New Features in ibaPDA v6.37.0

Author: iba Gent

Date: 15/12/2016

Table of contents

1 QPanel: Shapes

A new shape control was added to QPanel.

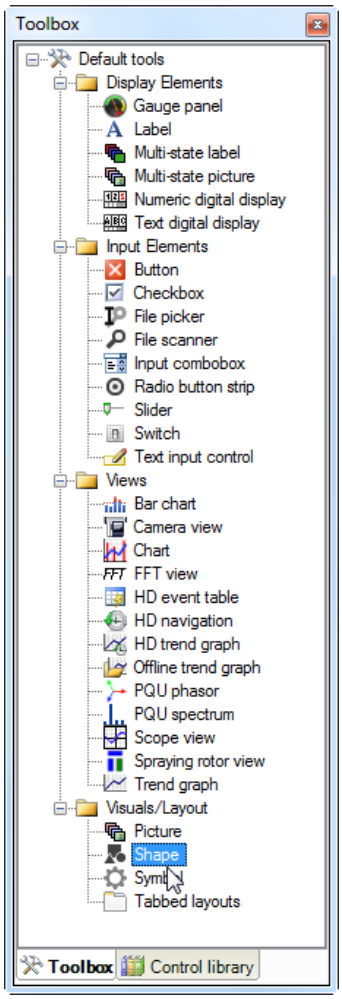


Figure 1 – The location of the shape tool in the toolbox.

There are four basic shapes supported:

- Rectangle
- Line
- Ellipse
- Polygon

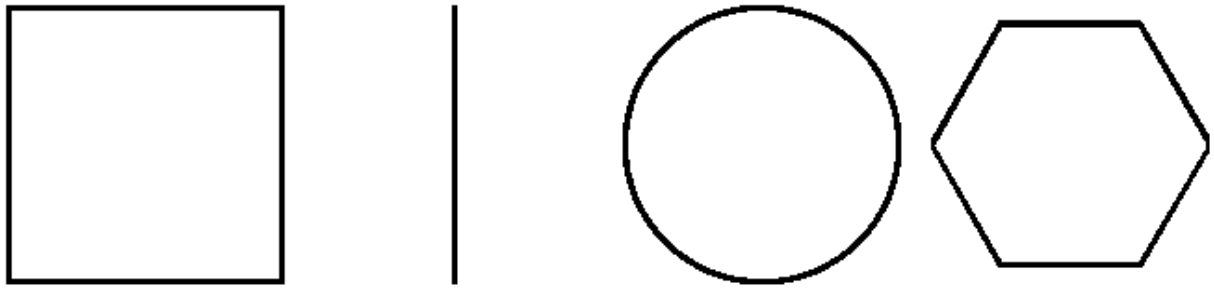


Figure 2 – The different types of shape.

The shape type can either be chosen from the property window, or by using the task menu in the designer, which can be opened by clicking the triangle in the top right corner.

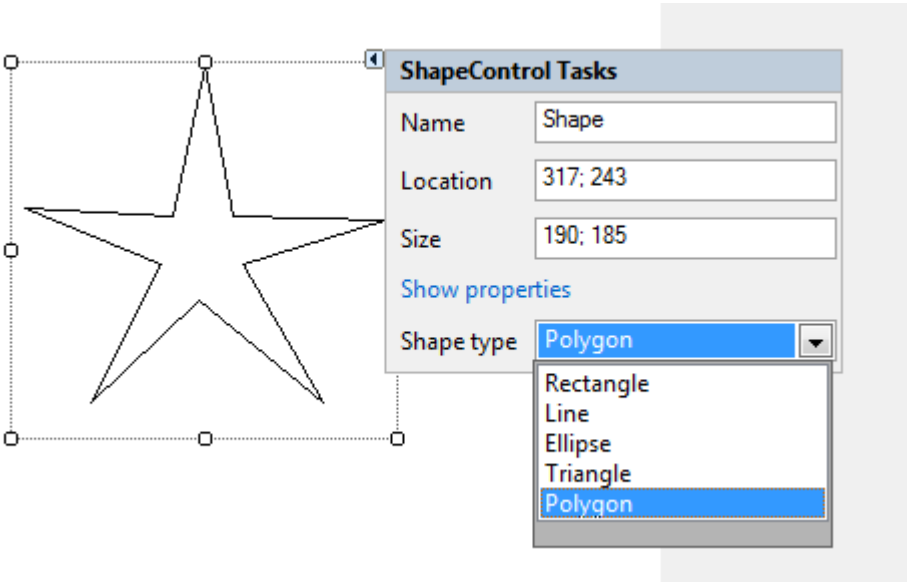


Figure x – Using the action menu to change the shape type.

1.1 Properties

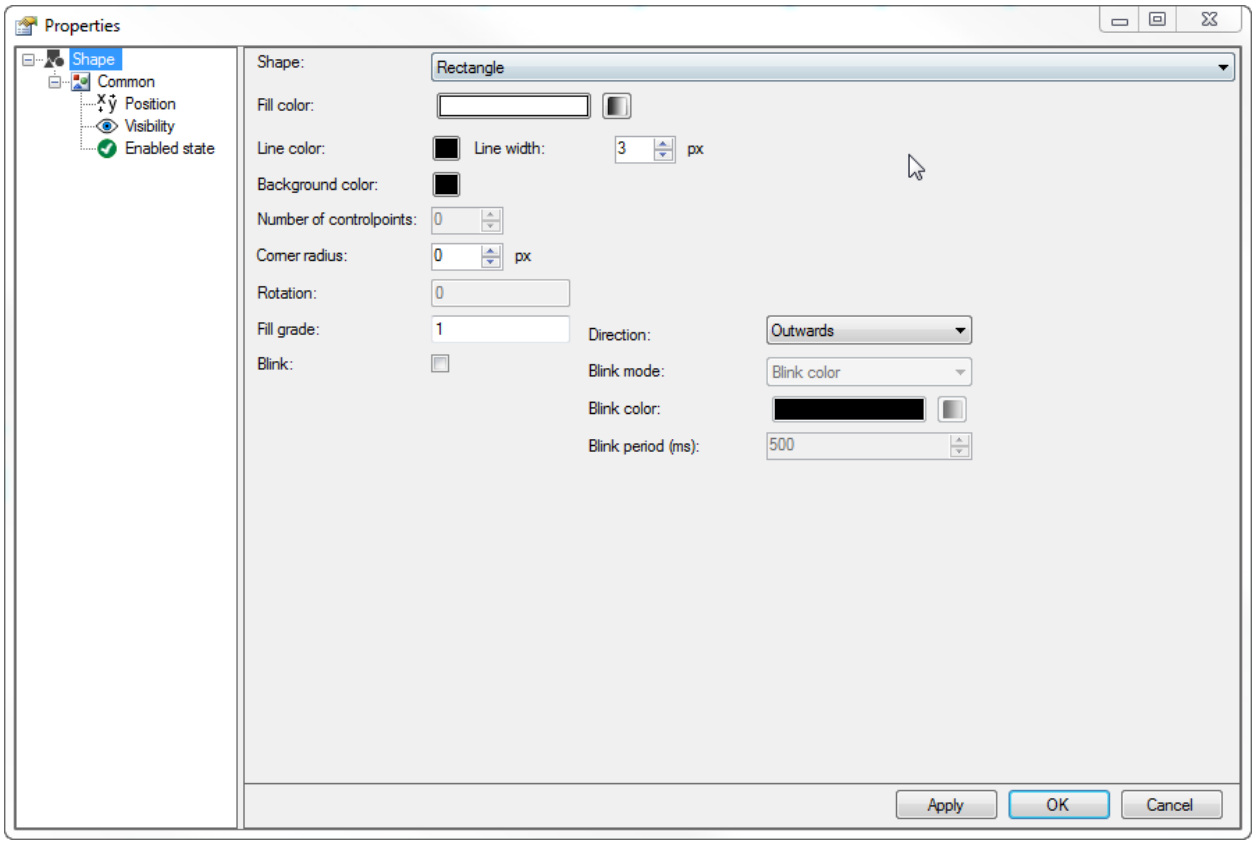


Figure x – The property window of a shape.

The following properties can be configured:

Shape type – Choose the shape type: rectangle, line, ellipse or polygon

Fill color – Choose a solid color or define a gradient. Not applicable when a line is chosen.

Line color/width – Choose the type of line used.

Background color – The background color, usually has to be set the match the background. This color is used when the fill grade is not set to 1.

Number of controlpoints – The number of moveable controlpoints when shape type is set to either line or polygon.

Corner radius – If the shape type is set to rectangle rounded corners can be used. This property defines the radius in pixels of the rounded corners.

Rotation – if the shape type is a polygon, the amount of rotation that is applied to the shape.

Fill grade/direction – If set to 1, the shape is filled entirely. If less than one, the shape is filled according the defined fill grade. It is possible to define the fill grade direction, this defines from which side the shape gets filled.

Blink – If enabled the shape will blink if the designer is disabled. There are two blinking modes: Transparent: the shape alternates between hidden and visible state. If hidden the background will be shown. Color: The shapes alternates between the color defined with fill color and the color defined with blink color. Blink period is the time in milliseconds of a cycle on and off.

1.1.1 Defining gradients

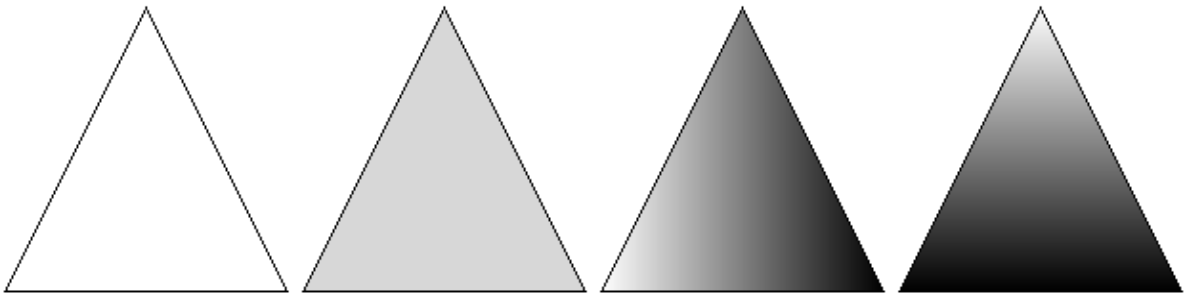


Figure 3 – Different fill colors: White, gray, horizontal gradient and vertical gradient

The gradient can be edited defined using the gradient slider:

Figure 4 – The gradient editor.

By double clicking on the squares at each end the color of that anchor point can be changed. To insert a new point, double click at any point on the gradient. An anchorpoint can be moved by dragging it with the mouse, note that the end points are fixed and cannot be moved. You can delete a custom anchorpoint by selecting it with left click and pressing ‘delete’.

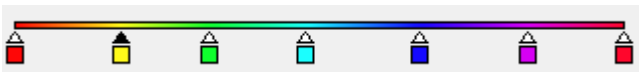


Figure x – Using the gradient editor to create a rainbow.

1.2 Line

By default the line shape can used to make horizontal or vertical lines, by stretching the bounding box of the control either horziontally or vertically.

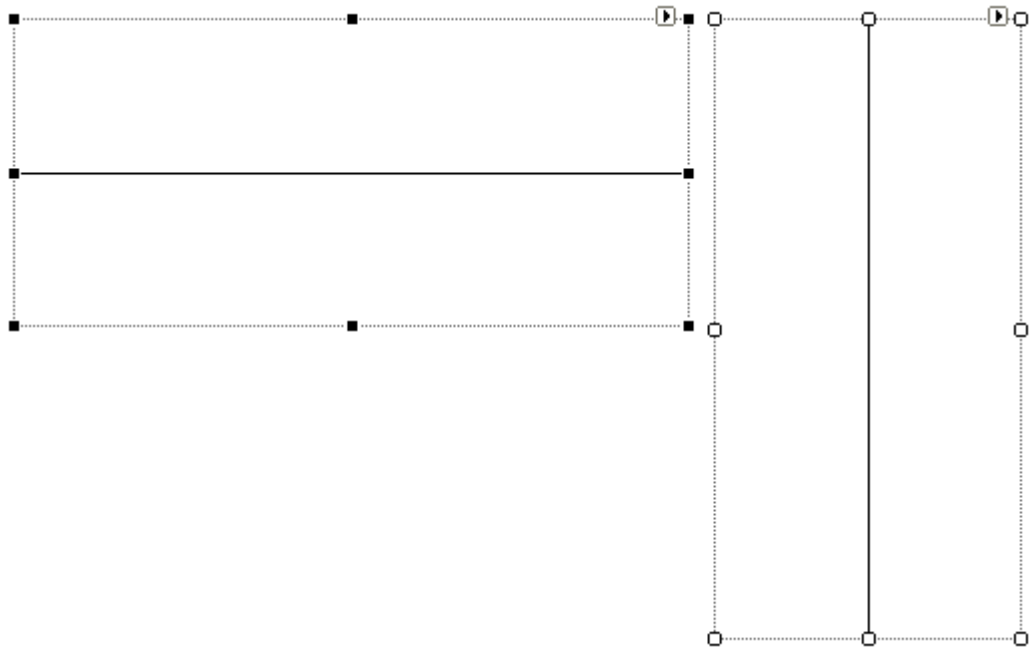


Figure x – Creating horizontal/vertical lines.

Custom lines can de created by setting the number of control points in the properties to a value of 2 or higher. After setting the required number of control points, the line can be edited by selecting the line in the designer and moving the handles of the control points around.

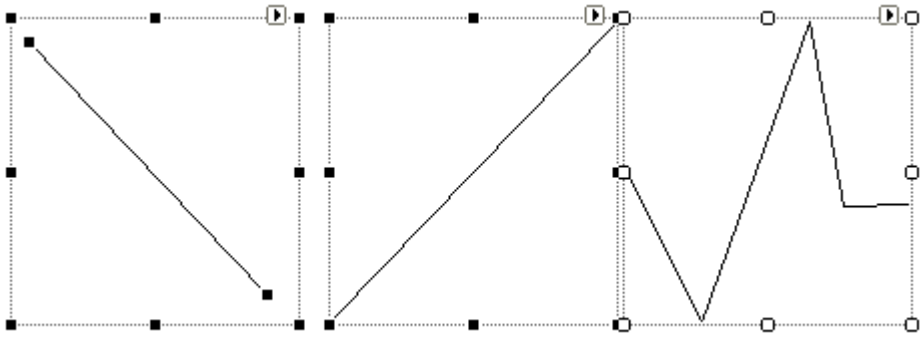


Figure x – Examples of custom lines.

When editing a custom line, new control points can be inserted by double clicking anywhere on the line. A control point can be removed by double clicking on a control point while holding shift.

1.3 Rectangle

To create recangles with rounder corners, set the corner radius property. This is the radius in pixel of the circle at the corners.

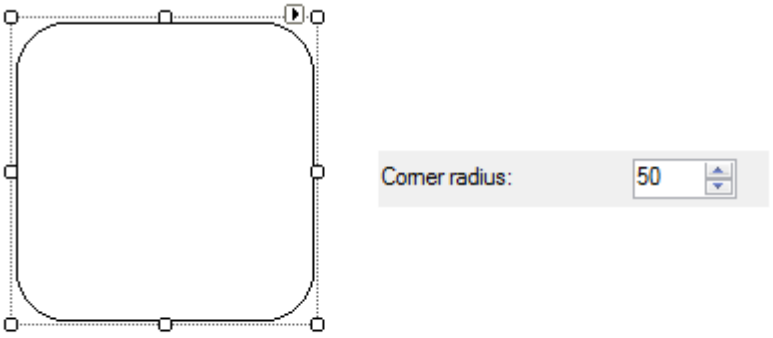


Figure x: Using rounded corners for a rectangle.

1.4 Polygon

Custom polygons can be created with the polygon shape. First define the number of control points required in the properties. After that the required shape can be designed by dragging the control points in the designer.

When editing a controlpoints, new control points can be inserted by double clicking anywhere on the line. A control point can be removed by double clicking on a control point while holding shift.

