Devices and Interfaces

On this page:

Devices in FactoryStudio are any live real-time data source. Typically a device is a PLC, another FactoryStudio, an OPC server, an external Asset Manager or Historian Server, or any 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 **protocol**. 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**. The data points provide the specific data values to be accessed using tags. 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 at this data point, such as polling rate, whether a read is performed on startup, and whether unsolicited input is accepted.

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

1) Define the equipment (and protocols) the project will use at Edit Devices Channels.

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

3) Map the Tags in you data model to addresses in the devices, at Edit Devices Points.

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

In order to expedite the configuration, FactoryStudio provides many Import Wizards that will automatically create the Tags 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 documentation.
Settings	Configure the settings for this channel. The values available 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. **Tx**: Time for the first byte of the message to be sent (in milliseconds) **RxStart**: Time between the last byte sent and the first received (in milliseconds) **RxFinish**: Time to fully receive the message (in milliseconds) **NextByte**: Time between every received byte (in milliseconds) **Retry**: Quantity of retries before prompt a timeout error (in milliseconds) **Unsolicited**: Maximum unsolicited communication time before prompt an error (in milliseconds)
IntialState	Select the initial state for this channel.
Remote Settings	Set 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 to a variety of hardware manufacturers, PLC and protocols.

The reason to include native protocols, besides OPC, are many, such as:

- cost reduction, as most protocols are not charged;
- easier configuration, as it is integrated on 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 settings for your nodes as usual through the Engineering Workspace. You can also import settings from an OPC server or from other 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 documentation. **For Modbus protocol:** ○ For a Serial interface, the SlaveID is the device 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 the device slave address on the Modbus network. Valid addresses are 1-247. ○ For a TCPIP interface: ▪ IP—Identification of the slave device address. ▪ Port—TCP port where the slave device is connected (default is 502). ▪ SlaveID—Device slave address on 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 in a single group OPC. In this way, only one connection is created with OPC server. ○ WaitAfterConnect—Time to communicate after the application is running. ⚠ OPC UA and OPCXmlDA protocols have a "Test"button to test connection. OPC UA also have a "Certificates" button to create new certificates for the system.|
BackupStation	Enter the information required to access the backup node, based on the protocol selected. When 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 has been done (read only field)
SyncStation	Information about the station that the import has been 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 the 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 configured for the project and your license for FactoryStudio.

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, based on the PLC and protocol for this data point and tag. ⚠ 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 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 types 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 access types. See Access Types, below.
Scaling	If you want to manipulate the tag value in some way, select the options you want. For the Equation option, when reading the data: ○ 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 to 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 to represent 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 by which values are to be read and written for each specific data point, such as polling rate, whether a read is performed on startup, and whether 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 to retrieve the address value.
ReadTrigger	Enter an object property to tell the system when to read the value.
ReadOnStartup	When selected, the system reads the value on startup.
ReadStatus	Enter an object property to receive the status of the read communication
ReadCompleted	Enter an object property to receive an indication that the reading is completed. The value will change between 0 and 1 every time a reading was completed.
Write	
WriteEventEnable	Select to enable writing of values to the PLC.
WriteEvent	Select when to write the value.
WriteTrigger	Enter an object property to tell the system when to write the value.
WriteStatus	Enter an object property to receive the status of the write communication.
WriteCompleted	Enter an object property to receive an indication that the writing is completed. The value will change between 0 and 1 every time a writing was completed.
Settings	
AcceptUnsolicited	When selected, the system accepts values from the PLC, even if the polling time has not yet elapsed.
UseStaticBlocks	Select when you want 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 drive uses this filed 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

When creating communication nodes and data points, you can import them if they are defined in another data source in the following ways:

- You can copy and paste the contents of a table from Excel. The tables can have different columns or order, as long you include the title of the column in the copy and paste operations. The system will put the data in the expected columns, even if the order is different in the source and target tables.
- 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 from runtime execution when it is necessary.

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

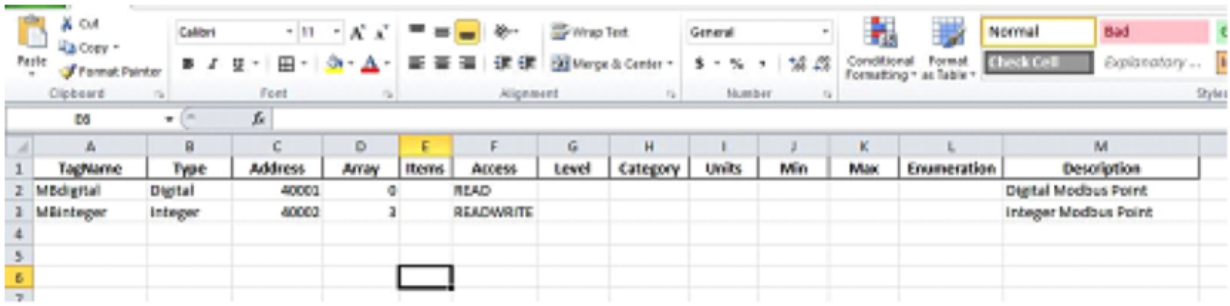
Importing from an OPC Server

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:

