User Manual

IndigoSCADA

Powerful SCADA with Modbus, OPC, IEC 60870-5-104/103/101, DNP 3.0 and MQTT support

Copyright

Contributors

apa

endejha

Feedback

Please direct any comments or suggestions about this document to: info at enscada.com

Publication date and software version

Published on 09th December 2024. Based on LibreOffice 4.0.3.

Note for the reader

This manual refers to **IndigoSCADA** software version revision **483**

Contents

Introduction

IndigoSCADA is a powerful **SCADA** useful for small scale projects where MODBUS, DNP3, OPC DA 2.05, OPC UA 1.03, IEC 60870-5-101, IEC 60870-5-103, IEC 60870-5-104, MQTT 3.1.1/Sparkplug B, IEC 61850 has to be used.

Features

IndigoSCADA has grown many important features during years:

- Use of data to generate daily/weekly/monthly management reports
- Historical and real time graphical presentation of data
- Real-time event and alarm notification
- Integrated real time and historical databases
- SQL editor allows for on-line maintenance of real-time and historical databases
- Multiple users with different access rights
- Built in IEC 60870-5-101 data types
- Small footprint SCADA system
- Support of multiple dedicated lines on the back end
- OPC DA 2.05, DNP 3.0, Modbus, MQTT 3.1.1/Sparkplug B,
- IEC 60870-5-101/103/104, IEC 61850
- Softlogic programming with C scripts
- Easy Backups
- Support of multiple HMI windows
- HMI designer

Known problems and limitations

None at the moment

Hardware requirements

To work with IndigoSCADA, you need a PC or a laptop with the following minimum specifications:

Operating system	***Minimum requirements***		
	Processor	RAM	Graphics
MS Windows Professional XP	800 MHZ	512 GB	XGA 1024 x 768 16-bit color depth
MS Windows Server 2003	2.4 GHZ	1 GB	XGA 1024 x 768 16-bit color depth
MS Windows Vista	1 GHZ	1GB	XGA 1024 x 768 16-bit color depth
MS Windows Seven	1 GHZ	1GB	XGA 1024 x 768 16-bit color depth
MS Windows 10 – 64 bit	1 GHZ	4GB	XGA 1024 x 768 16-bit color depth
MS Windows 11 – 64 bit	1 GHZ	4GB	XGA 1024 x 768 16-bit color depth

You also need

- Keyboard and mouse
- RS232 serial interface

Software requirements

IndigoSCADA is a 32-bit (Win32) application whose functionality has been designed especially for the following operating systems:

- Microsoft Windows 2000 Professional 32-bit
- Microsoft Windows XP Professional 32-bit with Service Pack 3
- Microsoft Windows Vista Home Premium 32-bit with Service Pack 2
- Microsoft Windows Vista Business 32-bit with Service Pack 2
- Microsoft Windows Vista Ultimate 32-bit with Service Pack 2
- Microsoft Windows Server 2003 Standard Edition 32-bit with Service Pack 2 used as a workstation computer
- Microsoft Windows 7 Ultimate/Enterprise 32-bit
- Microsoft Windows 7 Professional 32-bit
- Microsoft Windows 10 64-bit
- Microsoft Windows 11 64-bit
- VMWare support for virtual machines (for more details, see the next chapter)

IndigoSCADA has not been tested for the following operating systems:

- Microsoft Windows XP Professional 32-bit without Service Pack or with Service Pack 1 or Service Pack 2
- Microsoft Windows Vista Home Basic 32-bit
- Microsoft Windows Vista variants 32-bit without Service Pack or with Service Pack 1
- Microsoft Windows Server 2008 32-bit as a workstation computer
- Microsoft Windows 7 Ultimate/Enterprise and Professional 32-bit with Service Pack 1
- If you use these operating systems with **IndigoSCADA**, use it is at your own risk.

Virtual Environment

IndigoSCADA can be used with these popular virtualization software:

- VMware Workstation
- WMware Player
- VirtualBox

Installation

Installation of IndigoSCADA over existing IndigoSCADA Installation

It is not recommended to install **IndigoSCADA** over an existing installation because some previous installed files may conflict with newer files.

Please uninstall an existing installation of IndigoSCADA before installing a new software version.

Installation of IndigoSCADA from setup files

Please note the following restrictions and recommendations:

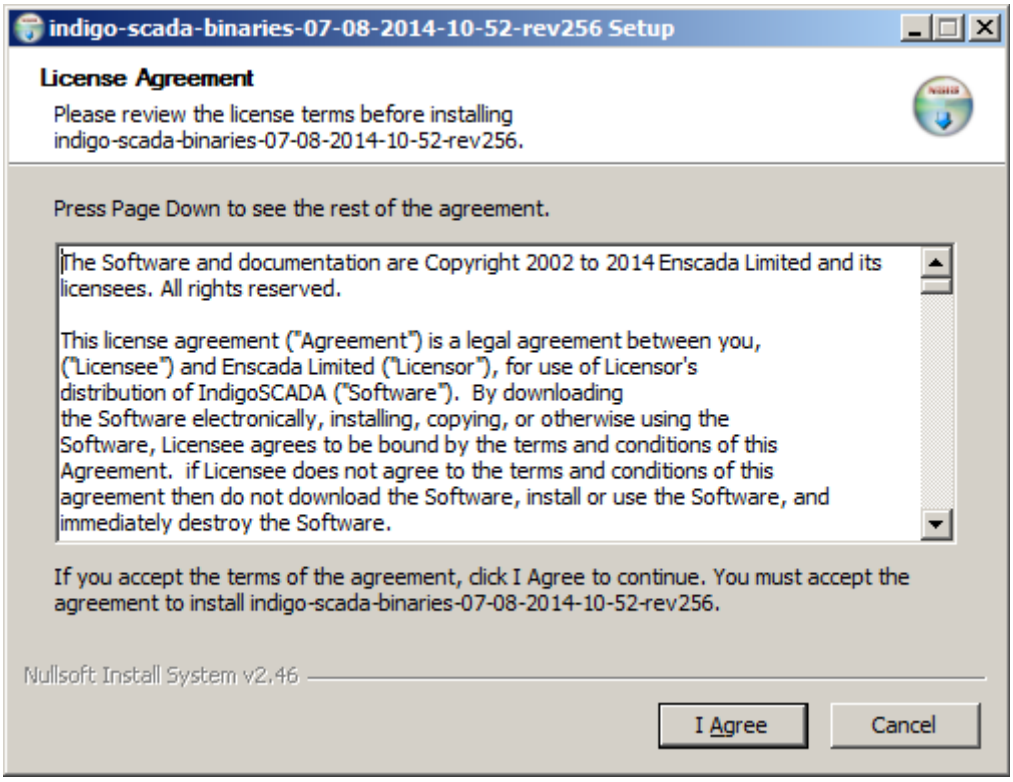
• Before starting the installation or update, please close all applications (such as

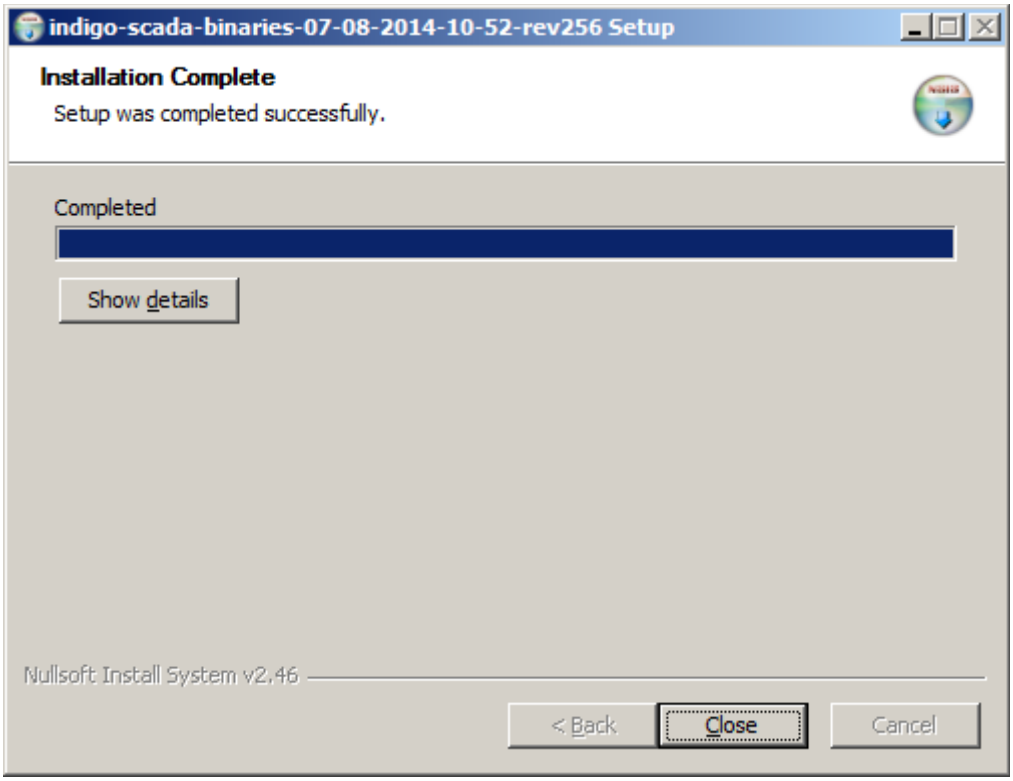
Microsoft Word) that may interfere with the installation

To start the installation of IndigoSCADA, double-click the setup.exe file that you downloaded from the Sourceforge site. Usually its name will be in the form of **indigo-scada-binaries-XX-XX-XXX-YY-YY-revZZZ.**

The **IndigoSCADA** setup program guides you through the installation. Follow the on-screen instructions of the setup program:

Then Accept the license pressing the **I Agree** button and choose as destination folder **C:** (the default installation directory)





After few seconds the installer will copy the required files and **IndigoSCADA** will be ready to be used.

Uninstall of IndigoSCADA

To remove **IndigoSCADA** from your PC, simply delete the destination folder **C:\scada** used during the installation process.

Firewall exceptions

Create the following exceptions on the firewall:

dspserver, rtsqlserver, sqlserver, rtps_mngr

IndigoSCADA project management

After the program is installed, all the project files are located in the folder **C:\scada\project** Project files have the following file extensions: .dbs .fdb .ui .db .ini

These file contains empty databases, except **modbus_database1.db** which contains a predefined modbus configuration with 24 points.

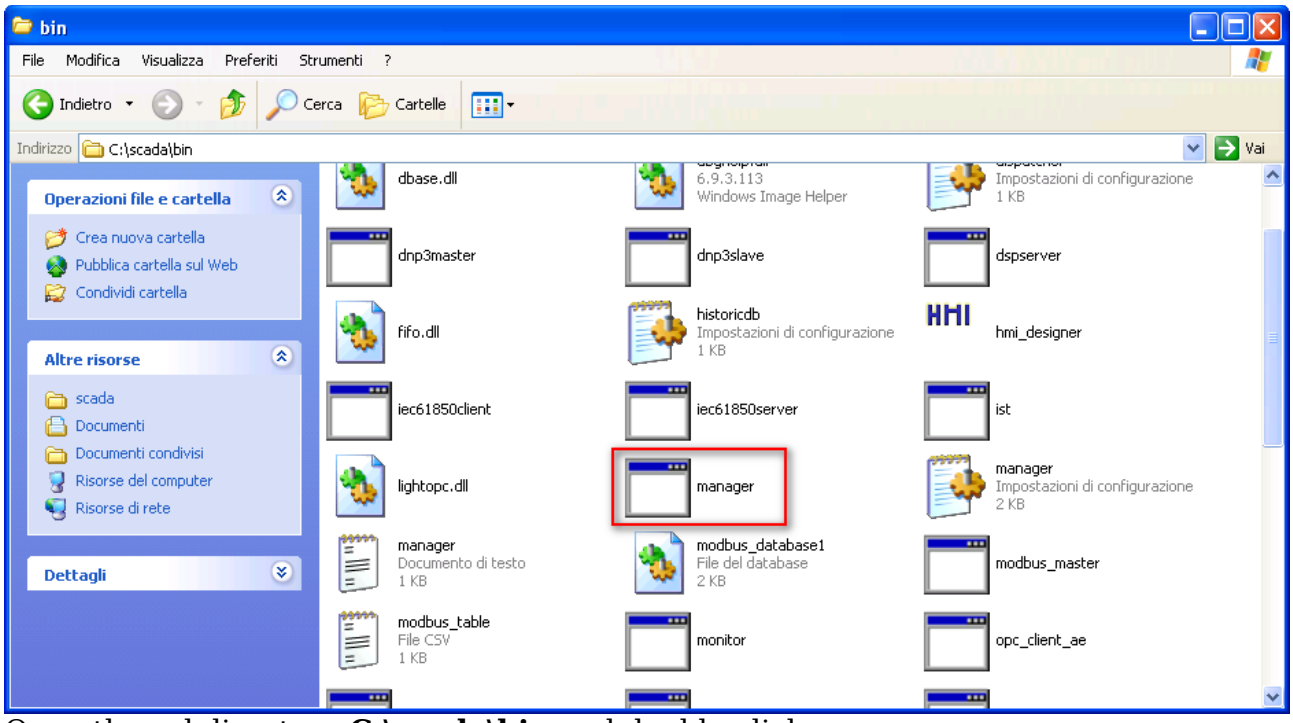
You may zip the **C:\scada\project** folder to do backup and restore of a project.

The empty databases of the IndigoSCADA distribution are contained in the **empty_project.zip** backup file.

Every time you want to create a new project, you need to create an empty directory, then unzip the files contained in **empty_project.zip** file into the empty directory. Set the path of the new directory in file **C:\scada\bin\project.ini**, restart manager.exe, and continue with the configuration of the empty project. Otherwise you can press the button **Set project folder** and select a folder where there is a project.

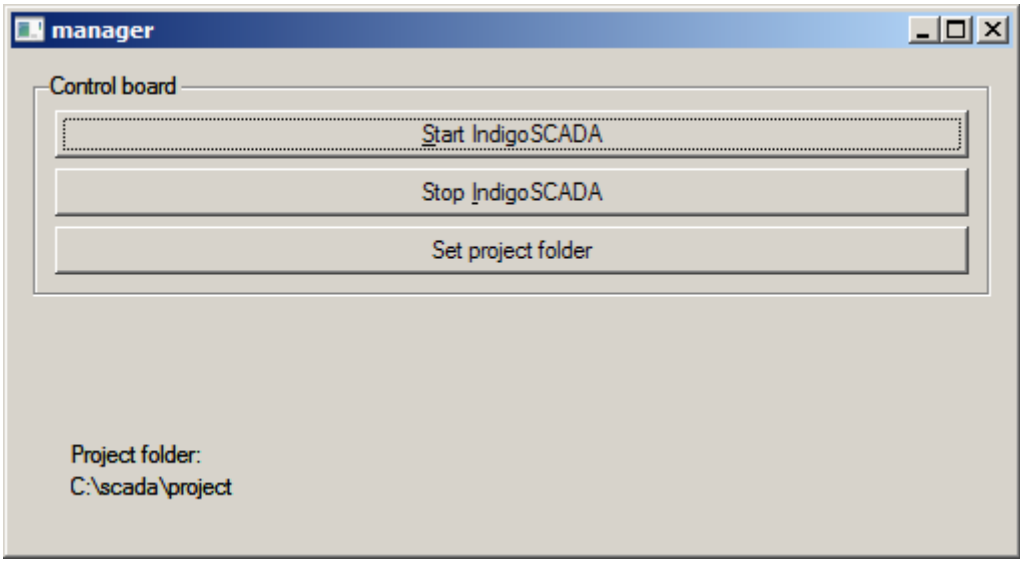
Starting IndigoSCADA

After the program is installed, it is possible to configure the empty project and start it to runtime.



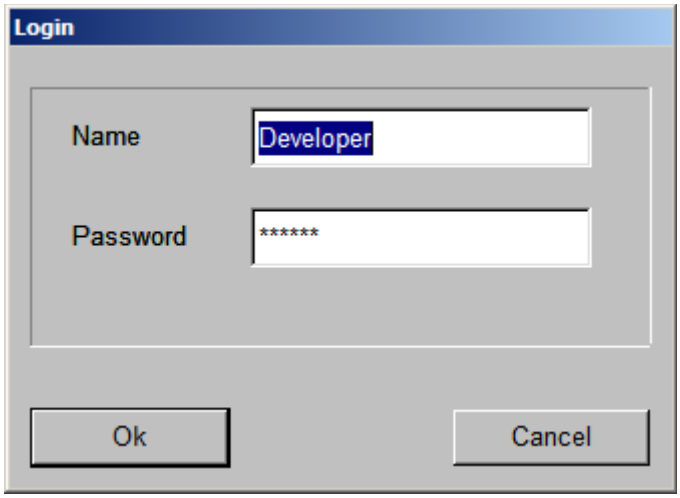
Open the subdirectory **C:\scada\bin** and double-click on **manager.exe.**

The main applet **Manager** shows up. This applet can be used to conveniently start and stop **IndigoSCADA** without worrying of opening or closing all the sub-processes it opens during the operation.



At the beginning there is only one **Administrator User** which can be used to access the program. After the first login, it will be possible to add new users and configure user permissions in detail.

Use **Developer** as Name and **qwerty** as password.



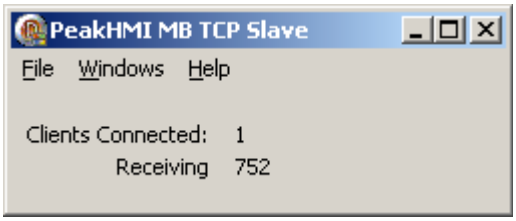
Insert a new password to change the default one or press then **Cancel** button to skip this phase.

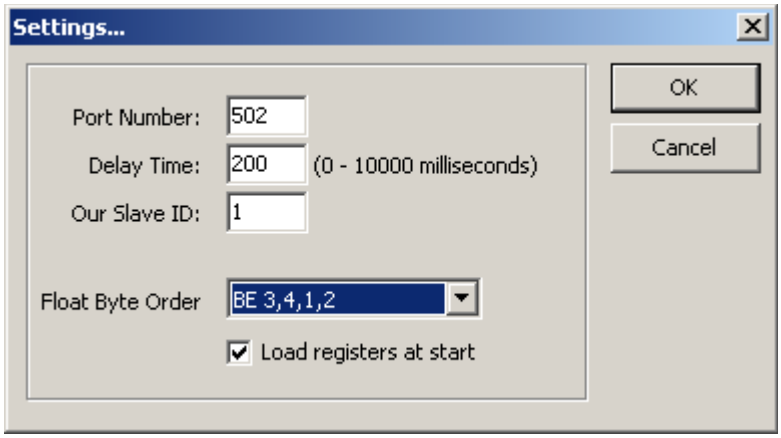
Quick tutorial with IndigoSCADA

In this tutorial we will use a **Modbus Simulator** called **PeakHMI,** which can be freely downloaded from **http://www.hmisys.com** and use it to subscribe a few data which will be used to animate some graphics object in one of the main tabs.

Setting PeakHMI simulator

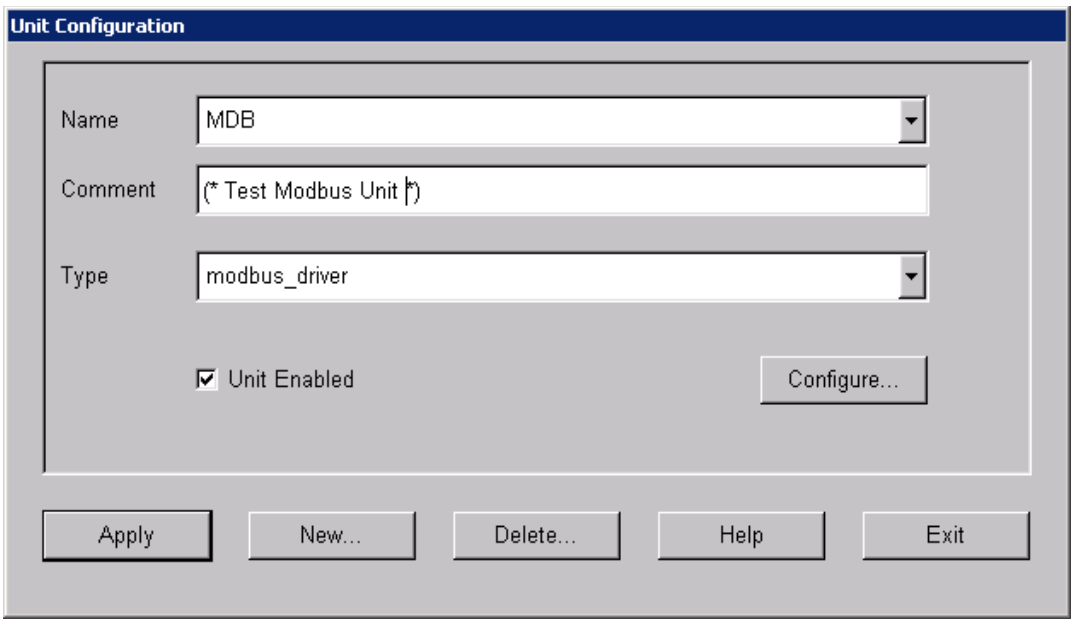
Start **MODBUS TCP slave** and next, set up the communication preferences by click on File->Settings menu



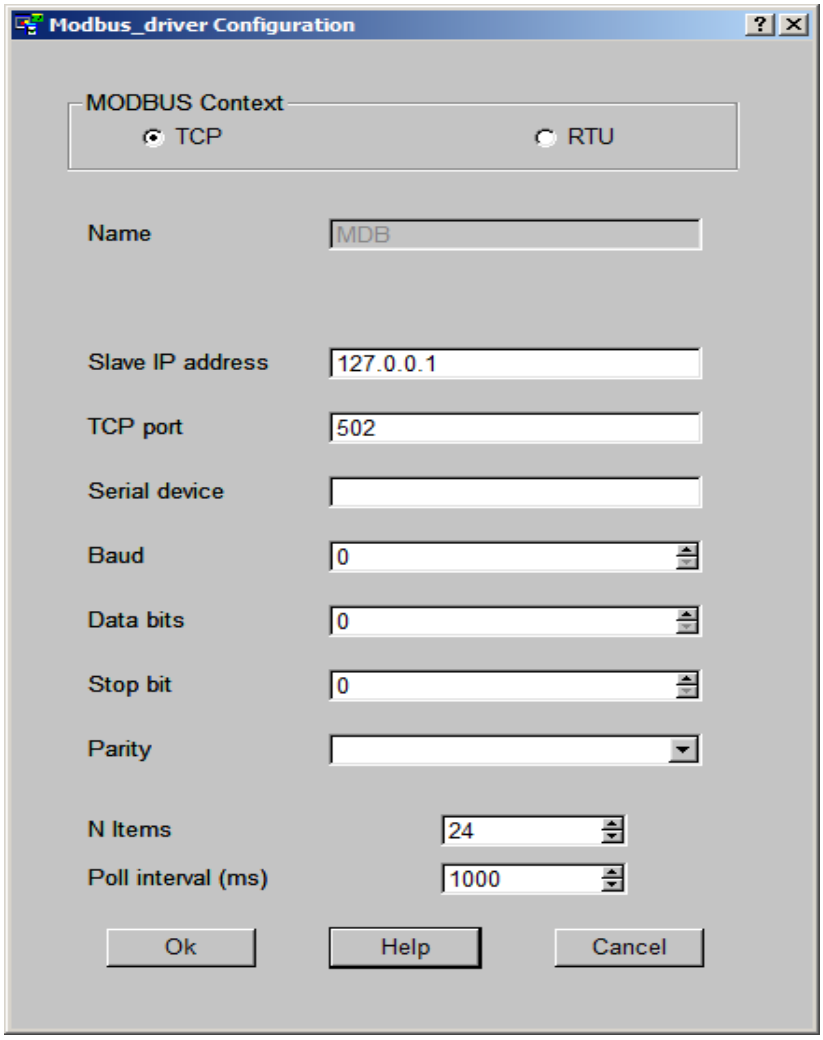


and setting **502** as Port and Our Slave ID = 1.

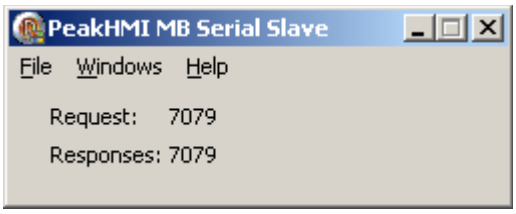
Then let's create a virtual device defined by **IndigoSCADA** as a **Unit**. Select **Configure Units** from the menu **Configure,** press the button **New** and add a new Unit. Name it **MDB** (=MODBUS) for example and enable it by setting the relative checkbox.



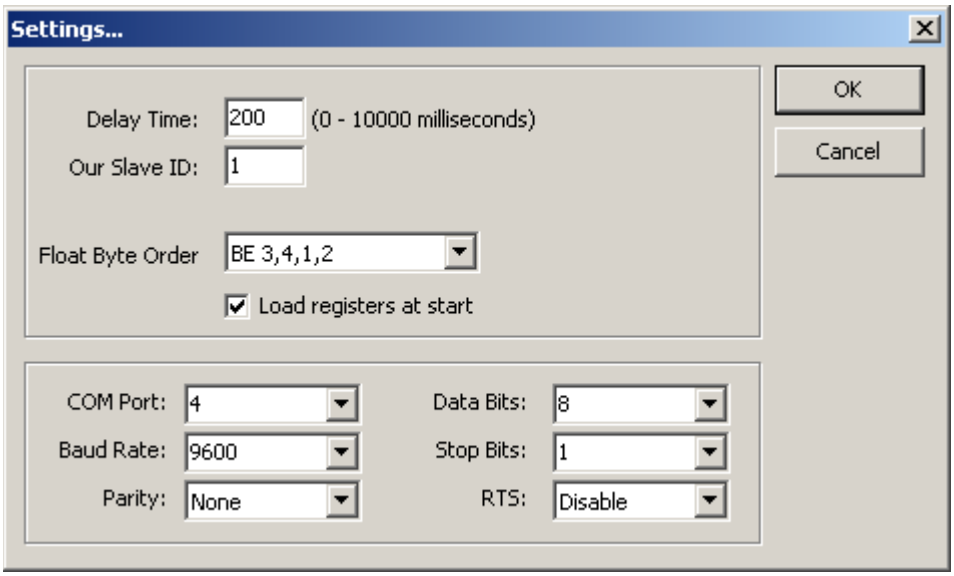
Choose **modbus_driver** as Type and press the **Configure** button to set some specific options. Select **TCP** as Context, **127.0.0.1** as IP address, **502** as Port, **24** is the number of total sampling points (N items) belonging to this unit and **1000** ms as Poll interval.



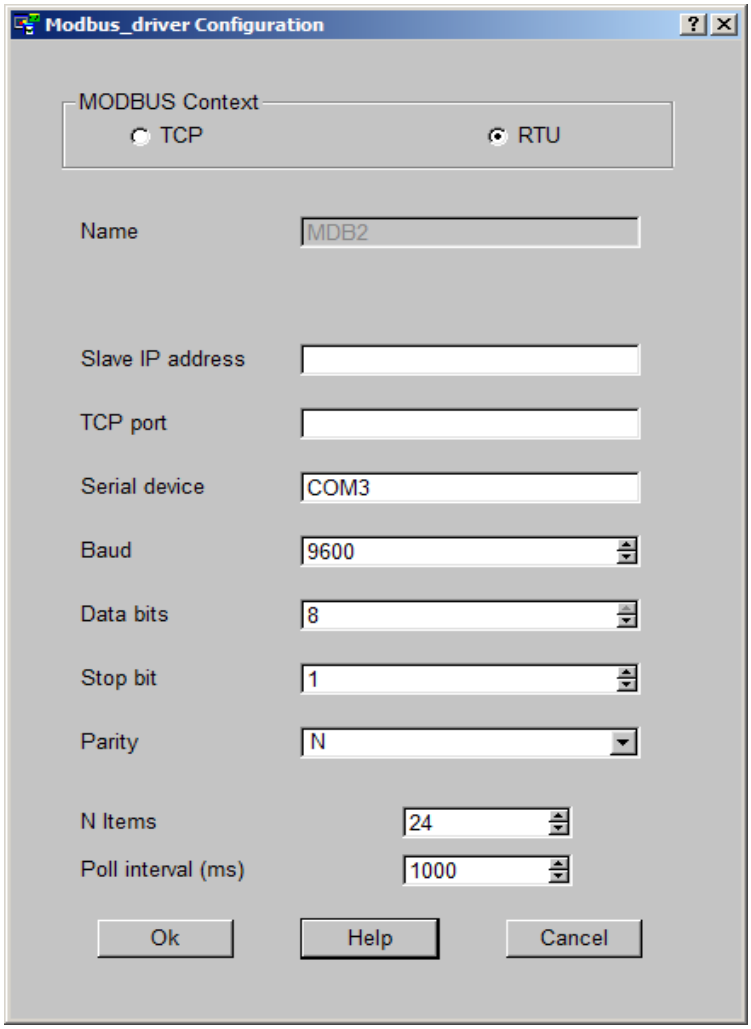
If you want to use a serial slave simulator: start **MODBUS RTU serial slave** simulator:



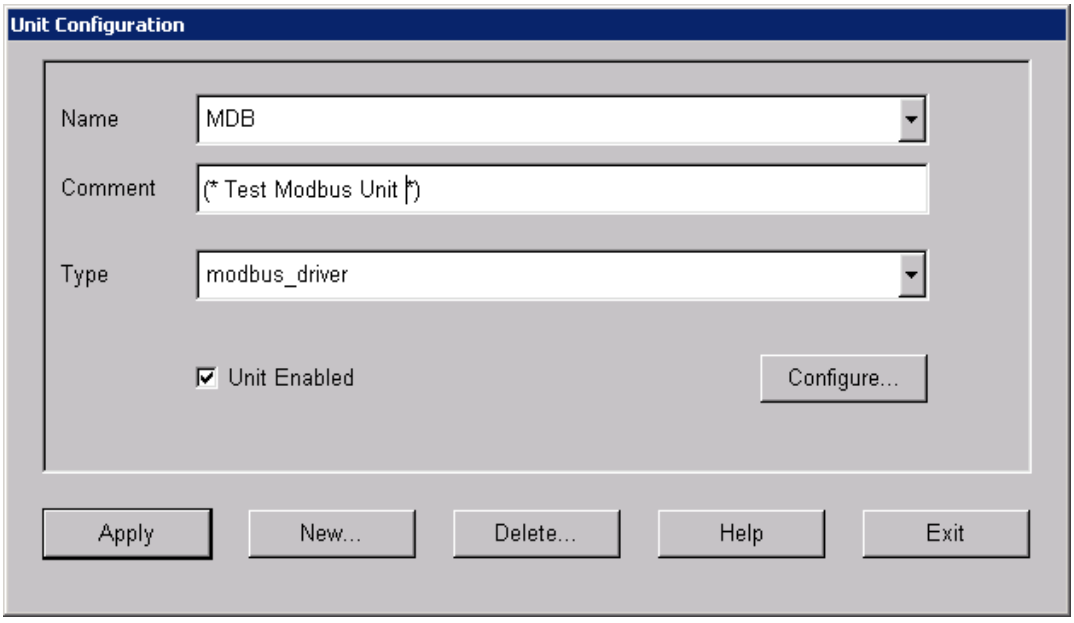
and next, set up the communication preferences by click on File->Settings menu and configure the slave parameters 9600, 8, N, 1, Our Slave ID 1:



In IndigoSCADA select **RTU** as Context, **COM3** as Serial device, **9600** as Baud, **8** as Data bits, **1** as stop bit, **N** as parity (N= None, E= Even, O= Odd parity), **24** is the number of total sampling points (N items) belonging to this unit and **1000** ms as Poll interval.

