Devices and Interfaces

The devices in FactoryStudio are any live real-time data source. Typically, a device is a PLC, another FactoryStudio project, an OPC server, a PI System, or any other equipment that has a communication protocol.

Data sources are connected to FactoryStudio through **Channels**. Each channel has an **interface** type (e.g. RS-232, TCP/IP) and a device-specific **pro tocol**. A channel may access multiple stations (e.g. devices) using a common protocol.

Each stations is called a **Node**. Each node has one or more data **Points**. Data points provide the specific data values for tags to access. Each data point is bound to a specific tag.

Finally, each data point is associated with an **Access Type**. This defines the rules for reading and writing values for this data point, such as the polling rate, whether a read is performed on startup, and whether unsolicited input is accepted.

In Summary, the Device Module configuration is executed in 3 steps:

1) Define the equipment and protocols the project will use, in Edit Devices Channels.

2) Defined the Nodes or the PLC stations related to each channel, in Edit Devices Nodes

3) Map the tags in the data model to addresses in the devices, in Edit Devices Points.

Optionally, you can customize or create new AccessTypes by mapping groups of tags, that similar communication requirements, to the same AccessType.

In order to expedite the configuration, FactoryStudio provides many Import Wizards that automatically create the Tag and Device mapping using the PLC configuration or another available source of information. Link to current list of Import Wizards.

Channels

Channels in FactoryStudio are the protocols you use to communicate with your PLCs. Many built-in protocols are available. You must set up a channel for each protocol you need to use.

To configure channels:

- Go to **Edit > Devices > Channels**.
- Click **Create New**.
- The Create New Channel window displays.
- Enter or select information, as needed.

Column	Description
Channel Name	Enter a name for the channel. The system lets you know if the name is not valid.
Protocol	Select the protocol this channel uses.
Interface	Select the interface type for this channel. ◦ Serial—Use to configure the serial parameters for RS232/485 networks. ◦ MultiSerial—Use for configurations with multiple RS-232 ports. ◦ TCPIP—Use for Ethernet or wireless networks.
Description	Enter a description for this channel. For more information about the configuration for common protocols and interfaces, click **Help** at the top of the tab.|

- Click **OK**. The channel is added as a new row in the table.
- Enter or select information, as needed.
- To add or remove a column, right-click the column heading area and select or deselect columns.

Column	Description

Protocol Options	Configure the options for this protocol. The protocol options are dependent upon the selected protocol. Select the protocol, from the dropdown list at the top of the page, and press the HELP button to access the specific protocol's documentation.
Settings	Configure the settings for this channel. The available values depend on the interface the channel is using. ⚠ The settings here must match the settings on the slave device. ◦ For a serial interface, typically keep the defaults. ◦ For a MultiSerial interface, enter the number of RS-232 ports to use in the Ports field. ◦ For a TCPIP interface: ▪ AcceptUnsolicited—Accept unsolicited input from the slave. ▪ ListeningPort—TCP port where the slave device is connected (default is 502). ▪ NodeConnection—Number of parallel requests sent to each node (asynchronous communication). ▪ MaxSimultaneousConnections—Maximum number of concurrent connections. ▪ ShareNodeSameIP—Several slaves are connected to a single IP address. For example, RS485/Ethernet Converters. ▪ UseSingleThread - Use a single thread for the same IP nodes. ▪ UsePingToCheckConnection - Check for connection before sending a packet.
Timeout	Configure the timeout options for this channel. Typically, keep the default value.
IntialState	Select the initial state for this channel.
Remote Settings	Set the primary IP and backup IP to configure the server addresses for this device module
Driver Version	The version of the current driver being used.
[Other columns]	For definitions of other columns that are available in many tables, see Common Column Definitions.

- Continue adding as many channels as you need.
- If needed, right-click a row to cut, copy, paste, or delete the row.

Protocols

Connectivity is a key feature of the FactoryStudio platform. The system has built-in support for many industry standard protocols, such as OPC and Modbus. FactoryStudio also includes many native communication interfaces for a variety of hardware manufacturers, PLC, and protocols.

There are many reasons for including native protocols as well as OPCs:

- cost reduction because most protocols are not charged;
- easier configuration because it is integrated in the system;
- higher access to protocols functions, performance, and diagnostics features that would not be possible using external components.
- improved technical support

Tatsoft is continuously adding new communication drivers.

