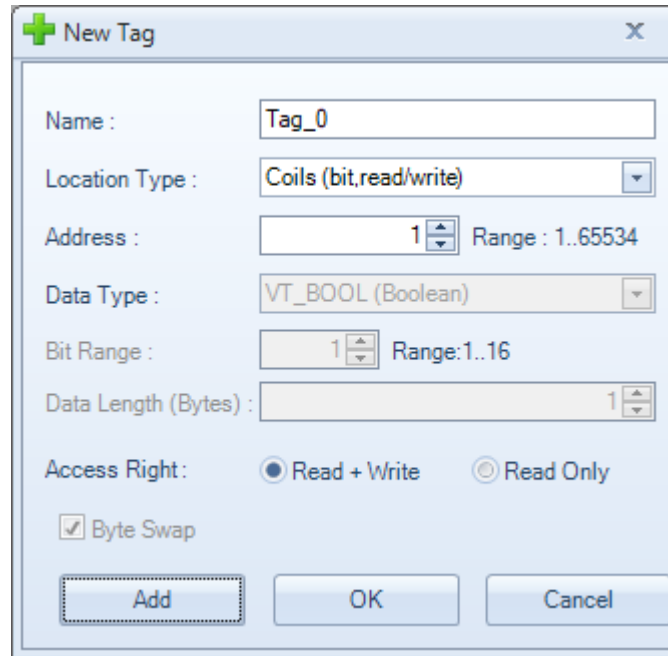


Figure 15: New Tag Block Dialog Box

Parameter	Description
Base Tag Name	The pattern used to generate the name of the tags
Location Type	<p>The Modbus address type, which can be:</p> <ul style="list-style-type: none"> • Coils • Discrete Inputs • Holding Registers • Input Registers <p><i>Refer to the Location Type table for more details.</i></p>
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	<p>The type of data, which can be:</p> <ul style="list-style-type: none"> • 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

	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • False (unchecked): Higher byte is sent/received first for string values. • True (checked): Lower byte is sent/received first for string values.

Table 9: Tag Block Properties

SAVING THE CONFIGURATION

You can 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.

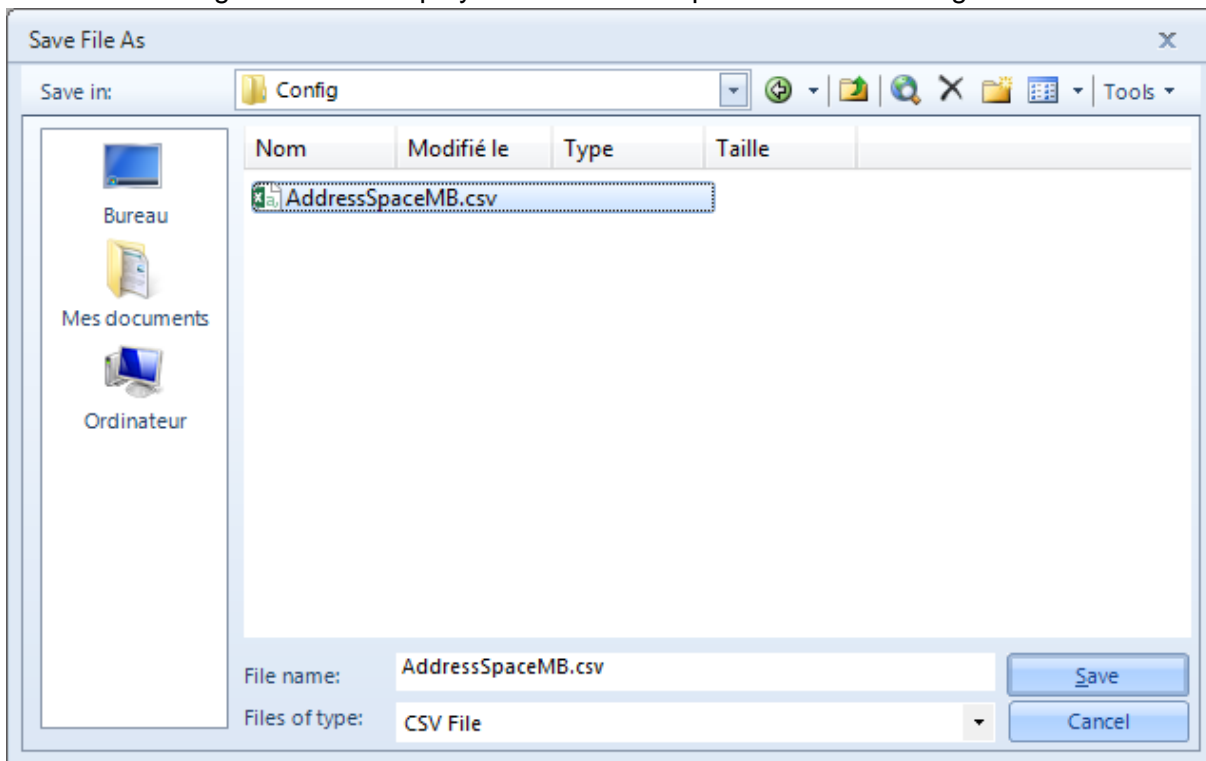


Figure 16: Save File As Dialog Box

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 10: 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	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	<p>This option is used with bits within registers referenced as Boolean</p> <ul style="list-style-type: none"> • 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.