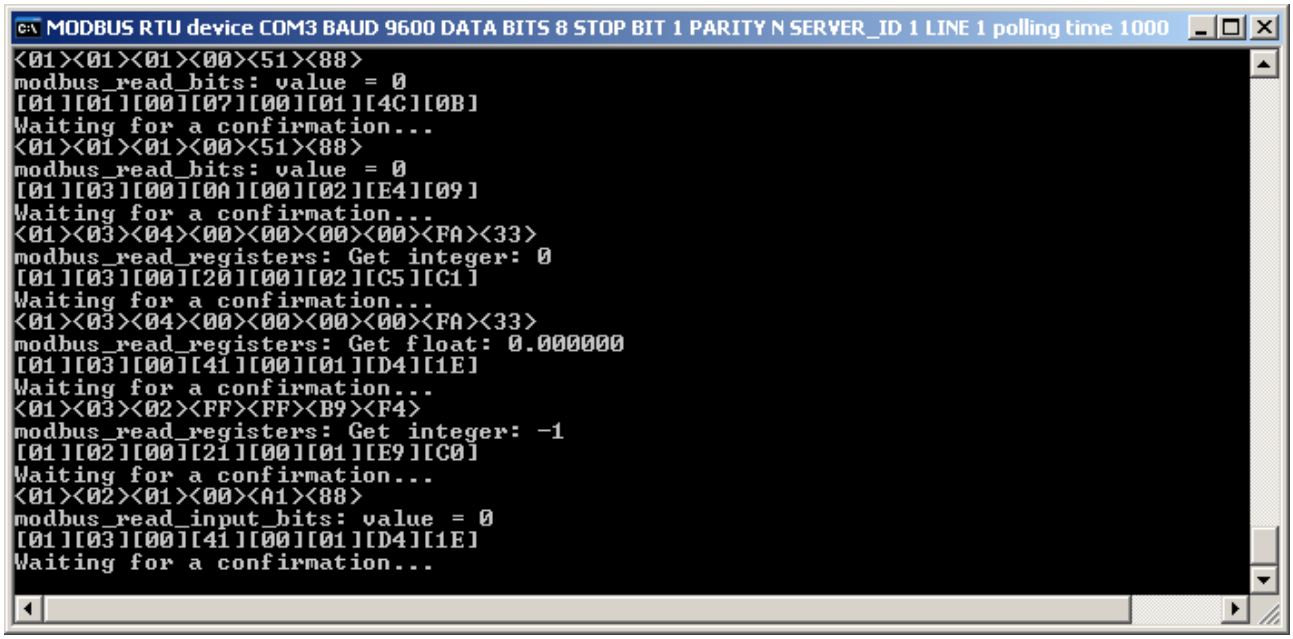


Press **Apply** button and wait it return active, than press **Exit** button.



Choose **Restart Monitor** from the **System Control** menu. Confirm the Restart operation by pressing the **Yes** button.

In the **Modbus** console you will see that the client is requesting data and

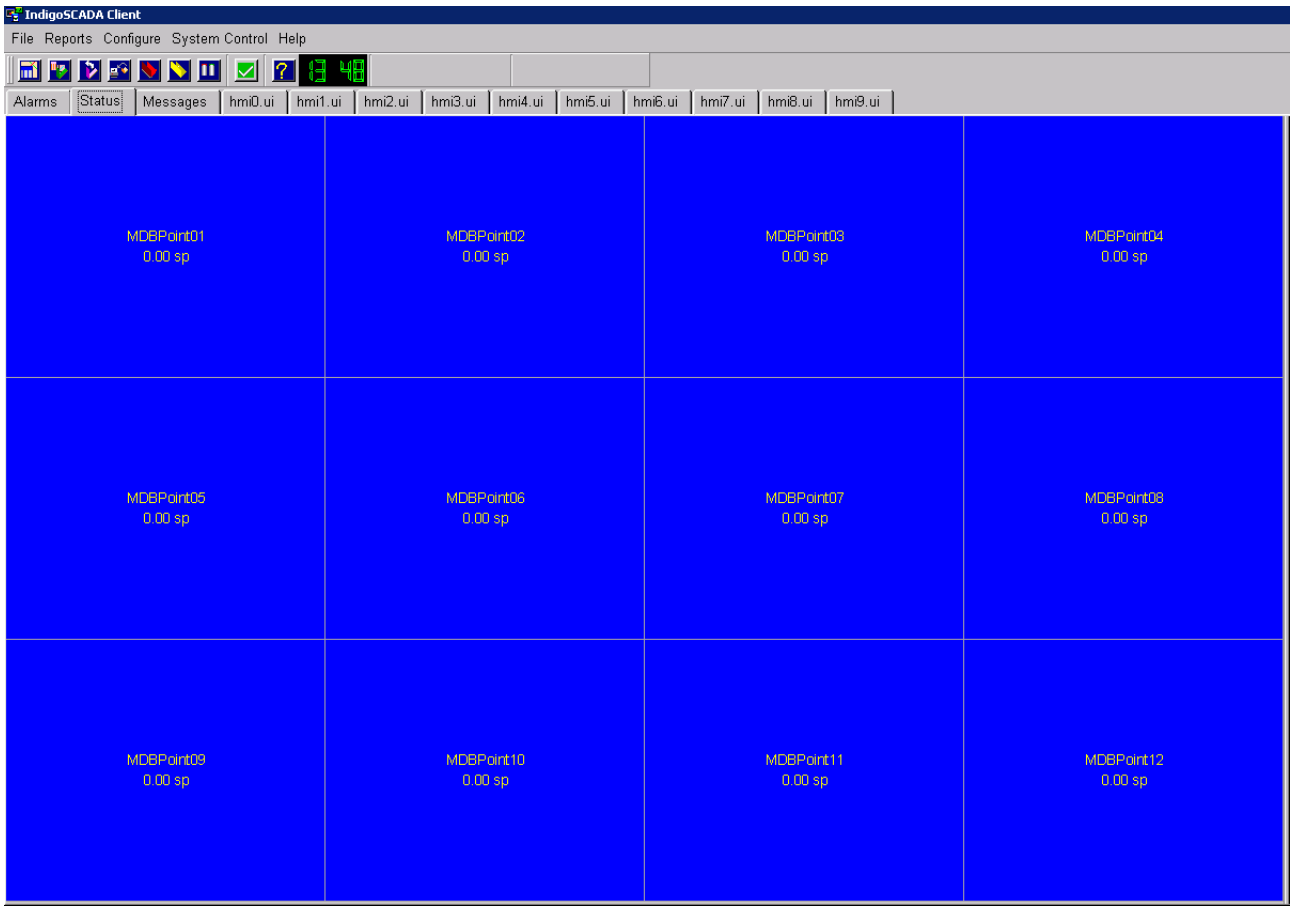


that new **Status tab** is now populated with 24 green rectangles, each one showing the value of one specific **Modbus** data from **MDBPoint01** to **MDBPoint24**.

Reading and writing Modbus parameters

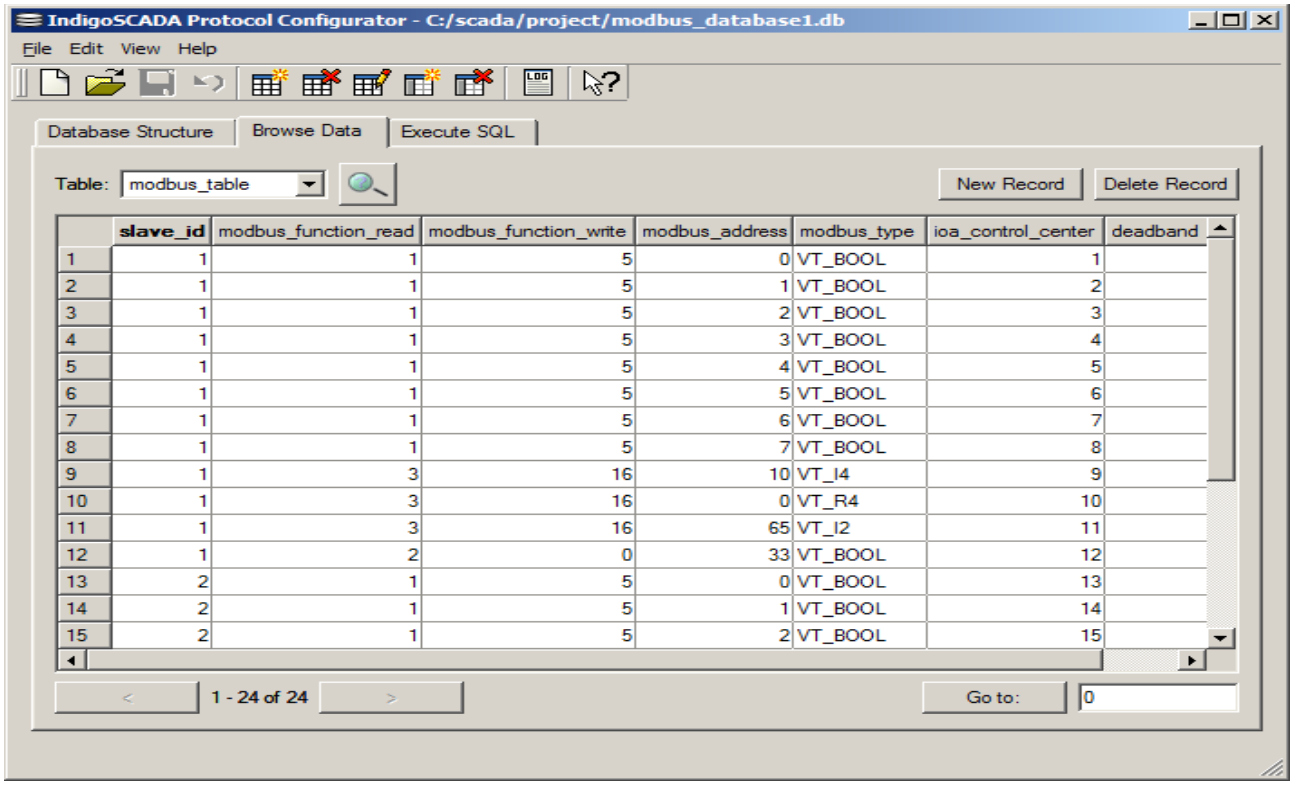
In **IndigoSCADA Status tab**, green is associated with OK status, light blue with not acknowledged alarm states and red with acknowledged alarm states.

If we interrupt the Modbus communication in **PeakHMI**, then all the rectangles will become blue because data values are now invalid.



We used 24 items of data in the Unit options **MDB** because **IndigoSCADA** comes already with a predefined set of 24 Modbus points, (named **MDBPoint01, MDBPoint02**, etc.) for quickly test it with **PeakHMI.**

Select **Configure Protocol** from the **Configure** menu and then open the file **C:\scada\project\modbus_database1.db.** Select the **Browse Data** tab and look to the predefined definitions of the 24 points.



	slave_id	modbus_function_read	modbus_function_write	modbus_address	modbus_type	ioa_control_center	deadband
1	1	1	5	0	VT_BOOL	1	
2	1	1	5	1	VT_BOOL	2	
3	1	1	5	2	VT_BOOL	3	
4	1	1	5	3	VT_BOOL	4	
5	1	1	5	4	VT_BOOL	5	
6	1	1	5	5	VT_BOOL	6	
7	1	1	5	6	VT_BOOL	7	
8	1	1	5	7	VT_BOOL	8	
9	1	3	16	10	VT_I4	9	
10	1	3	16	0	VT_R4	10	
11	1	3	16	65	VT_I2	11	
12	1	2	0	33	VT_BOOL	12	
13	2	1	5	0	VT_BOOL	13	
14	2	1	5	1	VT_BOOL	14	
15	2	1	5	2	VT_BOOL	15	

The first five columns refers to typical Modbus parameters. For every point, it is defined the Slave_id, Read function, the Write Function, Address and so on.

The other column is specific for the IEC 870-5-104 protocol because every signal is treated internally by **IndigoSCADA** as an IEC 870-5-104 protocol data.

The first record, ioa_control_center 1 is a Boolean data with modbus_address 0,

modbus function 1 is used so this point is a modbus coil output.

The column modbus_address is an offset relative to the first address of the function used.

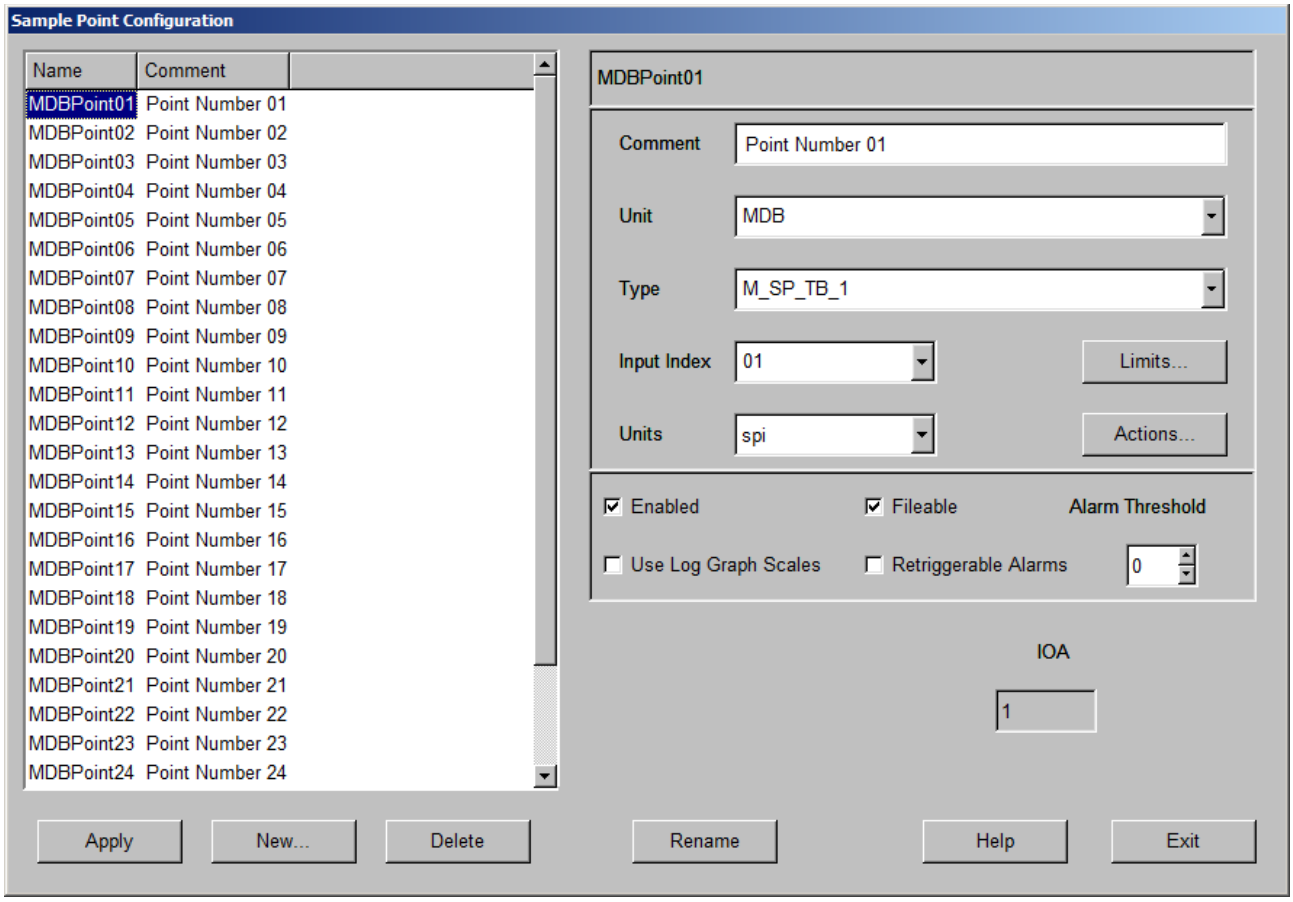
Record 1 has modbus_address 0, coils function 1 is used, so add the first coils address (000001) and you get actual modbus address 000001;

Record 11, has modbus_address 65, holding registers function 3 is used, so add the first holding registers address (400001) and you get actual modbus address 400066;

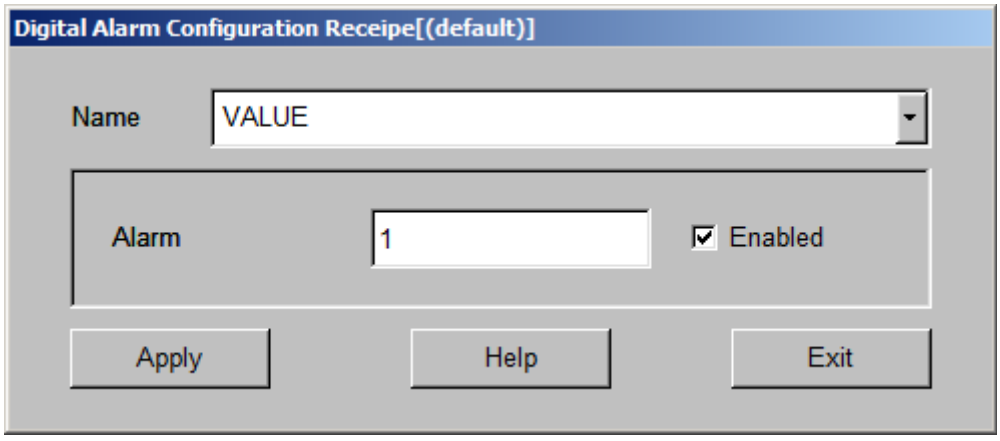
Record 12, has modbus_address 33, discrete inputs function 2 is used, so add the first discrete inputs address (100001) and you get actual modbus address 100034;

Activate alarming for MDBPoint01 with the following procedure:

MDBPont01 is a single point (M_SP_TB_1)

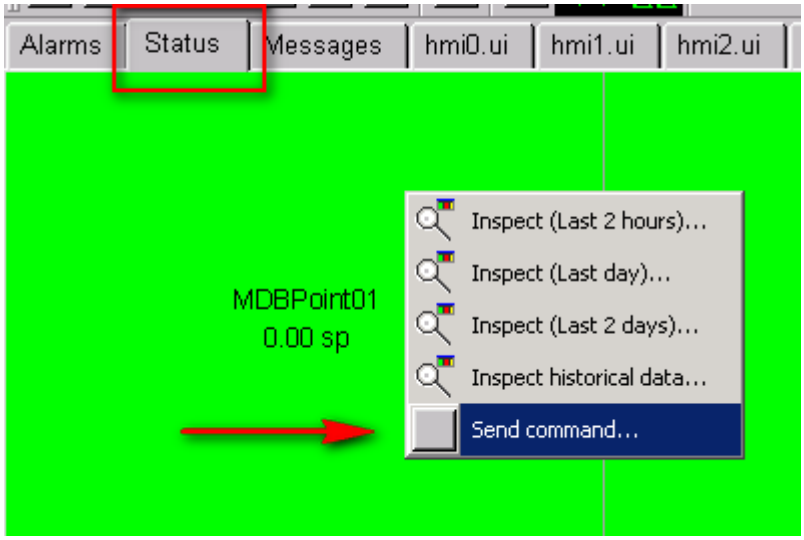


press “Limits...” button and set the value of the single point (0 or 1) to associate with the alarm state. In the next picture the value of 1 gives an alarm state.



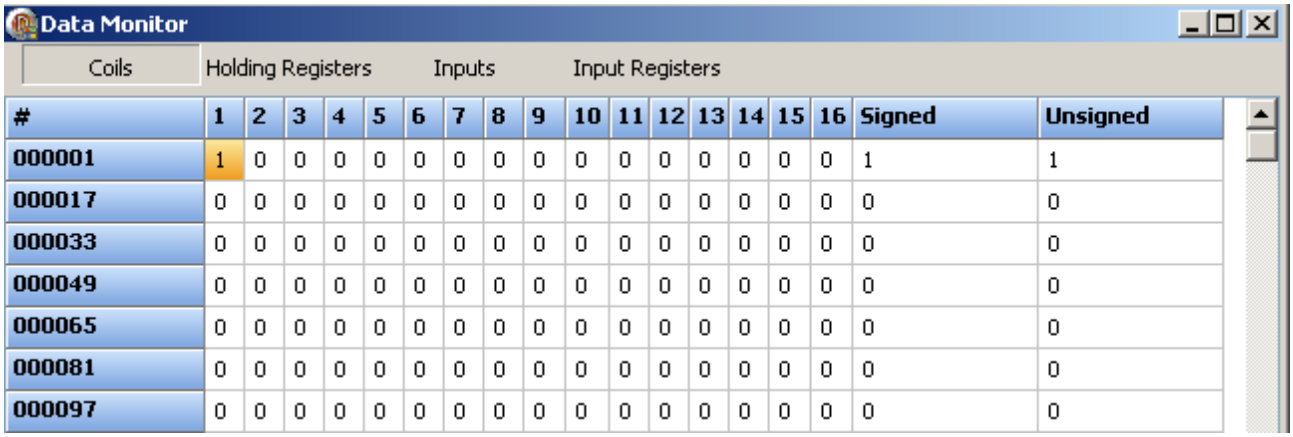
Restart monitor:

Return to the **Status tab,** right-click on the first green rectangle and choose **Send Command.**



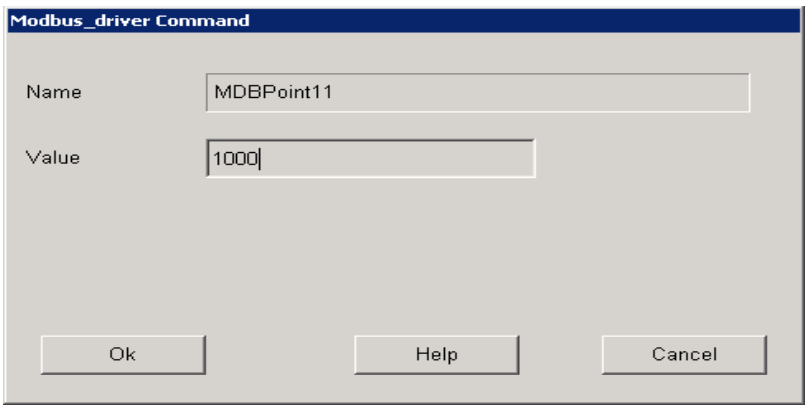
Write 1 in the **Value** text box and press **OK** button. The rectangle will become light blue and in the right part of the window a new Alarm will be notified.

Return back to **PeakHMI** and you will see that the data value of coils with offset zero (mapped with **MDBPoint01**) is now 1.



#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Signed	Unsigned
000001	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
000017	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
000033	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
000049	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
000065	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
000081	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
000097	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The same operation can be executed with **MDBPoint11**, which is an integer type. Send a command of 1000 for example,



and then return to the **PeakHMI** and select the **Holding Registers.**

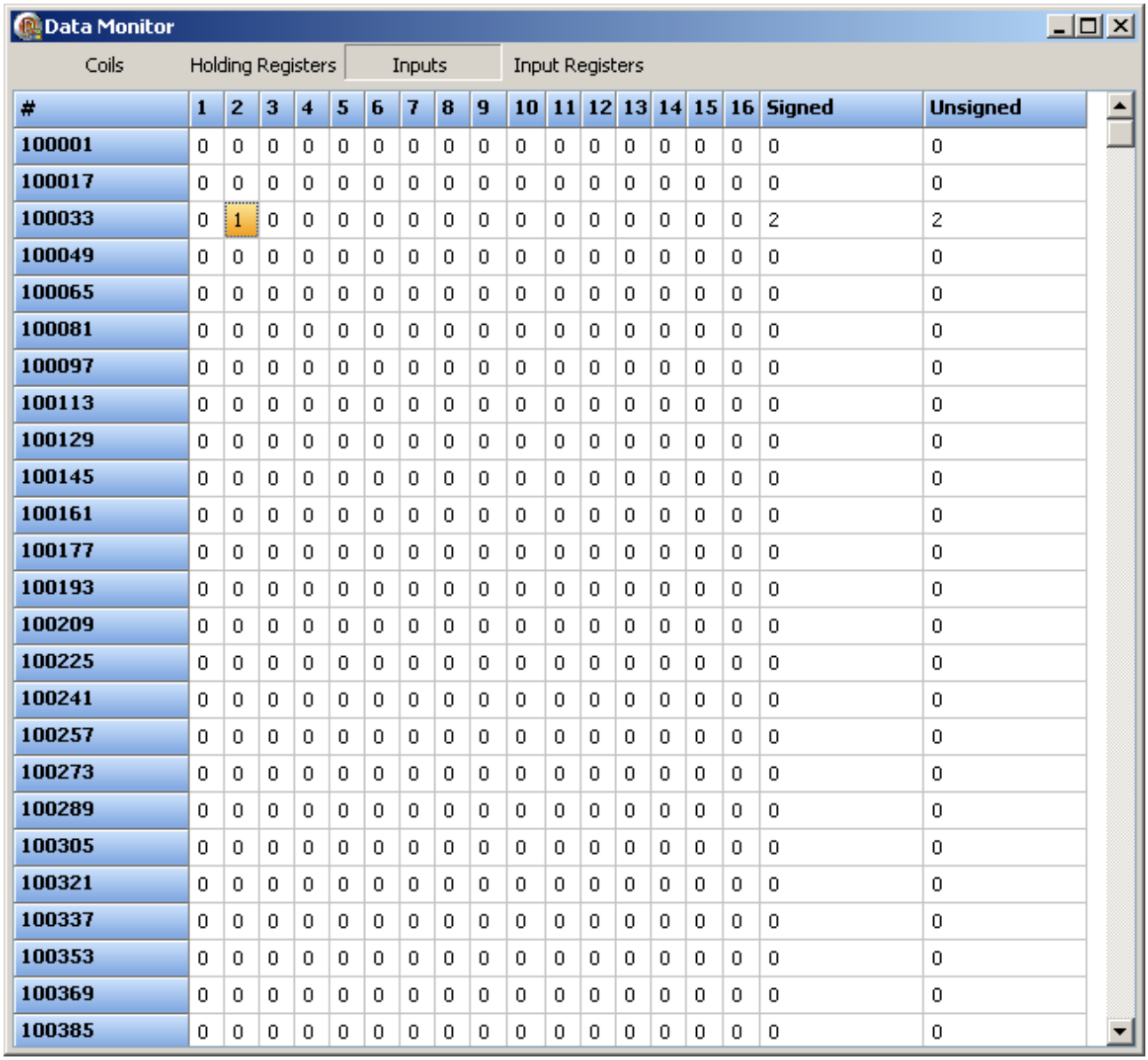
Data Monitor

Coils | Holding Registers | Inputs | Input Registers

#	Signed	Unsigned	Float	16	15	14	13	12	11	10	9	8	7
400050	0	0	00000000	0	0	0	0	0	0	0	0	0	0
400051	0	0	0	0	0	0	0	0	0	0	0	0	0
400052	0	0	00000000	0	0	0	0	0	0	0	0	0	0
400053	0	0	0	0	0	0	0	0	0	0	0	0	0
400054	0	0	00000000	0	0	0	0	0	0	0	0	0	0
400055	0	0	0	0	0	0	0	0	0	0	0	0	0
400056	0	0	00000000	0	0	0	0	0	0	0	0	0	0
400057	0	0	0	0	0	0	0	0	0	0	0	0	0
400058	0	0	00000000	0	0	0	0	0	0	0	0	0	0
400059	0	0	0	0	0	0	0	0	0	0	0	0	0
400060	0	0	00000000	0	0	0	0	0	0	0	0	0	0
400061	0	0	0	0	0	0	0	0	0	0	0	0	0
400062	0	0	00000000	0	0	0	0	0	0	0	0	0	0
400063	0	0	0	0	0	0	0	0	0	0	0	0	0
400064	0	0	00000000	0	0	0	0	0	0	0	0	0	0
400065	0	0	1,40129846432482E-42	0	0	0	0	0	0	0	0	0	0
400066	1000	1000	E8030000	0	0	0	1	0	1	1	1	1	1
400067	0	0	0	0	0	0	0	0	0	0	0	0	0
400068	0	0	00000000	0	0	0	0	0	0	0	0	0	0
400069	0	0	0	0	0	0	0	0	0	0	0	0	0
400070	0	0	00000000	0	0	0	0	0	0	0	0	0	0
400071	0	0	0	0	0	0	0	0	0	0	0	0	0
400072	0	0	00000000	0	0	0	0	0	0	0	0	0	0
400073	0	0	0	0	0	0	0	0	0	0	0	0	0

The value of the 400066th register will be **1000.**

In **PeakHMI** select **Inputs** and change the value 100034 to get a change in the value of **MDBPoint12**

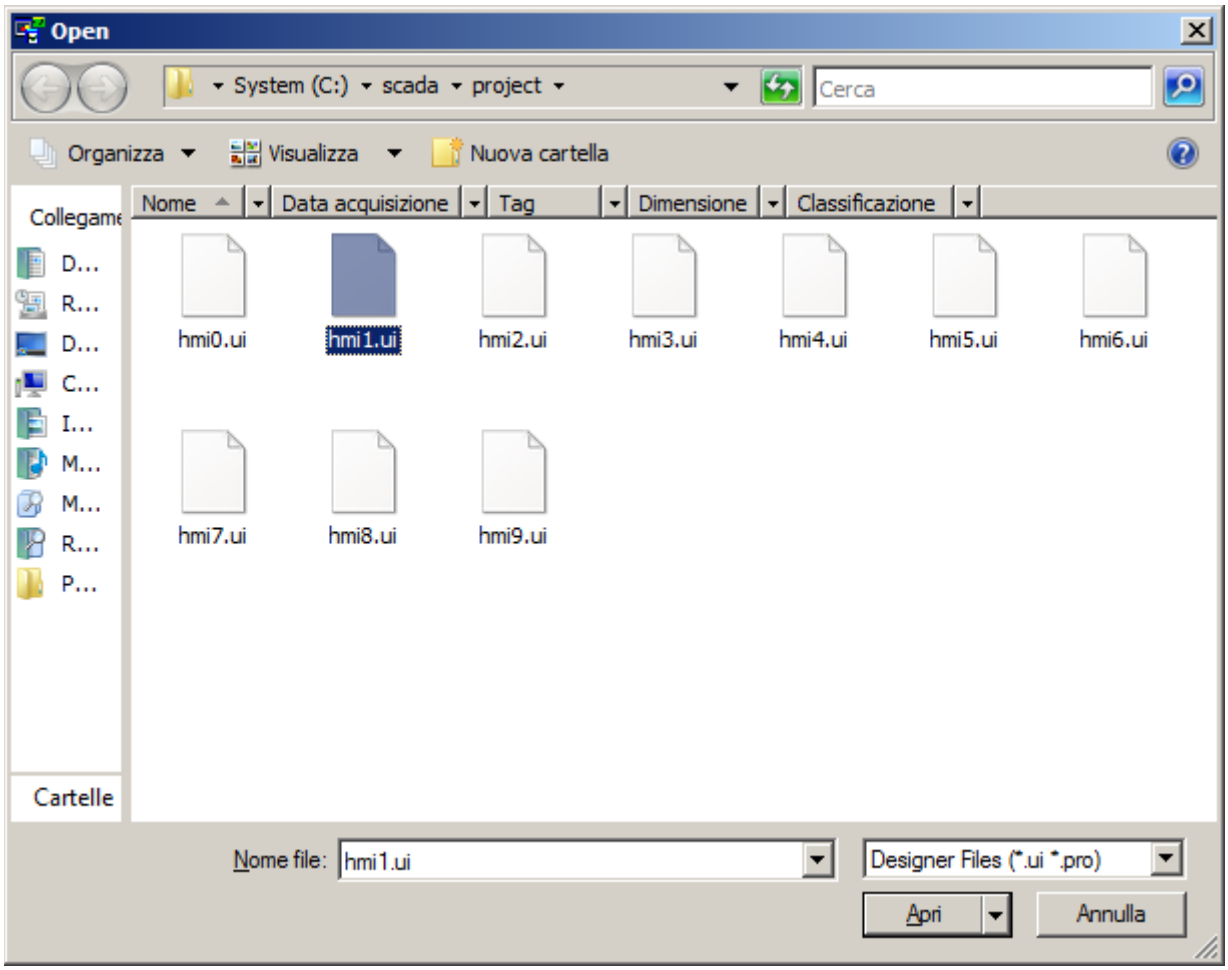


#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Signed	Unsigned
100001	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100017	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100033	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2
100049	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100065	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100081	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100097	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100113	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100129	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100145	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100161	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100177	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100193	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100209	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100225	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100241	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100257	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100273	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100289	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100305	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100321	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100337	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100353	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100369	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100385	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