This link is the list of drivers currently distributed.

Nodes

Nodes in FactoryStudio are the devices or PLCs on the network that you communicate with.

You can enter the settings for your nodes as usual through the Engineering Workspace. You can also import settings from an OPC server or from another data source. See "Importing PLC Addresses" below.

To configure nodes:

- Go to **Edit > Devices > Nodes**.
- Enter or select information, as needed.
- To add or remove a column, right-click the column heading area and select or deselect columns.

Column	Description
Name	Enter a name for the node. The system lets you know if the name is not valid.
Channel	Select the channel for this node. For more information about the configuration for common protocols, click **Help** at the top of the tab.
PrimaryStation	Enter the information required to access the primary node, based on the protocol selected. ⚠ The protocol options are dependent upon the selected protocol. Select the protocol from the dropdown list at the top of the page and press the HELP button to access the specific protocol's documentation. **For Modbus protocol:** ○ For a Serial interface, the SlaveID is the device's slave address on the Modbus network. Valid addresses are 1-247. ○ For a MultiSerial interface, select the number of the ComPort and enter the SlaveID, which is the device's slave address on the Modbus network. Valid addresses are 1-247. ○ For a TCPIP interface: ■ IP—Identification of the slave device's address. ■ Port—TCP port where the slave device is connected (default is 502). ■ SlaveID—Device slave address on the Modbus network. Valid addresses are 1-247. ○ **For OPC interfaces:** ○ Service URL—Defines the location of the OPC server. ■ You must configure the DCOM settings to access an external OPC server. Contact support for assistance. ○ RefreshRate—Server refresh rate. ○ AllTemsSameGroup—Adds all items to a single group OPC. This way, only one connection is created with the OPC server. ○ WaitAfterConnect—The time the application waits, after a project starts, to attempt to communicate. ⚠ OPC UA and OPCXmlDA protocols have a "Test"button to test the connection. OPC UA also has a "Certificates" button to create new certificates for the system.|
BackupStation	Enter the information required, based on the protocol, to access the backup node. When the backupstation is defined and a communication failure occurs on the primary station, the system automatically attempts to establish communication with the backup station.
SyncDate	Date that the import was done (read only field)
SyncStation	Information about the station where the import was done - Station IP, Port and Slot - (read only field)
SyncSettings	Information about the settings used by the system to import the file (read only field)
Description	Enter a description for this node.
[Other columns]	For definitions of other columns that are available in many tables, see Common Column Definitions.

- Continue adding as many nodes as you need.

Data Points

Data points define specific values for each node that can be accessed using tags. The number of data points you can configure is related to both the ProductModel that is configured for the project and your FactoryStudio license.

To configure data points:

- Go to **Edit > Devices > Points**.
- You can copy and paste tags from the **Tag > Objects** tab.
- Enter or select information, as needed.
- To add or remove a column, right-click the column heading area and select or deselect columns.

Column	Description
TagName	Enter a tag name or click ... to select a tag. You can also create a new tag.
Node	Select the node for this data point.
Address	Enter the register address which is based on the PLC and protocol for this data point and tag. ⚠ The Protocol options are dependent upon which protocol is selected. Select the protocol from the dropdown list at the top of the page, and press the HELP button to access the specific protocol's documentation.
DataType	Select the data type you want to use. Most protocols should use the native option. When native is used, the protocol will automatically handle the data conversion. Selecting a different data type overrides the defaults. Some options may not be applicable to the selected node. Make sure you know the applicable data types.
Modifiers	If the PLC uses a different byte order, select the options you want. You can change the position bit, byte, Word, or Dword of the data that is communicated.
Access Type	Select the access type for this data point. You can define and configure the access types. See Access Types, below.
Scaling	If you want to manipulate the tag value, select the options you want. When the data is read in the Equation option: ◦ Div—The system will divide the register value by what you enter here. ◦ Add—The system will add the amount you enter here as an offset of the result of the division. ◦ For a write operation, the calculations are the opposite (multiple by the Div value, then subtract the Add value).
Label	A text that represents a label on the point
[Other columns]	For definitions of other columns that are available in many tables, see Common Column Definitions.

- Continue adding as many points as you need.

Access Types

Access types define the specific methods for reading and writing the values of each data point. This could be the polling rate, whether or not a read is performed on startup, and whether or not unsolicited input is accepted. FactoryStudio comes with a few predefined access types that you can use, or you can create your own.