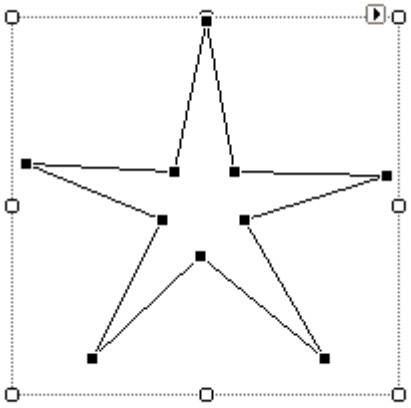


Figure x: A polygon in the designer

2 QPanel: Control library

The control library can be used to store configured controls for reuse at a later time or in a different layout or project. Control libraries can be exchanged between users.

The control library can be accessed in the toolbox of QPanel's designer. A second tab is created where library elements are placed.

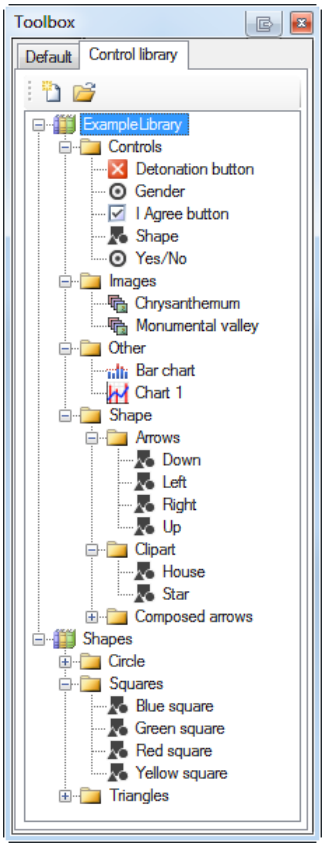


Figure x – The library toolbox with two loaded libraries.

2.1 Loading a library

Control libraries comes as files with the extension ".qpanellib". They can be loaded in toolbox by pressing the "Open" icon of the "Control library" tab, alternatively they can be dragged from Windows explorer.

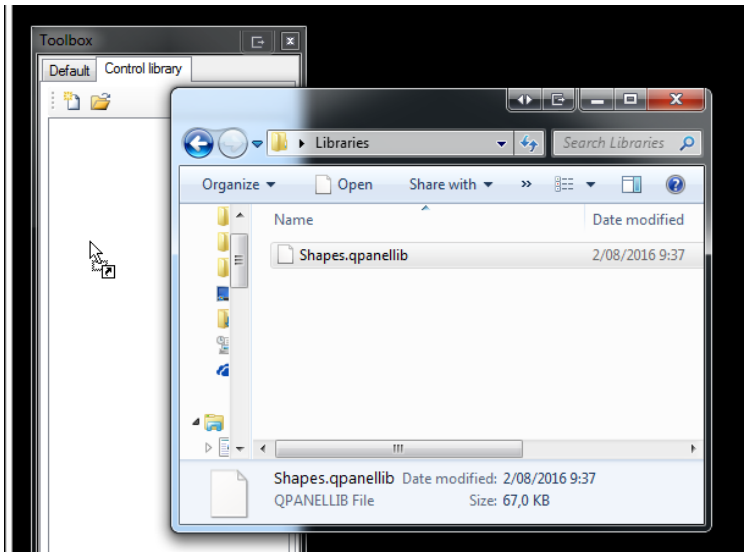


Figure x – Opening a library by dragging a library file into the toolbox.

2.2 Creating a new library

A new library can be created by either clicking the “New library” button or by dragging some control from the QPanel into empty space of the toolbox. The new library can be saved to disk by right clicking on the library and chosing “Save library” from the context menu.

2.3 Organizing your library

New controls can be added to the library by dragging them from the QPanel into a library in the toolbox. It is also possible to drag multiple controls at the same time, which will be added the library as a single item. Using the context menu which can be accessed by right clicking it is possible to remove or to rename items. To organize the library it is possible to add folders directly into the library or into other folders.

Changes to libraries will be only be saved to a file when explicitly clicking the save button. Changes will however remain when restarting ibaPDA.

3 QPanel: Editting multiple controls at the same time

A new editor has been added to edit the properties of multiple controls at the same time. Only basic properties can be edited using the new property editor, for example background color and size. More advanced properties can only be edited using the default property editor.

The new property editor can be activated from the designer when selecting multiple controls and chosing properties from the context menu.

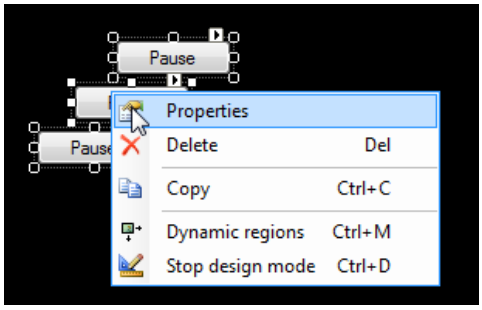


Figure x: Opening the new property editor

The property editor shows a list of available properties organized in two columns. Editting a value and clicking save will apply the new setting to every selected control.

There is also the possibility to open the property editor in a panel by clicking the property window button on the QPanel's toolbar. Changes to properties in this panel will be applied immediately. This panel responds to changes in selection of the QPanel and updates the available properties.

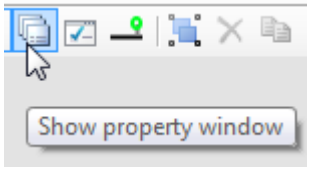


Figure x – Opening the property window as a dockable panel.

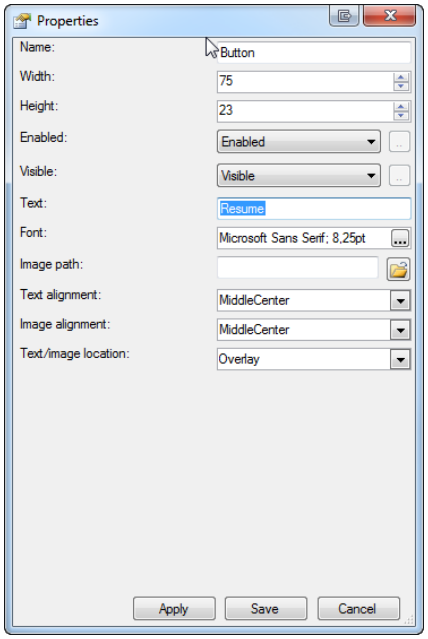


Figure x – properties of a button

A property can have multiple different values if multiple controls are selected. If this happens no value is shown, but instead the number of different values is shown. An edit button is available to override the property setting, effectively giving all selected controls the same value for that property.

Figure x – A property with multiple values.

Figure x – After clicking the edit button you can set a new value to all controls.

Some properties are shared between different controls, like background color or text. These properties can all be edited at once even if these properties are from different control types.

4 QPanel: Dynamic properties

With the newly introduced dynamic system it is possible to control properties dynamically with a signal. A dynamic consists of a selection of controls, properties and a signal. Insides the dynamic on number of states are defined. States map signal values to property values. A state is defined by a signal value and a property value for each property. The states are defined on increasing order of signal value. This means if a signal value increases continuesly the states are activated in order.

The states with their values and properties make a table, for example:

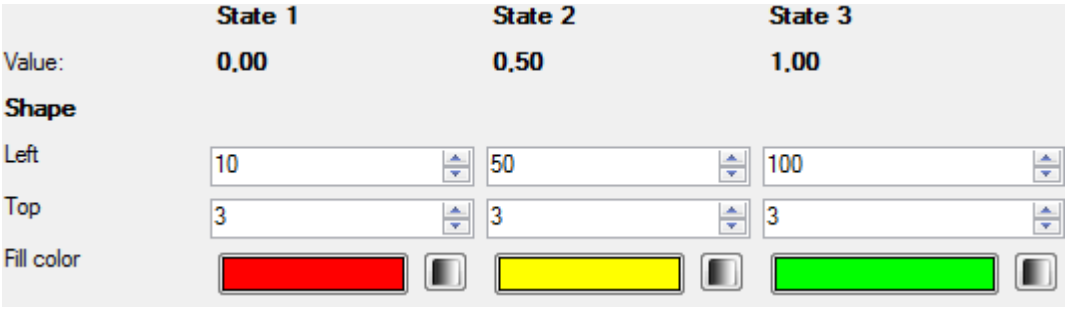


Figure x – Editing dynamics

This example dynamic maps a signal value 0 to the fill color red of a shape and the value 10 for the X-position. A signal value of 0.5 gets mapped to the fill color yellow and X-position 50. The signal value 1 gets mapped to fill color green and X-position 100.

4.1 Managing dynamics

Dynamics can be created using the dynamic editor which can be accessed in the design mode by opening the dynamics menu and choosing for “Dynamics Editor”

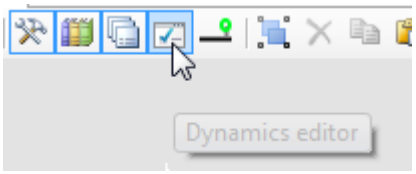


Figure x – Accessing the dynamics editor

To create a new dynamic inside the dynamic editor click  at the top of the window. Choose a name and select a signal for the newly defined dynamic. States can be defined by altering the QPanel while a state is selected in the dynamic editor.

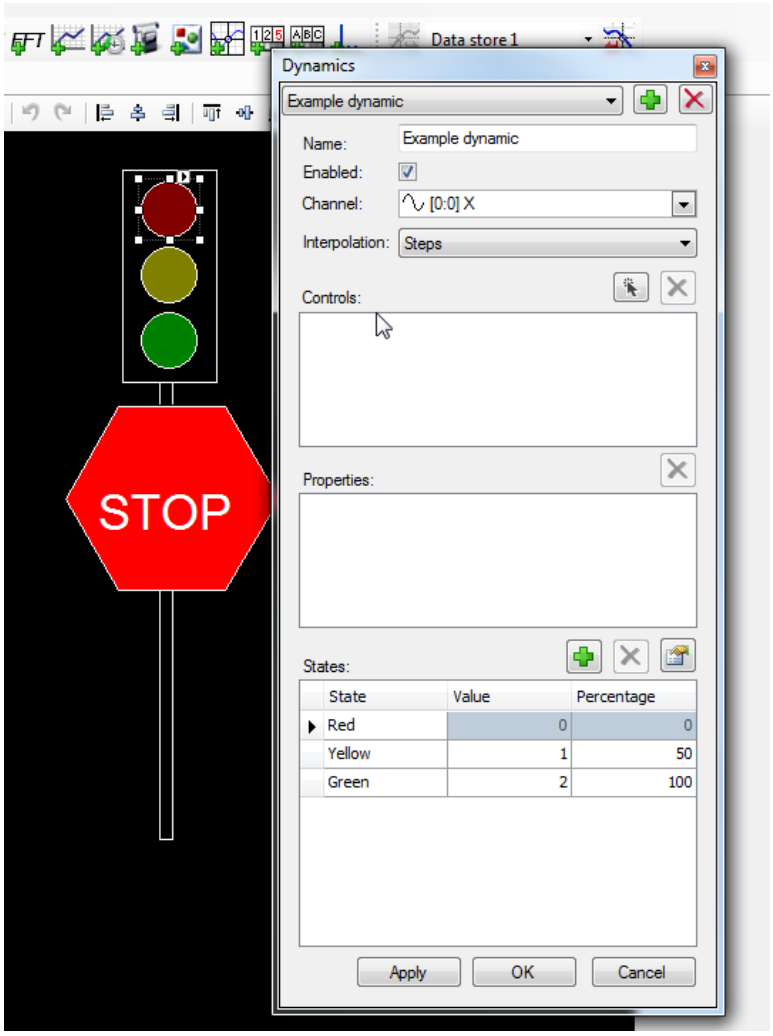


Figure x – The dynamic editor. Three initial states are created and their state values are configured.

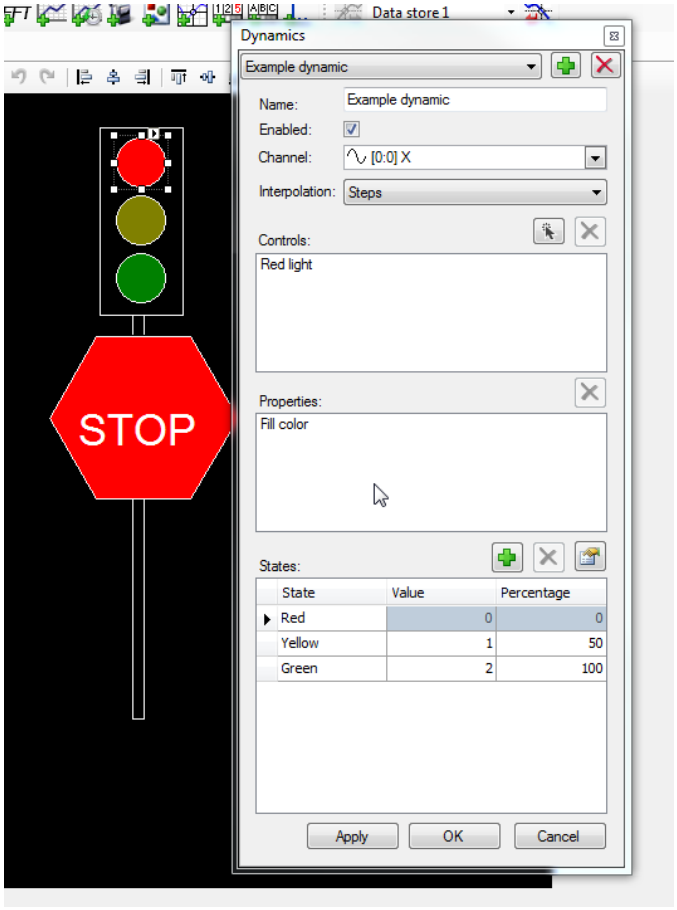


Figure x – After making the red light brighter the control and property is added to the dynamic.

When having multiple dynamics the drop down on the top of the top dynamics editor can be used to switch between different dynamics. To be able to distinguish dynamics a descriptive name be assigned to each dynamic. To delete a dynamic use the -button.

It is possible to enable/disable a dynamic.

4.2 Interpolation

The selection for interpolation determines how property values are set when the signal value is in between two defined states. Note that there is no extrapolation, if the signal value is outside the defined value range either the first or last frame is selected.The different interpolation modes are:

- Steps: No interpolation happens in between states. There is always one state selected. The selected state is the state with the highest value which is equal or lower than the signal value. The lower bound of a state range is the state's value, the higher bound is the value of the next state (exclusive).
- Linear: Property values are linearly interpolated if the signal value is in between two states. Not every property type can be interpolated (like fonts or drop down lists).
- Ease in: Interpolation is used, but it is non linear. The ease in-interpolation starts slow and ends fast. Easing can be used to animation look more natural.
- Ease out: Non linear interpolation which starts fast and ends slow

- Ease in and out: Non linear interpolation which starts and end slows and is fast in between.

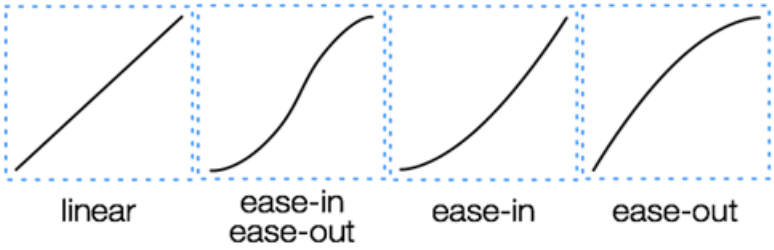


Figure x – the different interpolation modes

The  select controls button can be used to select the controls inside the Qpanel which belongs to the current dynamic. It is also possible to select a single control by clicking on its name in the control list.

In the state section of the property editor new states can be created and signal values can be assigned to states. To set property values for a frame, select a frame and change a property on the QPanel. When switching frames by selecting them in the frame section, the frame is applied to the QPanel when properties are modified on the QPanel the concering property and controls, they are automatically added to the dynamic.