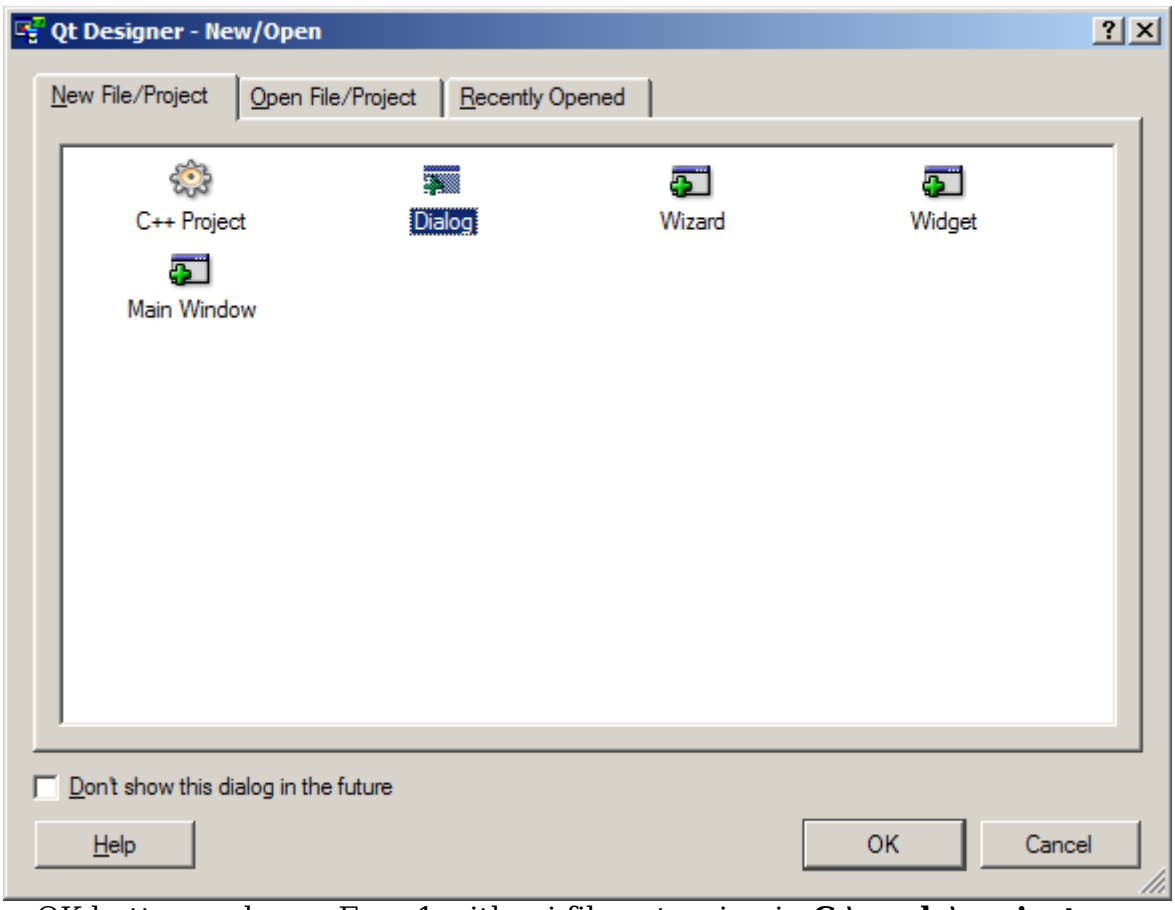
Animate graphics objects with process data

Binding a graphic object with a process data is very easy. Let's begin with a Boolean type (True/False) to animate three object: a rectangular led, a counter and circular led.

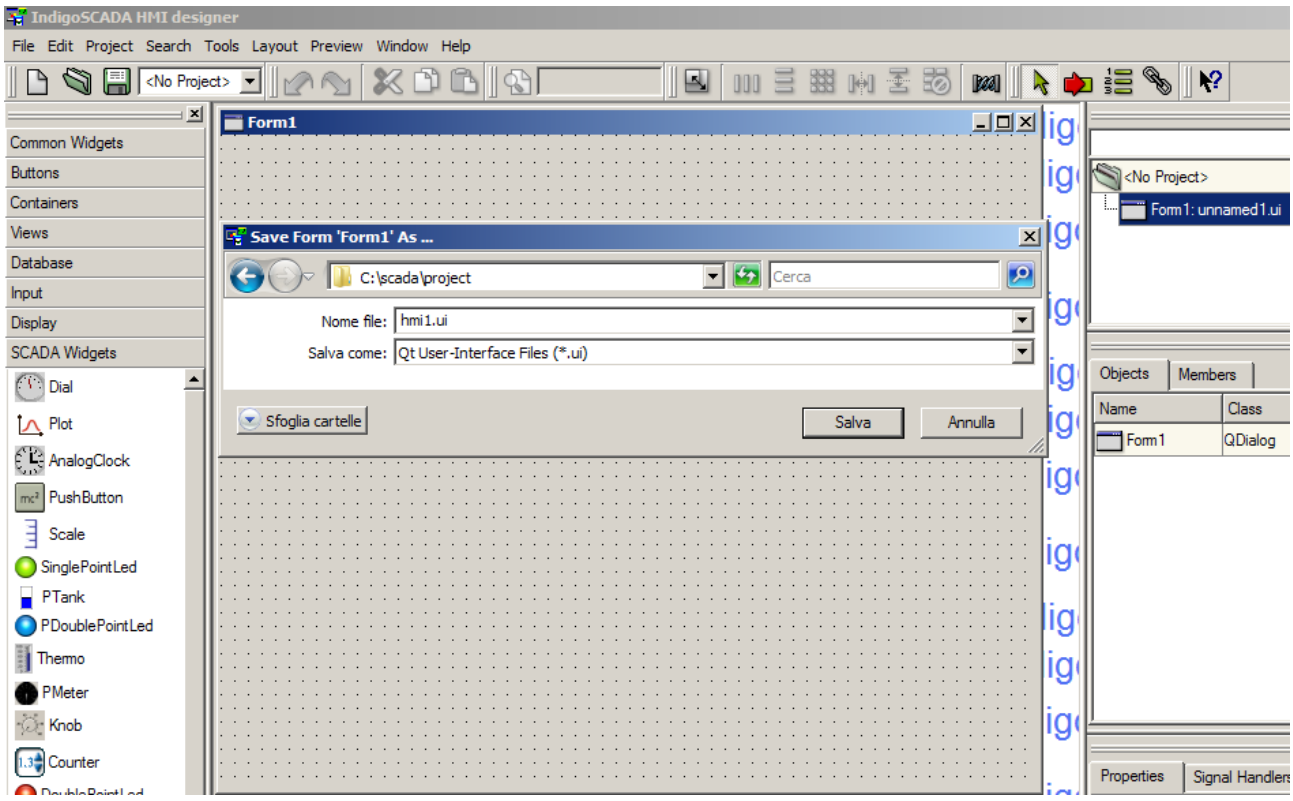
Open the **HMI Designer** by choosing **Configure HMI** from the **Configure** menu, select the **OpenFile/Project Tab** and open one of the .ui files present in **C:\scada\project** (**hmi1.ui** for example)



or create a new dialog:

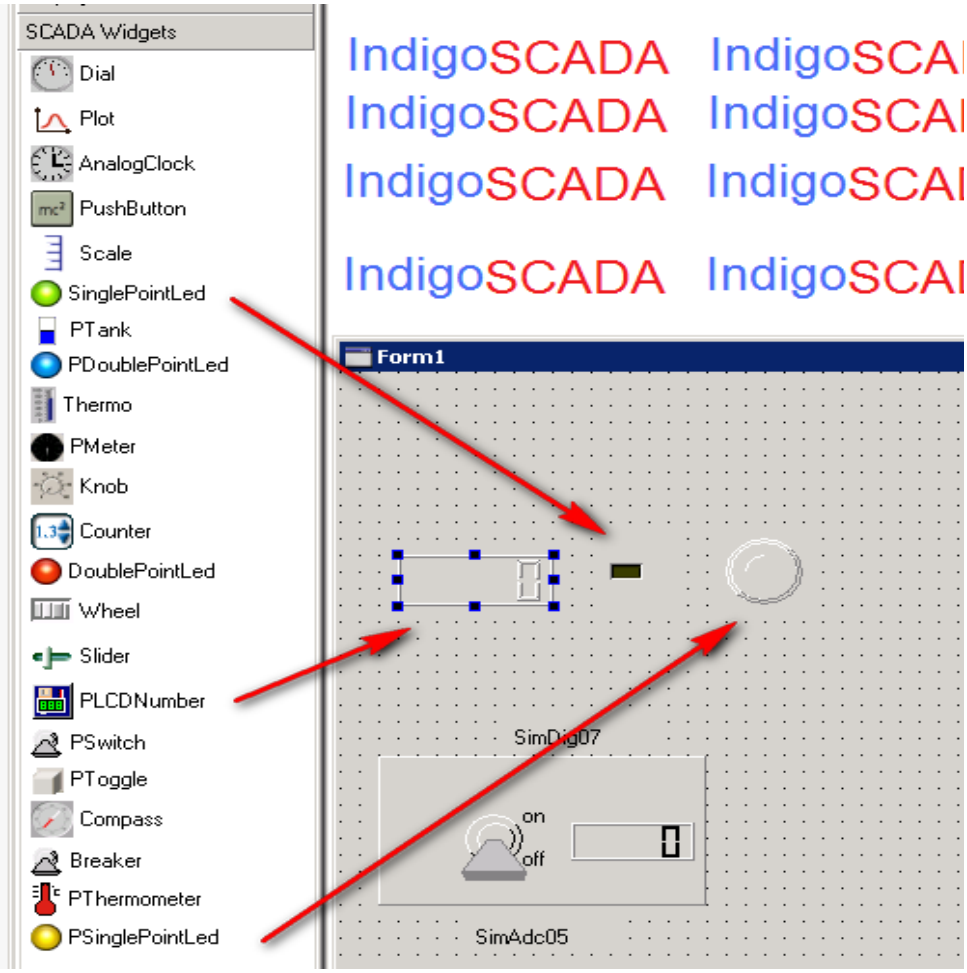


Press OK button and save Form1 with .ui file extension in **C:\scada\project**

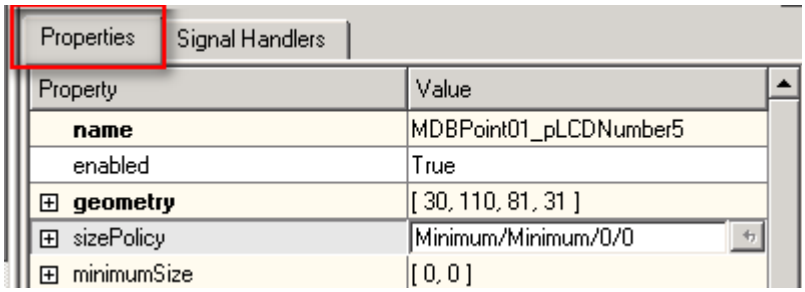


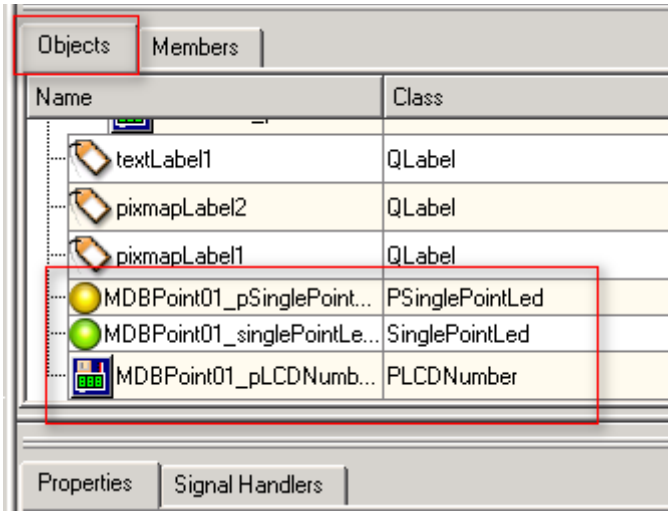
Then, “draw” these three objects from the **SCADA Widgets** toolbar onto the map:

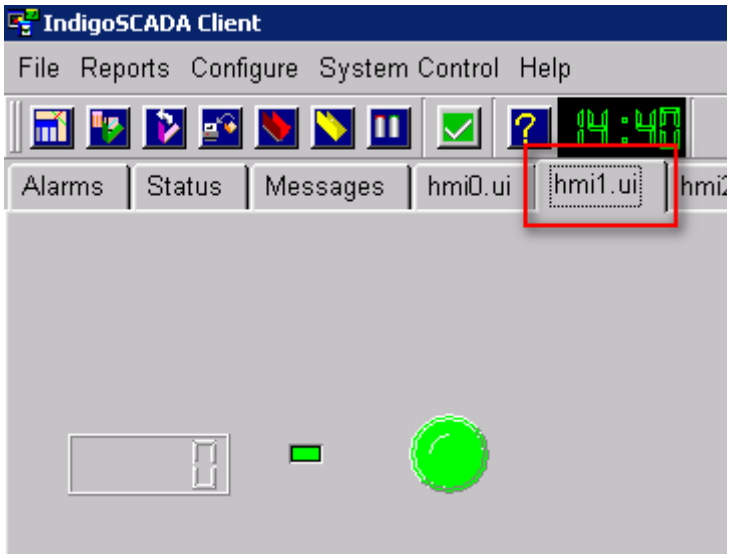
- PLCDNumber
- SinglePointLed
- PsinglePointLed



Rename these objects with format UnitNamePointXX_oldname (es. **pLCDNumber5** becomes **MDBPoint01_pLCDNumber5** and so on) using the Property Palette. Do this for all three objects.

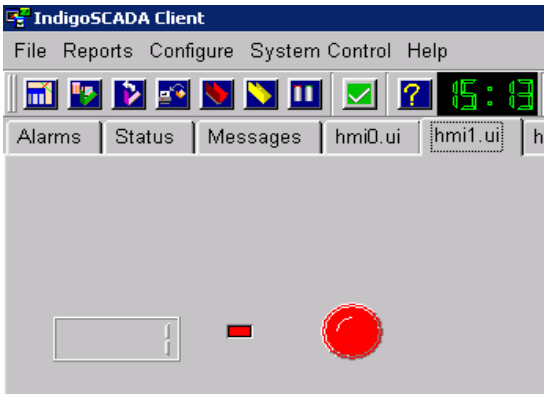






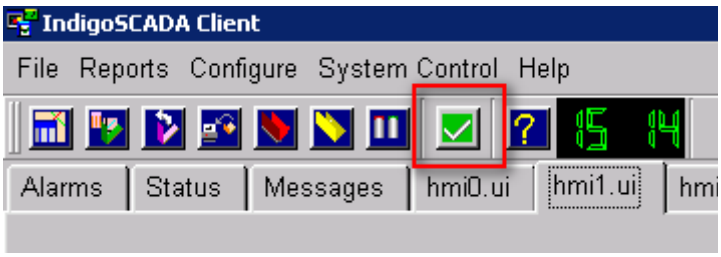
Save the map and exit **IndigoSCADA Client** by choosing **Exit** from the **File** menu, After few seconds, the client will be opened again and in the **hmi1.ui** tab those three new objects will be inside.

Now open the **PeakHMI** Data monitor window, select the Windows->Register data menu and change values of the coils

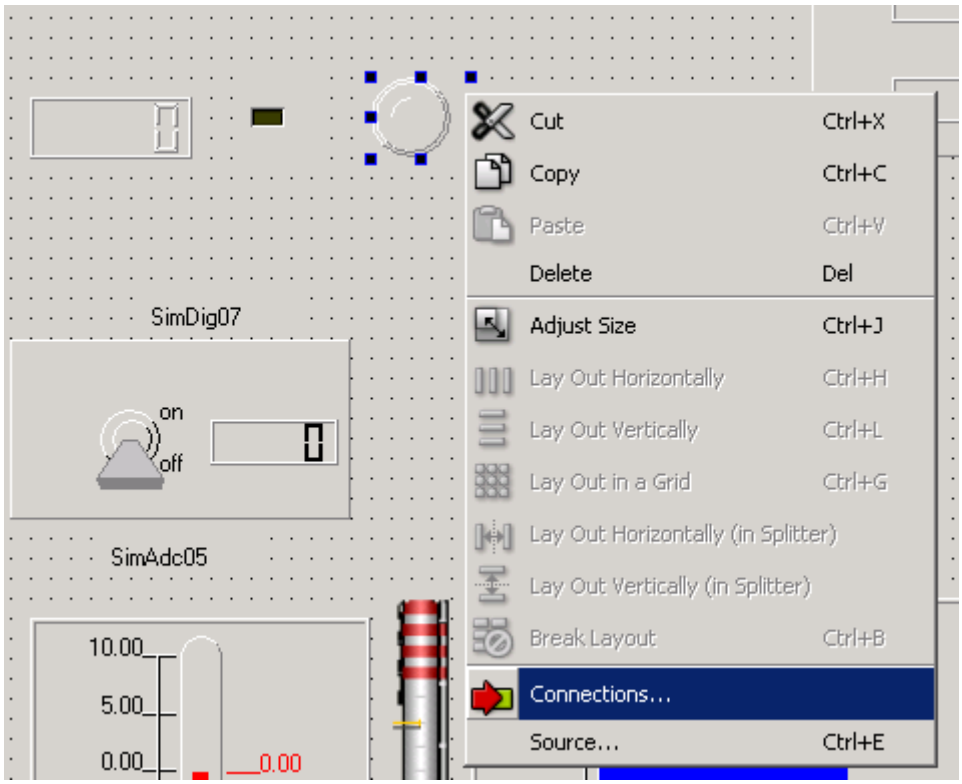


The graphic object will begin to flash because Point is in alarm state.

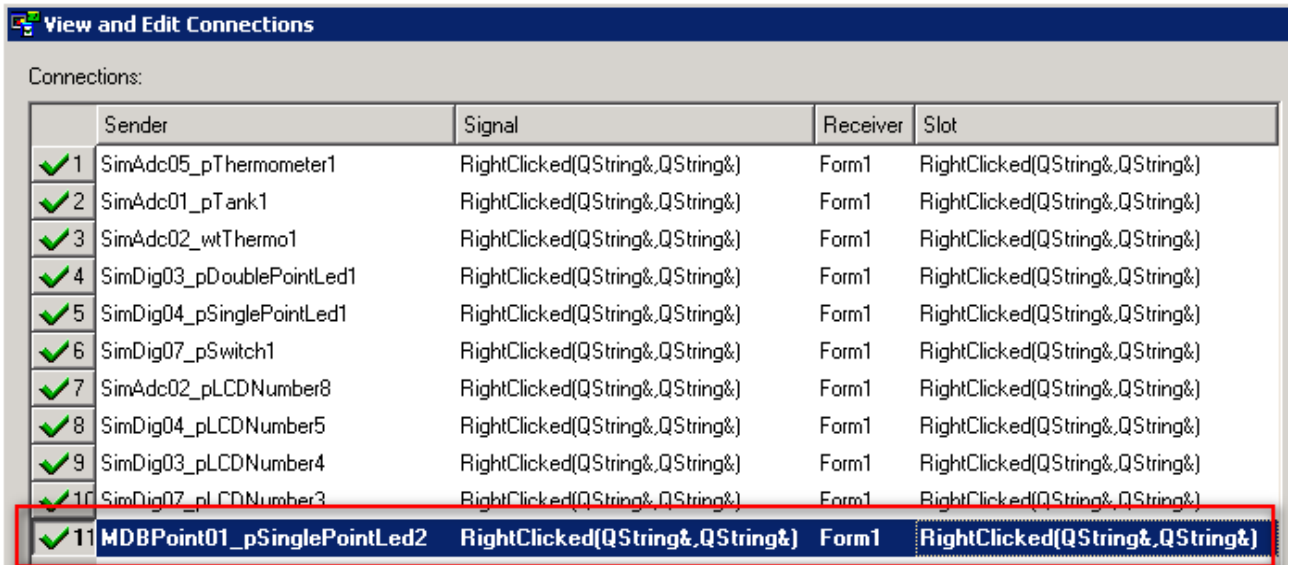
Press the **Acknowledge All Alarms** button to stop the animation.



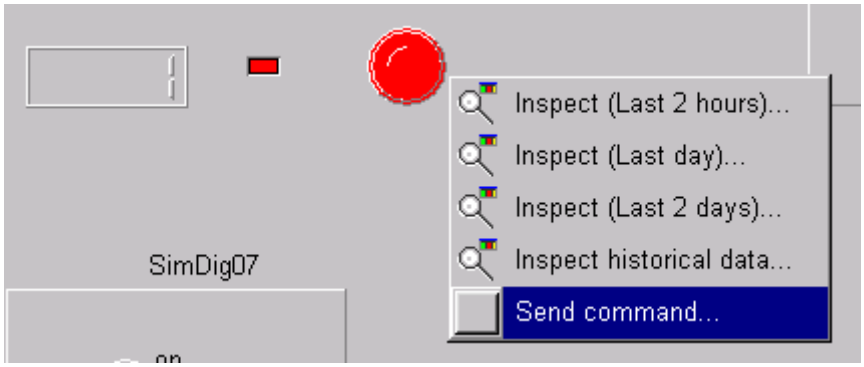
At the moment there is no contextual menu associated with these object: pressing the right button on it won't do nothing. So, now we will enable it. Open again the map **hmi1.ui** with the **HMI Designer**, right-click on the pSinglePointLEd and choose **Connections**

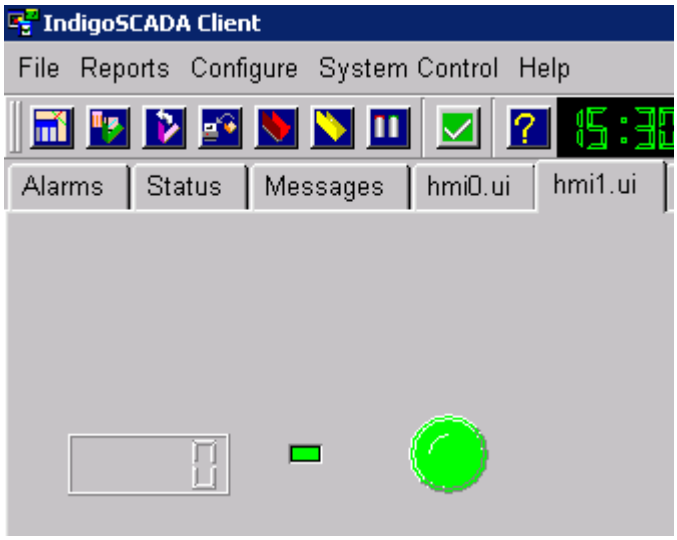


Add a new row as shown in the picture below.



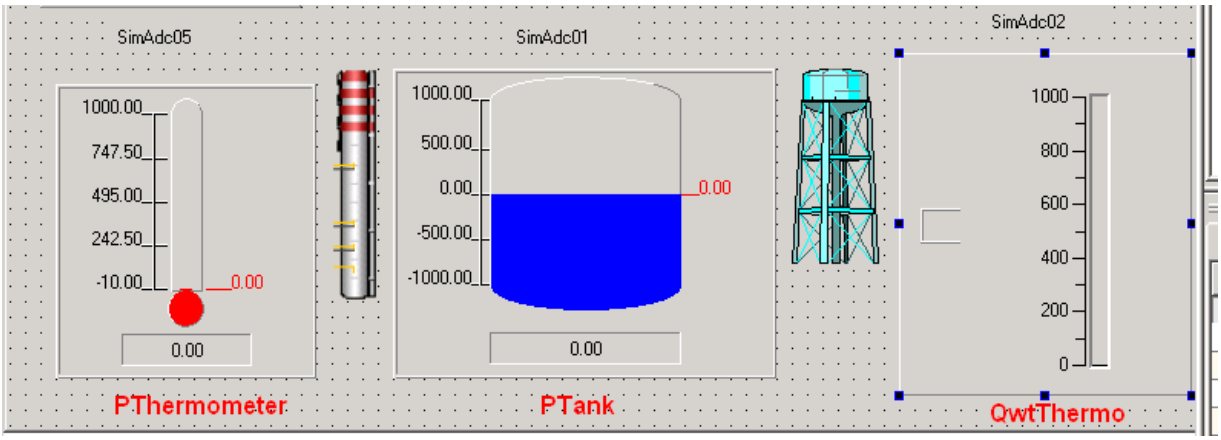
Save the map, close the **IndigoSCADA Client**, and now it should be possible to interact with the object. Right-click on the object and send a command with a value of zero.

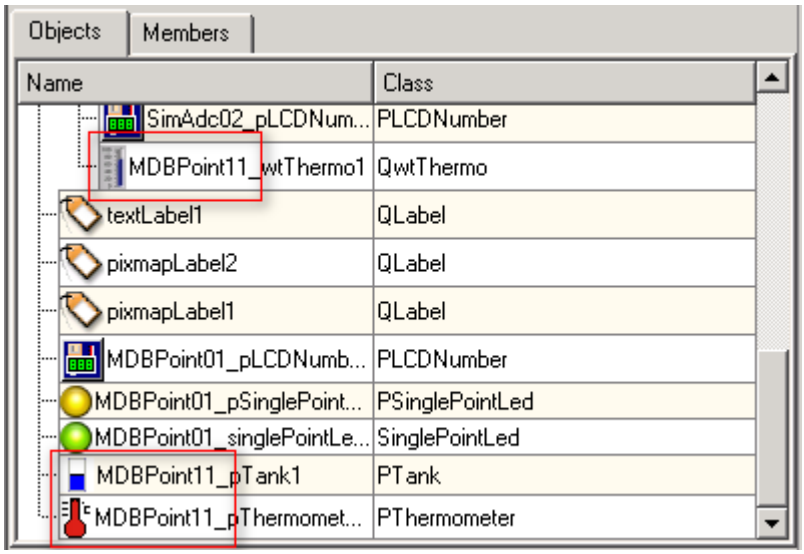




Using the same procedure, it is possible to animate graphic objects connecting them with analogue process data.

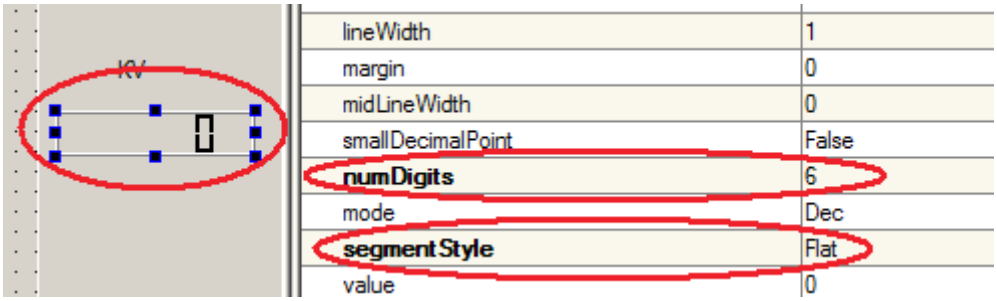
Open the **HMI Designer**, and rename the three objects **Pthermometer, Ptank** and **QwtThermo** using the formula UnitNamePointXX_oldname**.**





Set type of MDBPoint11 to M_ME_TE_1, accordingly to **modbus_database1.db** column iec_type_read.

The PLCDNumber object require the setting of **numDigits,** 6 digits to display correctly **MDBPoint11** which is a 16 bits signed integer (VT_I2).