To configure access types:

- Go to **Edit > Devices > AccessTypes**.
- Do one of the following:
- To edit an existing access type, double-click a field.
- To create a new access type, click **Create New**.
- Enter or select information, as needed.

Column	Description
Name	Enter a name for this access type.
Read	
ReadPolling	Select when you want to enable read polling.
ReadPolling Rate	Enter how often the address value is retrieved.
ReadTrigger	Enter an object property that tells the system when to read the value.
ReadOnStartup	When selected, the system reads the value on startup.
ReadStatus	Enter an object property that receives the status of the read communication.

ReadCompleted	Enter an object property that receives an indication that the reading is completed. The value will change between 0 and 1 every time a reading is completed.
Write	
WriteEventEnable	Select to enable the writing of values to the PLC.
WriteEvent	Select when the value is written.
WriteTrigger	Enter an object property that tells the system when to write the value.
WriteStatus	Enter an object property that receives the status of the write communication.
WriteCompleted	Enter an object property that receives an indication that the writing is completed. The value will change between 0 and 1 every time a writing is completed.
Settings	
AcceptUnsolicited	When selected, the system accepts values from the PLC, even if the polling time has not yet elapsed.
UseStaticBlocks	Select to use the block command field
BlockCommand	Enter a definition for each block that will be created. Check the driver documentation to see if the specific driver uses this field and the valid values.
Description	Enter a description for the access type.
[Other columns]	For definitions of other columns that are available in many tables, see Common Column Definitions.

Importing PLC Addresses

Communication nodes and data points that have been defined by another data source can be imported in the following ways:

- You can copy and paste the contents of an Excel table. As long you include the title of the columns when you copy and paste the contents of the table, the tables in your project can have different columns or can be ordered differently. The system will put the data in the expected columns, even if the order is different in the source.
- You can import the data from csv files.
- For Rockwell ControlLogix devices, you can import from L5k definition files.
- For OSIsoft® PI database, there is a FactoryStudio version to share definitions.
- A programming API is available that can populate the tables from code, even during runtime if it is necessary.

If your PLC or field device has an open database or a file with available addresses and you want the PLC and FactoryStudio addresses to have tight integration, contact support.

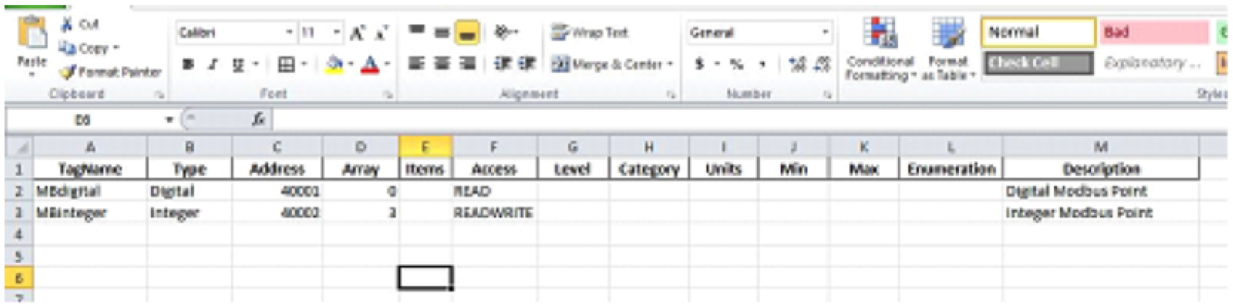
Importing from an OPC Server

After you create an OPC communication node, you can select the node and click **Import** to import the OPC database for the project. FactoryStudio automatically creates the tags and communication points.