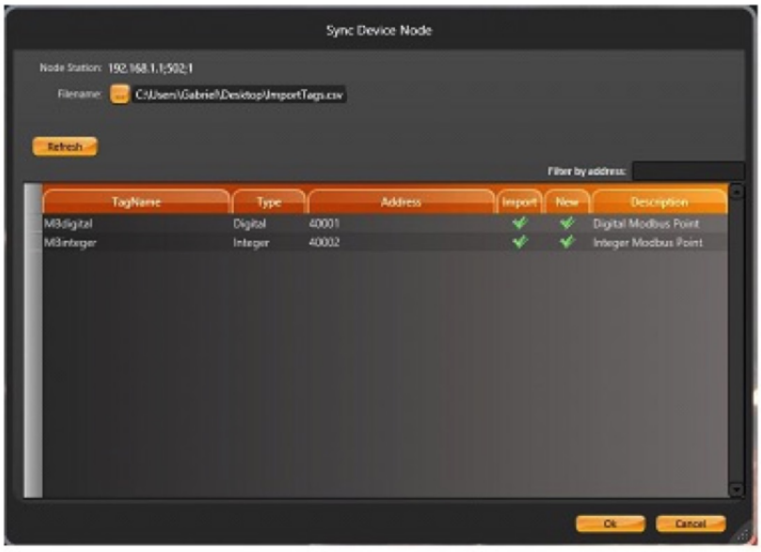
- Make a table in Excel with the Columns as shown below.



	A	B	C	D	F	F	G	H	I	J	K	L	M
1	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
4													
5													
6													
7													

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

- After you have chosen the Device Protocol and created a new Node, click on the **Import** button. Then choose the .CSV file with the Tags' information and click on the **OK** button.



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

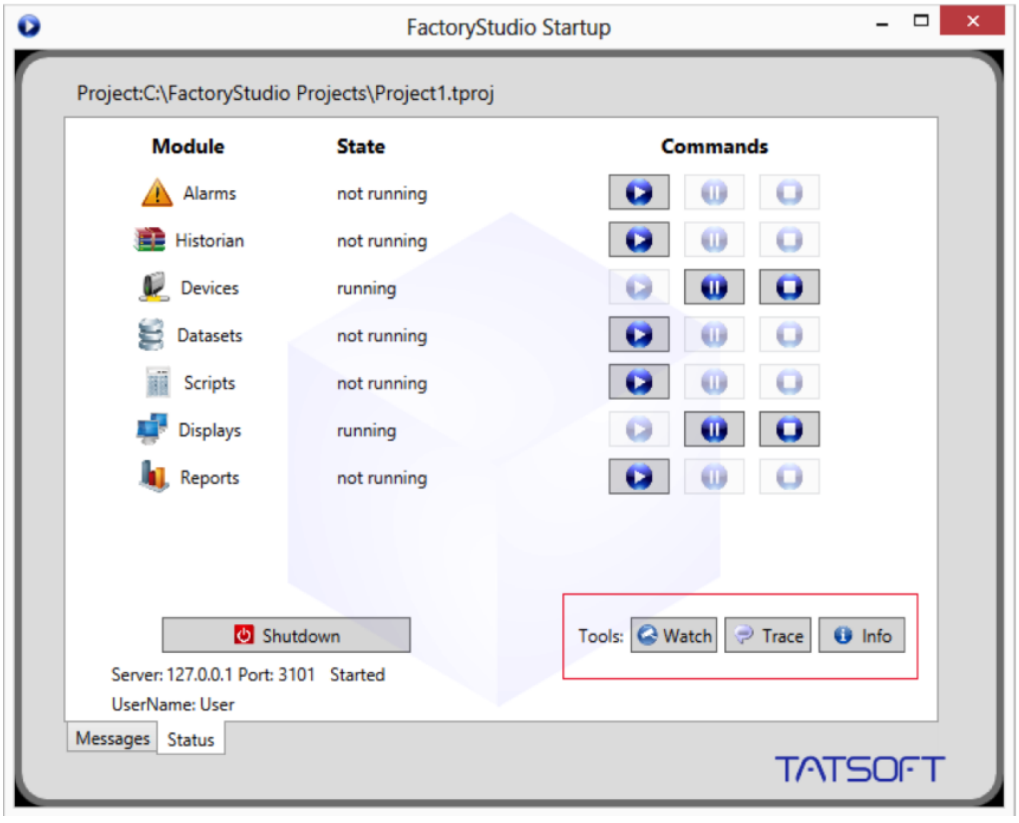
- **Edit > Device > Points**

After you have used the Import tool for the first time, the system will save the settings used, so the button shows now SYNC, which means the next time you use it, it will run a synchronization, verifying which addresses were previously imported and the new ones.