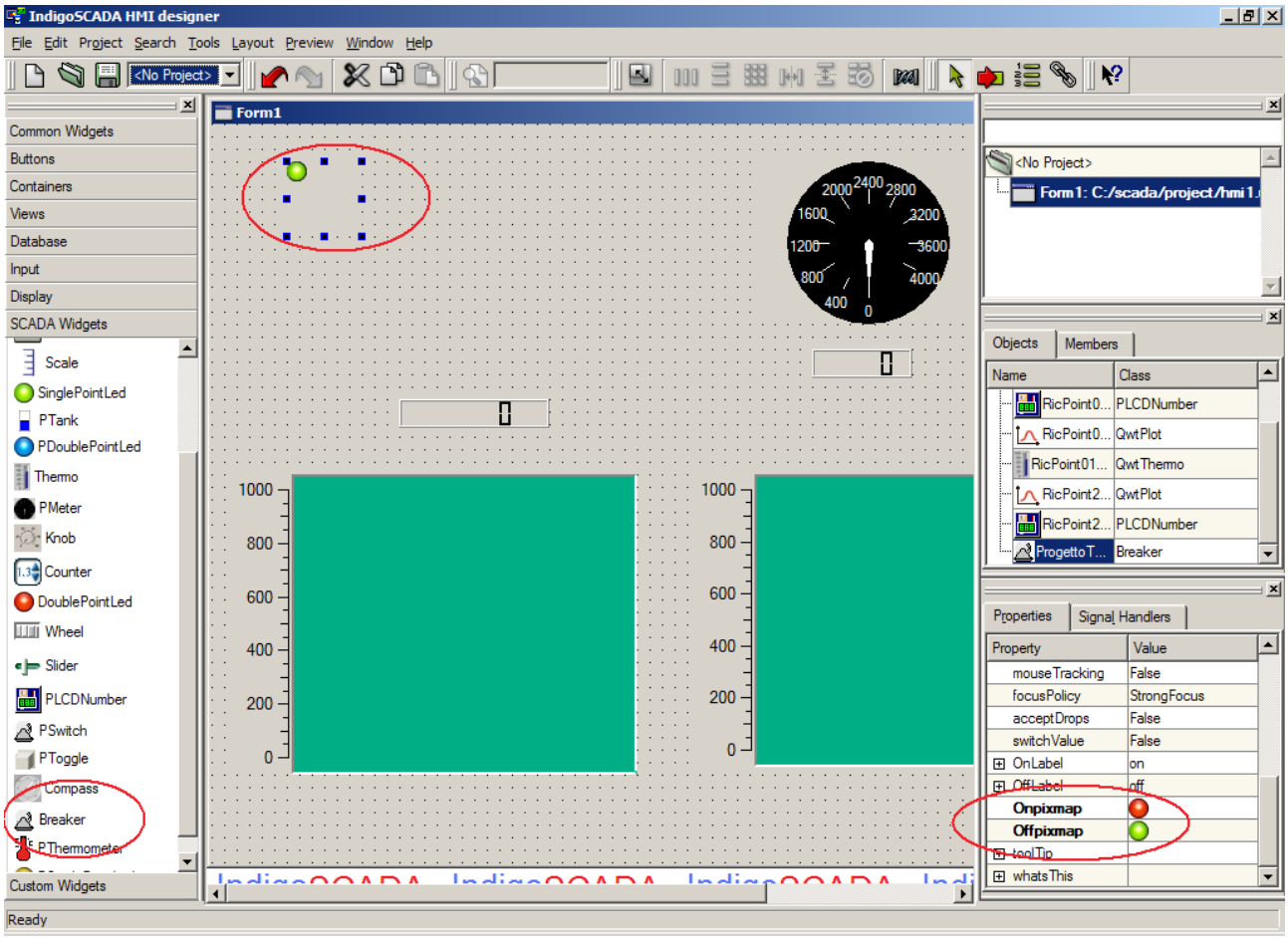


Dinamic bitmaps

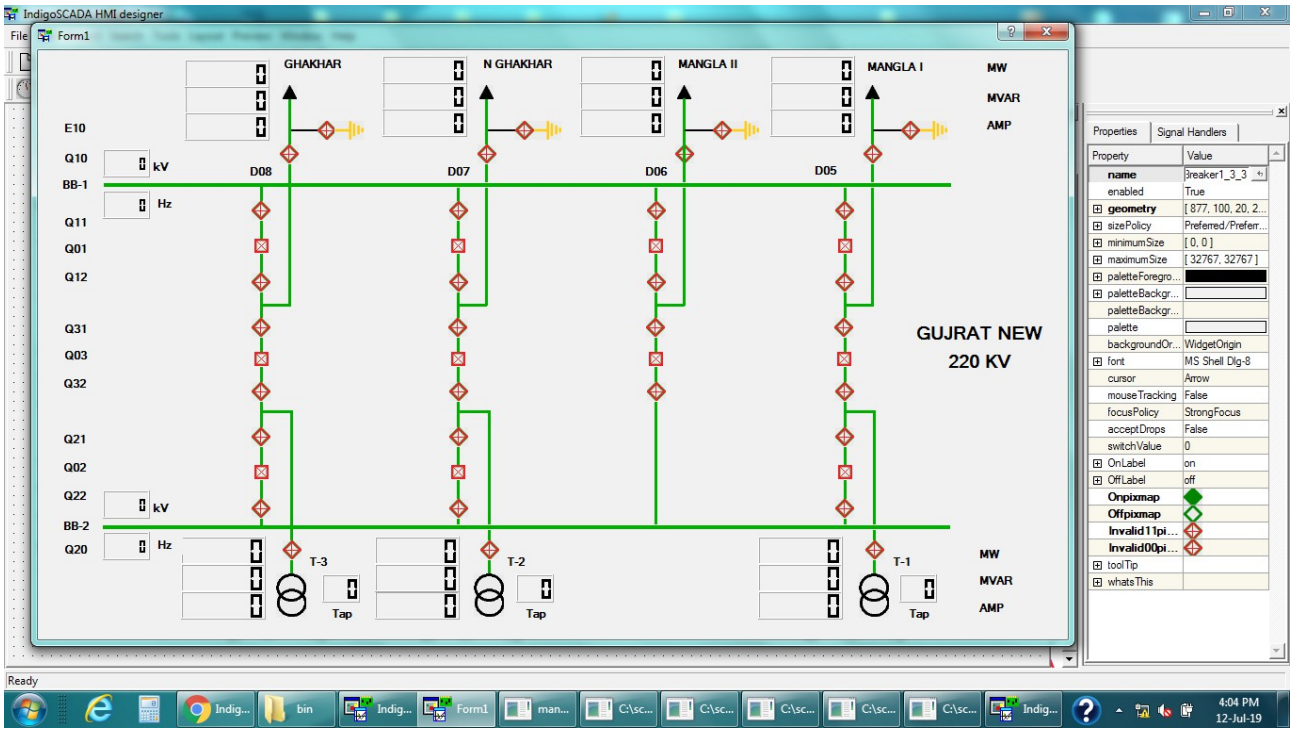
Use the **Breaker** widget to animate single point bitmaps. The bitmaps are selected in Properties Onpixmap and Offpixmap.

Onpixmap is shown in HMI when the value of the point is 1.

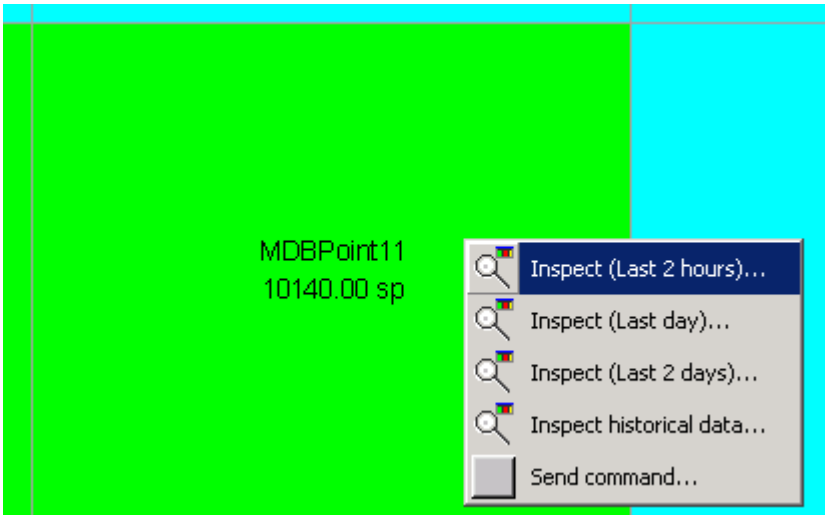
Offpixmap is shown in HMI when the value of the point is 0.



Use **DoubleBreaker** widget to animate double points bitmaps. The bitmaps are selected in Properties Onpixmap, Offpixmap, Invalid11pixmap and Invalid00pixmap.
Onpixmap is shown in HMI when the value of the point is 2.
Offpixmap is shown in HMI when the value of the point is 1.
Invalid11pixmap is shown in HMI when the value of the point is 3.
Invalid00pixmap is shown in HMI when the value of the point is 0.

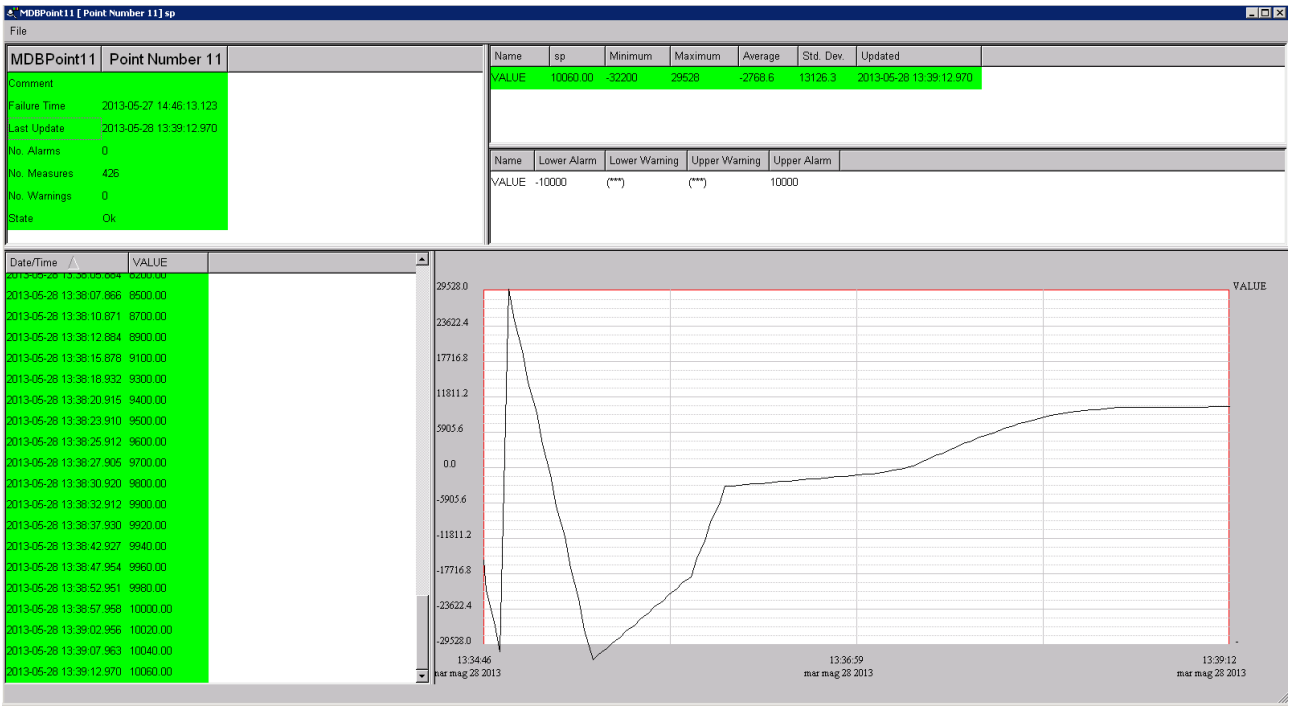


Historical data analysis



IndigoSCADA Client allows to inspect data in order to analyze the data and observe the trend of the process data. Go to the **Status** tab and right-click inside one the rectangles representing process data (e.g **MDBPoint11**) and choose **Inspect last two hours** for example.

The ongoing trend of **MDBPoint11** will be shown in a new window.

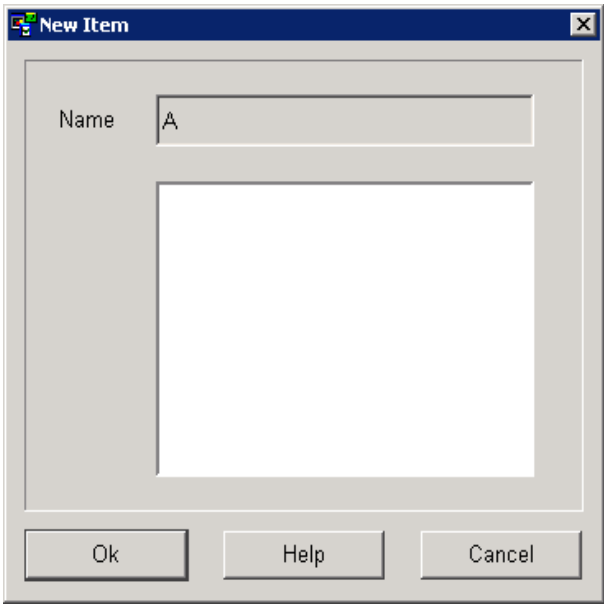


Alarms and Alarms group

As seen above, **IndigoSCADA Client** uses color to show the process data status. Sometimes, hovewer, it would be necessary to monitor many points at once and have one calculated point which is the OR combination of all these point. This point could summarize those point for example. This functionality is carried out by **IndigoSCADA client** using the **Alarm Group functionality**.

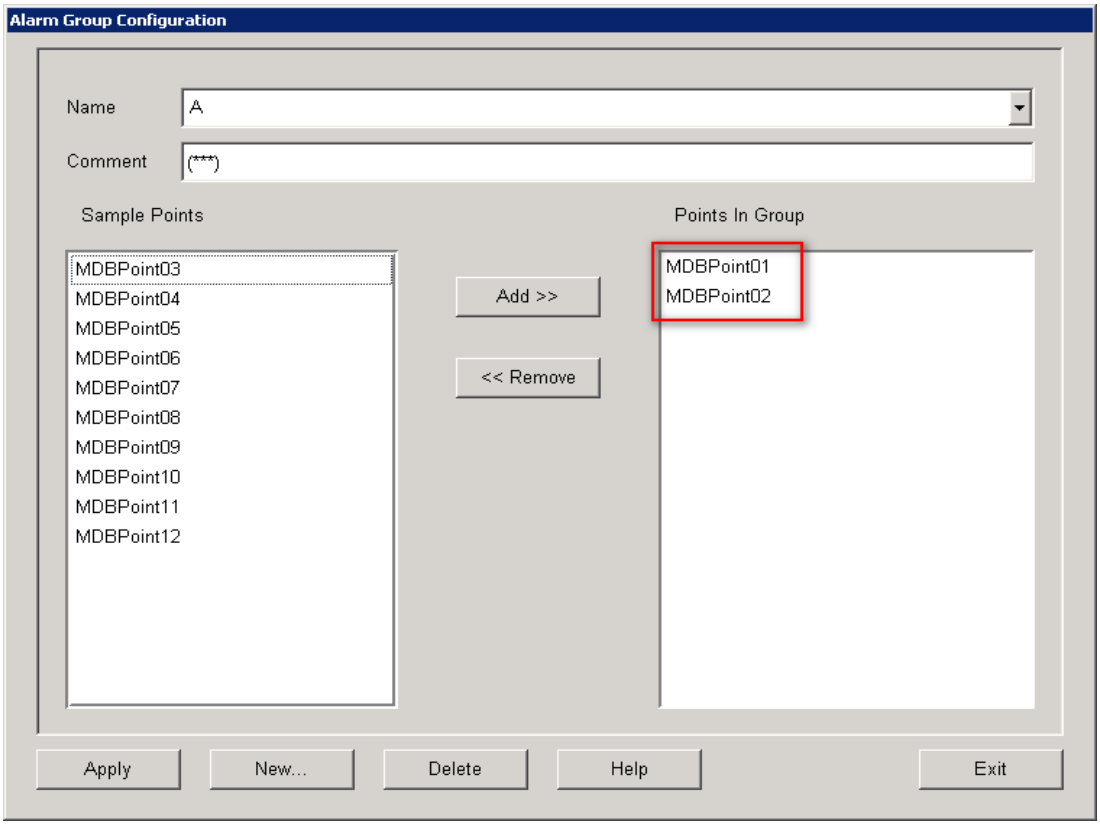
Suppose you need a calculated point called **A** (=Plant Alarm) which must true on whenever **MDBPoint01** or **MDBPoint02** are true**.**

Choose **Configure Alarms group** from the **Configure** menu and create a new point pressing the **New** button.



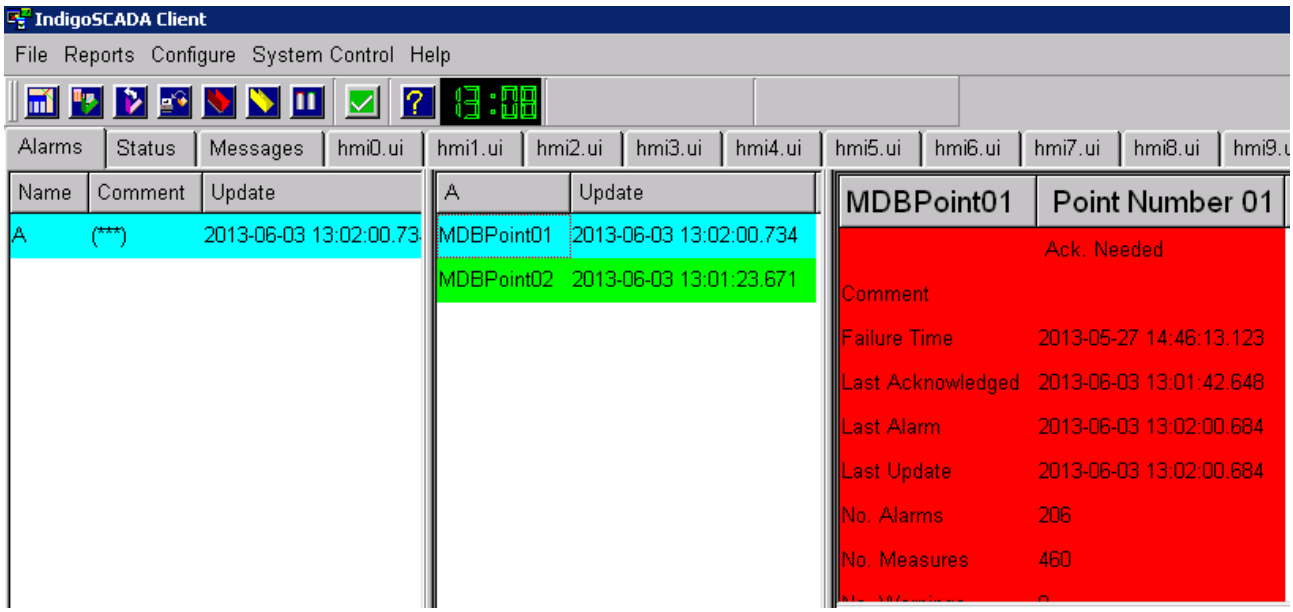
Now we will have a new group Alarm named **A.**

Add **MDBPoint01** and **MDBPoint02** to the list for the alarm **A**.

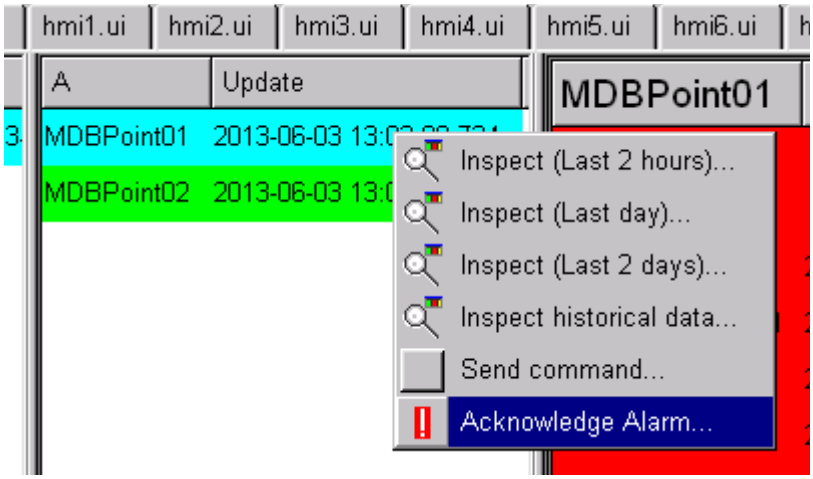


Apply these settings and close the window.

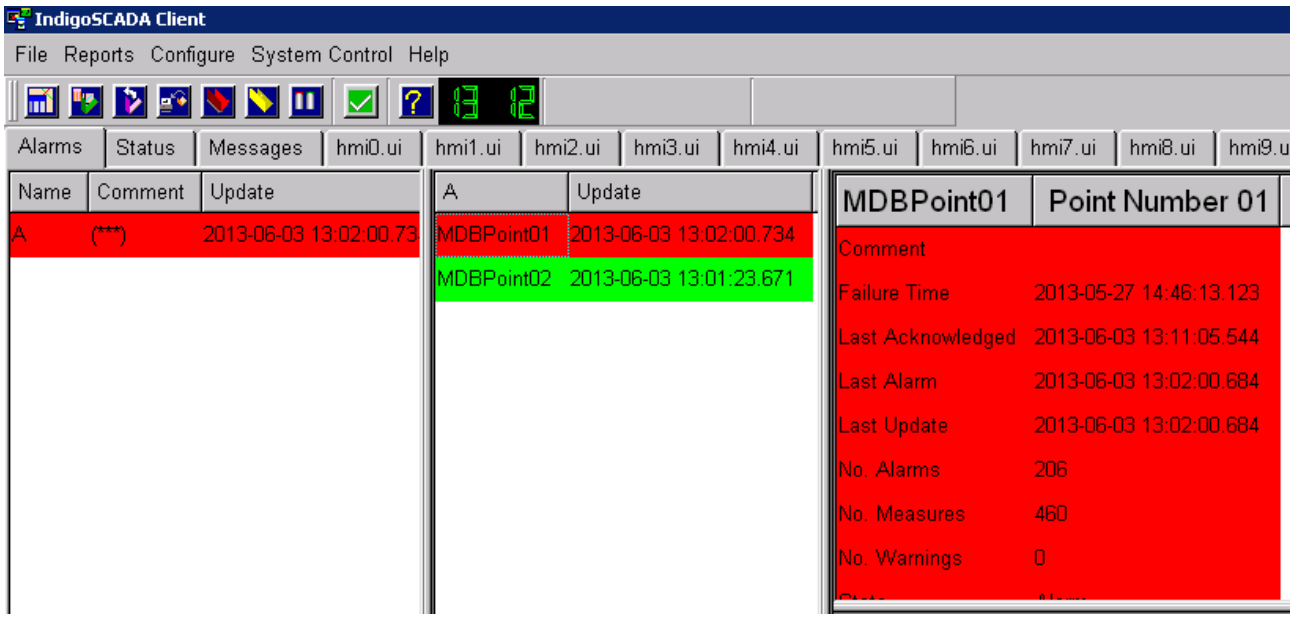
Change values in **PeakHMI** and look at the **Alarms tab.** In this example **MDBPoint01** has got value 1(light blue=alarm) and so the Plant Alarm **A** has become an alarm too.



Right-click the **MDBPoint01** and choose **Acknowledge Alarm.**



MDPoint01 changes his state in **Acknowledged Alarm** and Plant Alarm **A** changes this state too.



You can create as many alarm groups as you want.

Menu of IndigoSCADA

File menu

Reports menu

Make

Review Print

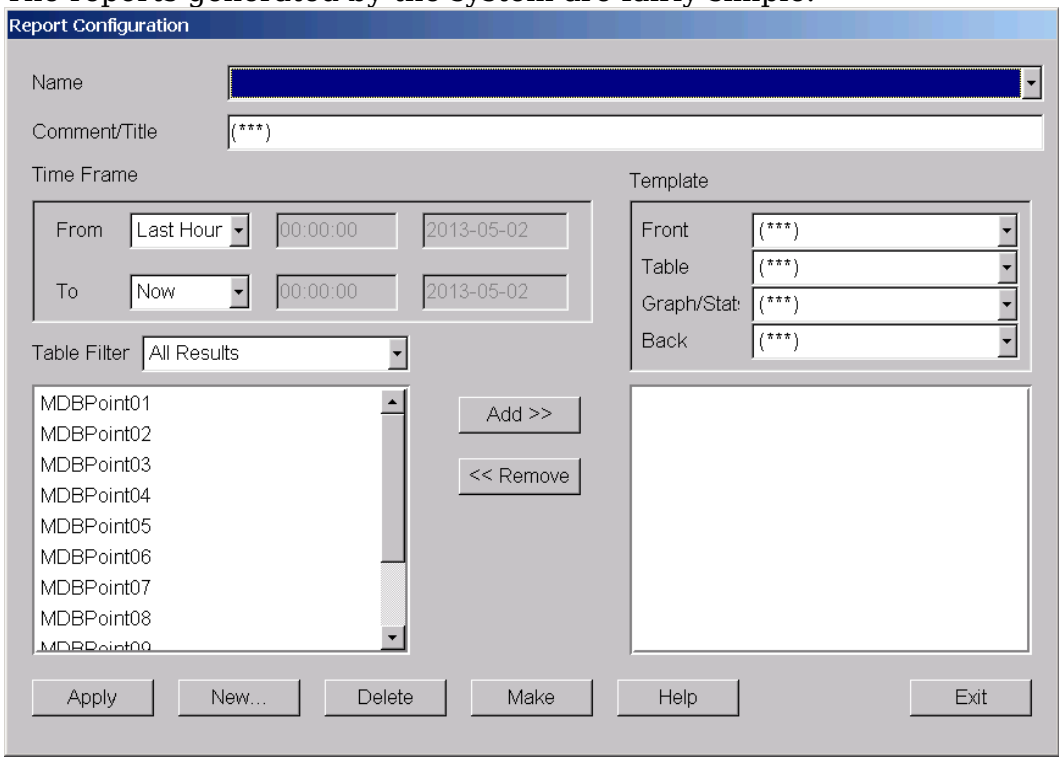
Alarms Report

Events Report

Audit

Report Configure

The reports generated by the system are fairly simple.



The **Name** field selects the report to be examined or modified.

The **Comment/Title** field is the title or description of the report.

The **From** and **To** fields define the time frame of the report. There are three parts to the **From** and **To** parts of the time frame, the type, a date and a time of day.

The type field can be for the From field:

- Last Hour
- Last Four Hours
- Last Twelve Hours
- Last Day (last 24 hours)

- Last Week (last 7 days)
- Date-Time (an explicit date and time, right click on the date field to get a calendar)
- Today (the date is current day and time of day is the given time)
- Yesterday (the date is previous day and time of day is the given time)

The type field for the **To** fields:

- Now. The current date and time.
- Date-Time (an explicit date and time, right click on the date field to get a calendar)
- Today (the date is current day and time of day is the given time)
- Yesterday (the date is previous day and time of day is the given time)

The **Add** button copies from the available sample points list (left) to the reports sample points list (right).

The **Remove** button copies selected sample points from the reports sample points to the available sample points.

The buttons at the bottom have the following function:

Apply. Any changes are made permanent for the currently selected report point. ***Unless Apply is pressed any changes are discarded***.

New. This creates a new report. The New Report dialog is displayed and the existing report listed. The new name must consist only of letters and numbers. The name must be unique (ignoring case).

Delete. The selected report is deleted after a confirmation.

Make. The selected report is generated and displayed.

Help. This displays this help page.

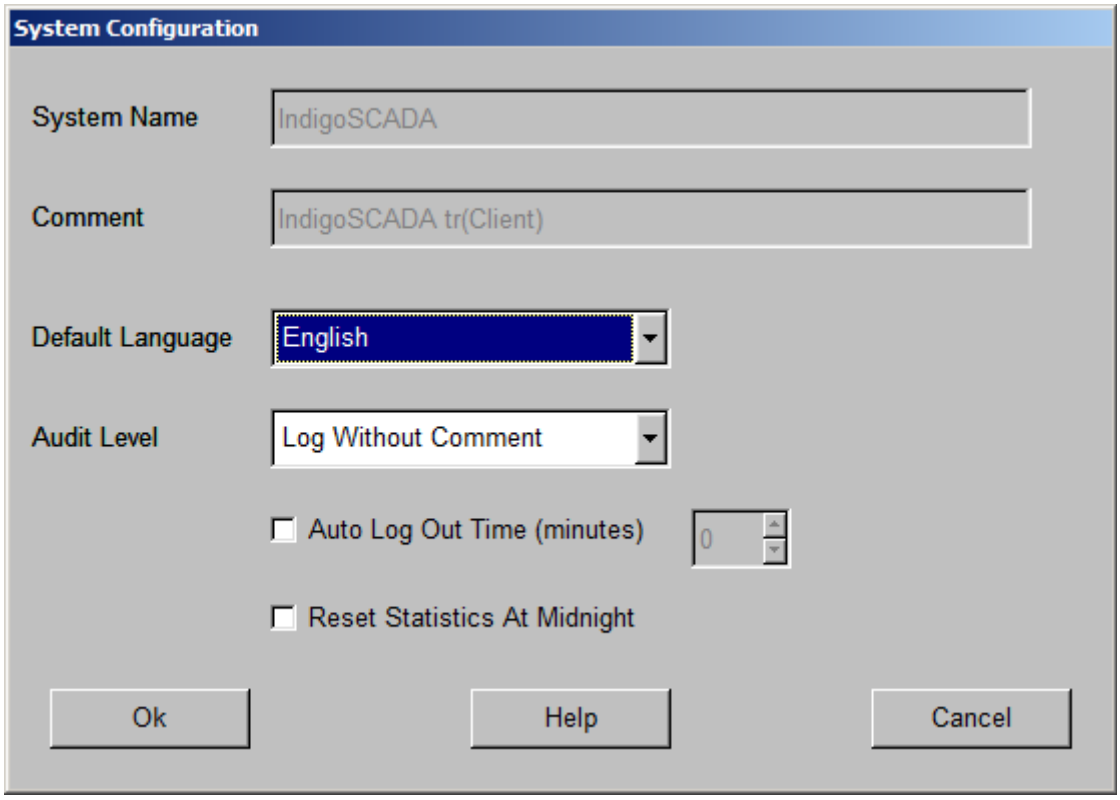
Exit. The dialog is closed.

Batch Editor

Pen Trace

Pen Trace is used to draw process data trend and make analysis.

Configure menu



Configure SystemThis dialog sets up system wide parameters.

The System **Name** is the name of the system. This is used in the application title bar.

The **Comment** is a long (free text) description of the installation.

Default Language sets the language to use when starting up. Users can select a different language to use after logging in.

Audit Level sets the type of change auditing that is performed. The setting may be **None** (for no auditing), **Log Without Comment** (changes are logged silently) and **Log With Comment** (requires the user to enter a reason for the change).

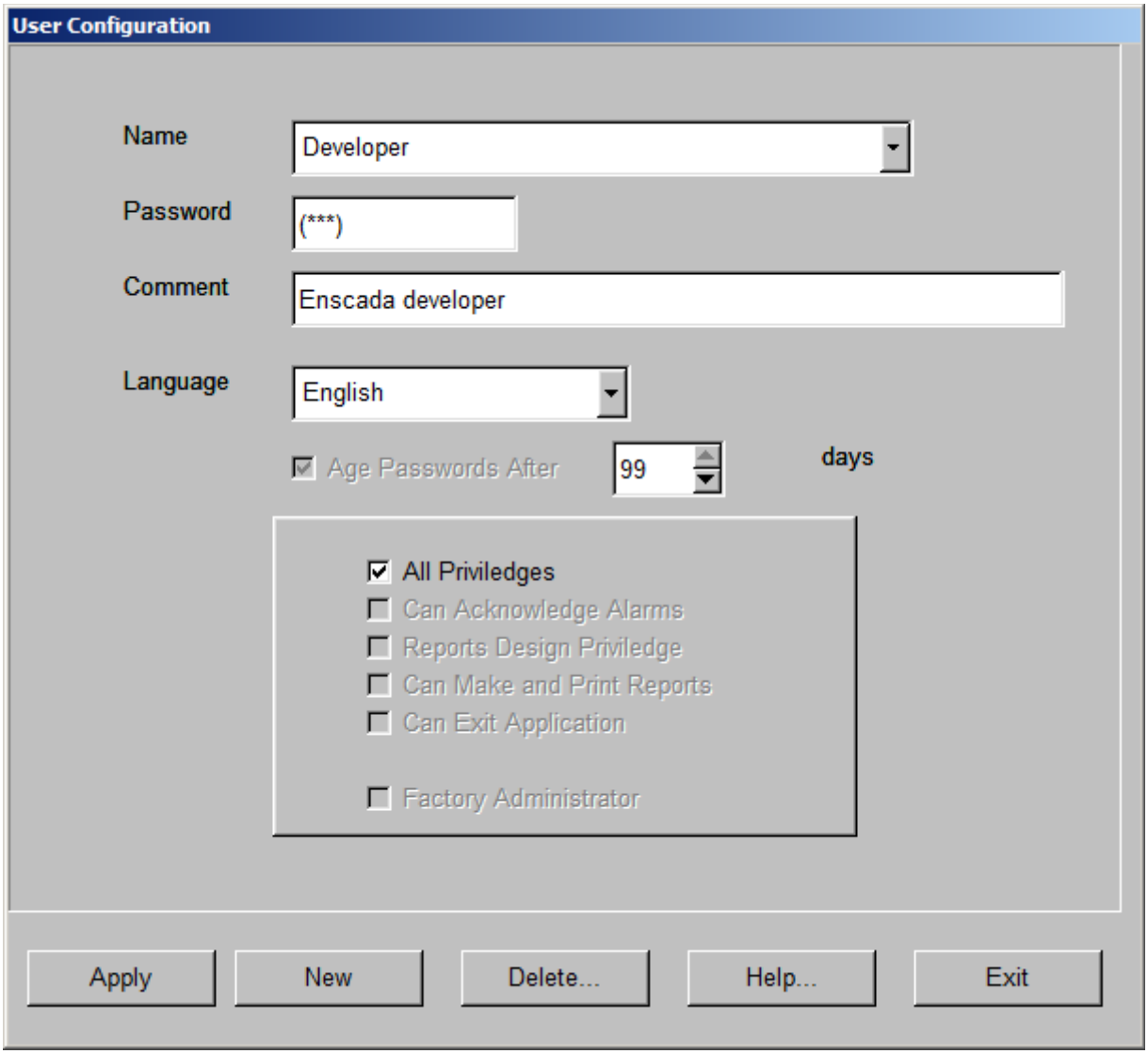
Auto Log Out Time. When checked a user is logged out automatically after the selected number of minutes.