Importing from Excel

To create and import Tags:

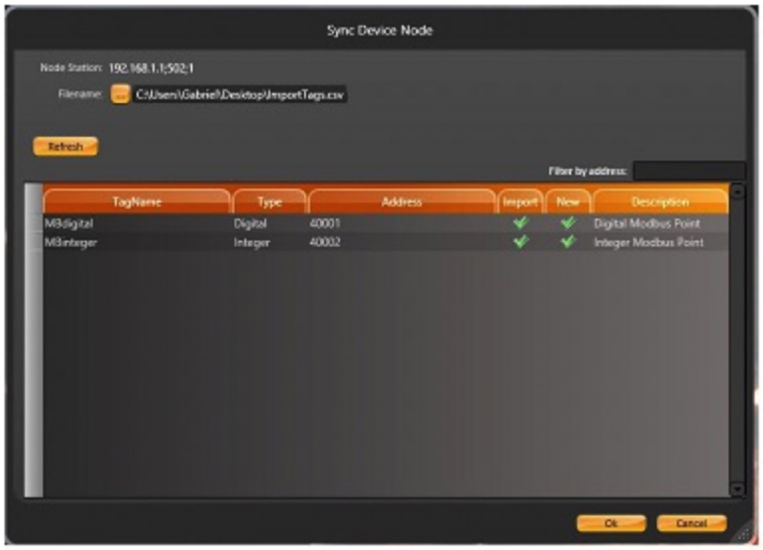
- In Excel, make a table with the columns that are shown below.



	TagName	Type	Address	Array	Items	Access	Level	Category	Units	Min	Max	Enumeration	Description
2	MBdigital	Digital	40001	0		READ							Digital Modbus Point
3	MBinteger	Integer	40002	3		READWRITE							Integer Modbus Point

To successfully import the tags, you need the TagName, Type, and Address columns.

- After you chose the device protocol and create a new node, click on the **Import** button. Then, choose the .CSV file that contains the tag's information and click on the ok button.



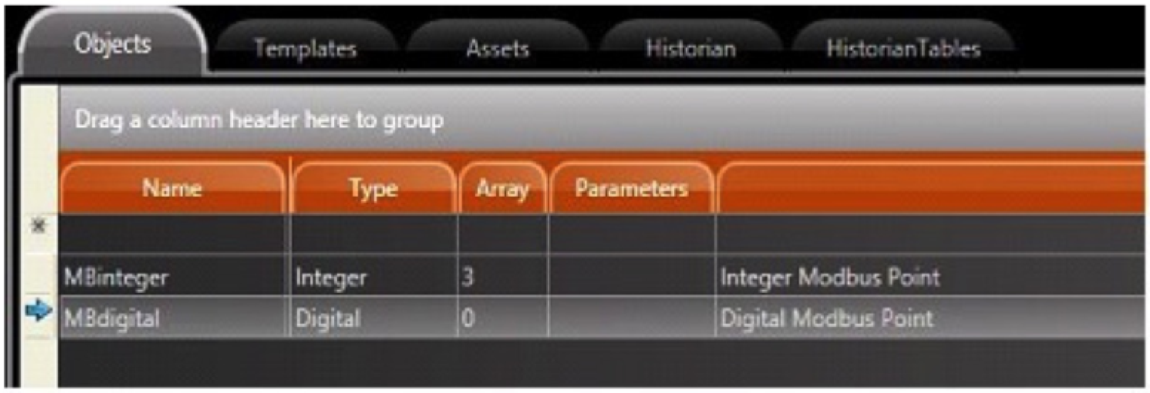
Sync Device Node

Node Station: 192.168.1.1;502;1

Filename: C:\Users\Gabriel\Desktop\ImportTags.csv

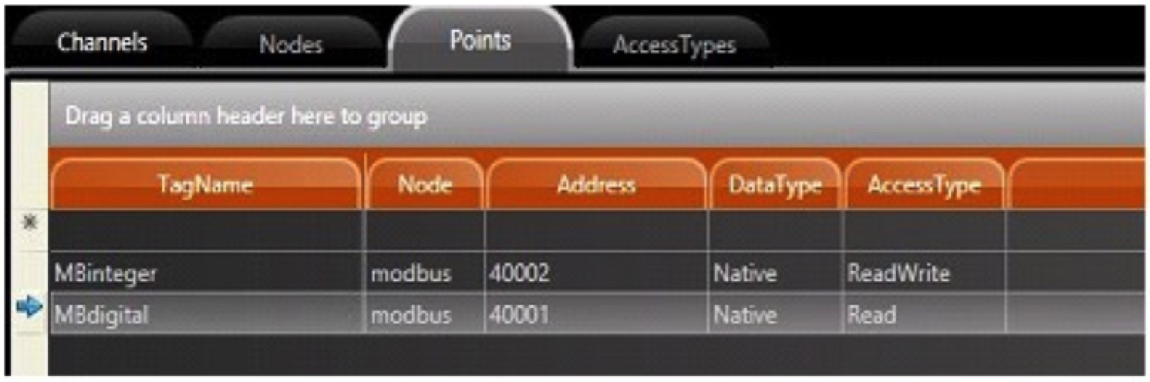
TagName	Type	Address	Import	New	Description
MBdigital	Digital	40001	✓	✓	Digital Modbus Point
MBinteger	Integer	40002	✓	✓	Integer Modbus Point