Alternatively to edit frames directly on the Qpanel, the dynamic can also be defined by editing the property values directly in a table view, where the frames are layed out horizontally and the controls with the properties are layed out vertically. To access the property editor, click  .

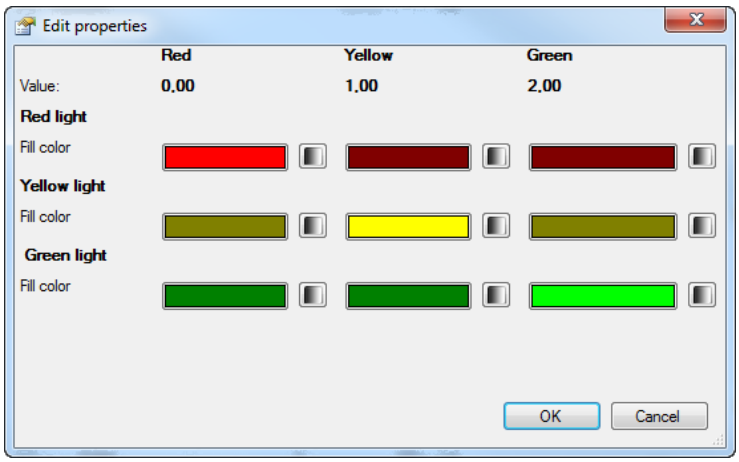


Figure x – The property editor of the traffic light dynamic.

4.3 Value line editor

Besides the property editor described above there is also the value line editor. The value line editor shows the defined states linearly on a line. This editor is more natural for a dynamic where interpolation is used, as it is possible to go in between states. The value line editor allows the user to simulate a signal value over time which gives insight in how a dynamic works. Note that the value line editor does not provide the exact same functionality as the dynamics editor. For instance changing the begin and end frames, or changing the signal channel is only possible with the dynamics editor.

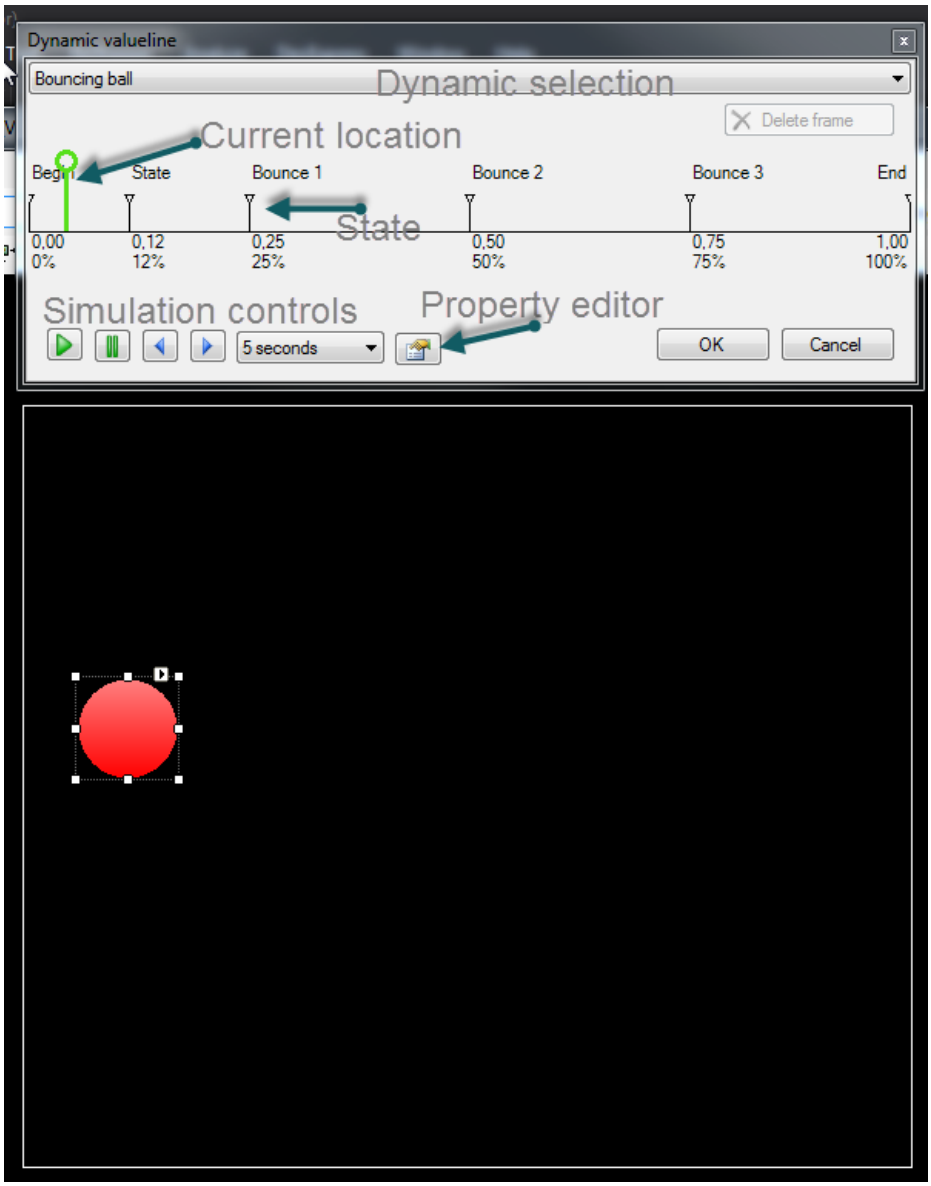


Figure x – The value line editor

The indicator allows the user to visit a signal value. The indicator can be moved by clicking anywhere on the value line. The state of the dynamic gets applied to the QPanel when the indicator is moved along the line.

The line turns blue if a state is selected or green when inbetween two states. If a state is selected the state can be edited directly by modifying the QPanel. If making changes while in between two states, a new state is inserted at the current selected position. States can be removed by selecting them and pressing the “Delete Frame” button. A state can be moved by dragging it over the line.

The simulations controls can be used to simulate a dynamic by evaluating the range of frame values over a configured time period. The blue arrow buttons can be used to go to a previous or next state.

In the property window of a control it is possible to view the dynamics in use by the selected control.

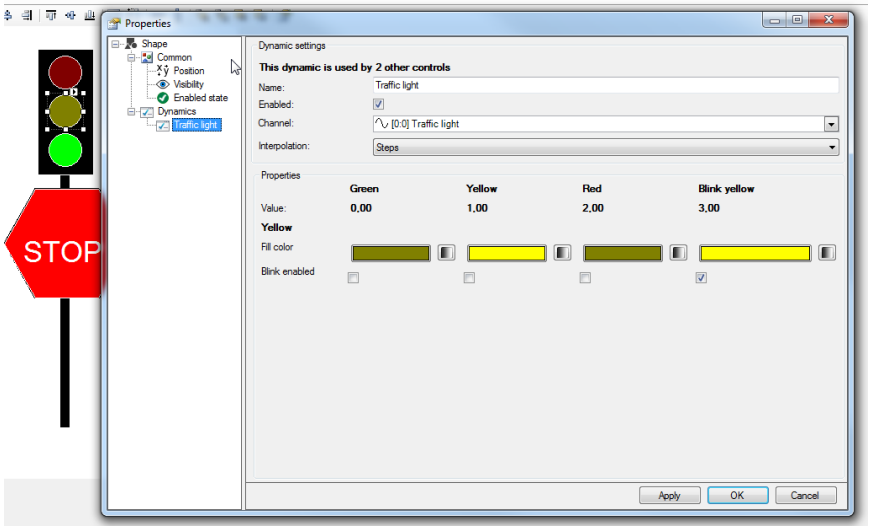


Figure x – Editing a dynamic using the property editor.

5 QPanel: Language dictionary

In 6.37 user texts for most controls can be supplied in multiple language. The text used is switched when the language of ibaPDA is changed.

Texts which can be translated have a language button at the right side of the text control on the property window.

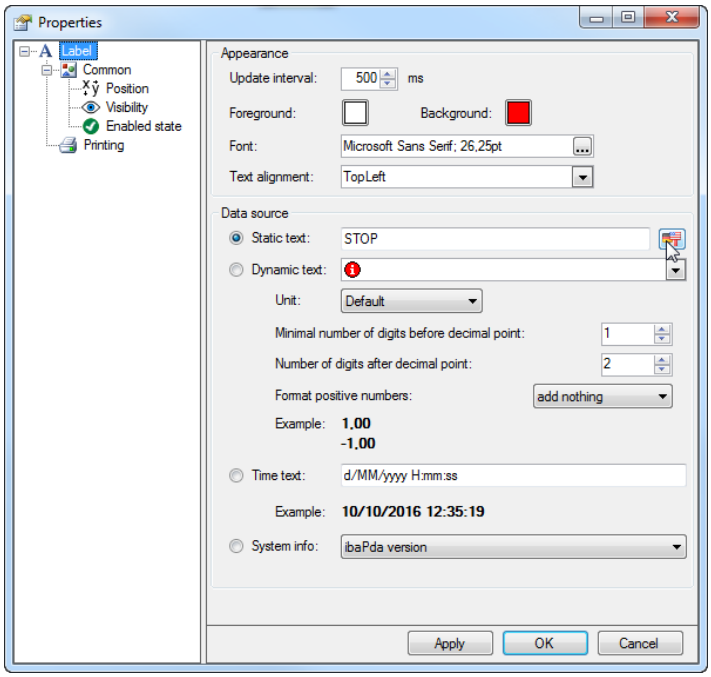


Figure x – A text property which is translatable in multiple languages.

A translatable text can be in two modes:

- *Static*: The text is not translated in other language, the same text is used no matter what language is chosen. 

- *Translated*: The text is a reusable entry from a dictionary. The text is supplied in multiple languages, in the input field the identifying key is displayed. This key can be used to reuse the same dictionary entry at different places. 

The translate-button  can be used to switch between the two modes and to edit the text and key in different languages. By pressing the button “Use static text” the mode is switched back to *static*.

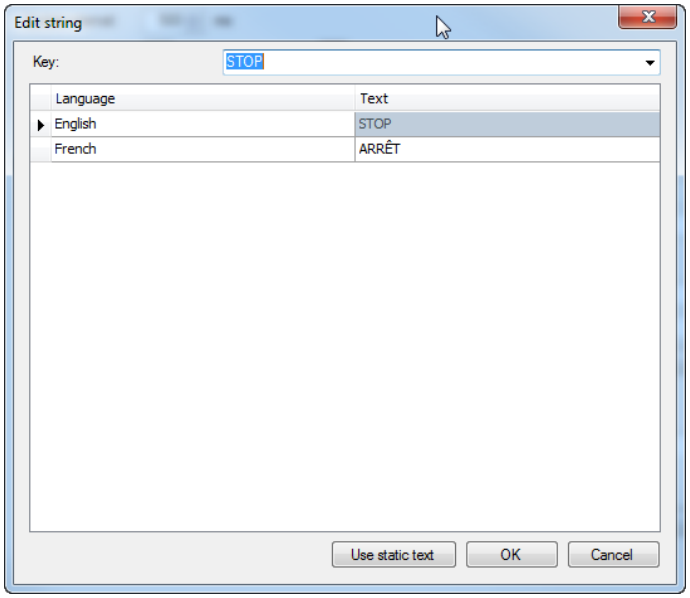


Figure x – Detail window of a translatable text.

When making a text translatable the system searches for existing entry in the dictionary to reuse. The currently selected language in ibaPDA is used to find such an existing entry. When adding translations to an element which has no translations yet, make sure ibaPDA is set to the language of the existing text. If reoccurring texts are present they automatically get matched to an existing dictionary entry.

The entire language dictionary can be edited by choosing the option “Language Dictionary” button from the View-menu.

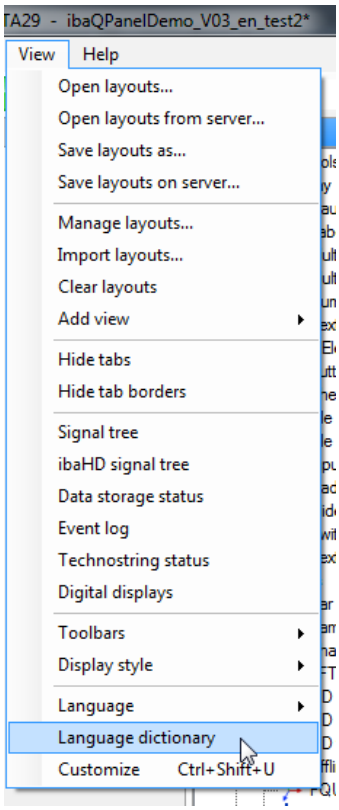


Figure x – Opening the language dictionary

In the language dictionary it is possible to edit/add/remove dictionary entries.

- A new language can be added by clicking the “Add language”-button.
- A language can be removed by clicking the “Remove language”-button. English is the default language and cannot be removed.
-  - Copies the entire dictionary to the clipboard. It can be pasted in for example in Excel.
-  - Adds entries to the dictionary from the clipboard.

6 QPanel: Grouping elements

In 6.37 it is possible to add a number of controls to a group. A group can be edited, copied/pasted or placed in a control library as if it is a single element. To create a group select two or more elements and select “Group” from the context menu or the toolbar or pressing control+G.

When selecting multiple elements and adding them to a control library, a group will automatically created.

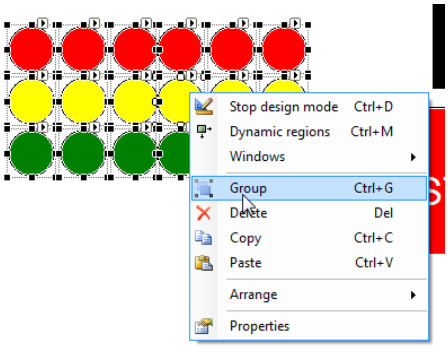


Figure x – Creating a group.

7 QPanel: New controls

New controls have been introduced.

7.1 Checkbox / switch

A checkbox can be bound to a writable signal. If the checked state is changed either a 0 or 1 is written to the configured signal. If the signal is changed from another source, the check state is updated to reflect the signal state. This means that if multiple checkboxes write to the same signal, their check states are synchronized.

The switch is a graphical variant of the checkbox. The switch has 4 states (checked/enabled, unchecked/enabled, checked/disabled, unchecked/disabled). The user can supply an image for every state. The default images used to do not have a distinguished image for the disabled states.

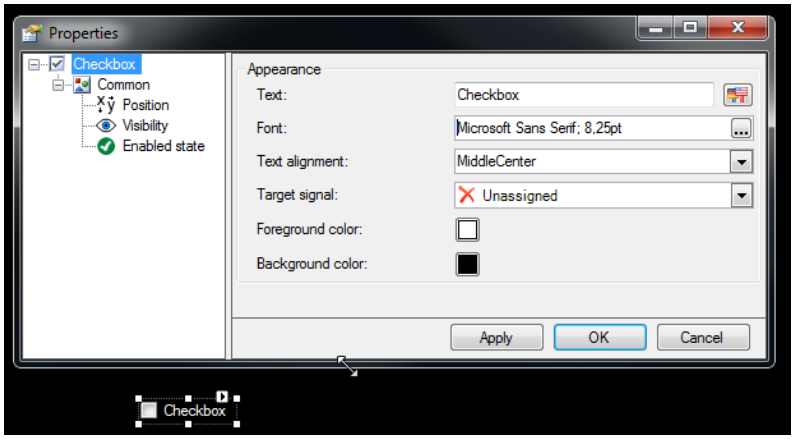


Figure x – The checkbox control

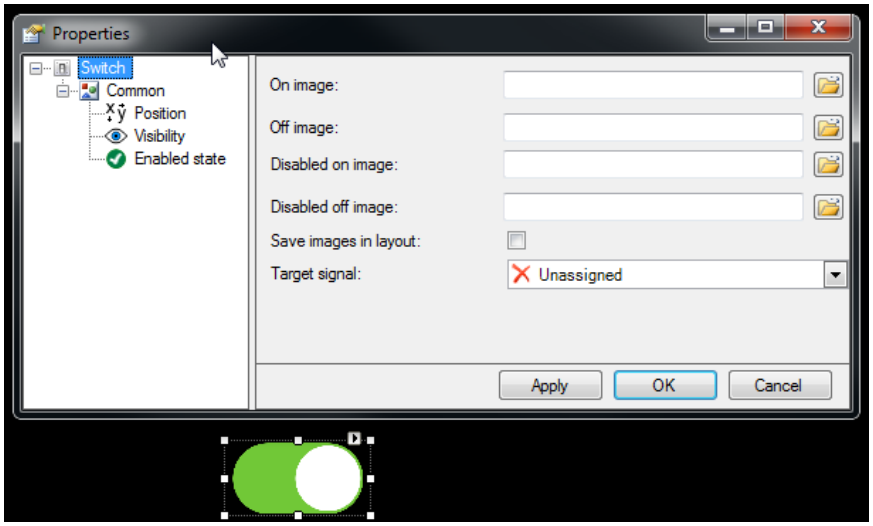


Figure x – The switch control

7.2 Radio button

Like the checkbox, the radio button can write a value to a configured. The user can configure the available options and set a value for each option. If the bound signal changes externally the checked option is changed to reflect the signal value.