Using Diagnostic Tools

After starting the project, from the Startup window, you can select some diagnostic tools. These are: PropertyWatch (Watch), TraceWindow (Trace), and ModuleInformation (Info).

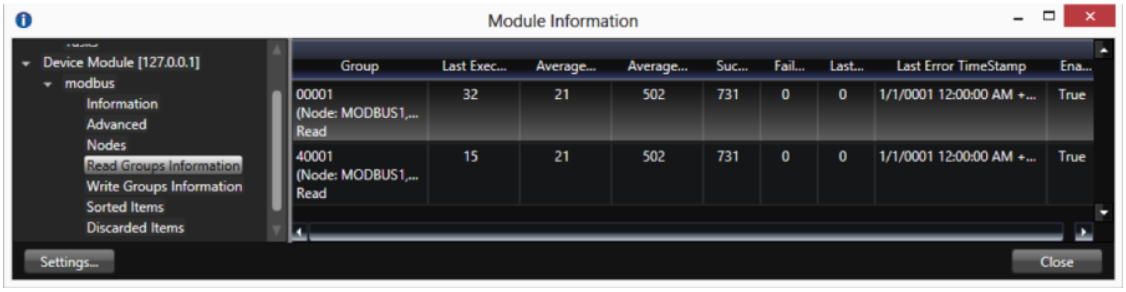
You can also start the Diagnostic tools at the Run-Test and Run-Startup pages, pressing the mouse left button over the icons of the desired tools. If the menus are enabled at the Displays you can also access the Tools menu.



Module Information

The Module Information tool provides information about the operation of the modules. If you choose the module Devices and a specific channel, you will have several information items about the functioning of the communication channel.

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



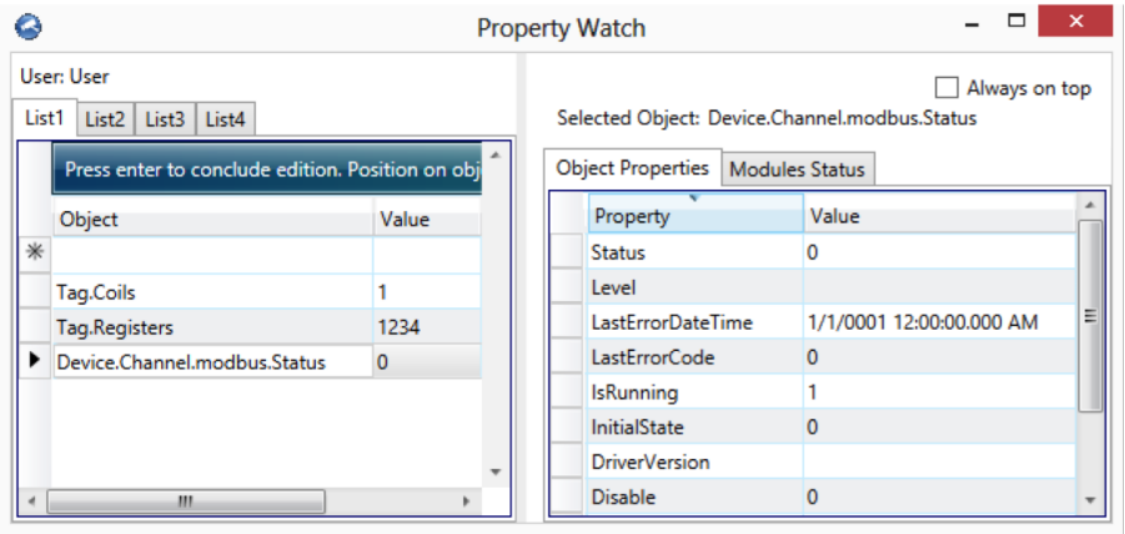
These are the typical steps when using the Module Information tool:

- Go to Read Groups Information, to look at the number of Successful and Failed communication events, in order to quickly identify the communication blocks.
- If you have systematic errors in all blocks, or status codes with negative values, typically it 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 connected with those blocks. Use the TraceWindow with Device information to collect information about those communication errors.
- For some protocols, such as OPC, the Discarded items will show wrong addresses in the configuration.