-
- The Tags and Points will be created automatically in **Edit > Tags.**



Name	Type	Array	Parameters	
MBinteger	Integer	3		Integer Modbus Point
MBdigital	Digital	0		Digital Modbus Point

- **Edit > Device > Points**



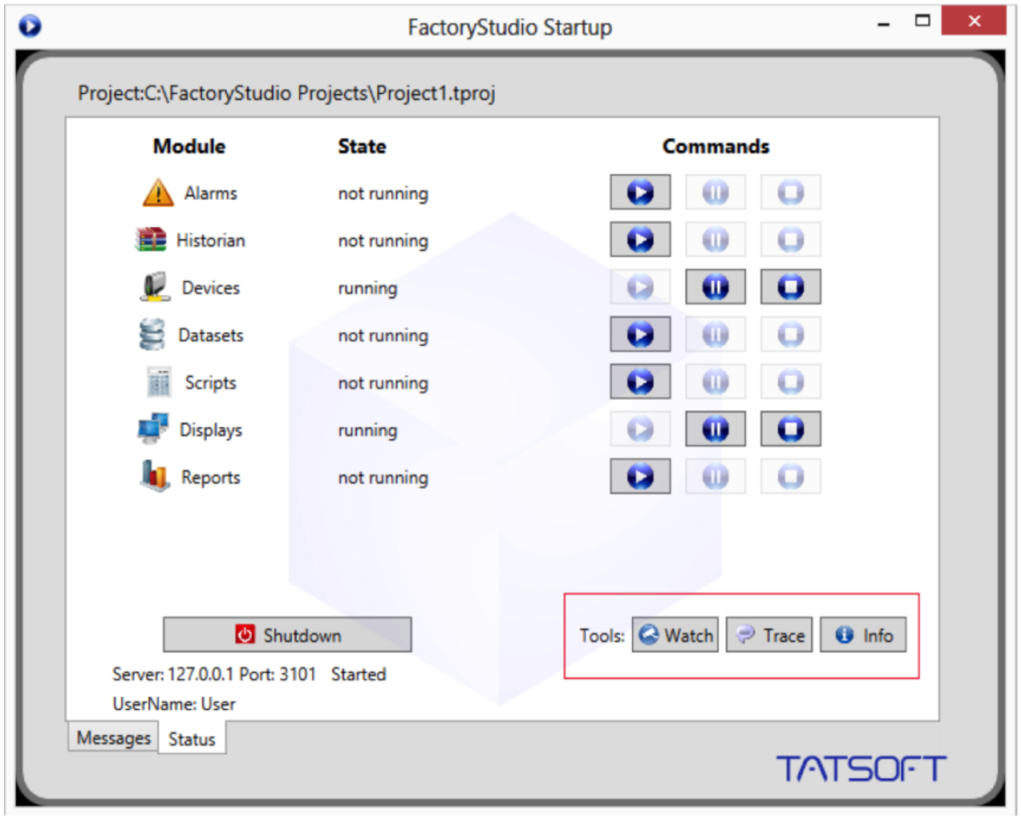
TagName	Node	Address	DataType	AccessType
MBinteger	modbus	40002	Native	ReadWrite
MBdigital	modbus	40001	Native	Read

After you use the Import tool for the first time, the system will save whatever settings you used. The import button switches to now be called the "Sync" button. This make the button execute a synchronization that verifies the previously imported addresses and the new ones.

Using Diagnostic Tools

After you start the project in the Startup window, select the diagnostic tools. These are: PropertyWatch (Watch), TraceWindow (Trace), and ModuleInformation (Info).

You can also start the diagnostic tools on the Run-Test and Run-Startup pages by clicking on the desired tool's icon with the lift mouse button. If menus are enabled for the display, you can also access the tools menu.