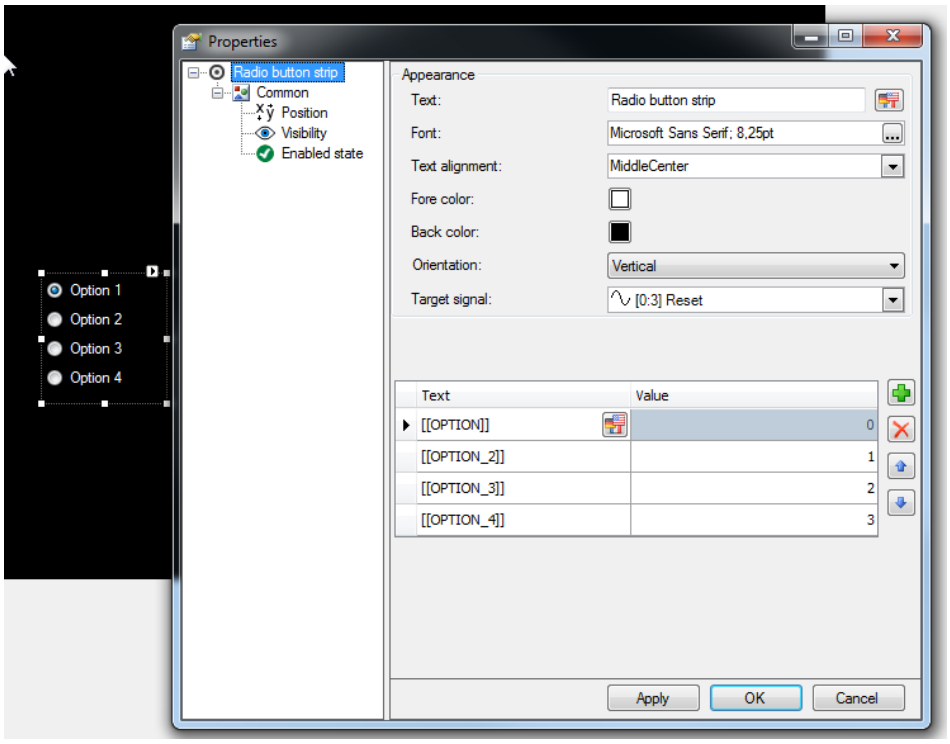


Figure x – The radio button

7.3 File picker

The file picker enables the user to locate a file on the file system by the “incremental search”-idiom. The selected file’s path is written to a configured technostring. The technostring can be used to load file in the offline trend graph or picture control or alternatively be displayed in a label.

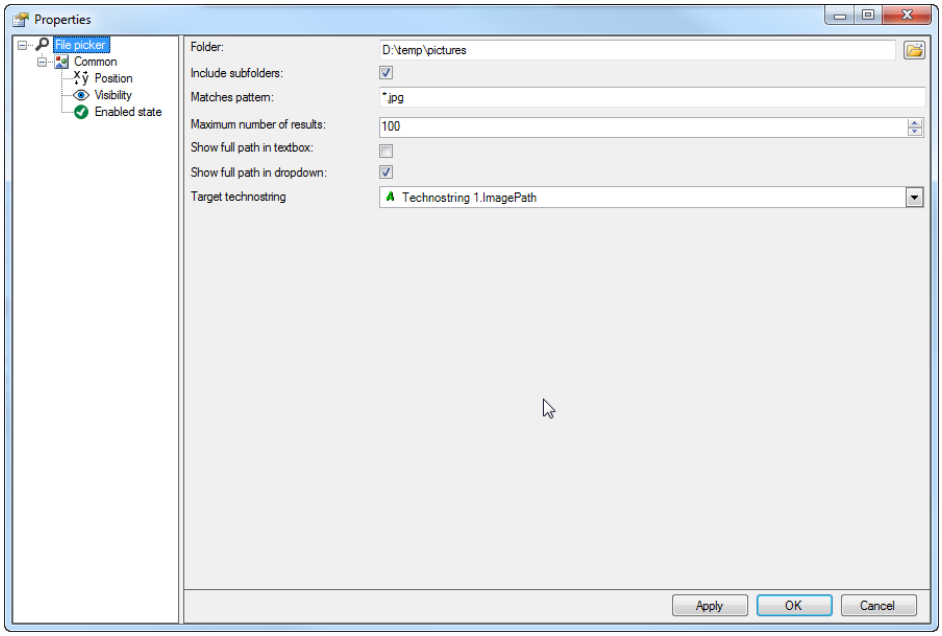


Figure x – The property window of the file picker

The user can configure the folder where to search for files and wether subfolders should be included in the serach. In addition the user can configure a file name pattern, which is used to filter the search result. The pattern can be used to limit the results to a certain file extension. The user can set a limit to the number of search result displayed. With “Show full path in

textbox” and “Show full path in dropdown” the user can customize how the search results are displayed.

7.4 File scanner

The file scanner is a control that can be placed on the QPanel which is not visible unless in design mode. The control can be used to watch the file system and in response to changes on the file system, write a path to a technostring.

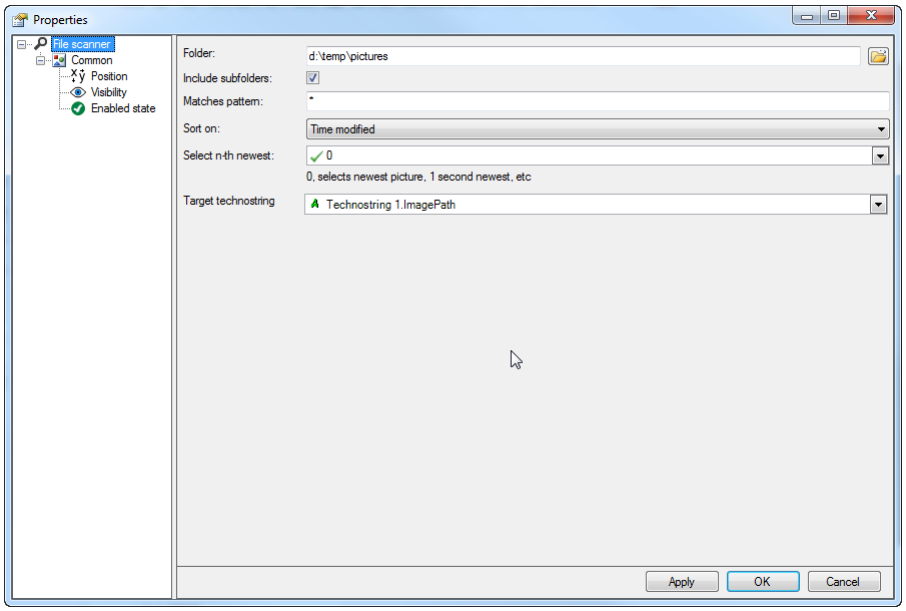


Figure x – The property window of the file scanner.

The file scanner watches the configured folder, and if configured also the subfolders. If a change occurs in the watched folders it looks for the newest or n-th newest file according to either the time modified or time created. The file selected is determined by the “Select n-th newest” property. If set to 0 the newest file gets selected, with 1 the second newest, etc.

7.5 Symbol

The symbol control is used to display vector images in scalable vector graphics (SVG). Basic transformations, like scaling and rotation can be applied to a symbol.

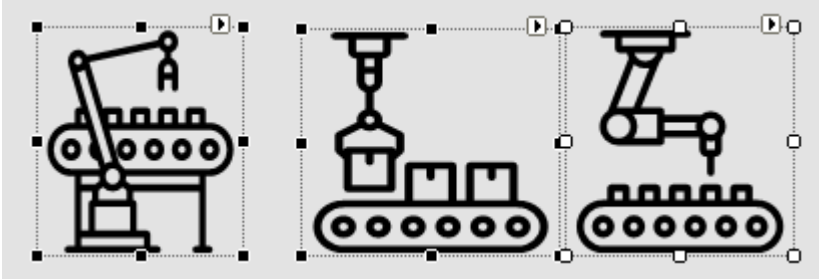


Figure x – Example symbols

In the property window the user can override a number of properties, like the fill color. In order to override a property the user has to check the “Override” checkbox next to property and define a new property value.

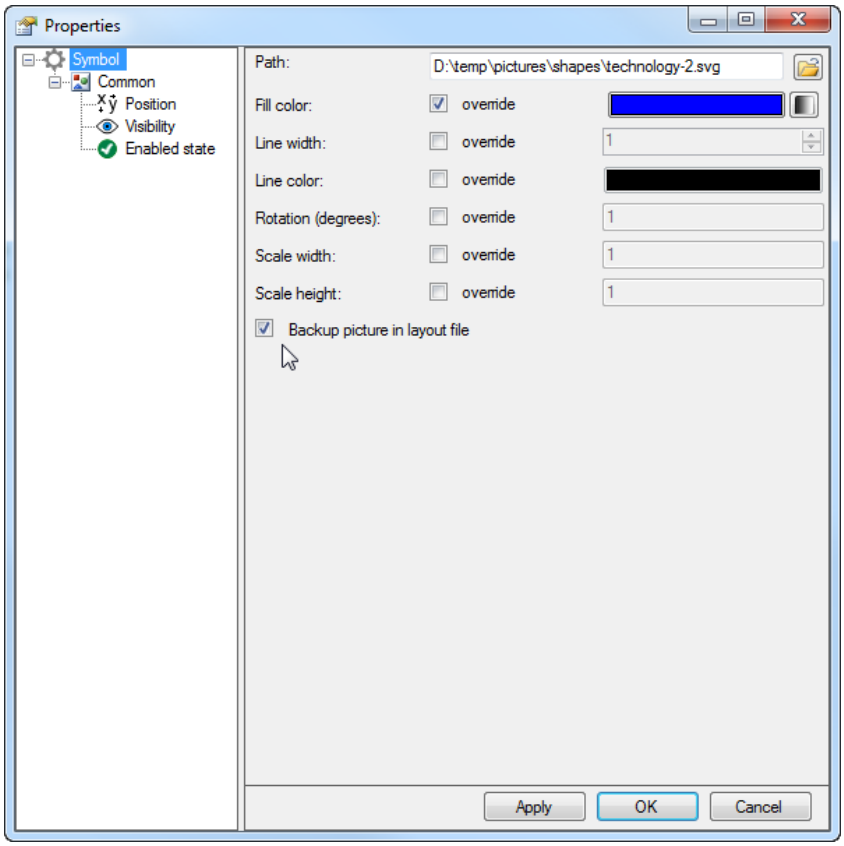


Figure x – The symbol's property window.

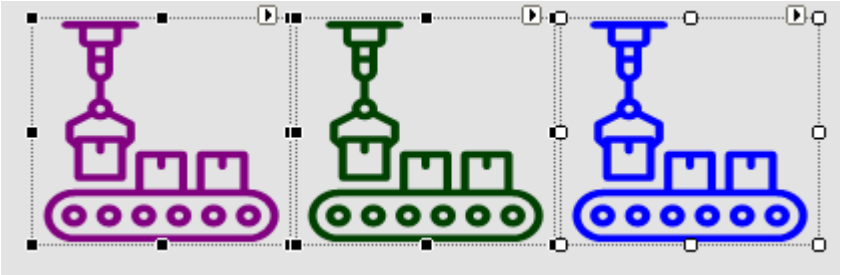


Figure x – A symbol with different overrides for the property "fill color".

8 Toshiba TC-net 100/1G hardware

The TC-net hardware interface for ibaPDA allows you to measure values from a TC-net through a Toshiba TC-net 100 or TC-net 1G PCI board. The following boards are supported:

Board type	PCI express	Network type
JTNI11/JTNI12		TC-net 100
JTNI21/JTNI22		TC-net 100
JTNI31/JTNI32		TC-net 100
JTNI41/JTNI42		TC-net 100
JTNI61/JNTI62	x	TC-net 100
JTGI23	x	TC-net 1G

Table 1 - Supported Toshiba TC-net PCI boards

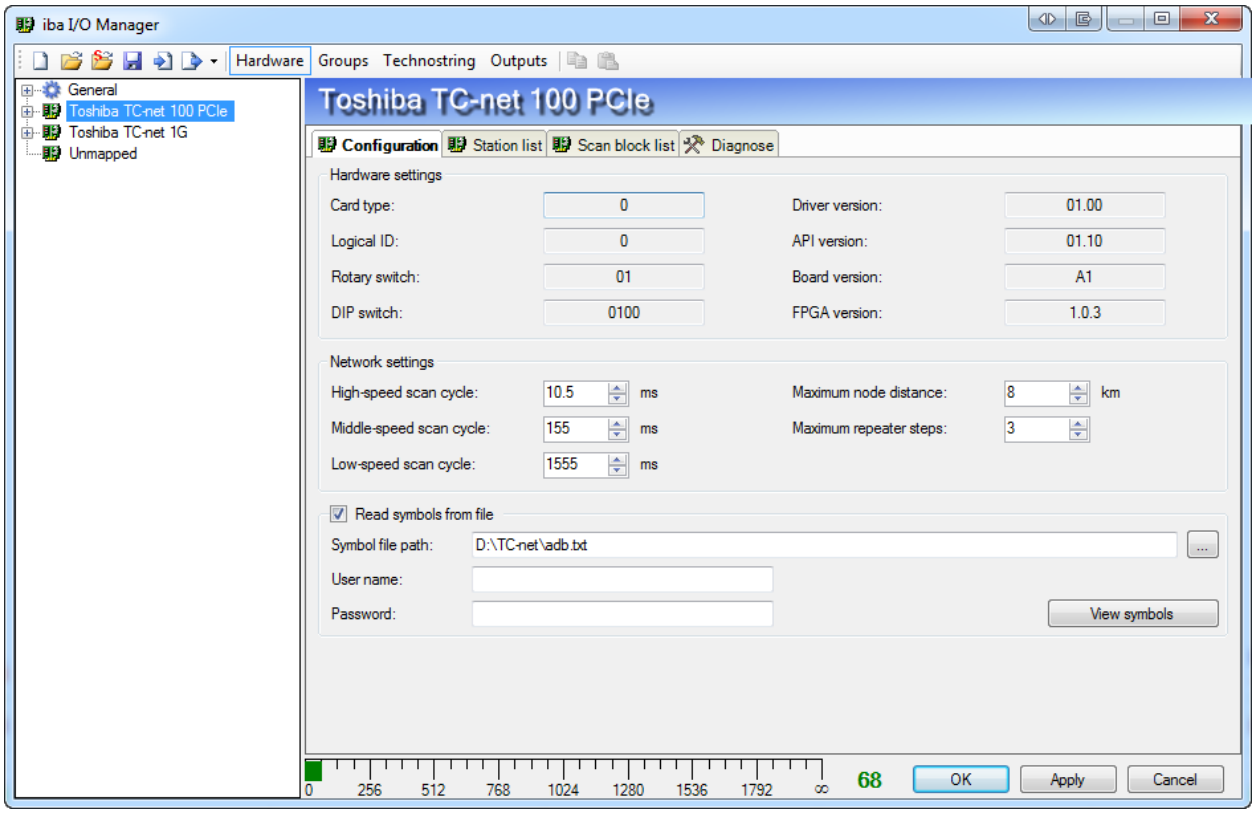
To benefit from faster transfer rates provided by DMA (only in the case of PCI express boards), ibaPDA does not directly read data from the TC-net board via the PCI bus but makes use of the official Toshiba API which supports DMA transfers. For the API to function properly, the official Toshiba device driver supplied with the hardware needs to be installed. ibaPDA does not install a device driver for TC-net boards.