Press the **Ok** button to accept and set the changes, then exit.

Press the **Cancel** button to discard any changes and exit.

Configure Users

This dialog allows users to be added, removed and configured.



The **Name** field selects which user is being examined or edited.

The **Comment** field is a free text description of the user.

The **Language** field selects the language to use for this user after login.

The group of checkboxes select which privileges the user has. These control what options are displayed to the user:

All Privileges. The user is a system administrator with all options and no restrictions. Note that although the application will not prevent access or modification of the system's configuration the privileges granted by the database system (when starting the application) may not permit any changes to the databases.

Reports Design Privileged. The user can design reports.

Can Acknowledge Alarms. The user can acknowledge alarms.

Can Make and Print Reports. The user can generate, view and print reports.

The buttons have the following function:

Apply. Any changes are made permanent for the currently selected user. ***Unless Apply is pressed any changes are discarded***.

New. This creates a new user. The New User dialog is displayed and the existing users listed. The new user name must consist only of letters and numbers. The name must be unique (ignoring case).

Delete. The selected user is deleted after confirmation.

Help. This displays this help page.

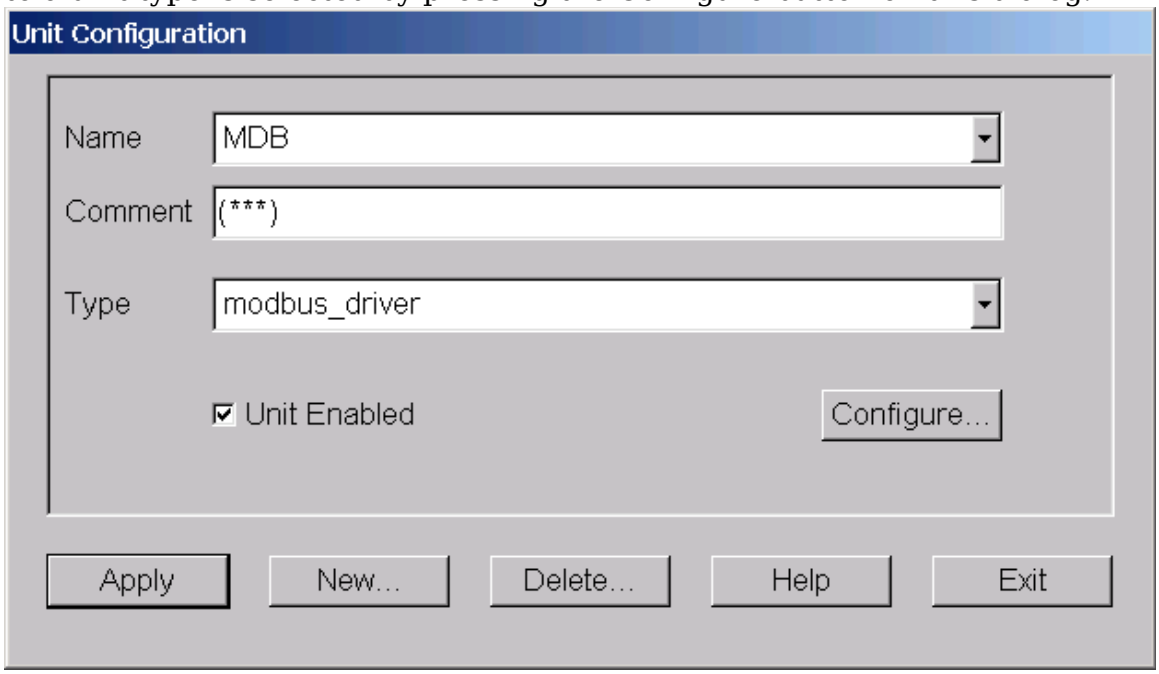
Exit. The dialog is closed.

Configure Units

A Unit is a piece of measuring equipment.

Each unit has one type of communication protocol associated with it.

This dialog provides the configuration common to all units. The configuration specific to a unit type is selected by pressing the Configure button on this dialog.



The **Name** field selects which unit is being examined and modified.

The **Comment** field is a free text description of the unit.

The **Type** field shows what sort of unit it is. The type determines the driver protocol used to interface to the equipment and to provide the configuration services.

If **Unit Enabled** is checked then the unit is by default enabled. Otherwise it is not enabled, by default. This can be overridden when a recipe is loaded.

The **Configure** button opens the specific configuration dialog for the selected unit of the selected type.

The buttons at the bottom have the following function:

Apply. Any changes are made permanent for the currently selected unit. ***Unless Apply is pressed any changes are discarded***.

New. This creates a new unit. The New Unit dialog is displayed and the existing units listed. The new unit name must consist only of letters and numbers. The name must be unique (ignoring case).

Delete. The selected unit is deleted after confirmation. All associated sample points are deleted if requested. The results associated with these sample points is not deleted.

Help. This displays this help page.

Exit. The dialog is closed.

Configure Sample Points

This dialog configures sample points. Each sample point represents an input into the system that measures one or more physical values. Each of these physical values is named with a **tag name**.

The list on the left hand side shows the configured sample points and their comments.

The **Comment** field allows a free text description of the sample point to be entered.

The **Unit** field selects the unit associated with the sample point. All sample points must be connected to a unit.

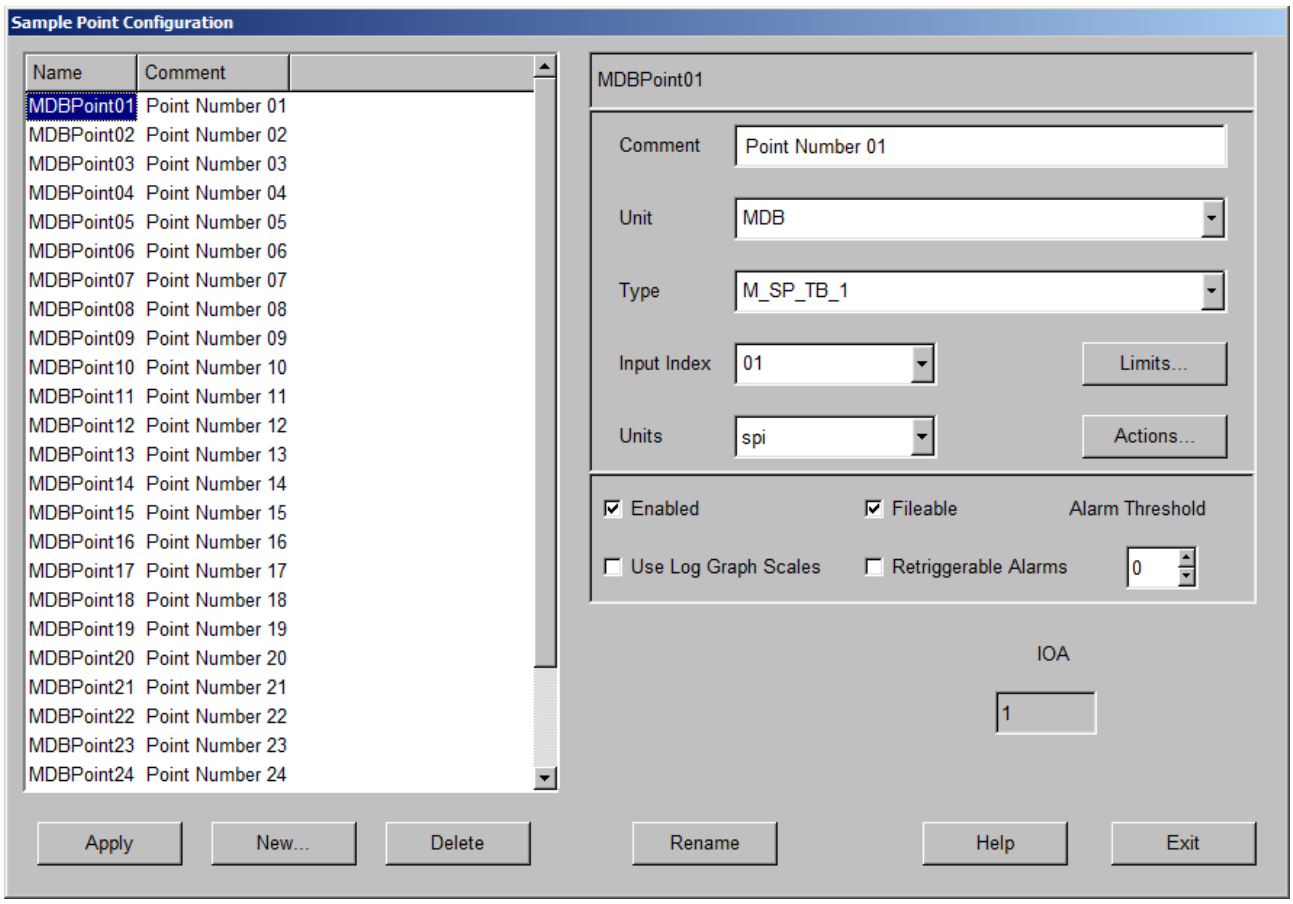
The **Type** field selects the type of input. This in turn (with the Unit field) selects the specific configuration for the sample point.

The **Input Index** field selects the actual input of a given type into the given unit that maps on to this sample point. The choices are determined by the Type and Unit fields.

The **Units** field sets the physical units for the sample point.

Pressing the **Limits** button opens the alarm threshold configuration dialog.

Pressing the **Actions** dialog opens the event action dialog. This allows certain commands to be performed when a sample point is measured and/or a sample point goes into alarm. This is an advanced user's function and although providing a considerable degree of flexibility it can cause serious damage if it is incorrectly used.



The **Enabled** checkbox indicates if the sample point is enabled by default. This can be overridden by loading a recipe.

The **Fileable** checkbox indicates if the data from the sample point is to be filed. Normally this should be checked, however there is the possibility that some data does not need filing (e.g. free disk space).

The **Use Log Graph Scales** check box indicates if the graphs for this sample point should use logarithmic scales. This is normally set for particle counts type measurements.

The **Retriggerable Alarms** checkbox indicates if a new alarm is registered for each measurement in alarm. If not checked then an alarm is only generated when the sample point enters alarm.

The **Alarm Threshold** field sets the number of consecutive alarms to be detected before the alarm is logged. While waiting for the alarm threshold to be breached the sample point is in a warning state.

The panel below the checkboxes is for the specific configuration fields. These are specific to a particular unit/type combination and allow for additional configuration (for example conversion factors).

The buttons at the bottom have the following function:

Apply. Any changes are made permanent for the currently selected sample point. ***Unless Apply is pressed any changes are discarded***.

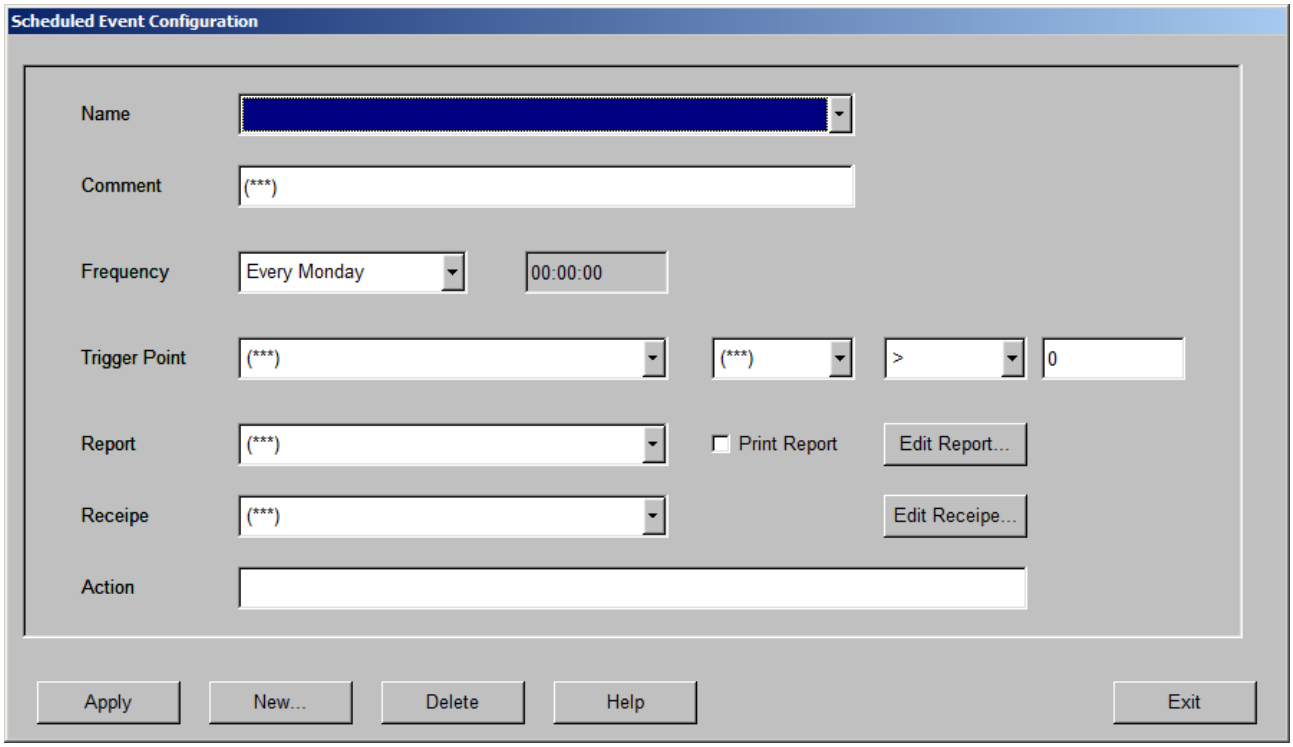
New. This creates a new sample point. The New Sample Point dialog is displayed and the existing sample points listed. The new name must consist only of letters and numbers. The name must be unique (ignoring case).

Delete. The selected sample point is deleted after a confirmation.

Help. This displays this help page.

Exit. The dialog is closed.

Configure Scheduled Events



The **Name** field selects which event is being examined or edited.

The **Comment** field is a free text description of the event.

The **Frequency** fields set how often the event is run and the time of day or period between events.

The **Trigger Point** parameters allow an event only to run when the given sample point-tag value meets a given criterion.

The **Report Field** selects which report to generate (or (***) selects no report).

If the **Print Report** checkbox is checked then the report is printed after generation.

Clicking the **Edit Report** button opens the report design dialog.

The **Recipe** selects the recipe to load when the event is run (or (***) selects no recipe).

Clicking the **Edit Recipe** button opens the recipe design dialog.

The **Action** field may be set to an action to be performed. If the action starts with a ! then the command is assumed to be a shell command, otherwise it is an SQL command performed on the current values database. This is an advanced user command, being very powerful it can cause considerable damage to a system. For this reason the application must never run with root privileges.

The buttons at the bottom have the following function:

Apply. Any changes are made permanent for the currently selected event. ***Unless Apply is pressed any changes are discarded***.

New. This creates a new event. The New Event dialog is displayed and the existing events listed. The new name must consist only of letters and numbers. The name must be unique (ignoring case).

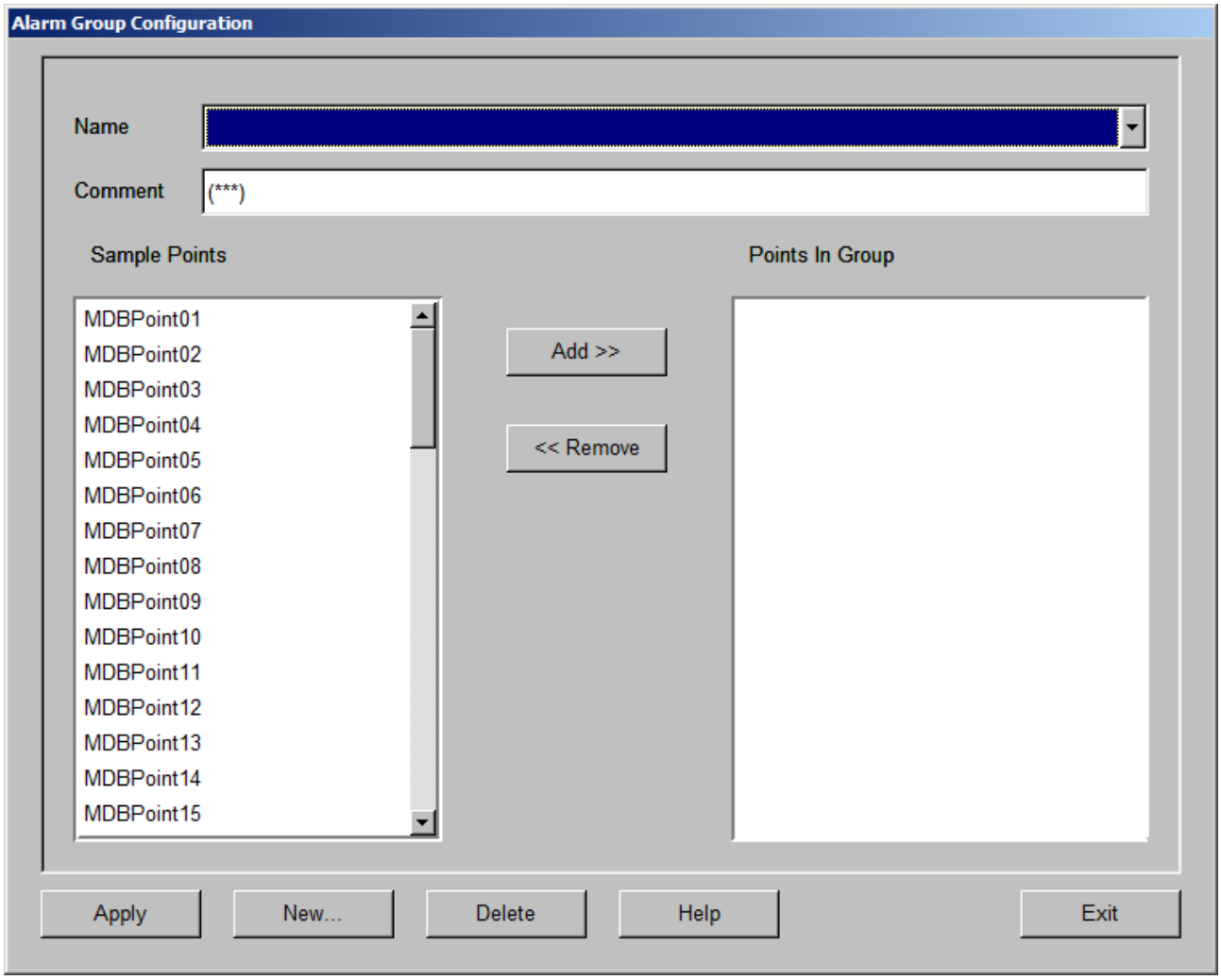
Delete. The selected event is deleted after confirmation.

Help. This displays this help page.

Exit. The dialog is closed.

Configure Alarm Groups

Alarm groups allow the alarm state of a collection of sample points to be collected together and handled as a single entity. The state of a group is the highest alarm state of any of the alarm group's members. Alarms can be acknowledged as a group



The **Name** field selects the alarm group to be examined or modified.

The **Comment** field has a free text description of the alarm group.

The **Sample Points** list is the list of sample points not in the alarm group.

The **Points In Group** list has the list of sample points in the group.

Clicking the **Add** button copies the selected sample points from the **Sample Points** list to the **Points In Group** list.

Clicking the **Remove** button removes the selected sample points in the **Points In Group** list to the **Sample Points** list.

The buttons at the bottom have the following function:

Apply. Any changes are made permanent for the currently selected alarm group. ***Unless Apply is pressed any changes are discarded***.

New. This creates a new alarm group. The New Group dialog is displayed and the existing alarm groups listed. The new name must consist only of letters and numbers. The name must be unique (ignoring case).

Delete. The selected alarm group is deleted after a confirmation.

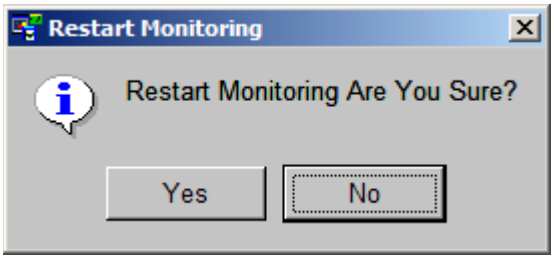
Help. This displays this help page.

Exit. The dialog is closed.

System control menu

Restart Monitor

This menu entry is used to restart monitor and make **IndigoSCADA client** aware of new Units(= protocol devices) added using the **Configure Units** functionality

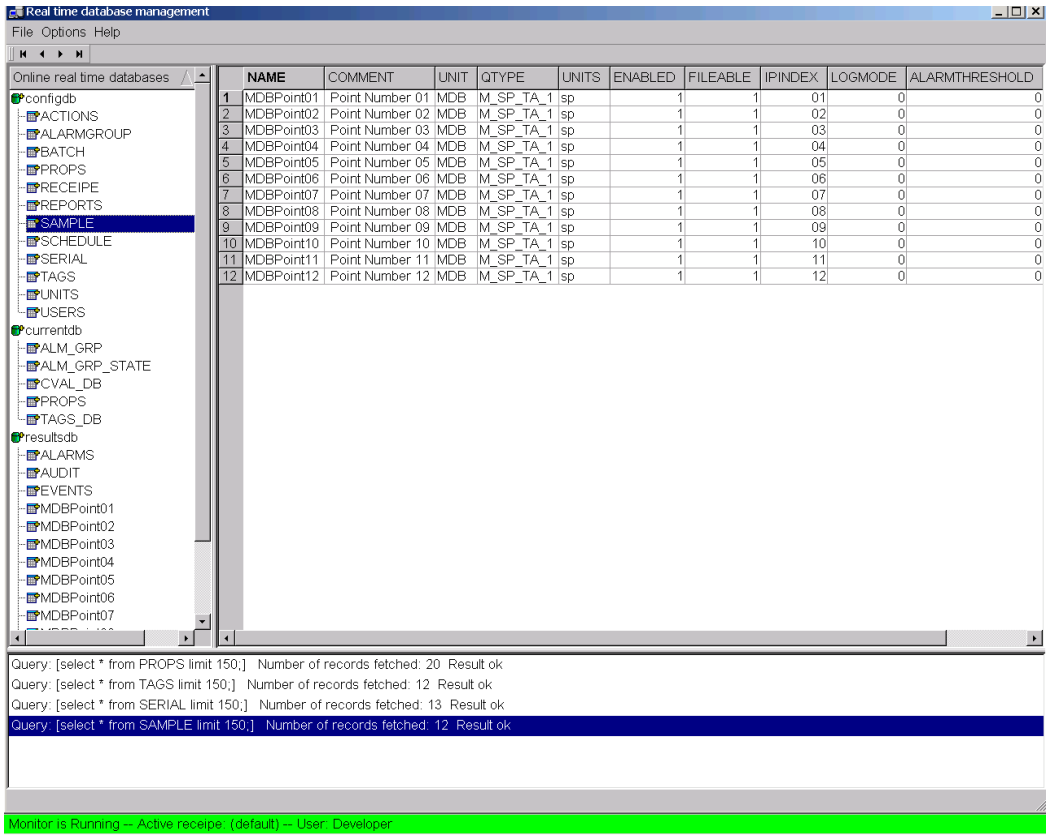


present in the **Configure** menu.

Press **Yes** button to confirm the restarting.

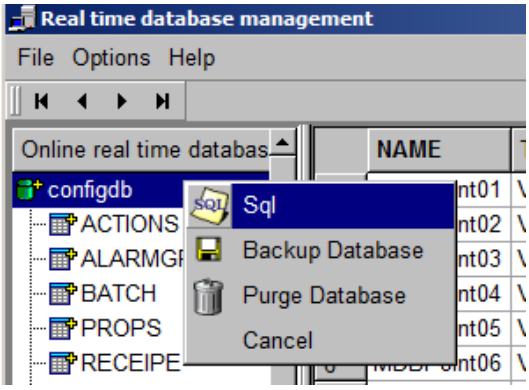
Realtime database Management

Realtime database management is a sub-program used to inspect all realtime data of **IndigoSCADA.** Using a SQL language it is possible to query data and administrate the internal database.

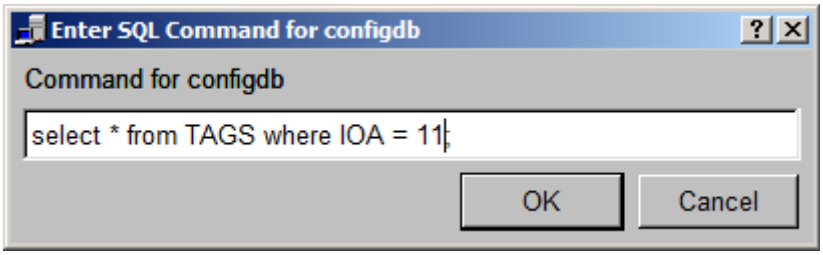


Execution of a SQL command.

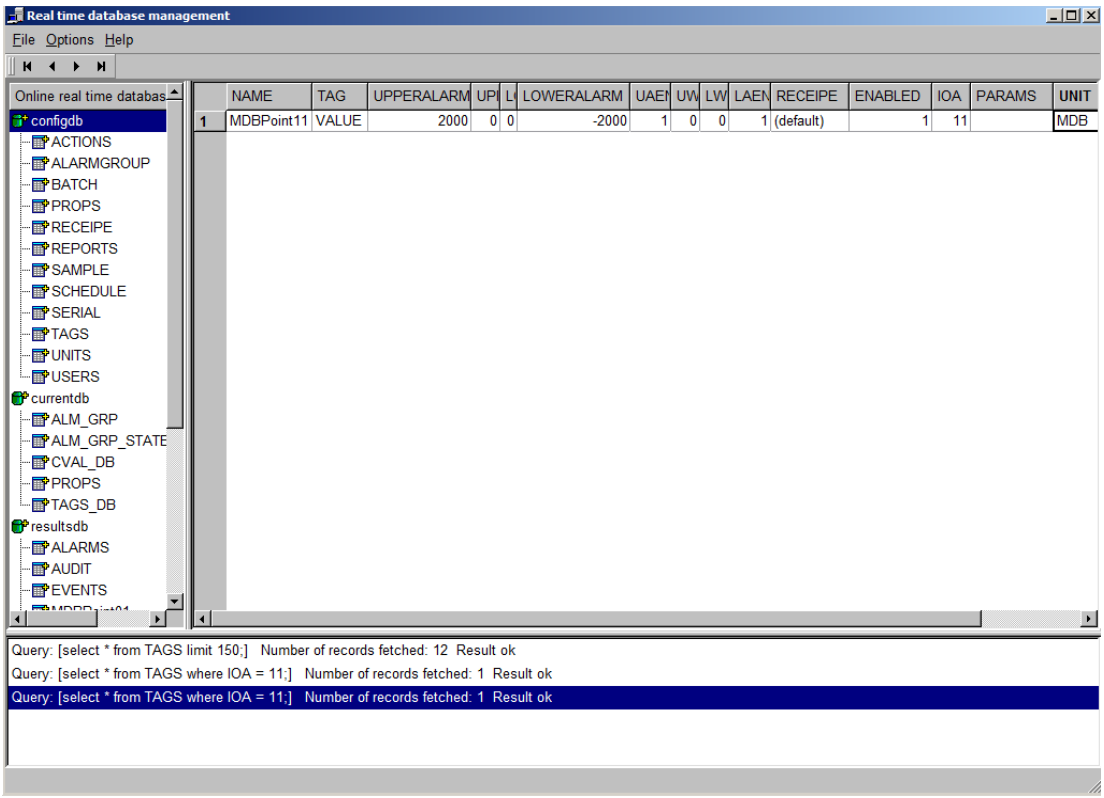
Right clicking with the mouse on the configdb database icon, the popup window shows up:

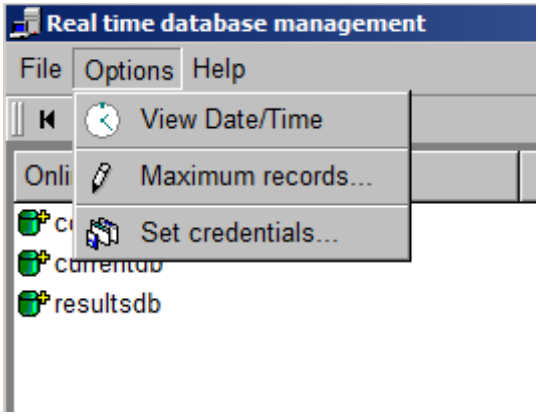


Insert a **SELECT** query similar as shown below

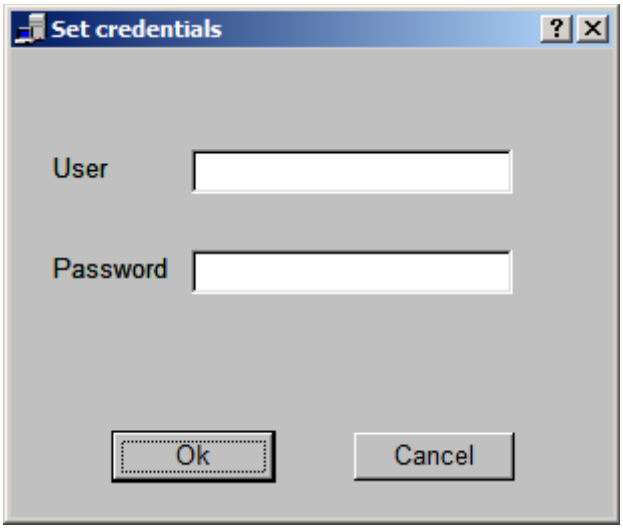


The result shows how is recorded the **MDBPoint11** sample point in the real time configuration database (configdb) **TAGS** table:



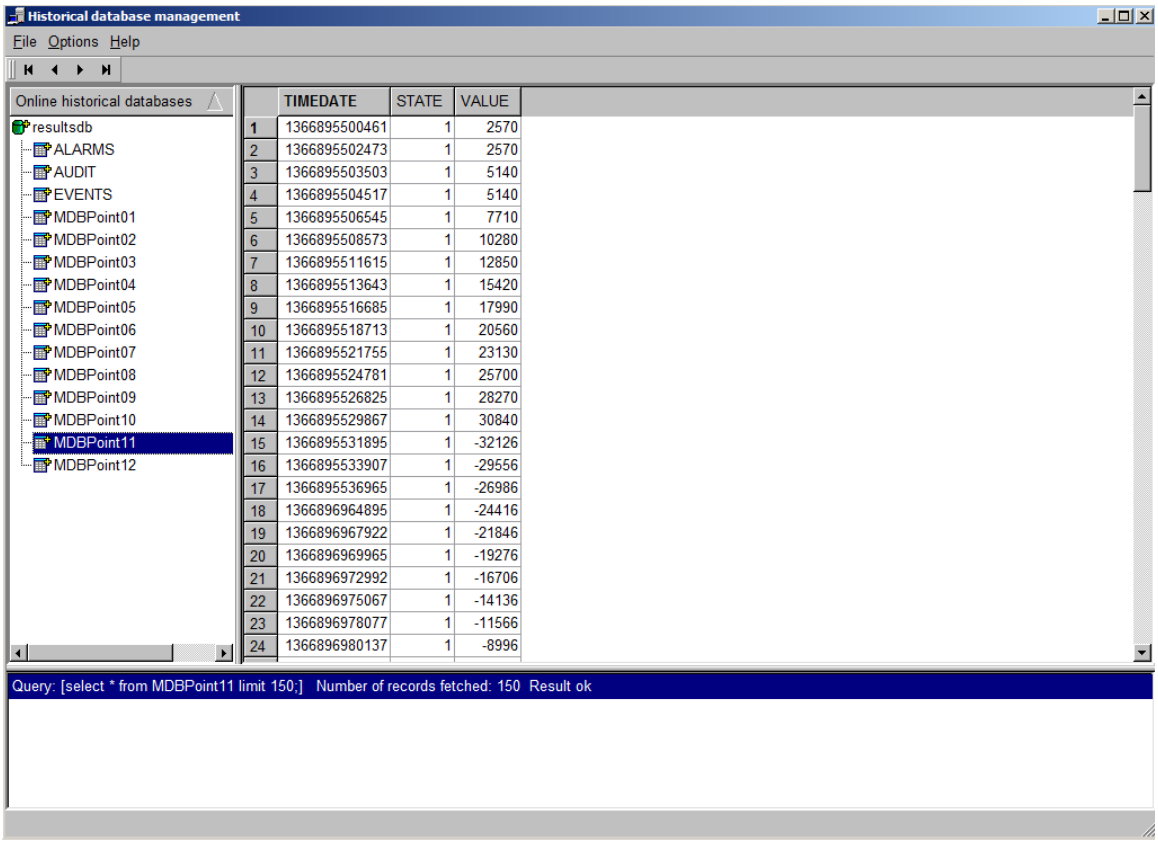


Select the **Options Set credentials** menu, it opens the following dialog where you can set the user and password for accessing the real time database:



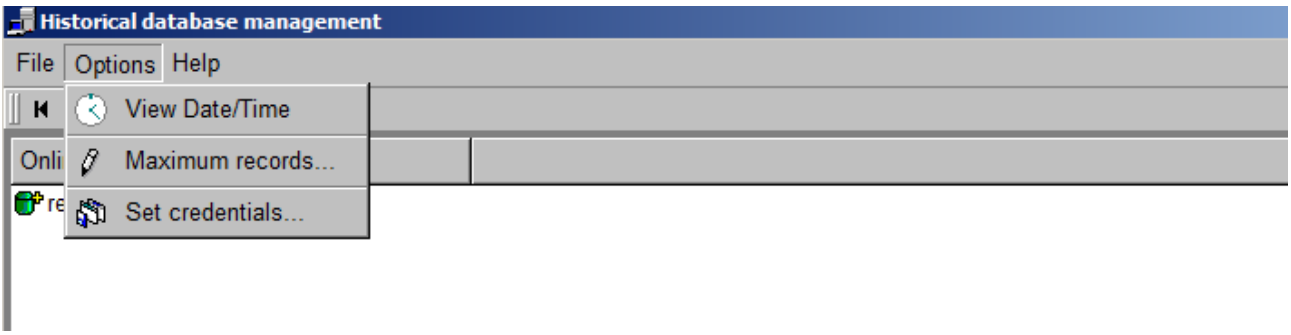
Note	The **Backup SQL** function works in this revision. The restore function is done with the SubSQL program.