Project:C:\FactoryStudio Projects\Project1.tproj

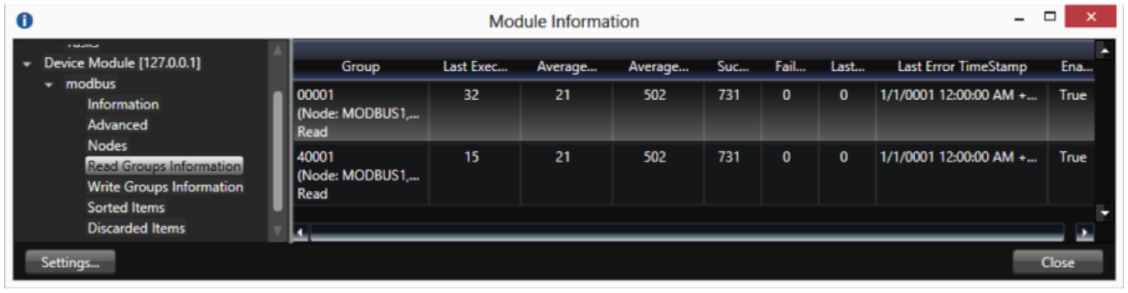
Module	State
Alarms	not running
Historian	not running
Devices	running
Datasets	not running
Scripts	not running
Displays	running
Reports	not running

Server: 127.0.0.1 Port: 3101 Started
UserName: User

Module Information

The Module Information tool provides information about the operation of the modules. If you choose the devices module and a specific channel, you will see information regarding the function of the communication channel.

The "Read Groups Information" is important because it provides information about the virtual reading groups, the run time of each item, the quantity of the readings, and the readings that have failed. It also reports on the code and date/time of the last error.



Group	Last Exec...	Average...	Average...	Suc...	Fail...	Last...	Last Error TimeStamp	Ena...
00001 (Node: MODBUS1,... Read	32	21	502	731	0	0	1/1/0001 12:00:00 AM +...	True
40001 (Node: MODBUS1,... Read	15	21	502	731	0	0	1/1/0001 12:00:00 AM +...	True

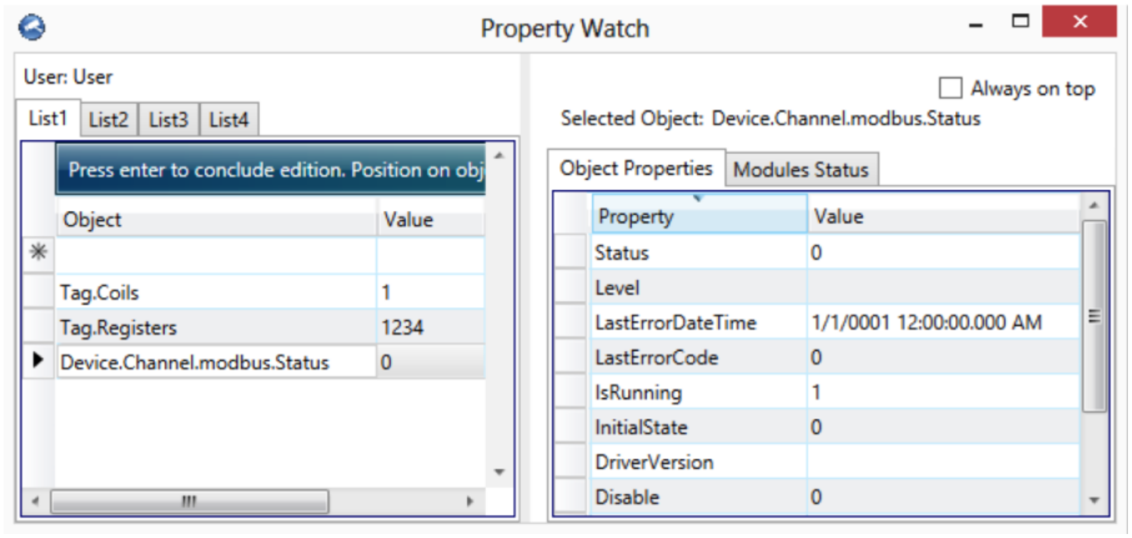
These are the typical steps to use the Module Information tool:

- Go to Read Groups Information to look at the number of successful and failed communication events and in order to quickly identify the communication blocks.
- If you have systematic errors in all of the blocks or status codes with negative values, it typically means you cannot access the remote device. Verify if the node address is right.
- If you have specific blocks with systematic errors, verify the tags and addresses that are connected to the blocks. Use the TraceWindow with the Device information to collect information about the communication errors.
- For some protocols, like OPC, the discarded items will show as a wrong addresses in the configuration.