As can be seen from Table 1, there are three categories of hardware: TC-net 100 PCI boards, TC-net 100 PCI express boards and TC-net 1G PCI express boards. For each of these three categories, the Toshiba-developed API is supported by ibaPDA.

Due to limitations of the Toshiba APIs, it is only possible to support two different boards for each API type. This implies that the maximum amount of TC-net boards in the ibaPDA server machine is 6: 2 TC-net 100 PCI boards, 2 TC-net 100 PCI express boards and 2 TC-net 1G boards.

Note that the Hardware TC-net license needs to be activated on your ibaPDA dongle.

When opening the I/O Manager in ibaPDA, the detected TC-net boards will be automatically listed in the tree structure.



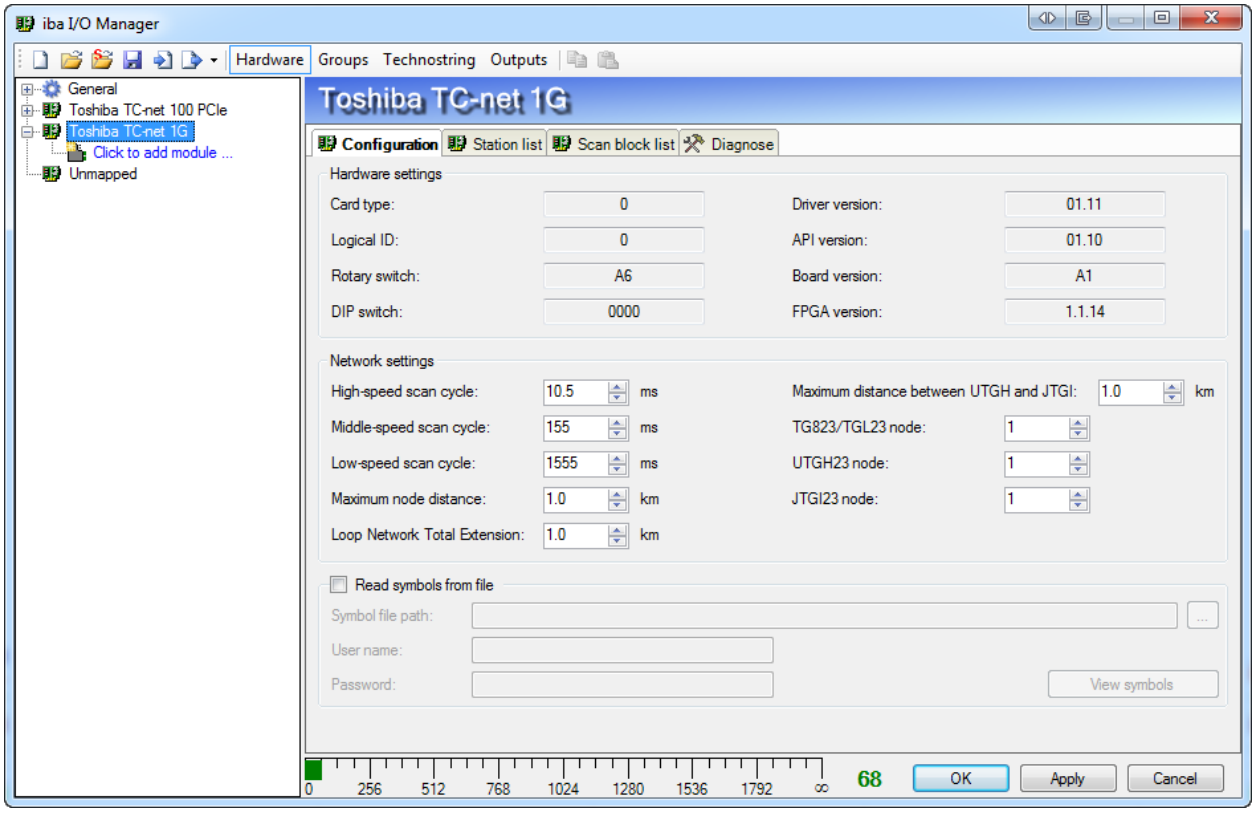
When selecting the interface node of a TC-net 100 board, the settings shown in the figure above will appear. The **Hardware settings** group contains diagnostic information on the TC-net board:

- ***Card type***: Toshiba-specific card type
- ***Logical ID***: used internally to distinguish between multiple TC-net boards from the same API type (TC-net 100, TC-net 100 PCI express or TC-net 1G). This is either 0 or 1.
- ***Rotary switch***: value of the two hex switches on the PCI board to identify the board in the TC-net.
- ***DIP switch***: setting of the DIP switch on the TC-net board (only available for PCI express)
- ***Driver version***: version of the Toshiba device driver
- ***API version***: version of the Toshiba API that interfaces between the Toshiba driver and ibaPDA
- ***Board version***: version of the hardware
- ***FPGA version***: version of the FPGA firmware on the TC-net board (only available for PCI express

Below that is the **Network settings** group which contains parameters that need to be properly set in order to enter the TC-net. More information on these settings can be found in the Toshiba documentation.

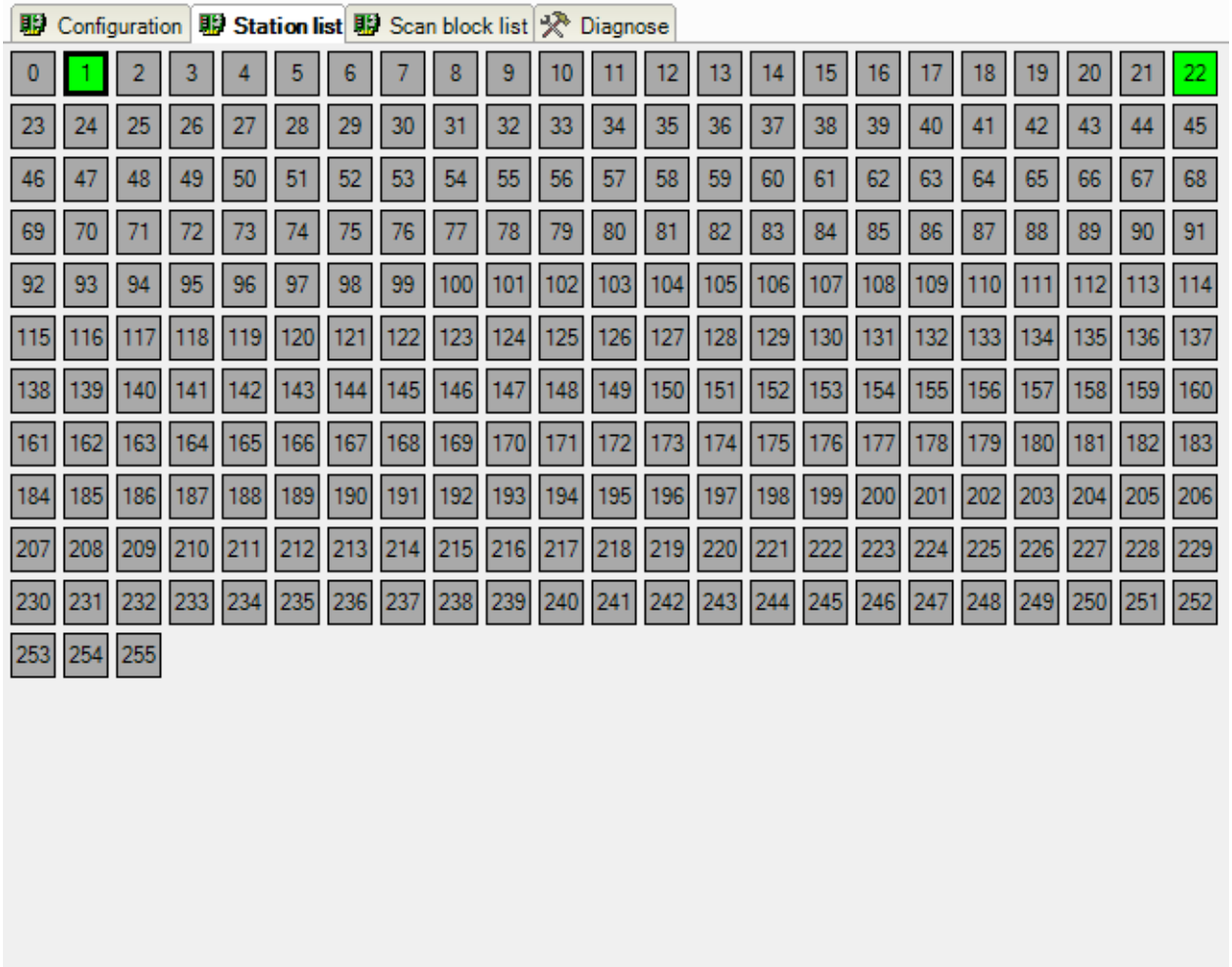
Finally, there's a possibility to link an address book to the TC-net board by enabling the checkbox **Read symbols from file**:

- ***Symbol file path***: the full path to an address book file. For more information on how to generate an address book, we refer to the Toshiba documentation
- ***User name***: the user name that should be used when trying to access an address book file over a network
- ***Password***: the password that should be used when trying to access an address book file over a network



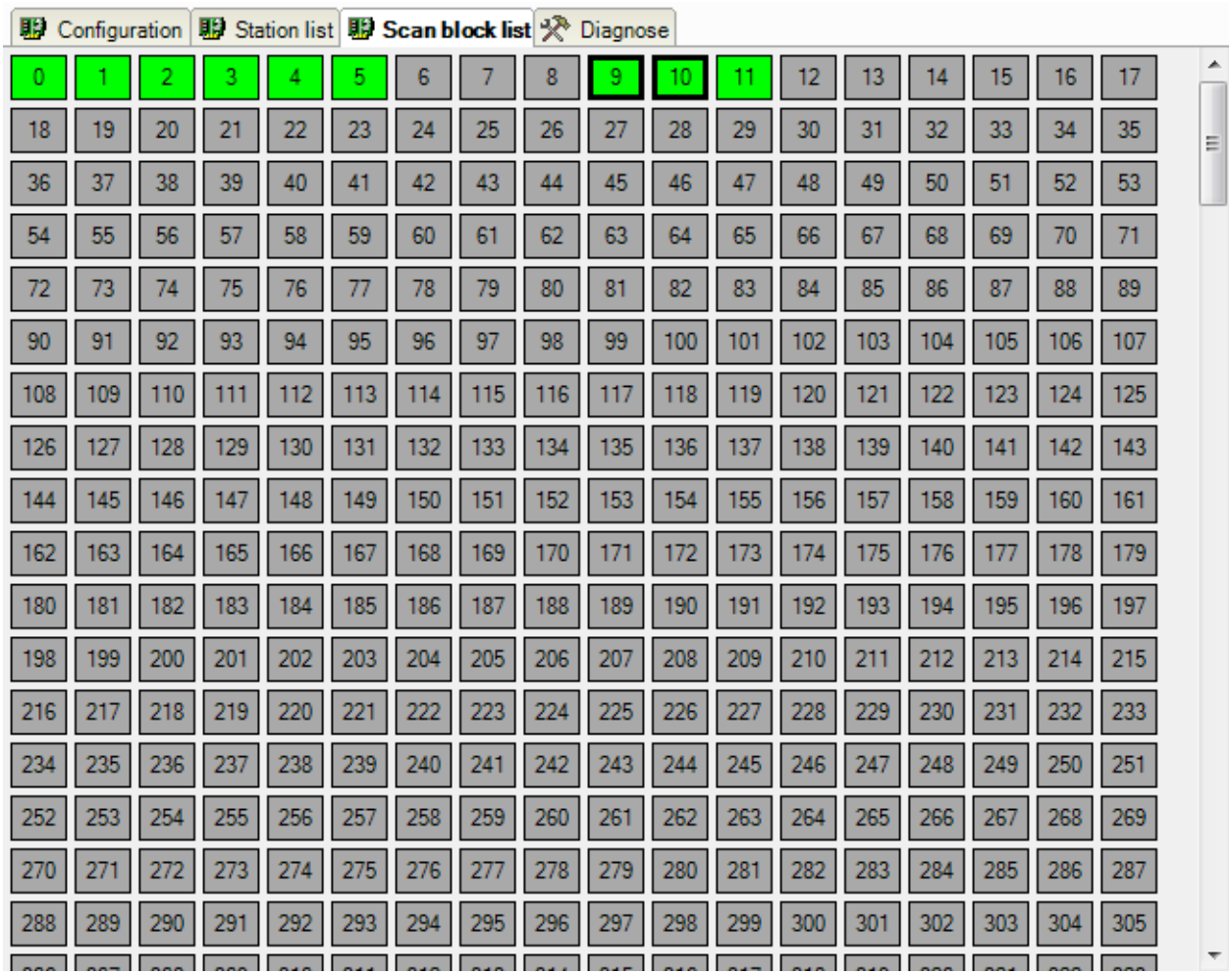
The interface node for a TC-net 1G board looks very similar to that of a TC-net 100 board with the exception of the **Network settings** group, for which we refer to the Toshiba documentation.

The **Station list** tab shows which of the 256 stations in the TC-net are active.



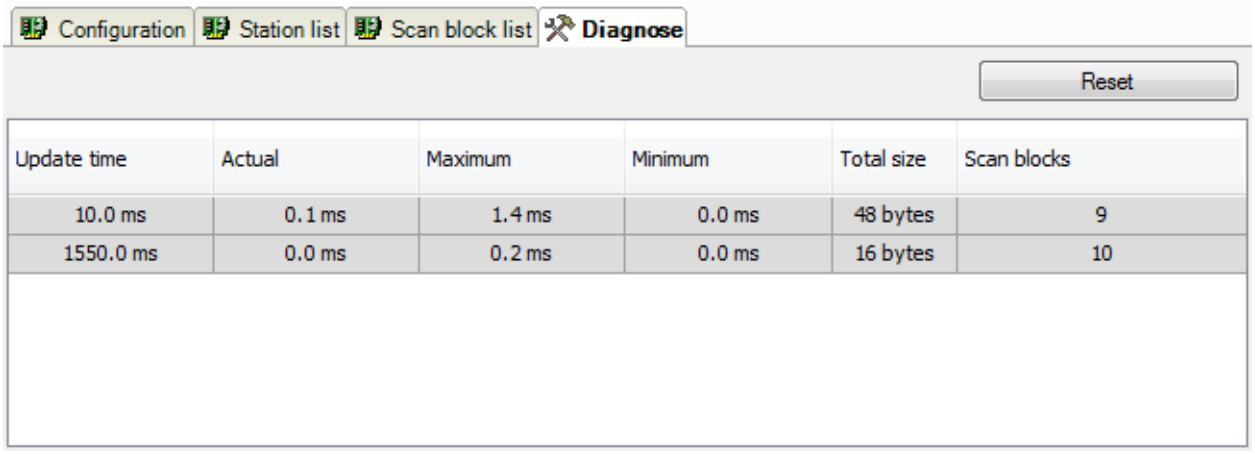
Active stations in the ring have a green background color while inactive stations have a gray background color. The station node used by the ibaPDA server has a bold border around it.

The **Scan block list** displays all scan blocks in the TC-net.



Active scan blocks in the ring have a green background color. In case they are linked to a TC-net module in ibaPDA, the border will be bold. For inactive scan blocks that are not linked to a TC-net module, the background color will be gray; for inactive scan blocks that are linked to a TC-net module, the background color will be red.

The **Diagnose** tab shows some basic information about how fast data is retrieved from the TC-net board.