		<ul style="list-style-type: none"> • 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	<p>This option is related to 32 bits and 64 bits OPC tags.</p> <ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	<p>This check box determines how floating points and long integer values are handled:</p> <ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • When unchecked, it means that the asynchronous communication will be used read/write data from/to Modbus device

		<ul style="list-style-type: none"> When checked, it means that the synchronous communication will be used read/write data from/to Modbus device
31	Diagnostic Enabled	<ul style="list-style-type: none"> 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 11: Devices Configuration Section Format

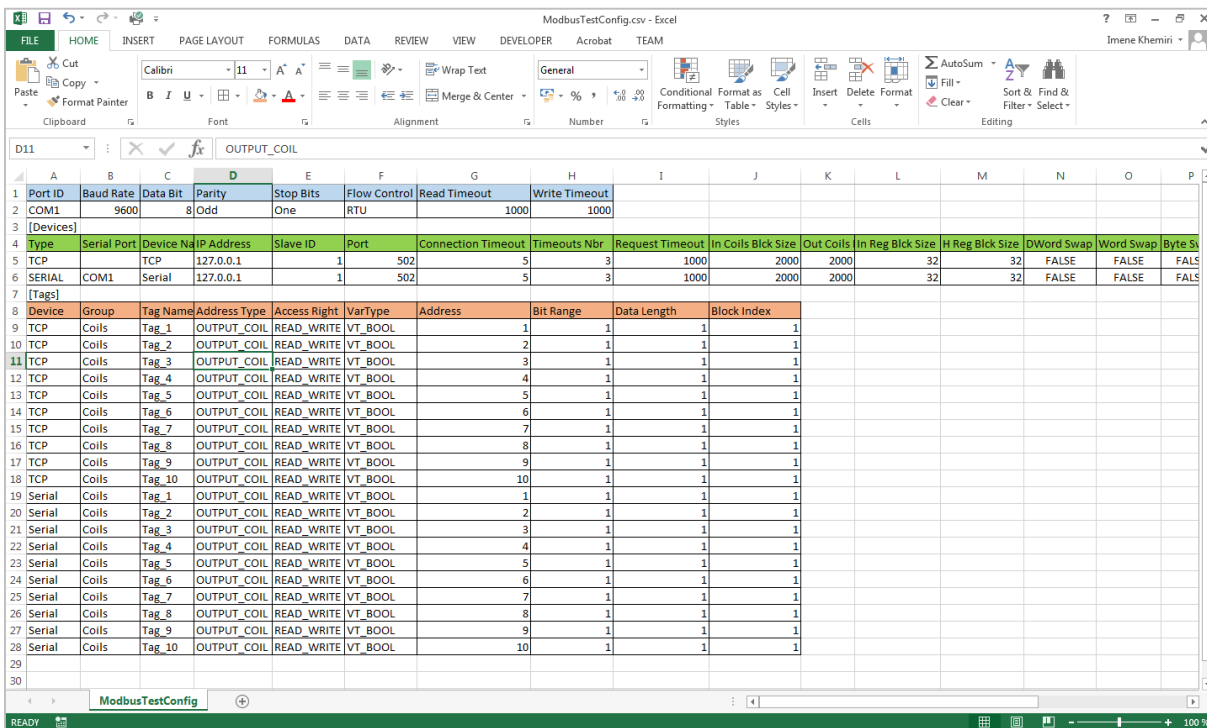


Figure 17: Configuration File Example

SETTING THE DEFAULT CONFIGURATION

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.

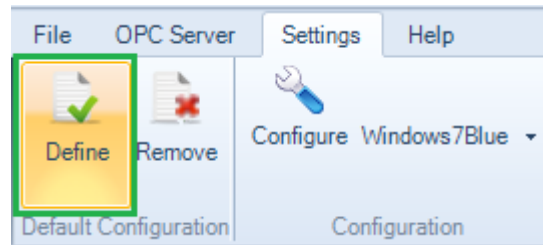


Figure 18: Set Default Configuration

STARTING THE OPC SERVER FOR MODBUS SERVICE

After setting the default configuration, select the OPC Server menu and click on the start button menu to start the Integration Objects' OPC Server for Modbus service.

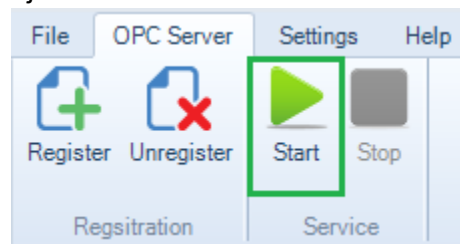


Figure 19: Start the OPC Server for Modbus Service

CONNECTING TO THE OPC SERVER FOR MODBUS

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:

Launch your OPC DA Client as administrator.

1. Enter the IP address of the machine where the OPC Server is installed
2. Connect to the OPC Server with the following ProgID:
IntegrationObjects.OPC.ModBus.1"
3. Add a group and select the items to be read

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](mailto: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 us on the Internet:

Online

- www.integrationobjects.com