Keep in mind that you can only READ from a field device when an Enterprise application is running in test mode. This makes it useful to run the application with ONLINE CONFIGURATION enabled because you do not need to start and stop the driver when modifying the configuration. You can modify PLC addresses, AccessTypes, and most of the application, and you can see the results in real-time on your running application. You can use the Startup window or the PropertyWatch tool to start and stop only one module, instead of restarting the entire runtime system.

Property Watch

Property Watch is a diagnostic tool used to access the tags and internal properties of the system for reading or writing. Type the name of the property in the Object column, and its value will be found in the Value column.



Selected Object: Device.Channel.modbus.Status

Object	Value
Tag.Coils	1
Tag.Registers	1234
Device.Channel.modbus.Status	0

Property	Value
Status	0
Level	
LastErrorDateTime	1/1/0001 12:00:00.000 AM
LastErrorCode	0
IsRunning	1
InitialState	0
DriverVersion	
Disable	0

For example, in the screen shown above, select Tag.Coils or Device.Channel.modbus.Status . The value of these objects will be shown. To the right side, additional properties are displayed for the selected object.

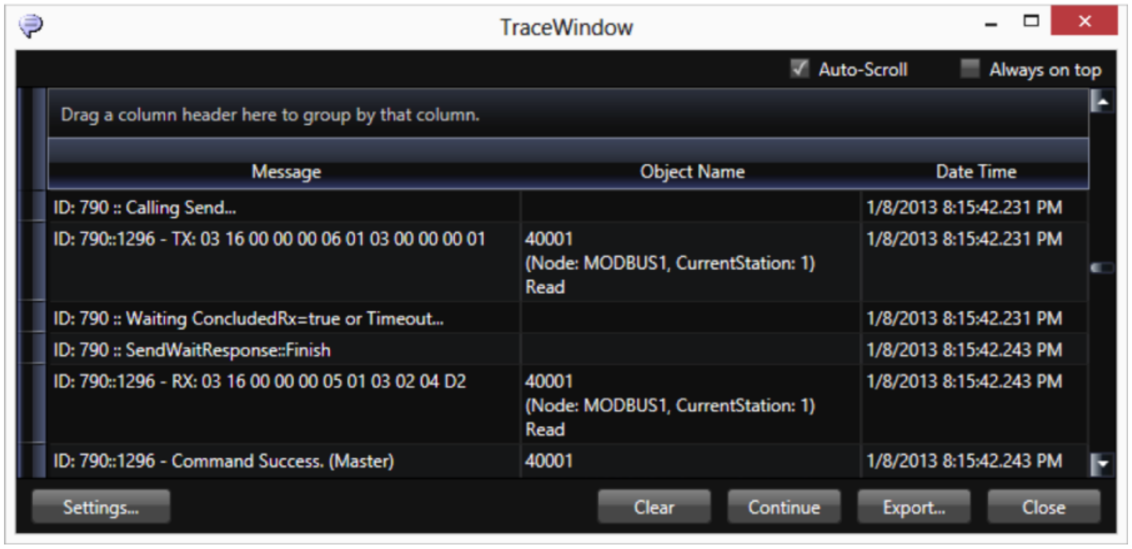
Devices negative error code

For Module Information and Property Watch, there are negative error codes that represents specific messages that can help to find the root cause of the error. In the Module Information, refer to the LastErrorCode column. On Property Watch, refer to @Device.Channel. ChannelName.Status. See below the list of negative error codes:

- *0 Success*
- *-1 BuildCommandException*
- *-2 ParseCommandUnsolicitedException*
- *-3 ParseReplyException*
- *-4 BuildReplyUnsolicitedException*
- *-5 ChannelException*
- *-6 NodeException*
- *-100 Base Send Error*
- *-101 Base SendAndWait Error*
- *-102 TCP Create Error 1*
- *-103 TCP Create Error 2*
- *-104 TCP Create SocketError*
- *-105 TCP Connect Callback Error*
- *-106 TCP Receive Error*
- *-107 UDP Create Error*
- *-108 UDP Receive Error*
- *-109 Serial Create Error*
- *-110 Serial Receive Error*
- *-111 TCP NotConnected*
- *-112 Start message timeout*
- *-113 Receiving bytes timeout*
- *-114 End message timeout*
- *-115 Connect timeout*
- *-200 ProtocolError*
- *-201 InvalidProtocol*
- *-202 InvalidStation*
- *-203 InvalidCommand*
- *-204 InvalidMsgSequence*
- *-205 InvalidCheckSum*
- *-206 InvalidAddress*
- *-207 InvalidModifiers*