When running an Enterprise application in TEST mode, keep in mind that in this mode we only READ from the field devices, even you have a configuration to write to field. It is very useful to run the application with ONLINE CONFIGURATION enabled, so you don't need to start and stop the driver when modifying the configuration. You can modify PLC addresses, AccessTypes, and most of the application, and 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, such as the Devices module, instead of restarting the entire runtime system.

Property Watch

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



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

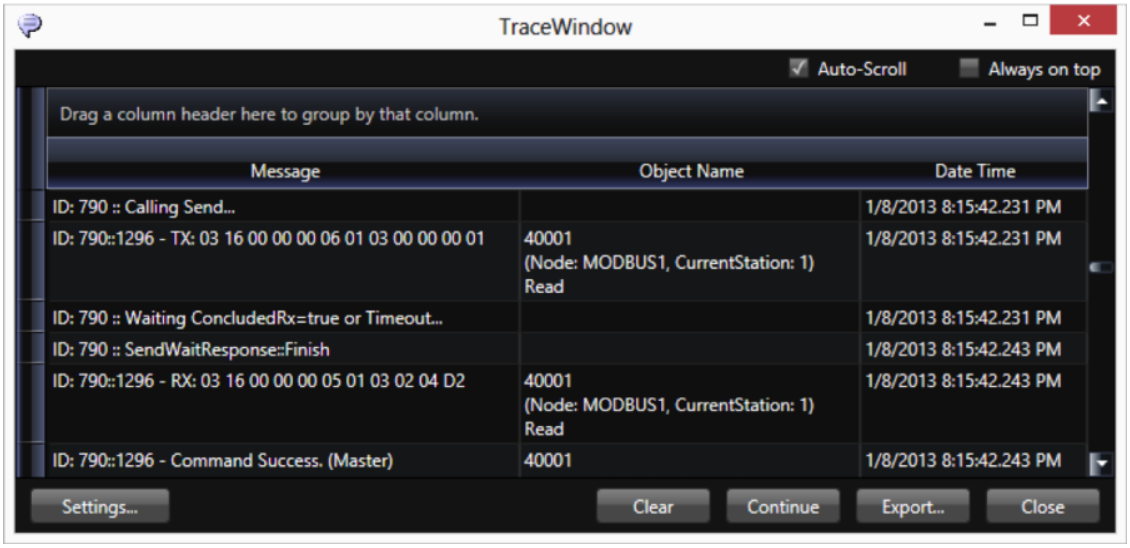
Devices negative error code

Even for Module Information and Property Watch there are some negative error codes that represents specific messages that can help to find the root cause of the error. On the Module Information, refer to the column LastErrorCode, and 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*

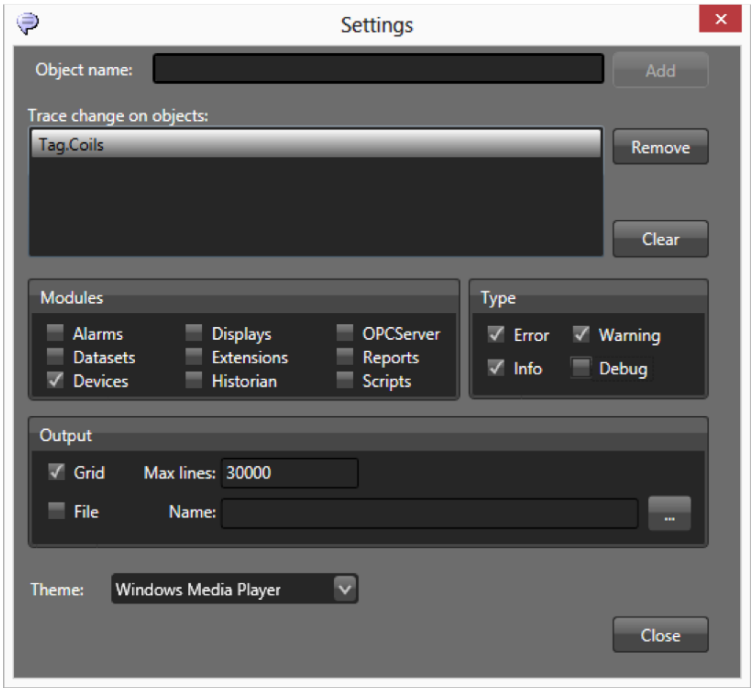
Trace Window

The Trace Window tool presents system messages in a data grid interface. If you enable the module Devices at the Settings button, you have information about the status of reads, writes, unsolicited input, TX frames (sent) and RX frames (received).



When checking the Devices CheckBox on the Settings, enable only the ERROR, INFO and Warning information, not the Debug information, otherwise 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 bring up a menu to select that object to include in the monitoring.



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

The Device Namespace

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

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

The **Device.Node** object lists all 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.