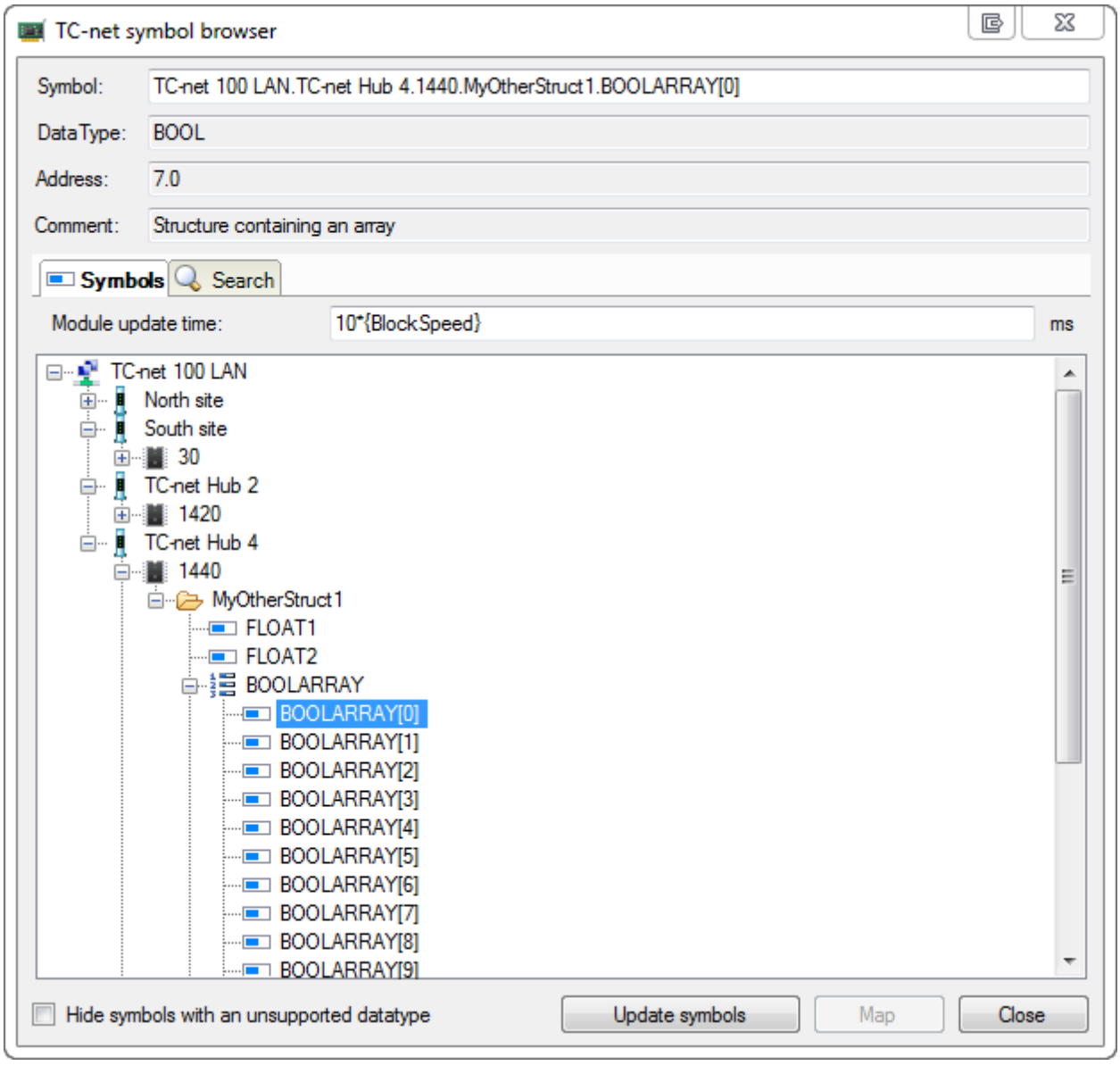
For each different update time value, an entry is shown with the following fields:

- ***Update time:*** the time interval between two read operations from the TC-net board
- ***Actual:*** the time it took to transfer all data for this update time from the TC-net board to ibaPDA memory
- ***Maximum:*** the maximum amount of time it took to transfer all data for this update time from the TC-net board to ibaPDA memory. This value can be reset by clicking the **Reset** button.
- ***Minimum:*** the minimum amount of time it took to transfer all data for this update time from the TC-net board to ibaPDA memory. This value can be reset by clicking the **Reset** button.
- ***Total size:*** the total amount of bytes that need to be transferred for every update time.
- ***Scan blocks:*** scan block Ids linked to TC-net modules with this update time.

The ***Maximum*** and ***Minimum*** values can be reset by clicking the **Reset** button.

8.1 TC-net symbol browser

When **Read symbols from file** has been checked and when clicking **View symbols** in the **Configuration** tab, the TC-net symbol browser will appear.



In the **Symbols** tab, a tree structure will appear containing all symbols parsed from the TC-net address book file. The top-level node is the TC-net which contains a number of station nodes which each in turn contain a number of scan blocks. These scan blocks then contain the actual variables that need to be measured.

When selecting a variable, the **DataType**, the **Address** (on a 16-bit WORD basis; e.g. 4.10 refers to the 10th bit of the 4th WORD or 9.3 on an 8-bit BYTE basis) and **Comment** as supplied by the address book file will be displayed.

Structures are supported, provided that the definition file (typically named [StructureName].typ) is located in the same directory as the address book file.

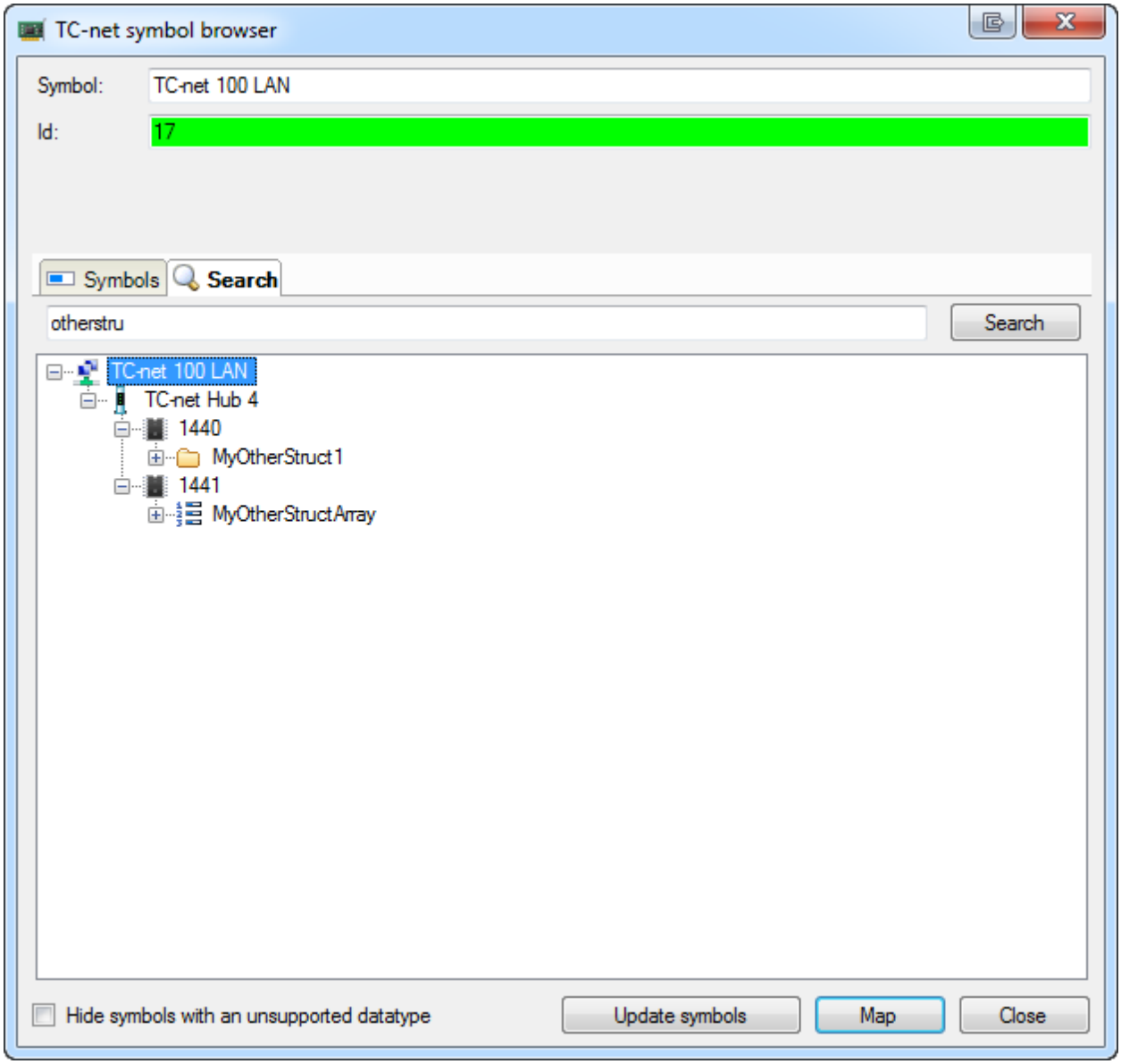
When selecting a scan block, the scan block **Speed** will be displayed. The value of this can be either ***Low***, ***Middle*** or ***High***.

For TC-net and station nodes, the **Id** is displayed.

By checking the **Hide symbols with an unsupported datatype** option, variables with a datatype that could not be parsed properly will not be displayed. Variables with an unsupported datatype are marked with the icon while supported variables are marked with the icon.

In case no symbols appear in the tree structure in the **Symbols** tab, click **Update symbols**. If still no symbols appear, there's probably something wrong with the format of the supplied address book file.

Instead of manually looking for a certain variable, it is possible to perform a search.



To search for variables, click the **Search** tab, enter part of the complete variable name you are looking for and click **Search**. The symbol browser will only list variables that match the supplied pattern. Note that the complete variable name contains the network name, station name and scan block Id as well. For example: if we were to enter ***TC-net*** in the search box and click **Search** the entire symbol tree would be shown since every variable starts its complete name with TC-net 100 LAN in this case.

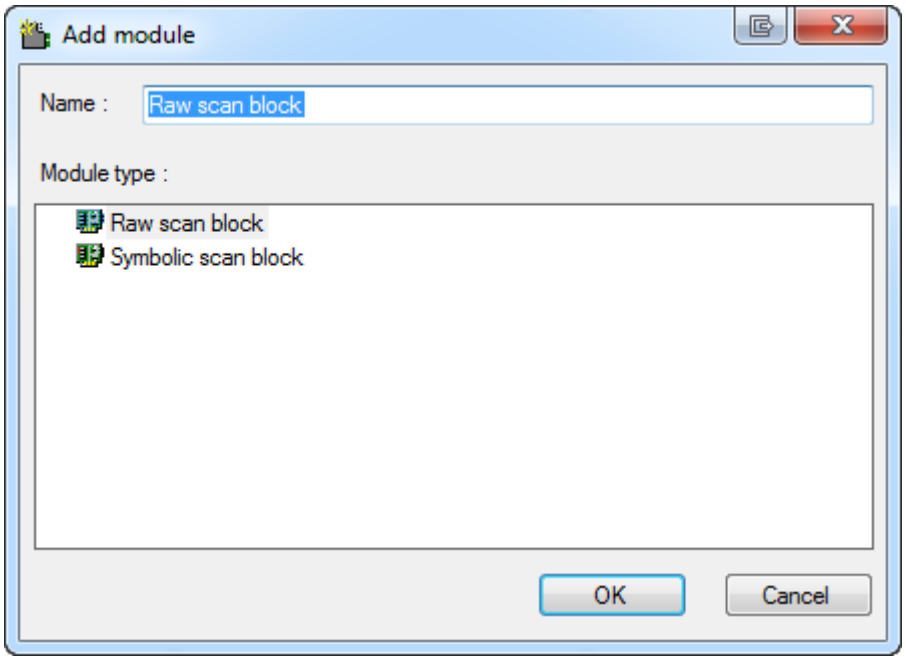
Going back to the **Symbols** tab, it is possible to automatically generate ibaPDA modules based on the address book. As we will see in Sections 8.2 and 8.3, a TC-net module is always one-to-one linked to a certain scan block.

When selecting a scan block in the symbol browser, the **Map** button will become enabled. When clicking the button, a new TC-net module will be generated containing all variables in that scan block. In case a module already exists linked to the selected scan block, the variables that are not yet present in the existing module are added to the module.

Apart from being able to select multiple scan blocks at once, it is also possible to select TC-net station or TC-net nodes. When mapping a TC-net station node, all scan blocks linked to the station node will be added; when mapping a TC-net node, all scan blocks linked to all stations in the TC-net will be added.

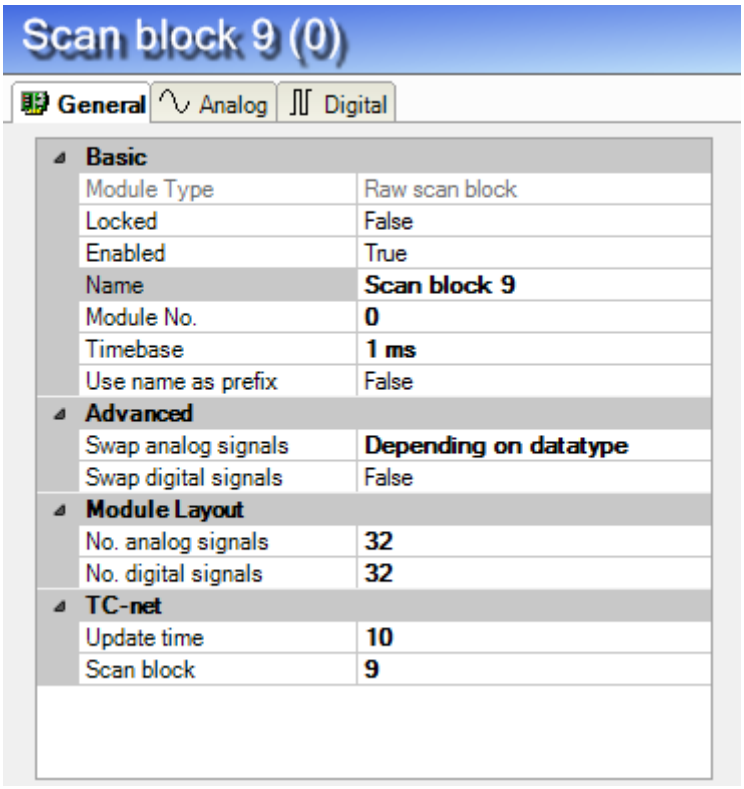
When mapping scan blocks, the update time provided in ***Module update time*** will be applied to all newly created modules. For more information on how to define the ***Module update time***, see Sections 8.2 and 8.3.

When expanding the interface node in the I/O Manager tree structure, new TC-net modules can be added. When clicking **Click to add module...** the following dialog will appear.



8.2 TC-net Raw scan block

When using this module type, no address book information is used and the addresses and datatypes of the requested variables need to be entered manually.



Apart from the standard module properties, the following extra settings are available:

- ***Swap analog signals:*** allows byte-swapping of signals with multi-byte datatypes. The following modes are available:

MODE	16 BIT	32 BIT
No swap	AB	ABCD
Depending on datatype	BA	DCBA
Swap 16 bit	AB	CDAB
Swap 8 bit	BA	BADC

- ***Swap digital signals:*** digital signals are interpreted as 16-bit WORDs. When this property is enabled, the WORDs will be byte-swapped.
- ***Update time:*** the time interval (ms) between two read operations from the TC-net board for this module. Either a constant value can be supplied or an expression containing one of the following placeholders:

PLACEHOLDER	MEANING
{HiScan}	High-speed scan cycle time defined for this board (in ms)
{MidScan}	Middle-speed scan cycle time defined for this board (in ms)
{LowScan}	Low-speed scan cycle time defined for this board (in ms)

For example: suppose we have a scan block that transmits data using the middle-speed scan cycle time. Instead of setting the update time to exactly that value we might want to

set it just a bit lower to make sure we never skip a cycle. To do this, we could set the ***Update time*** to ***0.9*{MidScan}***.