Trace Window

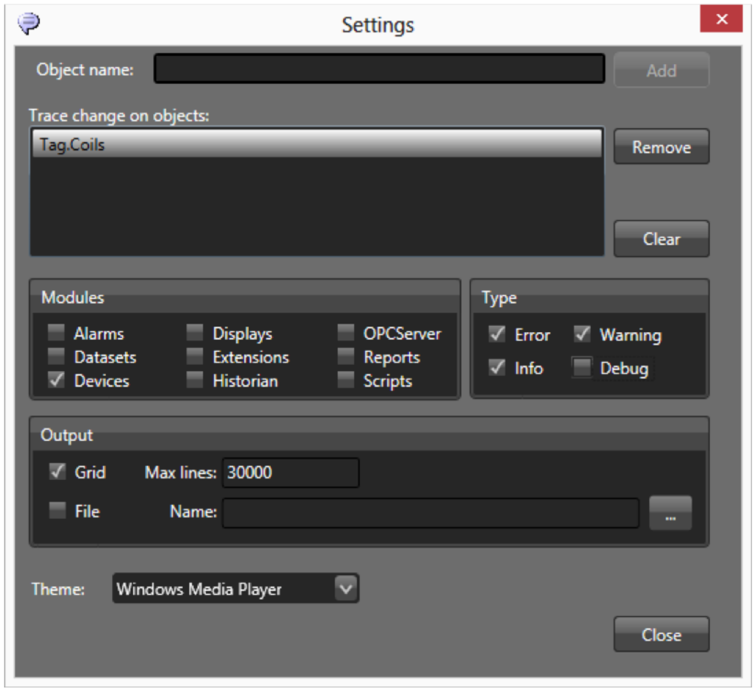
The Trace Window tool displays system messages in a data grid interface. If you enable the module devices in the settings, information will be available about the status of reads, writes, unsolicited input, TX frames (sent), and RX frames (received).



Message	Object Name	Date Time
ID: 790 :: Calling Send...		1/8/2013 8:15:42.231 PM
ID: 790::1296 - TX: 03 16 00 00 00 06 01 03 00 00 00 01	40001 (Node: MODBUS1, CurrentStation: 1) Read	1/8/2013 8:15:42.231 PM
ID: 790 :: Waiting ConcludedRx=true or Timeout...		1/8/2013 8:15:42.231 PM
ID: 790 :: SendWaitResponse::Finish		1/8/2013 8:15:42.243 PM
ID: 790::1296 - RX: 03 16 00 00 00 05 01 03 02 04 D2	40001 (Node: MODBUS1, CurrentStation: 1) Read	1/8/2013 8:15:42.243 PM
ID: 790::1296 - Command Success. (Master)	40001	1/8/2013 8:15:42.243 PM

When checking the devices CheckBox in the Settings, enable only the ERROR, INFO, and Warning information. Do not enable the Debug information, or you will create too much data. For ControlLogix devices, it is very important to use this tool, as the system will present here the invalid addresses on the configuration.

If you click on the settings button in the configuration dialog, you can select which message types and modules to display. You can see the data in the data grid or save it to a file. It is also possible to configure a tag in ObjectName, and click the Add button to display a menu to select that object and include it in the monitoring.



For more information on how to use the Diagnostic Tools, please see Using the Diagnostic Tools.

The Device Namespace

The **Device** namespace is the entry point for all objects related to the device module.

The **Device.Channel** object lists all of the configured channels and their runtime properties.

The **Device.Node** object lists all of the configured nodes and their runtime properties.

The **Device.AccessType** object lists the defined access types and has options to execute synchronous calls on reading and writing to the device.

The following tag properties are updated based on the device module:

tag.tagname.DevicePoint: Device point address connected with this tag

See Namespaces for the complete programming reference on runtime objects.