Historical database Management



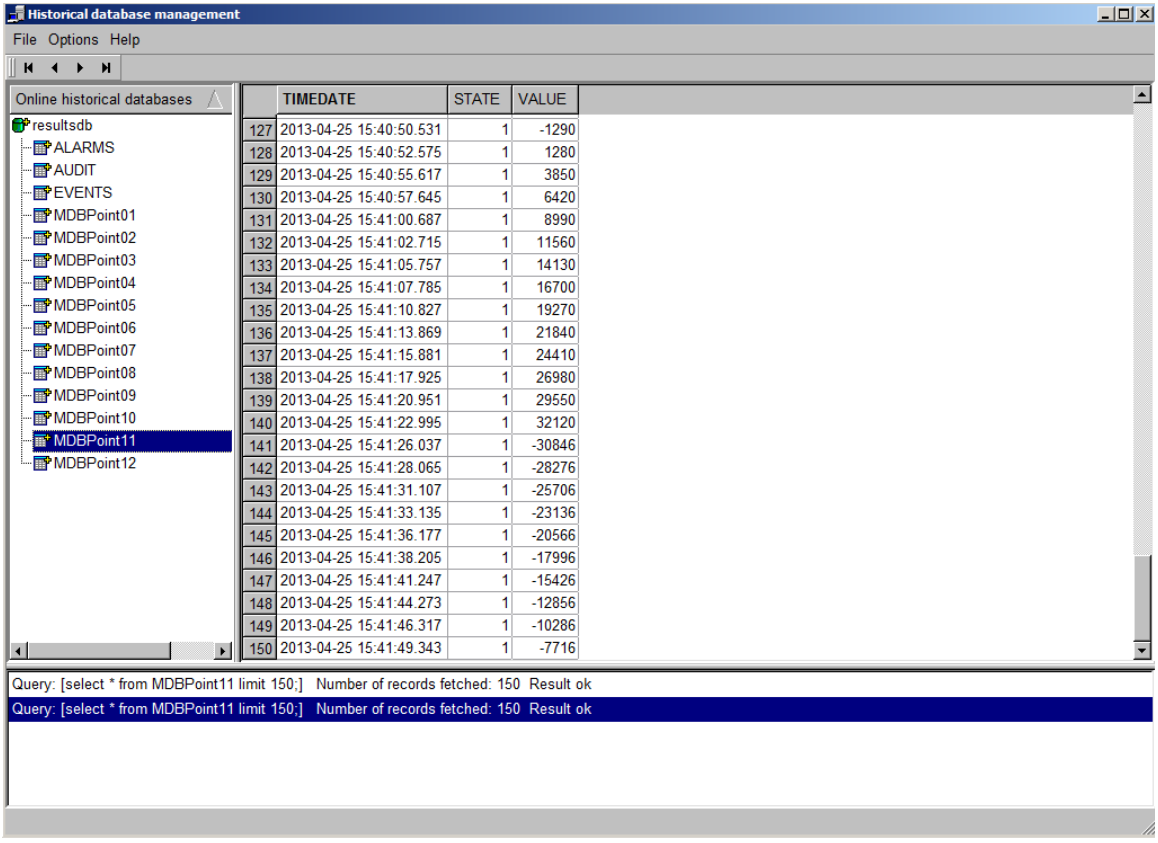
Historical database management is used to analyze all the process data samples from the beginning. It gives a low level access to all internal tables of **IndigoSCADA**

The **MDBPoint11** table is selected on the left pane.



Select the **Options View Date/Time** menu

Select again the **MDBPoint11** table on the left pane. **TIMEDATE** column now shows a readable date and time.



The **Options Set credentials** menu allows for the setting of user and password of historical database.

Help menu

Show the online help guide.

Main toolbar of IndigoSCADA

The main toolbar of IndigoSCADA is made up by 10 objects:

Report make

Report configure

Report review

Report print

Report alarm

Report events

Edit batches

Acknowledge all alarms

Help

Clock

Configuration

Ini files configuration

Edit **C:\scada\project\scada.ini** in order to configure the default browser.

The browser is used to show help windows and reports produced by the user.

[internet_browser]

path=C:\Program Files\Internet Explorer\IEXPLORE.EXE

In order to enable the recording of protocol communication to log files in **C:\scada\logs**, set log_to_file to 1

[drivers]

log_to_file=1

Advanced topics

IndigoSCADA Modbus protocol configuration

IndigoSCADA is using VT_ types (used in OPC) to describe the modbus type:

VT_BOOL is a bit

VT_I4 is a 32 bits signed integer

VT_UI4 is a 32 bits unsigned integer

VT_R4 is a 32 bits floating point

VT_R4SWAP is a 32 bits floating point with swap of words

VT_I2 is a 16 bits signed integer

VT_UI2 is a 16 bits unsigned integer

Modbus function codes:

The following modbus functions are used to read:

_FC_READ_COILS 0x01 for IndigoSCADA this is type VT_BOOL

_FC_READ_DISCRETE_INPUTS 0x02 for IndigoSCADA this is type VT_BOOL

_FC_READ_HOLDING_REGISTERS 0x03 for IndigoSCADA this can be type VT_I4, VT_R4, VT_I2, VT_UI4, VT_UI2

_FC_READ_INPUT_REGISTERS 0x04 for IndigoSCADA this can be type VT_I4, VT_R4, VT_I2, VT_UI4, VT_UI2

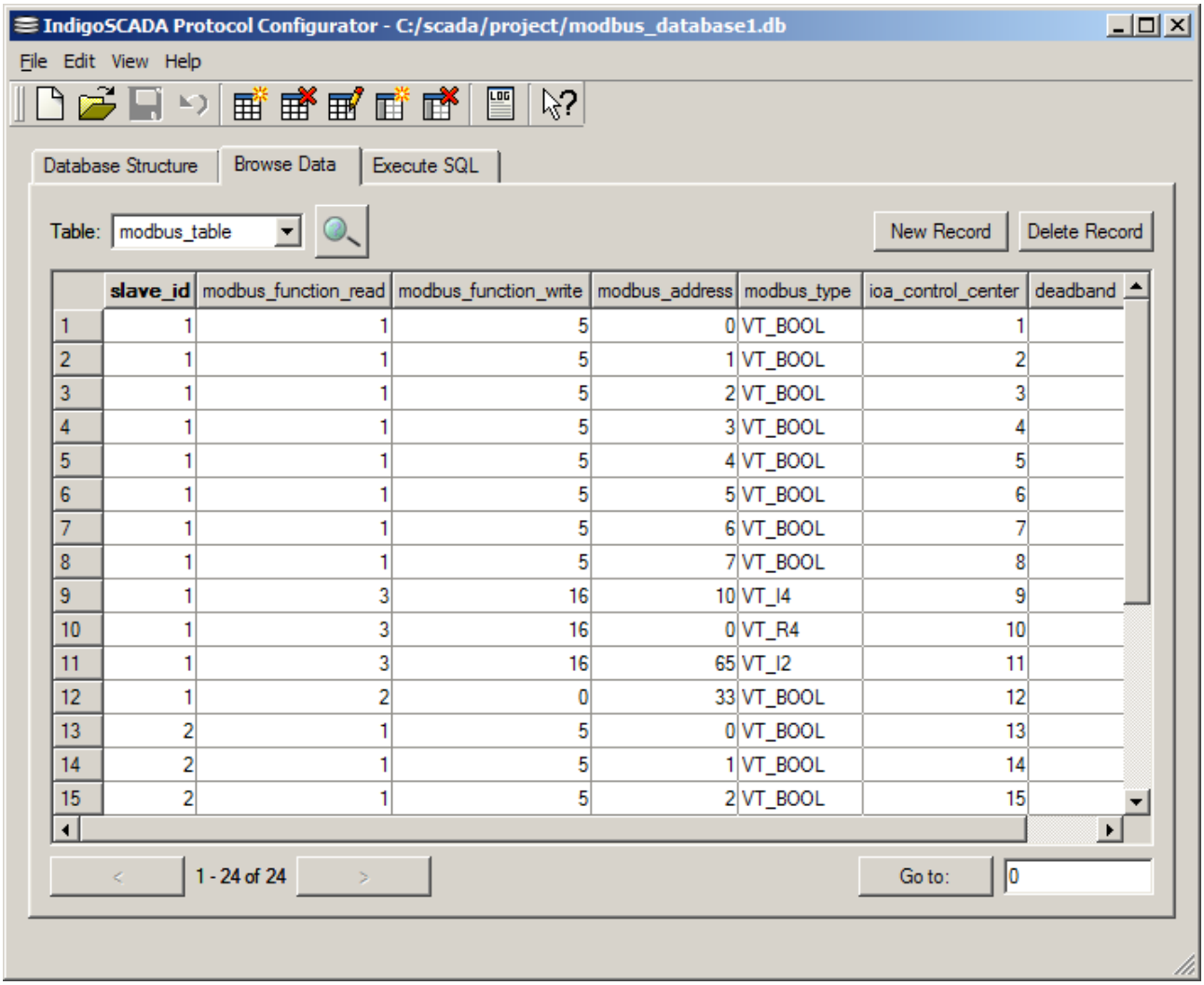
The following modbus functions are used to write:

_FC_WRITE_SINGLE_COIL 0x05 for Indigo this is type VT_BOOL

_FC_WRITE_MULTIPLE_REGISTERS 0x10 for Indigo this can be type VT_I4, VT_R4, VT_I2

Columns of Data Table in **IndigoSCADA Protocol Configurator** for Modbus protocol are used this way:

MODBUS part:
slave_id = ID of the device in daisy-chain
modbus_function_read = MODBUS function used to read ITEM



	slave_id	modbus_function_read	modbus_function_write	modbus_address	modbus_type	ioa_control_center	deadband
1	1	1	5	0	VT_BOOL	1	
2	1	1	5	1	VT_BOOL	2	
3	1	1	5	2	VT_BOOL	3	
4	1	1	5	3	VT_BOOL	4	
5	1	1	5	4	VT_BOOL	5	
6	1	1	5	5	VT_BOOL	6	
7	1	1	5	6	VT_BOOL	7	
8	1	1	5	7	VT_BOOL	8	
9	1	3	16	10	VT_I4	9	
10	1	3	16	0	VT_R4	10	
11	1	3	16	65	VT_I2	11	
12	1	2	0	33	VT_BOOL	12	
13	2	1	5	0	VT_BOOL	13	
14	2	1	5	1	VT_BOOL	14	
15	2	1	5	2	VT_BOOL	15	

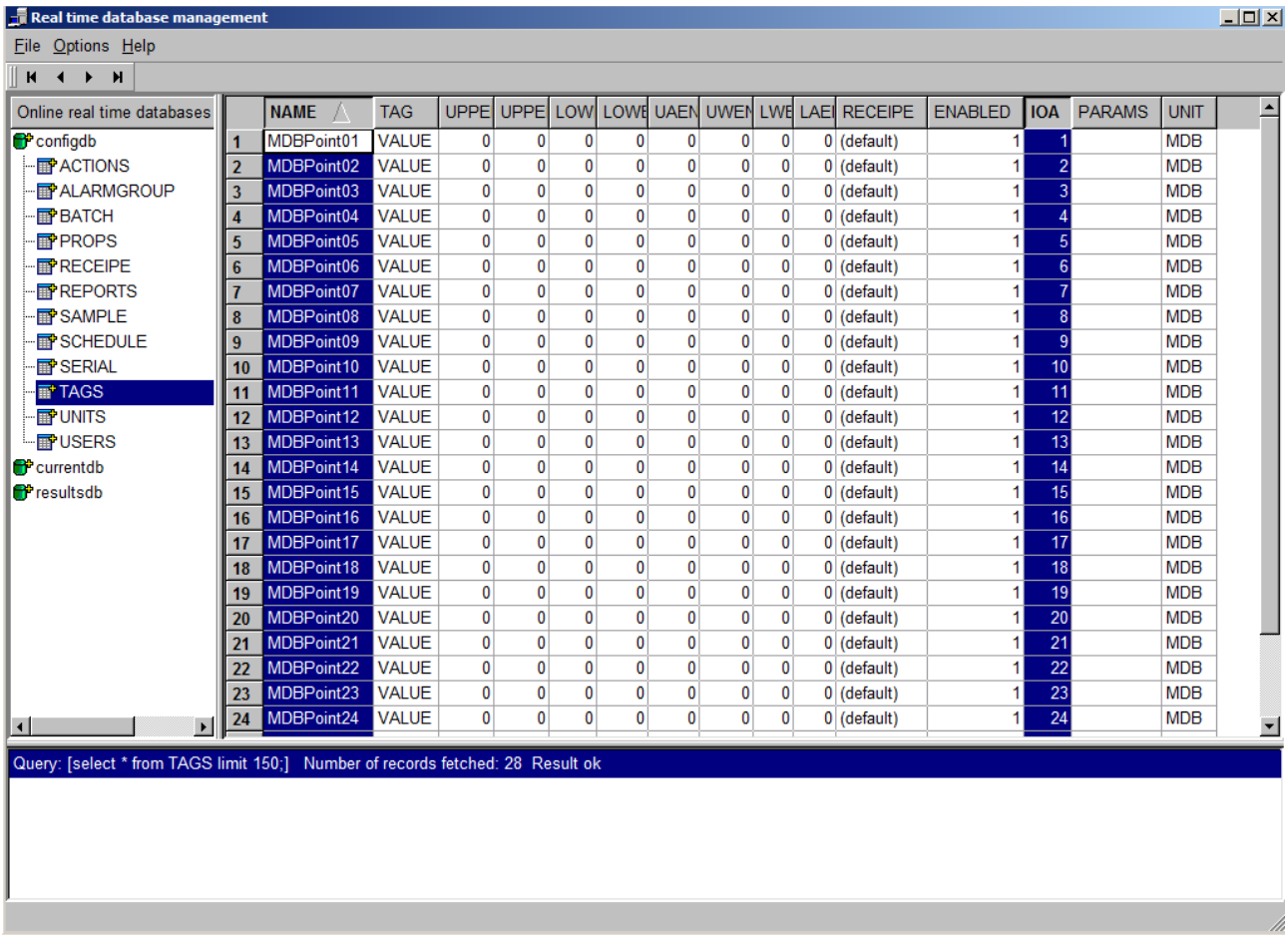
modbus_function_write = MODBUS function used to write the ITEM
modbus_address = MODBUS address of the ITEM, is an offset relative to the first address of the function used.

Record one has modbus_address 0, coils function 1 is used, so add the first coils address (000001) and you get actual modbus address 000001;
Record 11, has modbus_address 65, holding registers function 3 is used, so add the first holding registers address (400001) and you get actual modbus address 400066;
Record 12, has modbus_address 33, discrete inputs function 2 is used, so add the first discrete inputs address (100001) and you get actual modbus address 100034;

modbus_type = WINDOWS (or OPC) type, used to describe the ITEM type

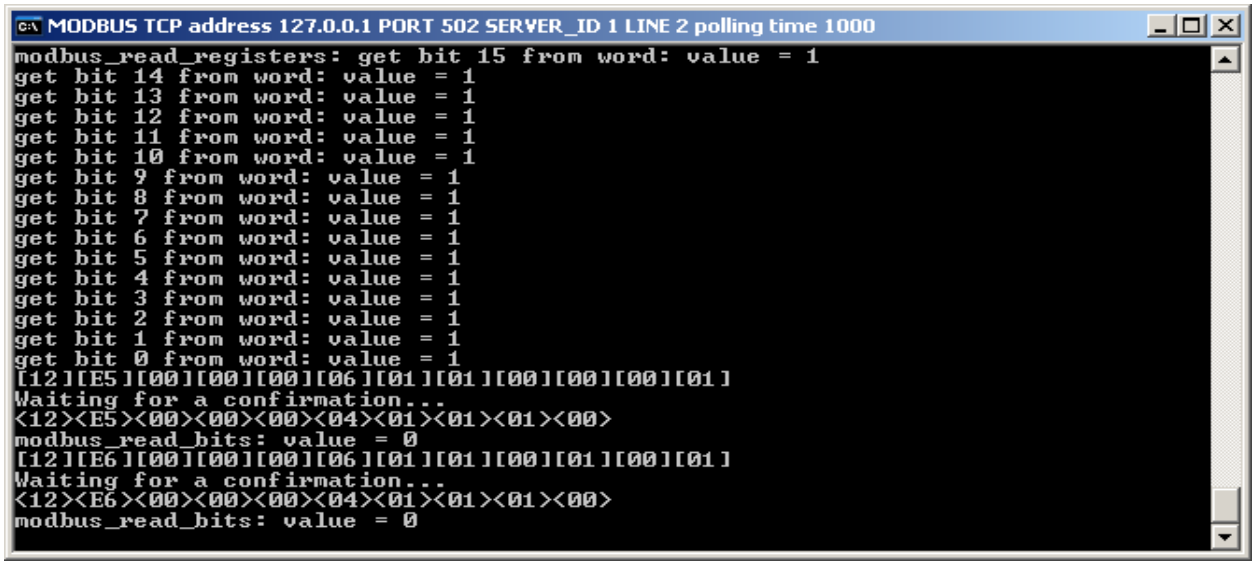
IEC104 part:

ioa_control_center = progressive number identifying the ITEM



The relation between modbus_table and TAGS table of configdb database is done by ioa_control_center column on modbus_table and IOA column on TAGS table, so for example MDBPoint11 has IOA 11and through modbus_table has modbus type VT_I2, modbus_address 65, modbus_function_read 3.

If you need another modbus unit (line 2), create for example a unit with name MDB2 and make a copy of file **C:\scada\project\modbus_database1.db** and rename the copy as **C:\scada\project\modbus_database2.db** Configure the unit and restart the monitor.



Advanced topics

IndigoSCADA OPC DA protocol configuration

Step 1:

Browsing of available OPC DA itemID's in the server

Open a command shell windows and run the command, that will create the file configuration_file.sql in C:\scada\project:

opc_client_da.exe -a server_IP_address -p OPCServerProgID -e configuration_file.sql -q {CLSID}

Example:

opc_client_da.exe -a 10.15.20.30 -p Enscada.OPC.Srv -e configuration_file.sql -q {78BB6572-B8BA-47F9-83B3-4EC99DF9B1A3}

Step 2:

Creation of protocol configuration database

Go to C:\scada\bin and run protocol_configurator.exe

Import the C:\scada\project\configuration_file.sql, selecting the C:\scada\project\configuration_file.sql with the menu: File->Import->Database from SQL file

At the question: Do you want to create a new database file to hold the imported data?

Answer: Yes

The database name should be your OPCServerProgID with extension .db

Save it in C:\scada\project

If you do not use OPCServerProgID.db name, the Step 3 will fail.

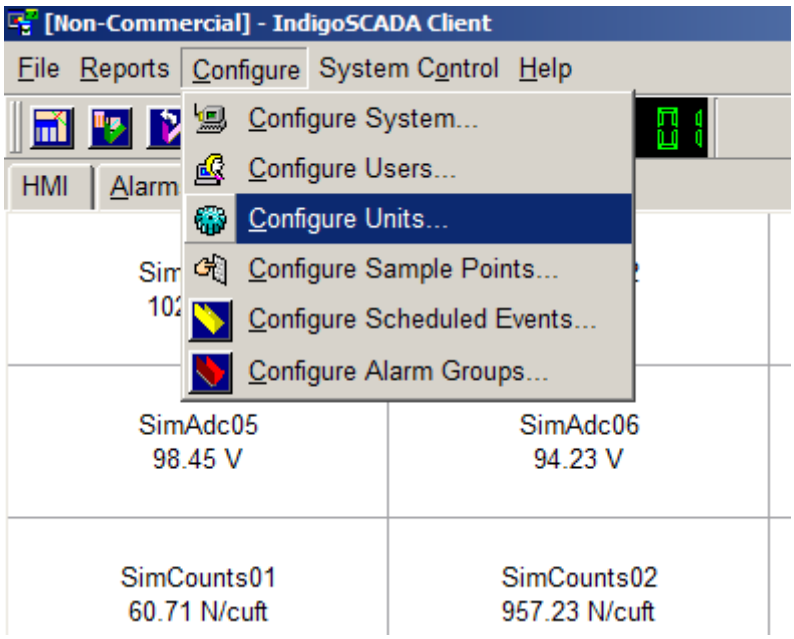
Example: Enscada.OPC.Srv.db

Wait for few seconds the Import Completed confirmation Window

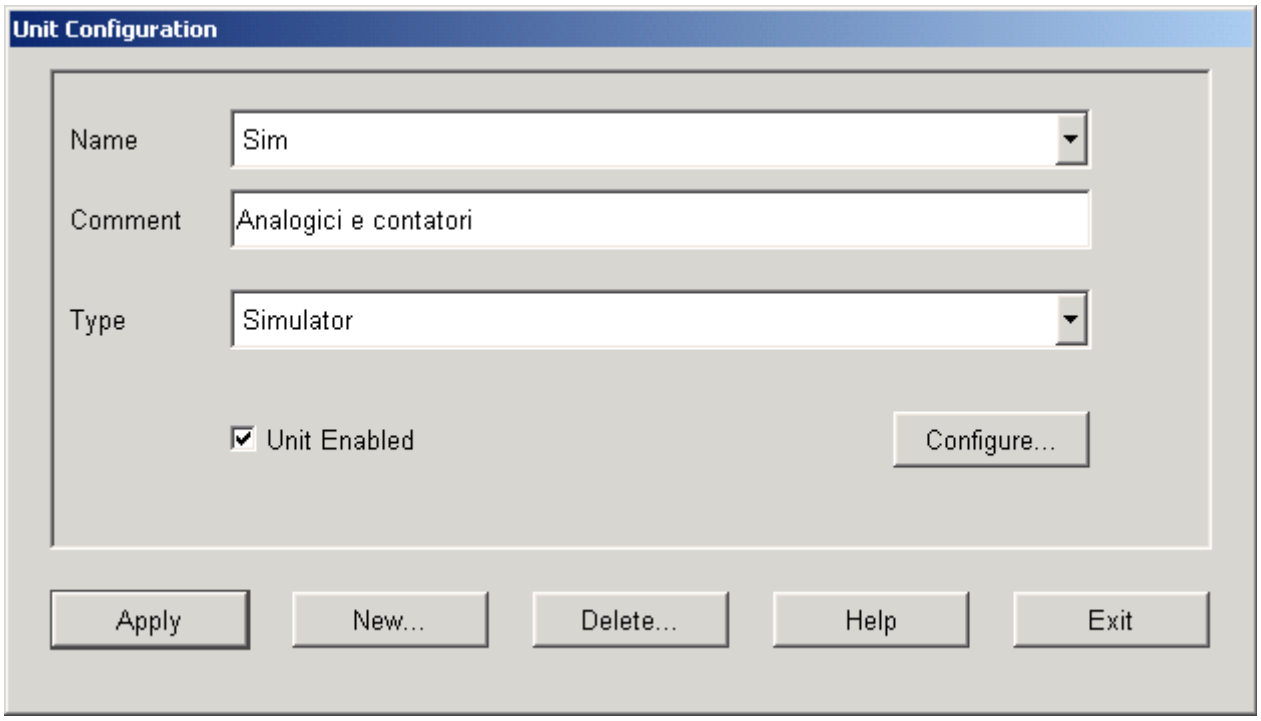
Step 3:

Creation of the new SCADA points

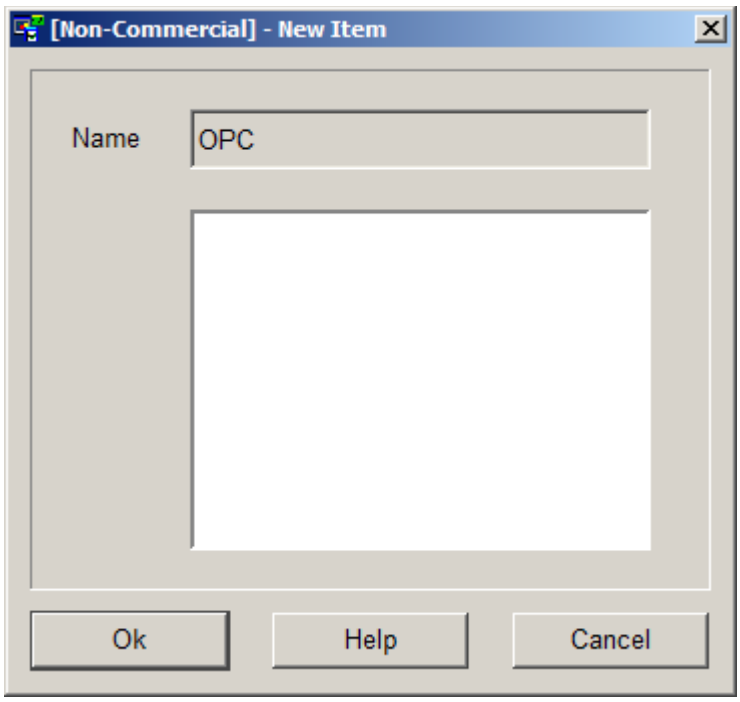
1 - Select the menu: Configure\Configure Units