- ***Scan block:*** the scan block Id associated with this module

When selecting the **Analog** tab, we get the following signal grid:

General | Analog | Digital

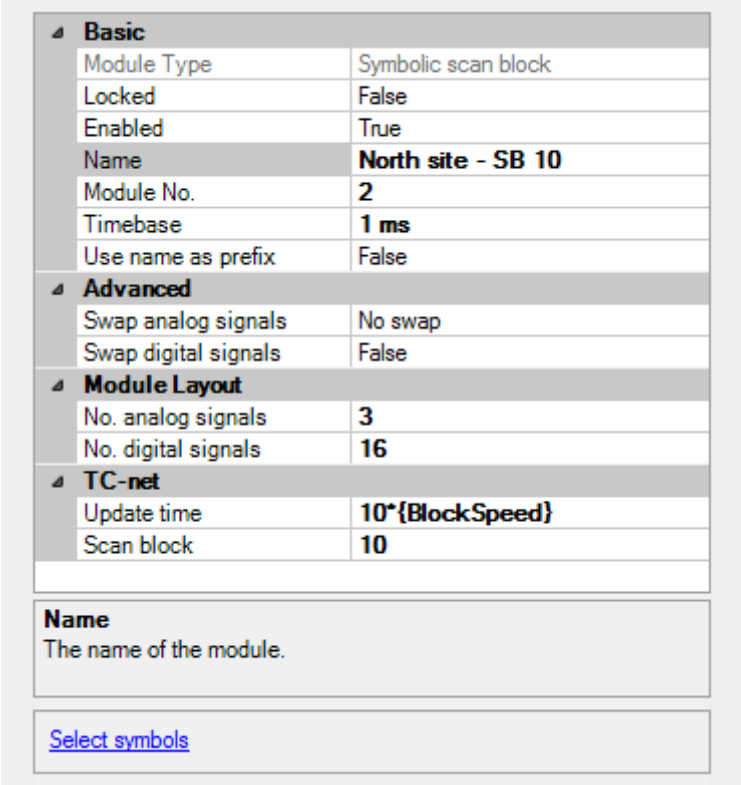
	Name	Unit	Gain	Offset	Address	DataType	Active	Actual
0			1	0	1	BYTE	☑	2
1			1	0	2	BYTE	☐	9
2			1	0	3	BYTE	☐	4
3			1	0	4	BYTE	☑	9
4			1	0	5	BYTE	☑	6
5			1	0	6	DWORD	☑	151521546
6			1	0	7	BYTE	☐	8
7			1	0	8	BYTE	☐	9
8			1	0	9	BYTE	☑	10
9			1	0	10	BYTE	☑	9
10			1	0	11	BYTE	☑	12
11			1	0	12	WORD	☑	2318
12			1	0	13	BYTE	☐	14
13			1	0	14	BYTE	☐	9
14			1	0	15	BYTE	☐	16
15			1	0	109	BYTE	☑	110
16			1	0	110	BYTE	☑	9
17			1	0	111	BYTE	☑	112
18			1	0	112	BYTE	☑	9
19			1	0	113	BYTE	☑	114
20			1	0	127	BYTE	☑	128
21			1	0	81	BYTE	☑	82
22			1	0	82	BYTE	☑	9
23			1	0	83	BYTE	☑	84
24			1	0	84	WORD	☑	2390
25			1	0	85	BYTE	☑	86
26			1	0	86	BYTE	☑	9
27			1	0	87	BYTE	☑	88
28			1	0	88	DWORD	☑	156895580
29			1	0	89	BYTE	☑	90
30			1	0	90	BYTE	☑	9
31			1	0	91	DWORD	☑	1544117769

Here we have to define for each signal we want to measure the **Address** within the scan block and the correct **DataType**. Note that a signal may not exceed the 128-byte boundary (i.e. the total size of a scan block).

A similar grid is shown for the **Digital** signals.

8.3 TC-net Symbolic scan block

This module type can only be used when a valid address book has been supplied.



Apart from the standard module properties, the following extra settings are available:

- ***Swap analog signals:*** allows byte-swapping of signals with multi-byte datatypes. The following modes are available:

MODE	16 BIT	32 BIT
No swap	AB	ABCD
Depending on datatype	BA	DCBA
Swap 16 bit	AB	CDAB
Swap 8 bit	BA	BADC

- ***Swap digital signals:*** digital signals are interpreted as 16-bit WORDs. When this property is enabled, the WORDs will be byte-swapped.
- ***Update time:*** the time interval (ms) between two read operations from the TC-net board for this module. Either a constant value can be supplied or an expression containing one of the following placeholders:

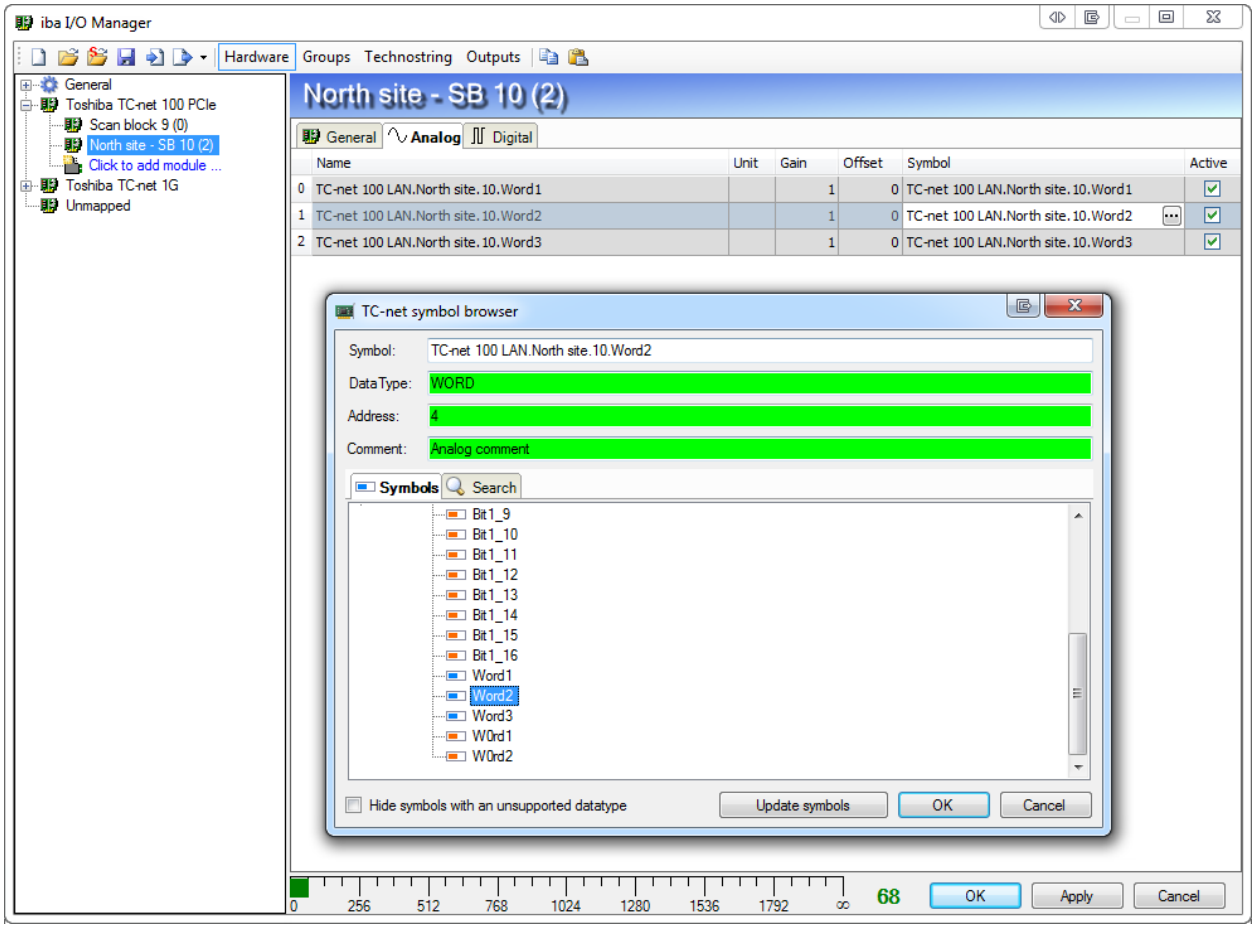
PLACEHOLDER	MEANING
{HiScan}	High-speed scan cycle time defined for this board (in ms)
{MidScan}	Middle-speed scan cycle time defined for this board (in ms)
{LowScan}	Low-speed scan cycle time defined for this board (in ms)
{BlockSpeed}	Speed property of scan block as defined in address book

For example: suppose we want to map a number of scan blocks using the symbol browser on the TC-net interface node that operate at different speeds. The provided update time expression in the symbol browser is applied to all scan blocks. However, we would like an update time of ***0.9*{MidScan}*** for middle-speed scan blocks and an update time of ***0.9*{HiScan}*** for high-speed scan blocks. To do this, we could set the ***Update time*** to ***0.9*{BlockSpeed}*** for every module.

- ***Scan block:*** the scan block Id associated with this module

At the bottom of the properties grid, there is a link **Select symbols** which opens the TC-net symbol browser where only variables contained within the scan block associated with this module are shown.

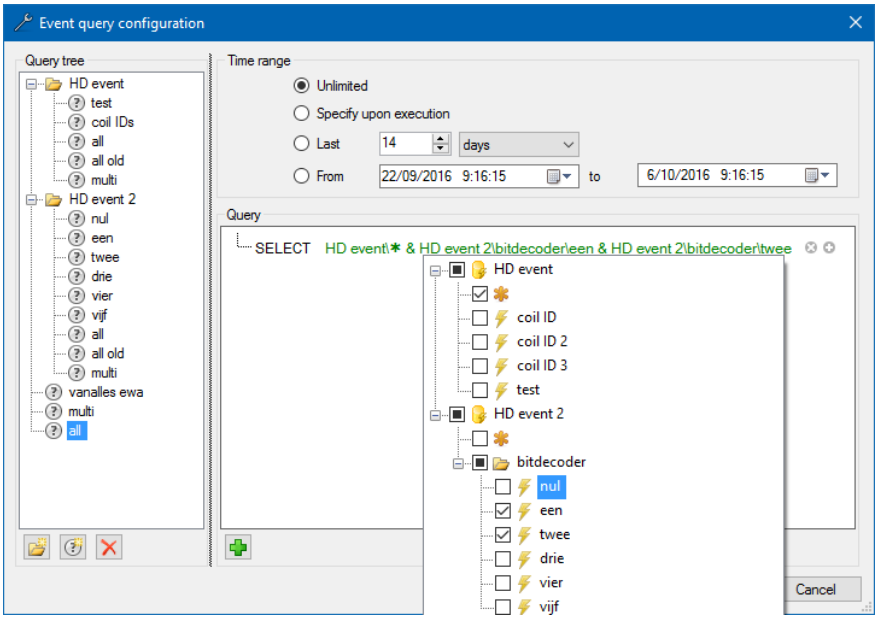
The **Analog** tab looks as follows:



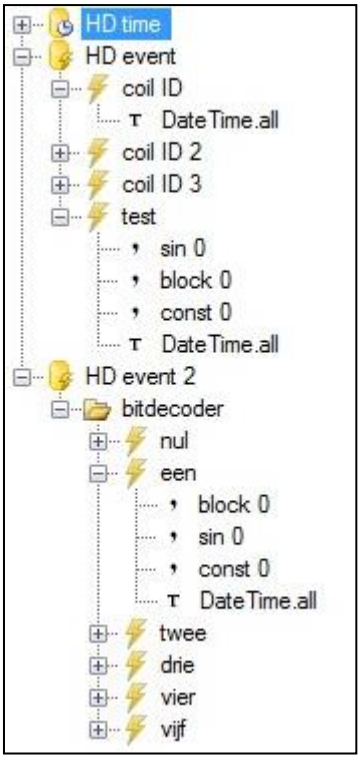
Instead of having to provide an address and datatype for each signal, only the symbolic variable name is required here. The symbol name can either be entered manually or selected using the TC-net symbol browser. When entering the **Symbol** column of a signal, a button will appear at the right side of the cell. Clicking this button will open the TC-net symbol browser where the currently selected symbol will be selected if available in the address book.

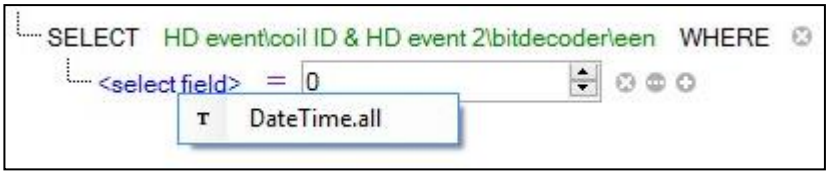
9 HD event table query

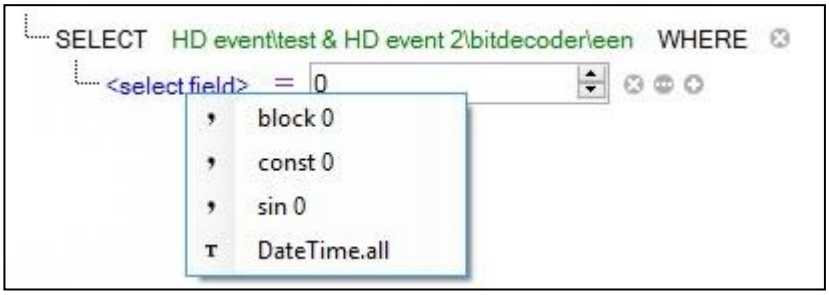
Using the HD event table it is now possible to query multiple events using a single select statement. The event tree popup in the event query configuration dialog displays checkboxes. An asterisk node is added for each store.



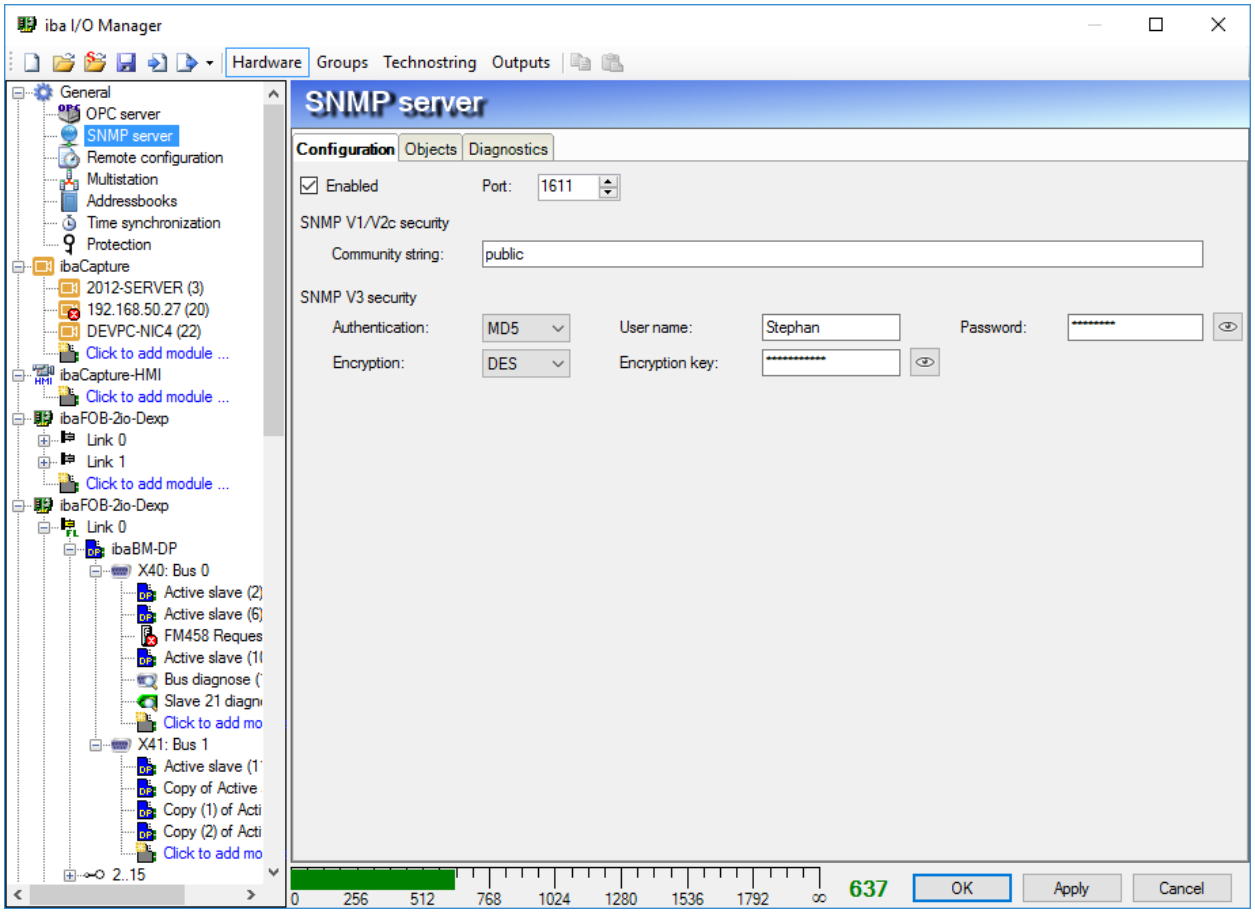
When multiple events (or an asterisk node) are checked, the expressions can only contain fields that are defined in all of the checked events. The event field popup will only display these.