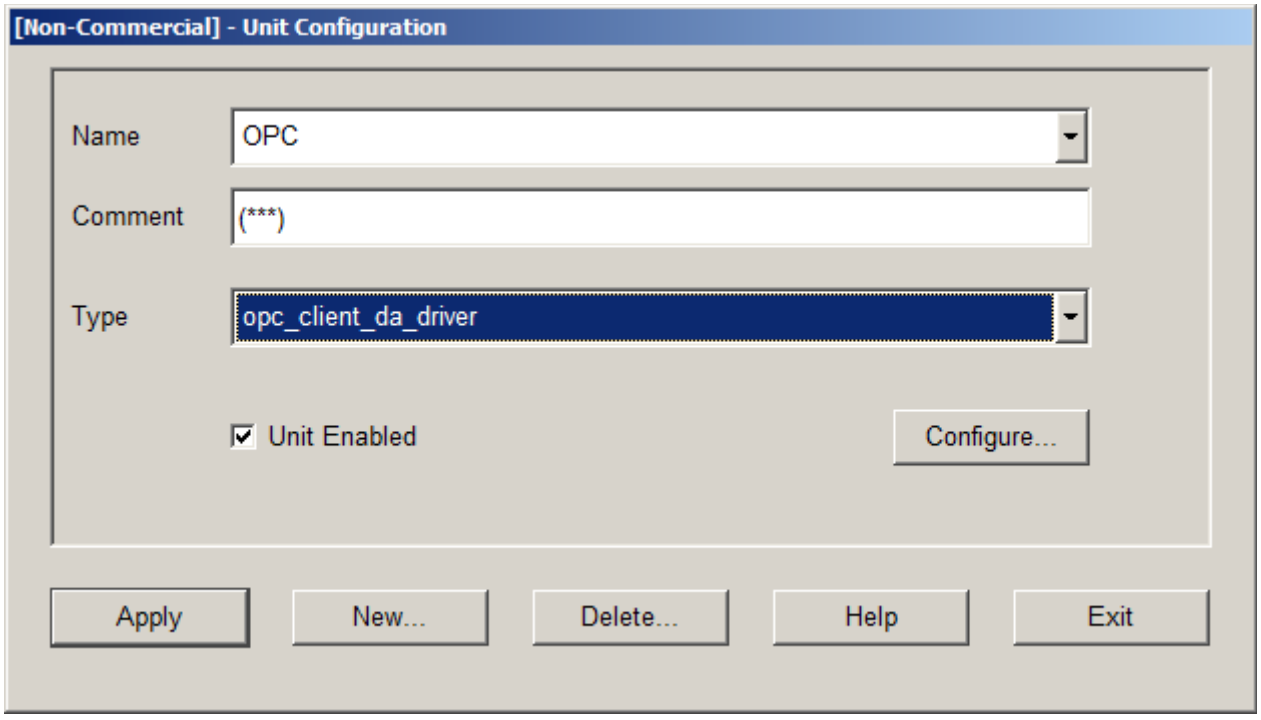
2 - Press button New



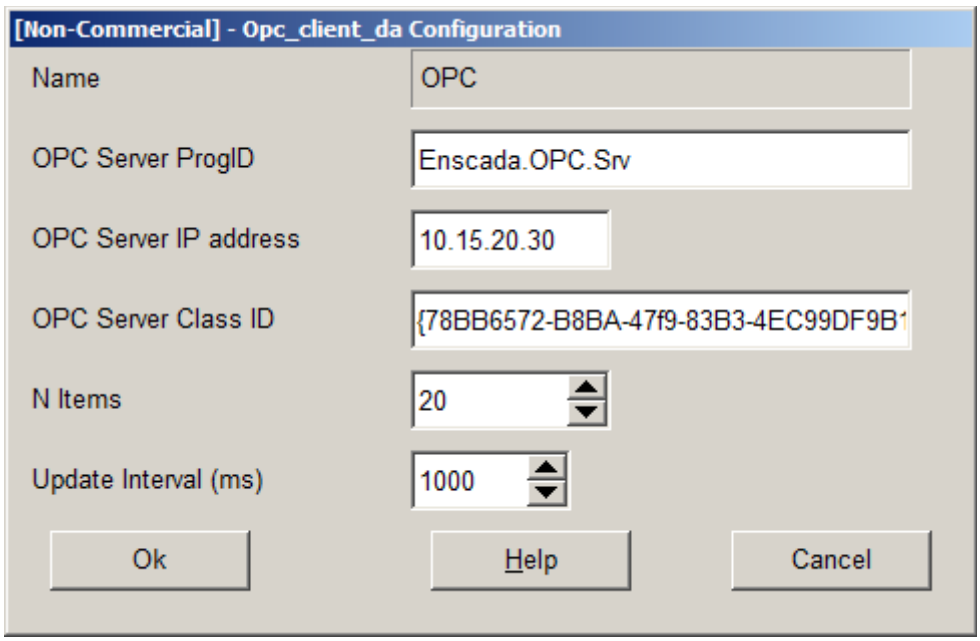
3 - Fill in the edit field Name: OPC



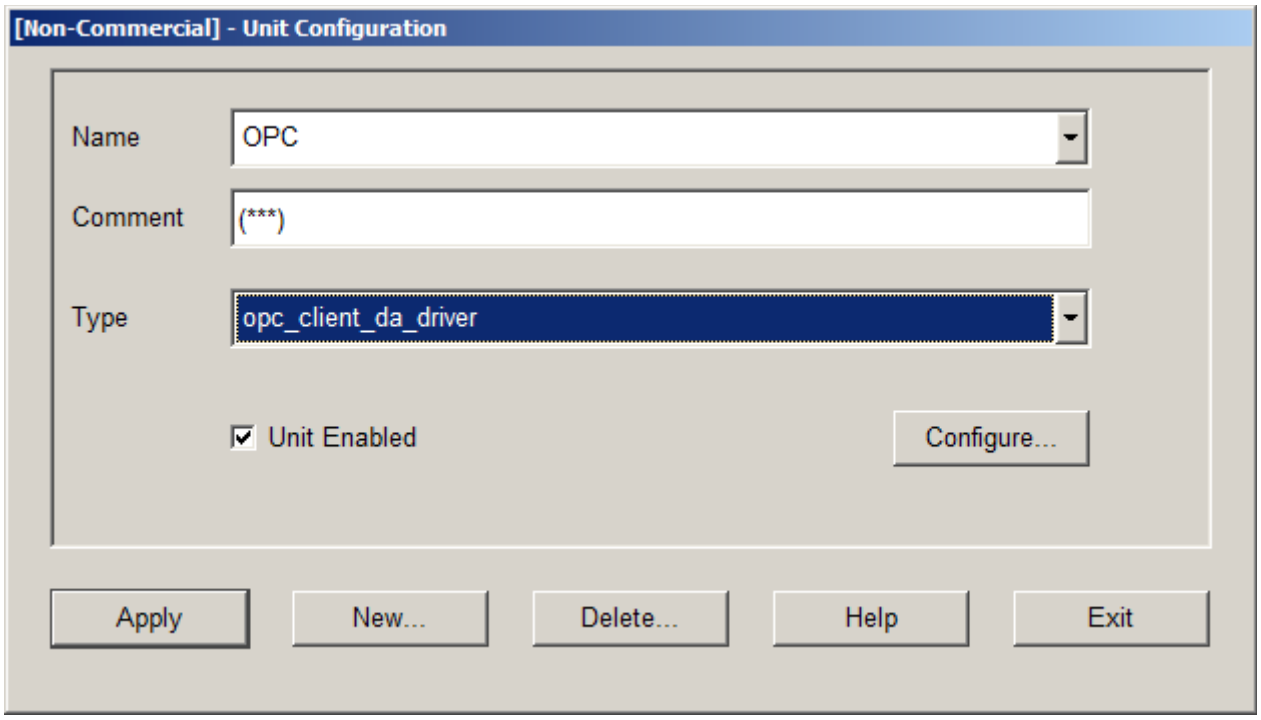
4 - Select Type opc_client_da_driver, flag Unit Enabled and press Configure button



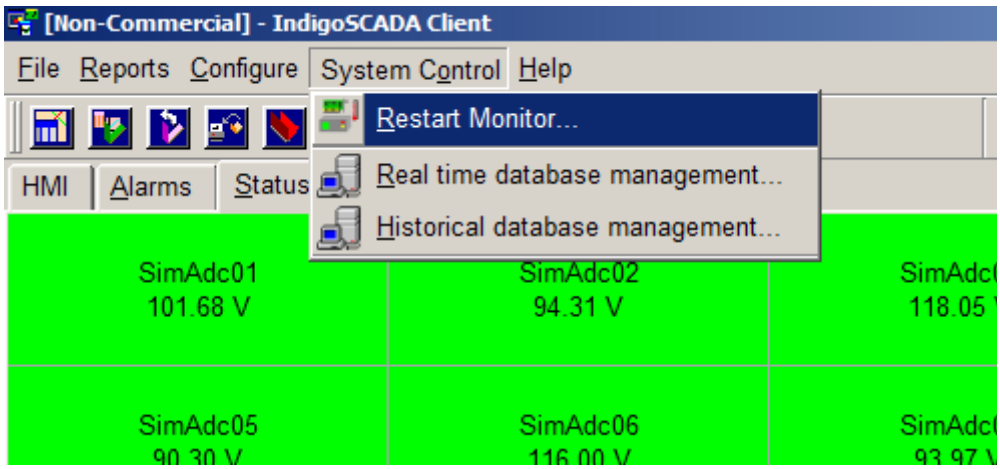
5 - Fill in the OPC Server ProgID, OPC Server IP address, OPC Server Class ID, number of Items and Update Interval in milliseconds. If IP address is left empty, local OPC DA server is used.
If OPC Server Class ID is left empty, it is necessary to configure opcenum.exe on the OPC server host.



6 - Press button Apply and wait that the Apply button returns active and then press Exit



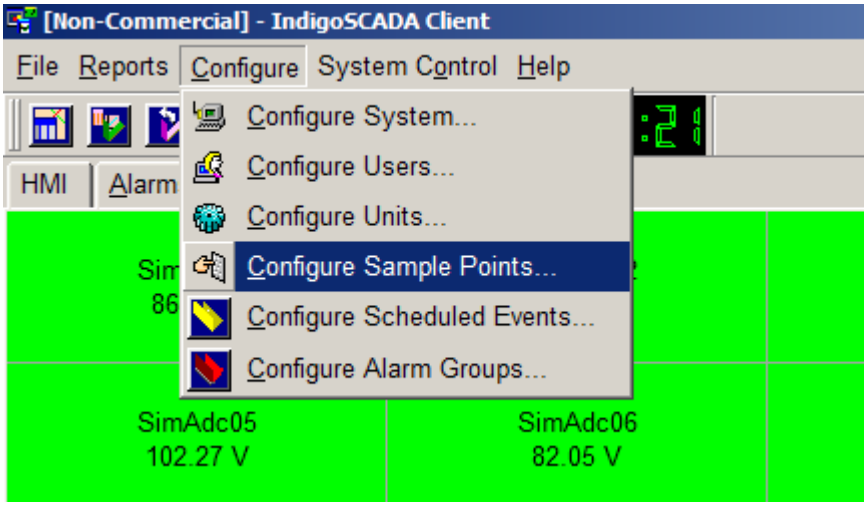
7 - Restart monitor.exe



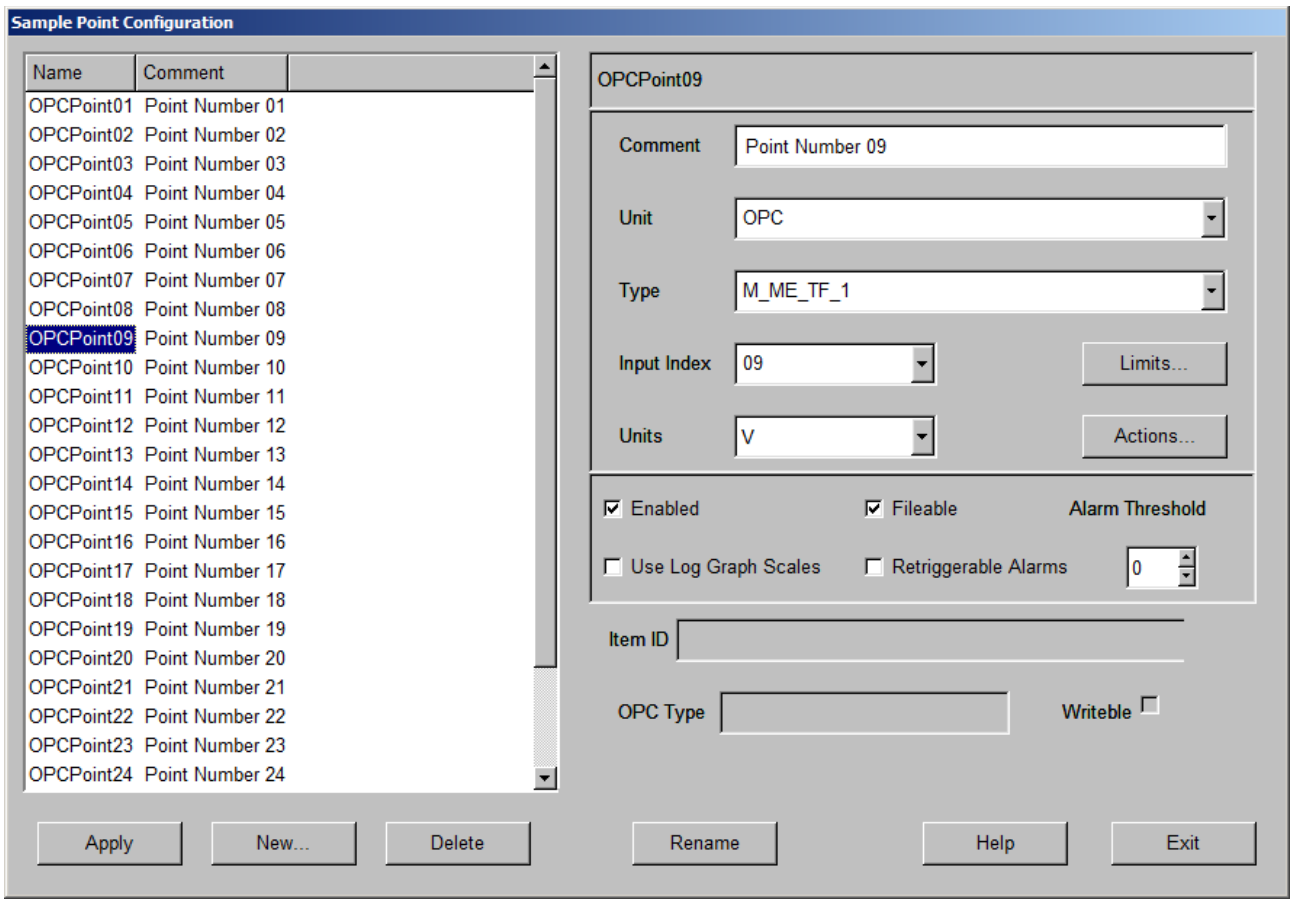
Step 4:

Configure each sample point

8 - Select the menu: Configure\Configure Sample Points



9 - Set Comment, Type, Units and the limits of each new point. Press button Apply



- Set type M_SP_TB_1 for digital OPC point (i.e VT_BOOL OPC type or any scada point that evaluate to 0 and 1)

- Set type M_ME_TF_1 for analog OPC point

Step 5:

HMI configuration with HMI designer

10 - Run the process C:\scada\bin\hmi_designer.exe

11 - Open the XML designer file C:\scada\project\hmi0.ui

11.1 - Or create from scratch your HMI (dialog) and save it as hmi0.ui in C:\scada\project

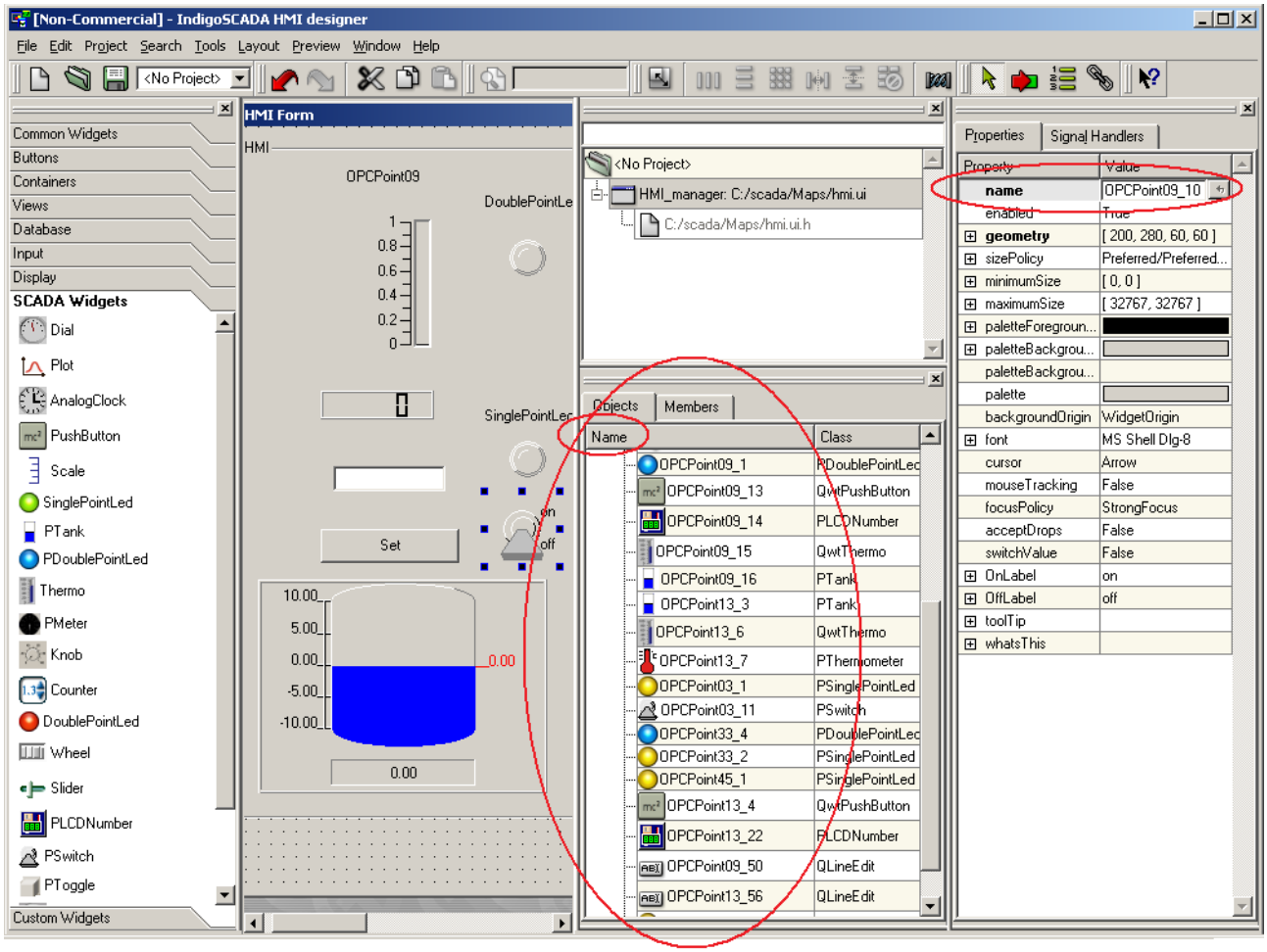
12 - Use the following widget classes

Widget	Class
SINGLE POINT LED	PSinglePointLed
DOUBLE POINT LED	PDoublePointLed
SINGLE POINT LED	SinglePointLed
DOUBLE POINT LED	DoublePointLed
SWITCH	PSwitch
THERMOMETER	QwtThermo
LCD NUMBER	PLCDNumber
COMMAND BUTTON	QPushButton
TANK	PTank
THERMOMETER	PThermometer
EDIT FIELD	QLineEdit
BREAKER	Breaker
DOUBLEBREAKER	DoubleBreaker
PLOT	QwtPlot
METER	Pmeter

13 - Widget names must contain the underscore _ as separator between sample point SCADA name and widget ID

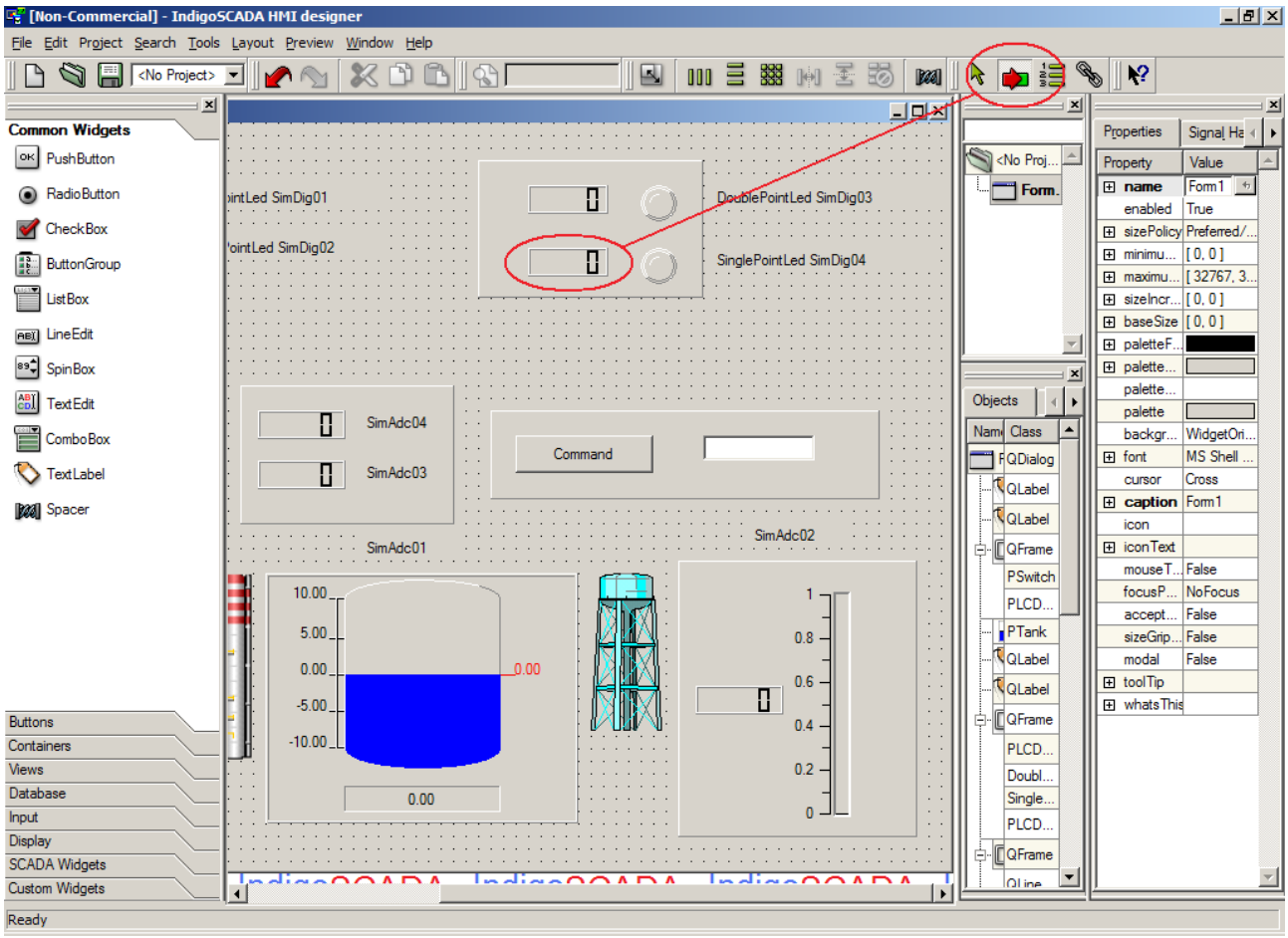
For example the widget with SCADA name OPCPoint09 and widget ID 10 has full name: OPCPoint09_10

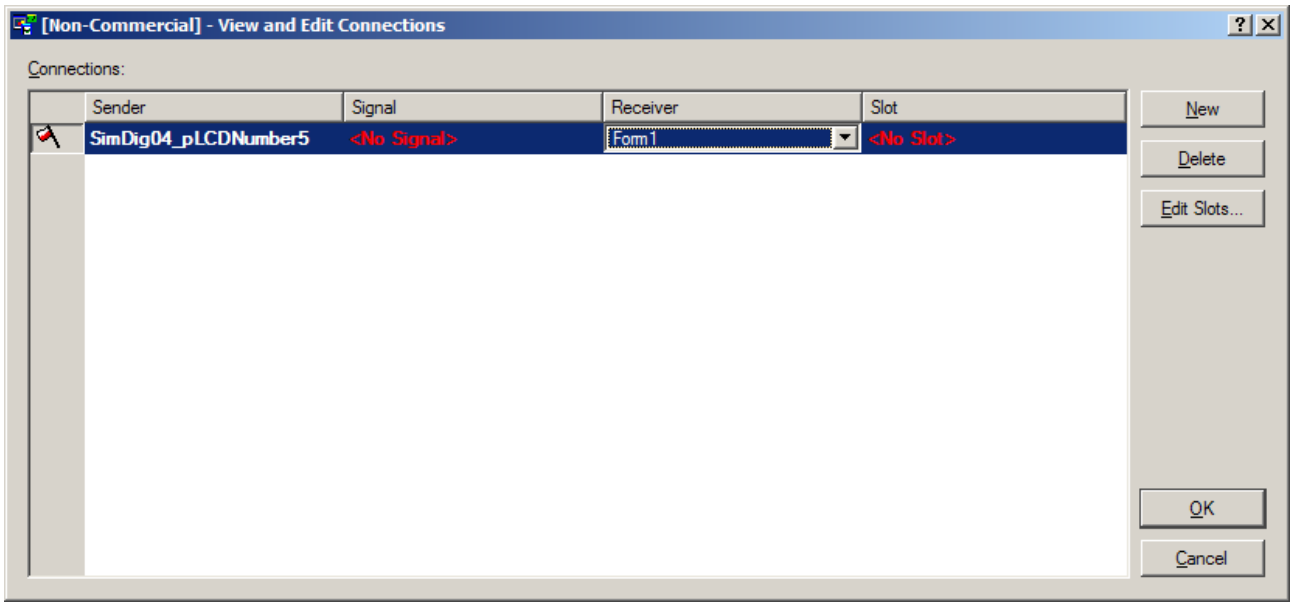
Please see next image 1

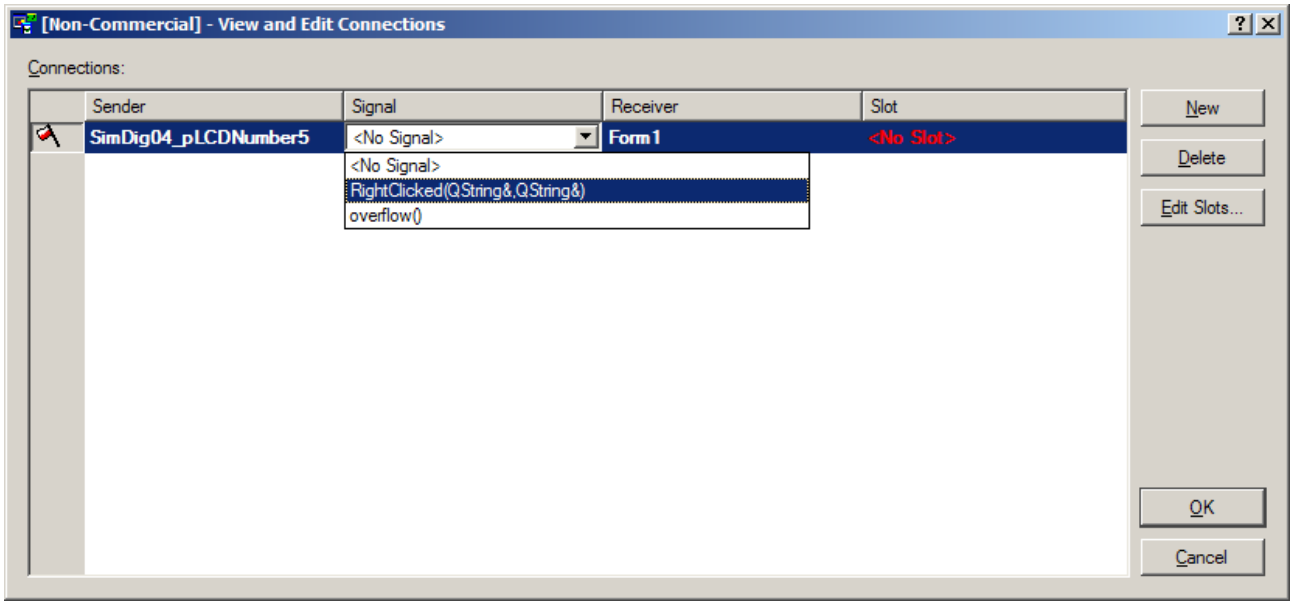


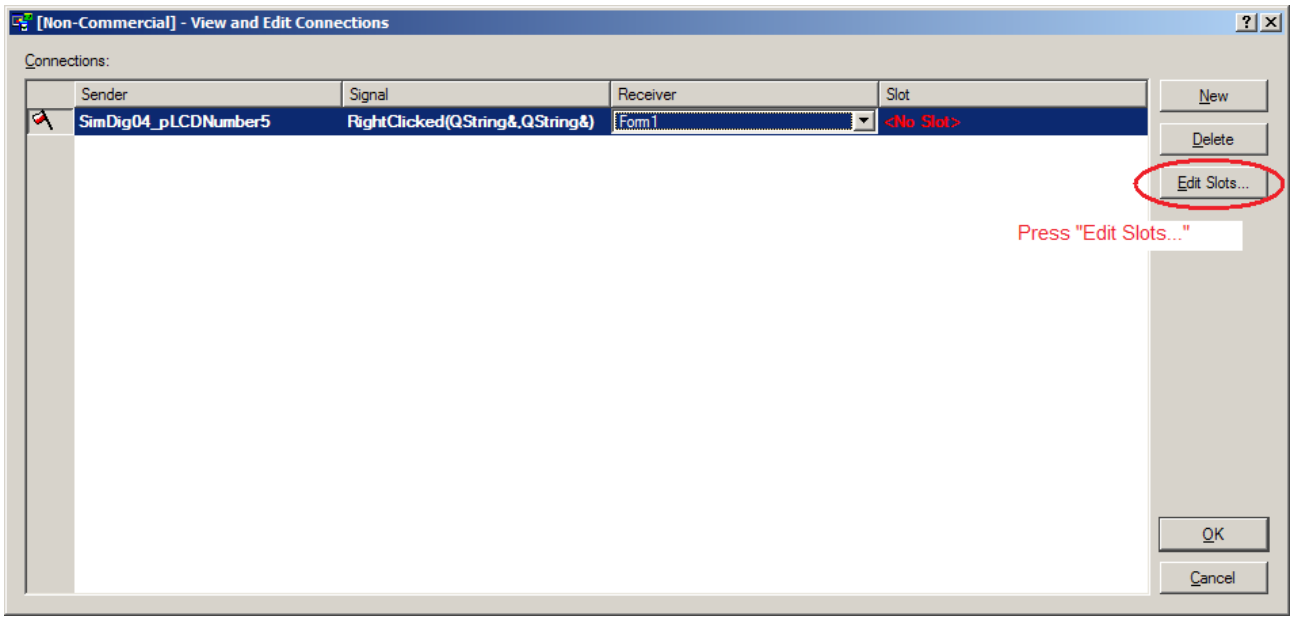
14 - To enable the inspect popup window, please follow the procedure:

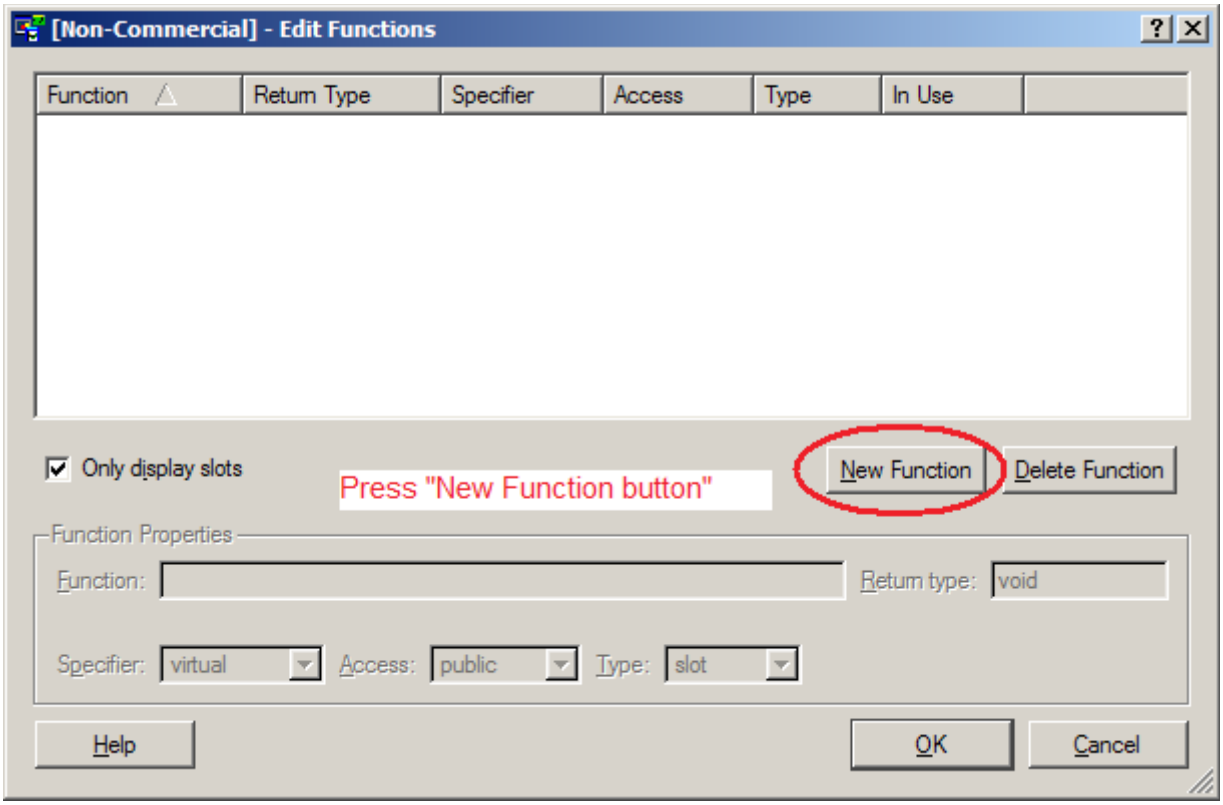
You have to connect the signals RightClicked(QString&,QString&) of the widgets with the slot RightClicked(QString&,QString&) of the HMI dialog (here Form1)

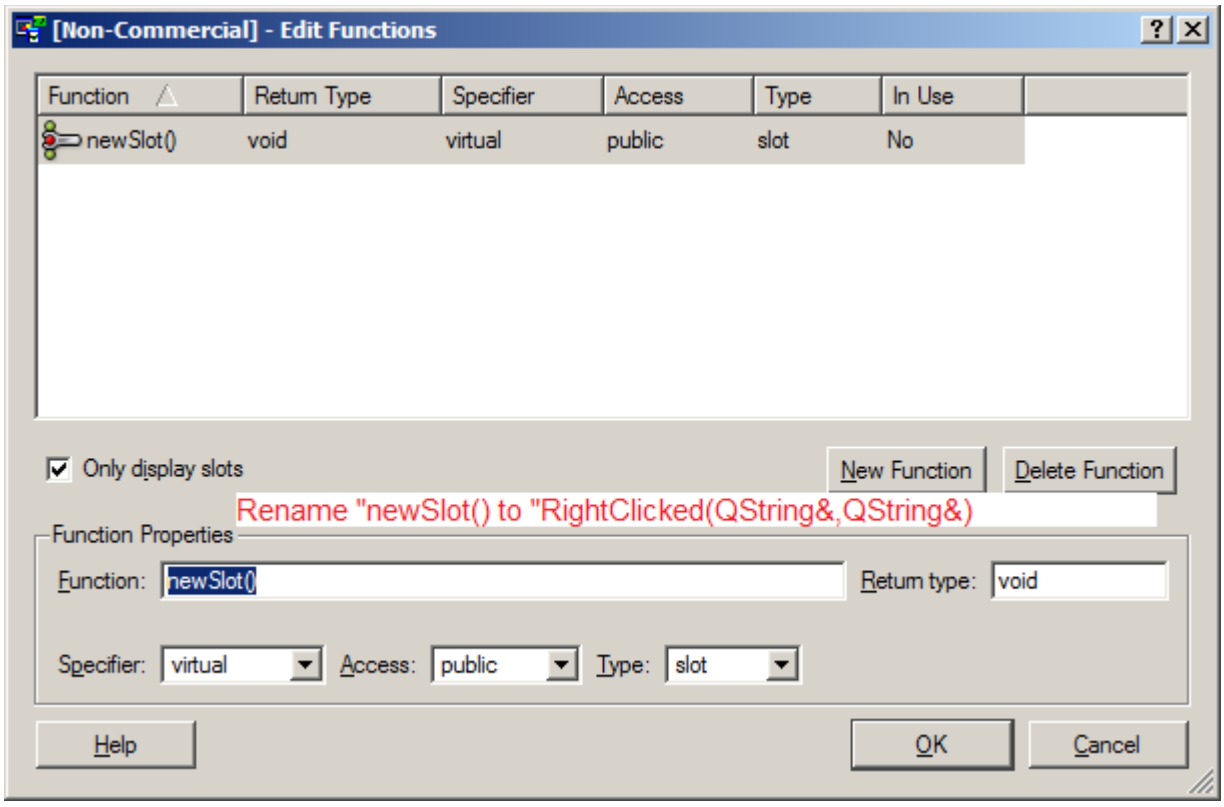

[Non-Commercial] - IndigoSCADA HMI designer
File Edit Project Search Tools Layout Preview Window Help
<No Project>
Common Widgets
PushButton
RadioButton
CheckBox
ButtonGroup
ListBox
LineEdit
SpinBox
TextEdit
ComboBox
TextLabel
Spacer
Buttons
Containers
Views
Database
Input
Display
SCADA Widgets
Custom Widgets
DoublePointLed SimDig03
SinglePointLed SimDig04
SimAdc04
SimAdc03
Command
SimAdc01
SimAdc02
10.00
5.00
0.00
-5.00
-10.00
0.00
1
0.8
0.6
0.4
0.2
0
Objects
Properties
Property
Value
Ready

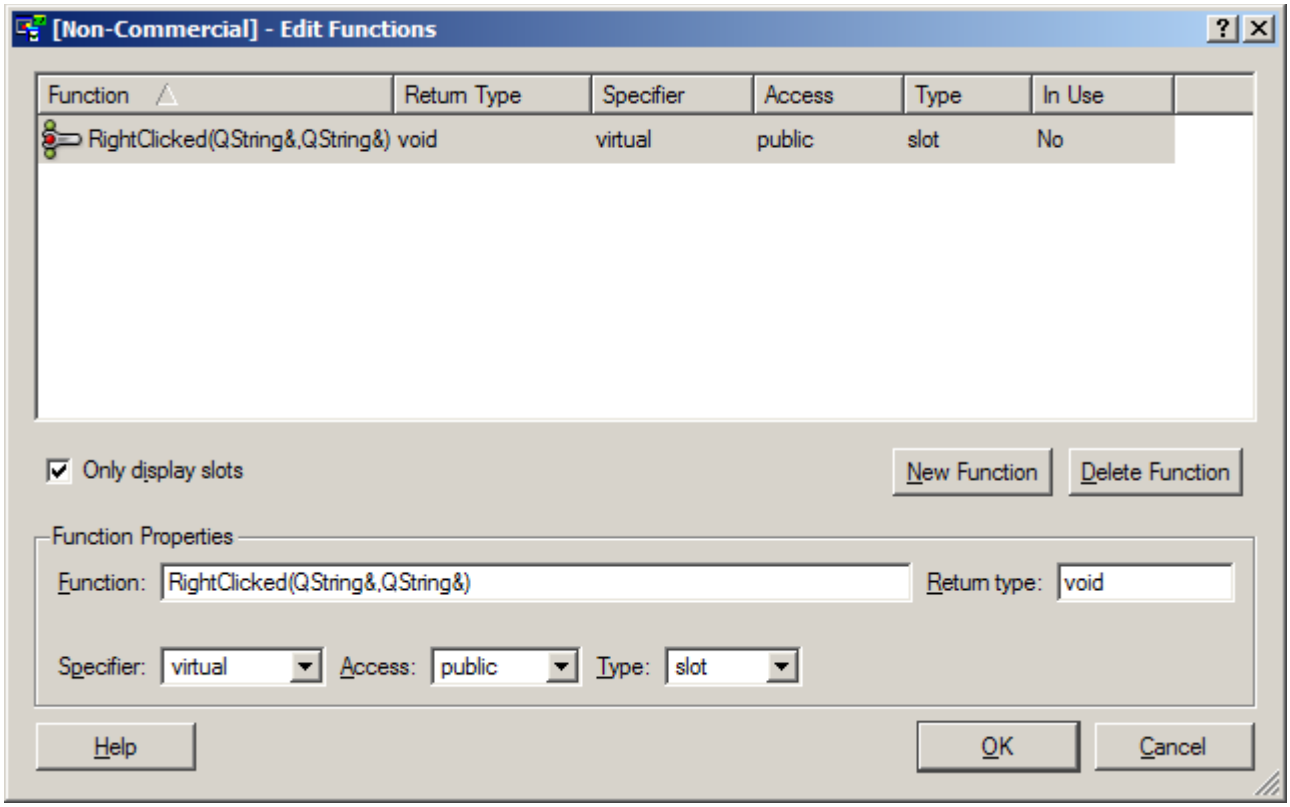

[Non-Commercial] - View and Edit Connections
Connections:
Sender
Signal
Receiver
Slot
SimDig04_pLCDNumber5
<No Signal>
Form1
<No Slot>
New
Delete
Edit Slots...
OK
Cancel

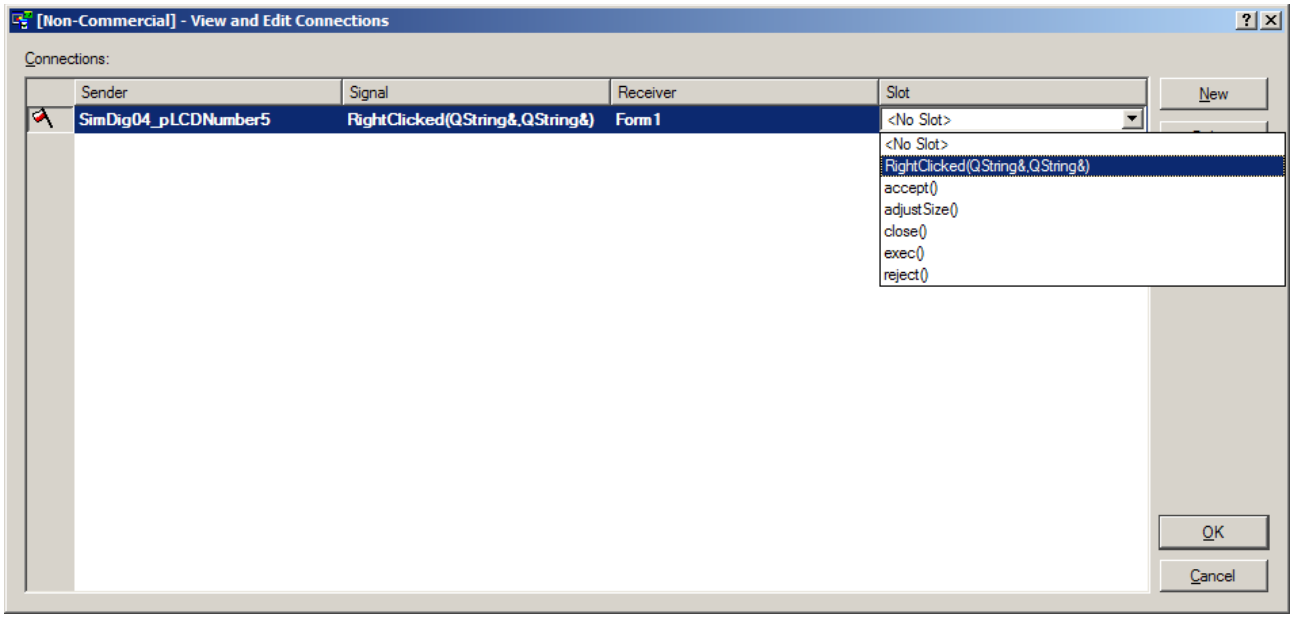

[Non-Commercial] - View and Edit Connections
Connections:
Sender
Signal
Receiver
Slot
SimDig04_pLCDNumber5
<No Signal>
Form1
<No Slot>
<No Signal>
RightClicked(QString&,QString&)
overflow()
New
Delete
Edit Slots...
OK
Cancel

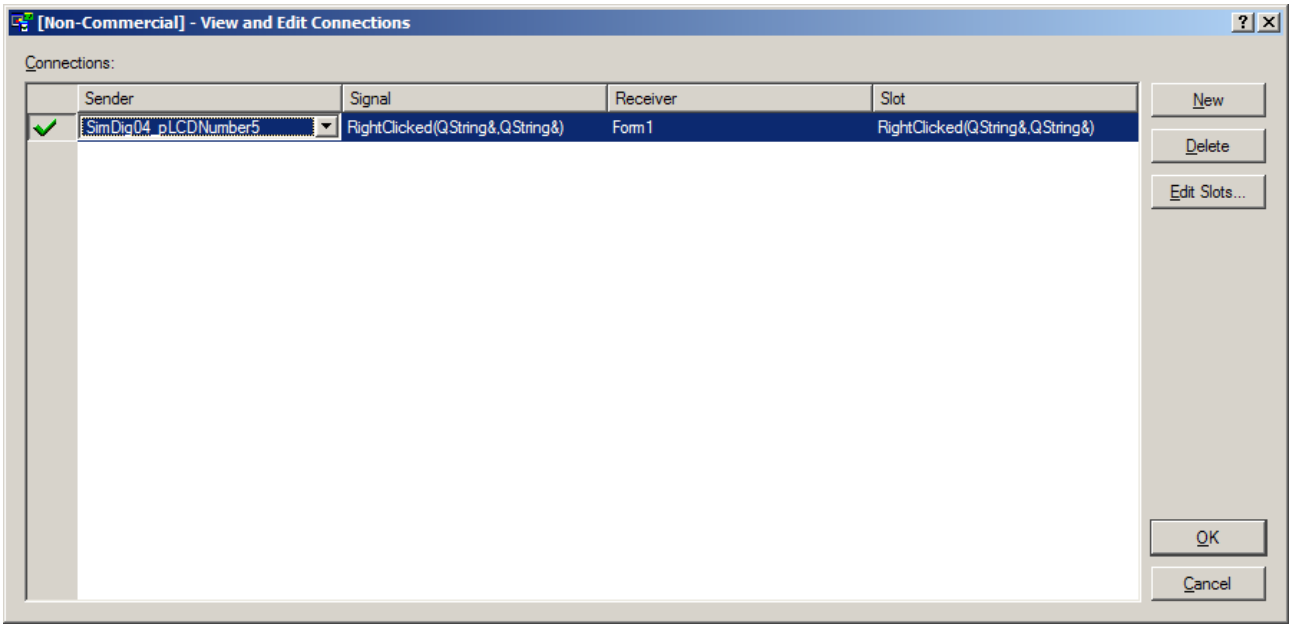

[Non-Commercial] - View and Edit Connections
Connections:
Sender
Signal
Receiver
Slot
SimDig04_pLCDNumber5
RightClicked(QString&,QString&)
Form1
<No Slot>
New
Delete
Edit Slots...
Press "Edit Slots..."
OK
Cancel

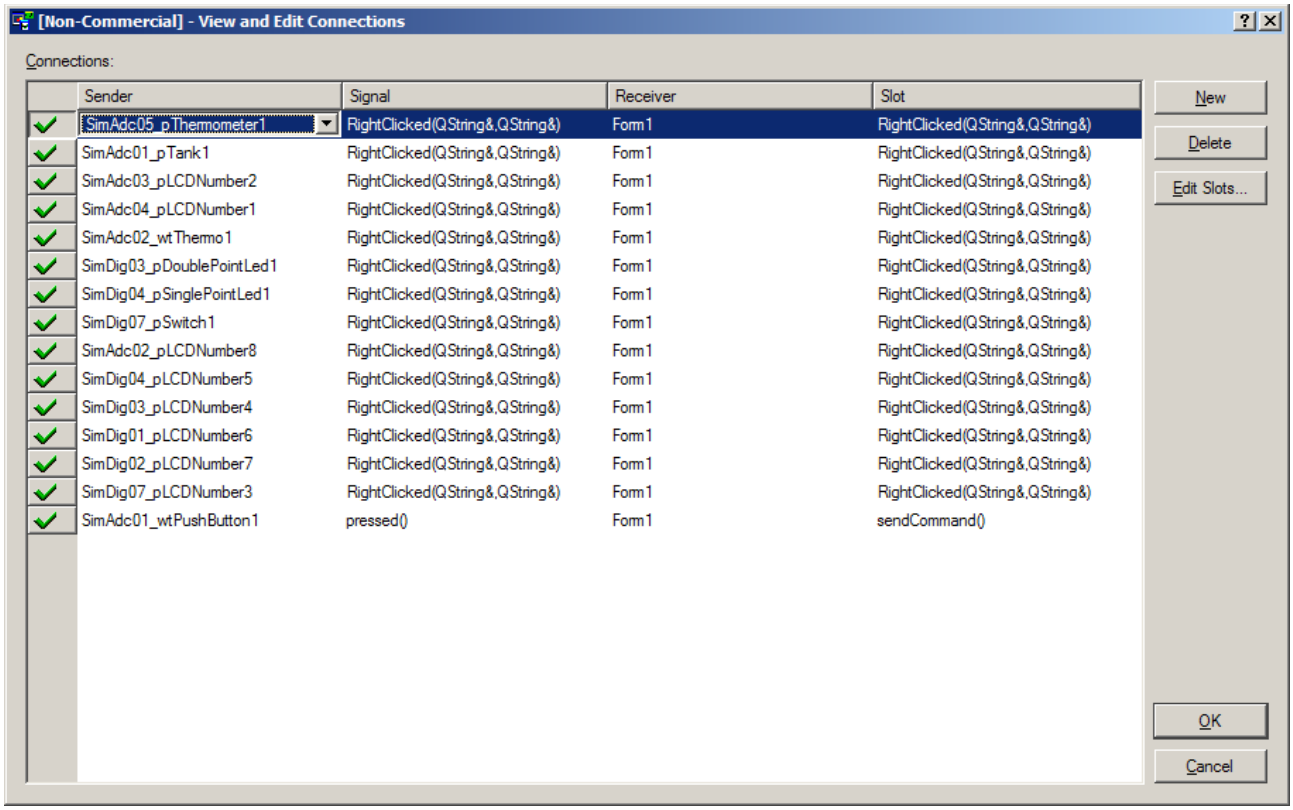

[Non-Commercial] - Edit Functions
Function
Return Type
Specifier
Access
Type
In Use
Only display slots
Press "New Function button"
New Function
Delete Function
Function Properties
Function:
Return type:
void
Specifier:
virtual
Access:
public
Type:
slot
Help
OK
Cancel

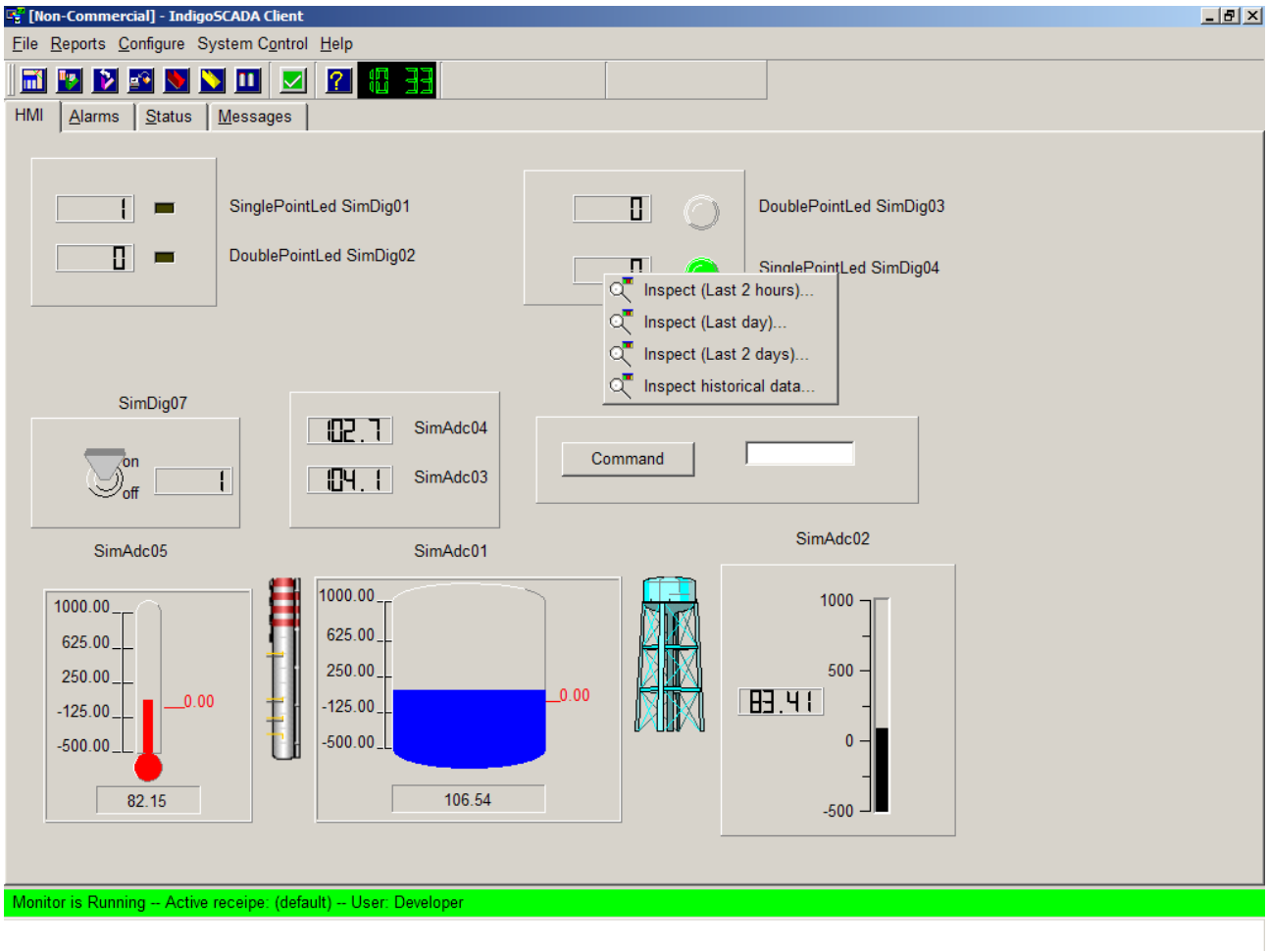

[Non-Commercial] - Edit Functions
Function
Return Type
Specifier
Access
Type
In Use
newSlot()
void
virtual
public
slot
No
Only display slots
New Function
Delete Function
Rename "newSlot() to "RightClicked(QString&,QString&)
Function Properties
Function:
newSlot()
Return type:
void
Specifier:
virtual
Access:
public
Type:
slot
Help
OK
Cancel


[Non-Commercial] - Edit Functions
Function
Return Type
Specifier
Access
Type
In Use
RightClicked(QString&,QString&)
void
virtual
public
slot
No
Only display slots
New Function
Delete Function
Function Properties
Function:
RightClicked(QString&,QString&)
Return type:
void
Specifier:
virtual
Access:
public
Type:
slot
Help
OK
Cancel


[Non-Commercial] - View and Edit Connections
Connections:
Sender
Signal
Receiver
Slot
SimDig04_pLCDNumber5
RightClicked(QString&,QString&)
Form1
<No Slot>
<No Slot>
RightClicked(QString&,QString&)
accept()
adjustSize()
close()
exec()
reject()
New
OK
Cancel


[Non-Commercial] - View and Edit Connections
Connections:
Sender
Signal
Receiver
Slot
SimDig04_pLCDNumber5
RightClicked(QString&,QString&)
Form1
RightClicked(QString&,QString&)
New
Delete
Edit Slots...
OK
Cancel


[Non-Commercial] - View and Edit Connections
Connections:
Sender
Signal
Receiver
Slot
SimAdc05_pThermometer1
RightClicked(QString&,QString&)
Form1
RightClicked(QString&,QString&)
SimAdc01_pTank1
SimAdc03_pLCDNumber2
SimAdc04_pLCDNumber1
SimAdc02_wtThermo1
SimDig03_pDoublePointLed1
SimDig04_pSinglePointLed1
SimDig07_pSwitch1
SimAdc02_pLCDNumber8
SimDig04_pLCDNumber5
SimDig03_pLCDNumber4
SimDig01_pLCDNumber6
SimDig02_pLCDNumber7
SimDig07_pLCDNumber3
SimAdc01_wtPushButton1
pressed()
sendCommand()
New
Delete
Edit Slots...
OK
Cancel


[Non-Commercial] - IndigoSCADA Client
File Reports Configure System Control Help
HMI Alarms Status Messages
SinglePointLed SimDig01
DoublePointLed SimDig02
DoublePointLed SimDig03
SinglePointLed SimDig04
Inspect (Last 2 hours)...
Inspect (Last day)...
Inspect (Last 2 days)...
Inspect historical data...
SimDig07
on
off
SimAdc04
SimAdc03
Command
SimAdc05
SimAdc01
SimAdc02
1000.00
625.00
250.00
-125.00
-500.00
0.00
82.15
106.54
1000
500
0
-500
Monitor is Running -- Active receipe: (default) -- User: Developer

Advanced topics

Configuration of IndigoSCADA as IoT gateway

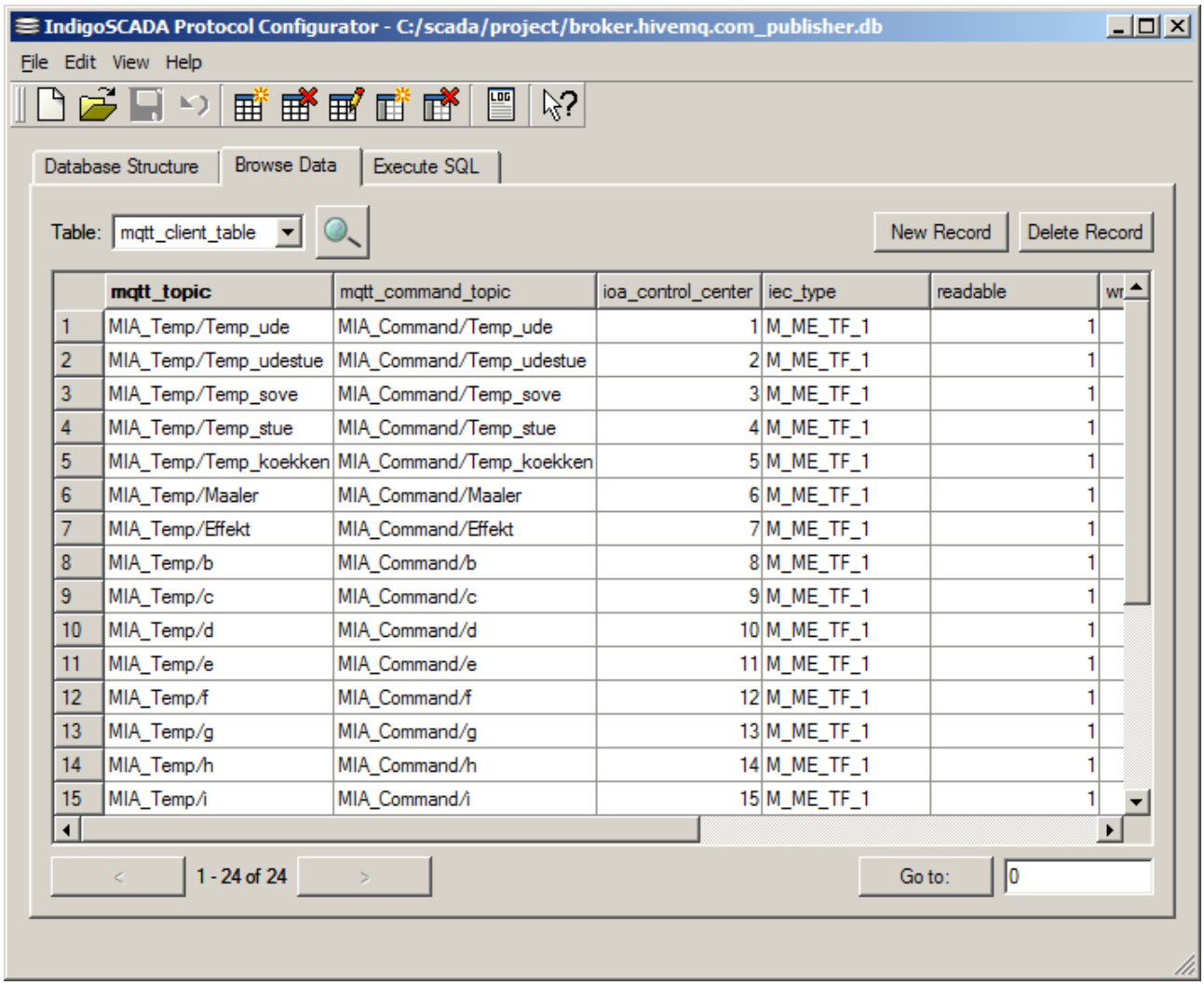
IndigoSCADA supports MQTT/Sparkplug B as payload encoding.

Open file C:\scada\project\manager.ini. Add the following section to activate MQTT client publisher for sending messages to broker broker.hivemq.com:

```
[Process6]
Process = mqtt_client_publisher.exe -a broker.hivemq.com -p MIA_Command/# -l 1 -t 1883
PauseStart= 1000
PauseEnd= 1000
UserInterface = Yes
Restart = Yes
```

Open file broker.hivemq.com_publisher.db with protocol_configurator.exe and configure each ioa_control_center for every mqtt_topic.

ioa_control_center has to match with the one of input protocol, eg. modbus



	mqtt_topic	mqtt_command_topic	ioa_control_center	iec_type	readable	wr
1	MIA_Temp/Temp_ude	MIA_Command/Temp_ude	1	M_ME_TF_1	1	
2	MIA_Temp/Temp_udestue	MIA_Command/Temp_udestue	2	M_ME_TF_1	1	
3	MIA_Temp/Temp_sove	MIA_Command/Temp_sove	3	M_ME_TF_1	1	
4	MIA_Temp/Temp_stue	MIA_Command/Temp_stue	4	M_ME_TF_1	1	
5	MIA_Temp/Temp_koekken	MIA_Command/Temp_koekken	5	M_ME_TF_1	1	
6	MIA_Temp/Maaler	MIA_Command/Maaler	6	M_ME_TF_1	1	
7	MIA_Temp/Effekt	MIA_Command/Effekt	7	M_ME_TF_1	1	
8	MIA_Temp/b	MIA_Command/b	8	M_ME_TF_1	1	
9	MIA_Temp/c	MIA_Command/c	9	M_ME_TF_1	1	
10	MIA_Temp/d	MIA_Command/d	10	M_ME_TF_1	1	
11	MIA_Temp/e	MIA_Command/e	11	M_ME_TF_1	1	
12	MIA_Temp/f	MIA_Command/f	12	M_ME_TF_1	1	
13	MIA_Temp/g	MIA_Command/g	13	M_ME_TF_1	1	
14	MIA_Temp/h	MIA_Command/h	14	M_ME_TF_1	1	
15	MIA_Temp/i	MIA_Command/i	15	M_ME_TF_1	1	

Support

For questions about **IndigoSCADA**, please email your request to the following e-mail or use the forum at Sourceforge site.

E-MAIL: info at enscada.com

SITE: https://sourceforge.net/projects/indigoscada/

SITE: https://www.enscada.com/a7khg9/indigoscada.html

FORUM: https://sourceforge.net/p/indigoscada/discussion/users/