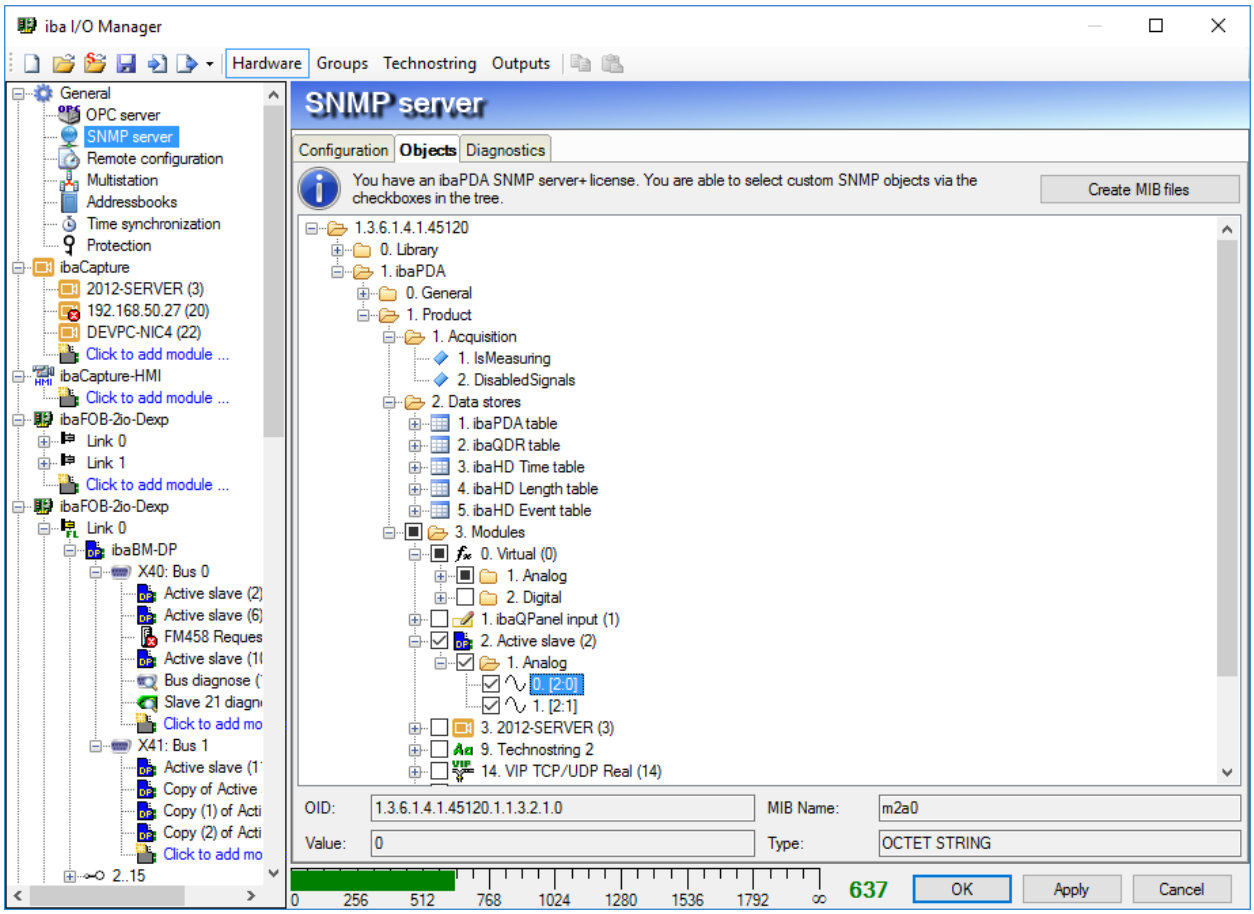




10 SNMP Server



IbaPDA is now an SNMP server. In the I/O manager you can enable the SNMP server. You can configure the port where the SNMP server is available on. IbaPDA supports the SNMP protocols V1, V2 and V3. For V1 and V2 you can only provide a community string that acts as a password. The default is public so that the SNMP server is accessible by any SNMP client. For V3 the security settings are a bit more elaborate. You can configure authentication and encryption.



On the *Objects* tab you can configure which data is available on the SNMP server. The tree shows the OID (object identification) tree. There are some SNMP objects that are always available. These objects don't have a checkbox. They are:

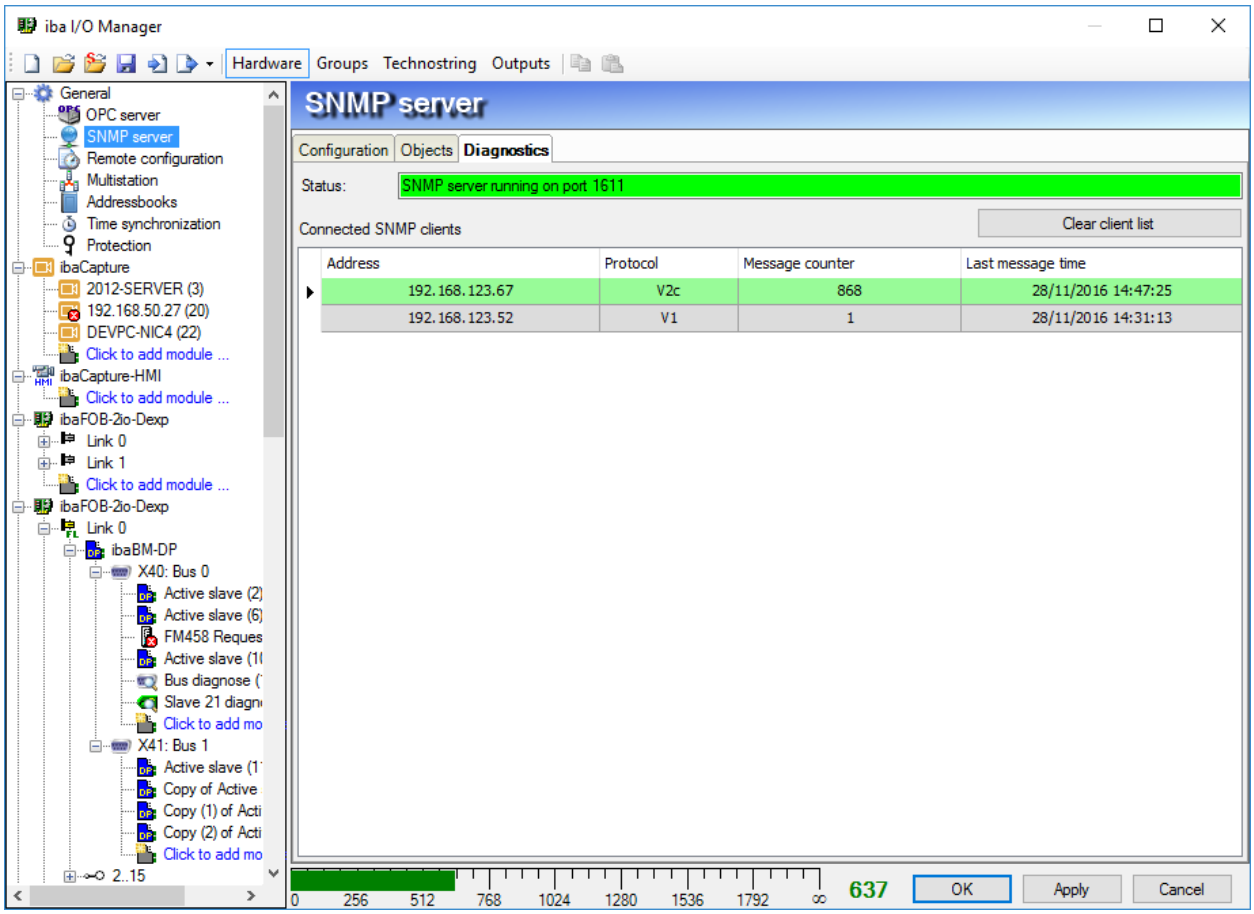
- Objects about the SNMP library
- Objects about ibaPDA general properties like the version and dongle information
- Objects about the acquisition like an IsMeasuring object
- Objects about the different data stores. These are grouped per data store type.

Next to these always available SNMP objects you can also configure custom SNMP objects. For the moment these are limited to signal objects. By checking the checkbox in front of a signal you make the signal available as an SNMP object. These custom SNMP objects are only allowed when you have an ibaPDA-Interface-SNMP-Server+ license. A message above the OID tree shows if you have a license or not.

On the top there is also a button to create the corresponding MIB files. The MIB files are generated by the ibaPDA server. So if you make any changes then you first have to apply these before the MIB files will contain the changes. There are 2 MIB files generated:

- IBA-GENERAL-MIB.txt: This contains the objects that are common to all iba products.
- IBA-PRODUCT-IBAPDA-MIB.txt: This contains the ibaPDA specific objects.

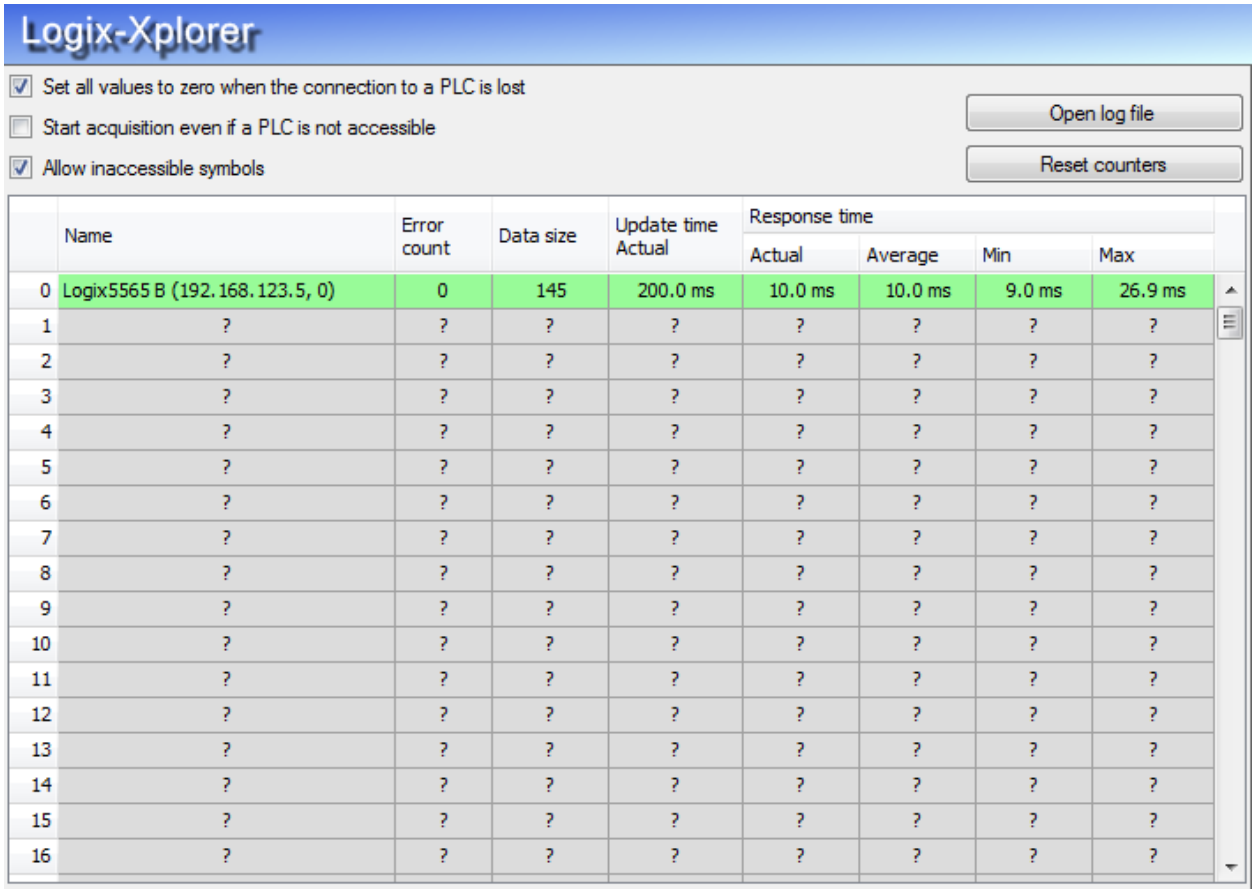
Below the OID tree you see the full OID and the MIB name of the selected node. You can also see the current value and the data type of the selected node.



On the *Diagnostics* tab you can see the current status of the SNMP server. It also shows a list of connected SNMP clients. Only the clients that have accessed the SNMP server within the last hour are shown. The clients that have accessed the SNMP server within the last 10 minutes are shown in green and the others in gray. Via the "*Clear client list*" button you can remove all the clients from the list.

11 Logix-Xplorer

The Logix-Xplorer interface in ibaPDA is used to measure data from Logix PLCs. It is an Xplorer interface which means that the data is cyclically read by ibaPDA instead of being sent by the PLC.



	Name	Error count	Data size	Update time Actual	Response time			
					Actual	Average	Min	Max
0	Logix5565 B (192.168.123.5, 0)	0	145	200.0 ms	10.0 ms	10.0 ms	9.0 ms	26.9 ms
1	?	?	?	?	?	?	?	?
2	?	?	?	?	?	?	?	?
3	?	?	?	?	?	?	?	?
4	?	?	?	?	?	?	?	?
5	?	?	?	?	?	?	?	?
6	?	?	?	?	?	?	?	?
7	?	?	?	?	?	?	?	?
8	?	?	?	?	?	?	?	?
9	?	?	?	?	?	?	?	?
10	?	?	?	?	?	?	?	?
11	?	?	?	?	?	?	?	?
12	?	?	?	?	?	?	?	?
13	?	?	?	?	?	?	?	?
14	?	?	?	?	?	?	?	?
15	?	?	?	?	?	?	?	?
16	?	?	?	?	?	?	?	?

The Logix-Xplorer interface shows a table of the available connections. Per Logix-Xplorer license you get 16 connections. A maximum of 240 connections is allowed. This means that maximum 15 licenses can be used. Each connection corresponds with a row in the table. The row is green when the connection is ok and data is being read. The row is orange when the connection is ok but the data is coming in slower than the configured update time. The row is red when the connection could not be established. The row is grey when there is no connection configured. The response time is the time it takes to read the data for a connection. The table shows the actual, average, minimum and maximum of the response time. The update time is the time between 2 reads. The data size shows how much data is read per read. You can use the “Reset counters” button to clear the counters for all connections.

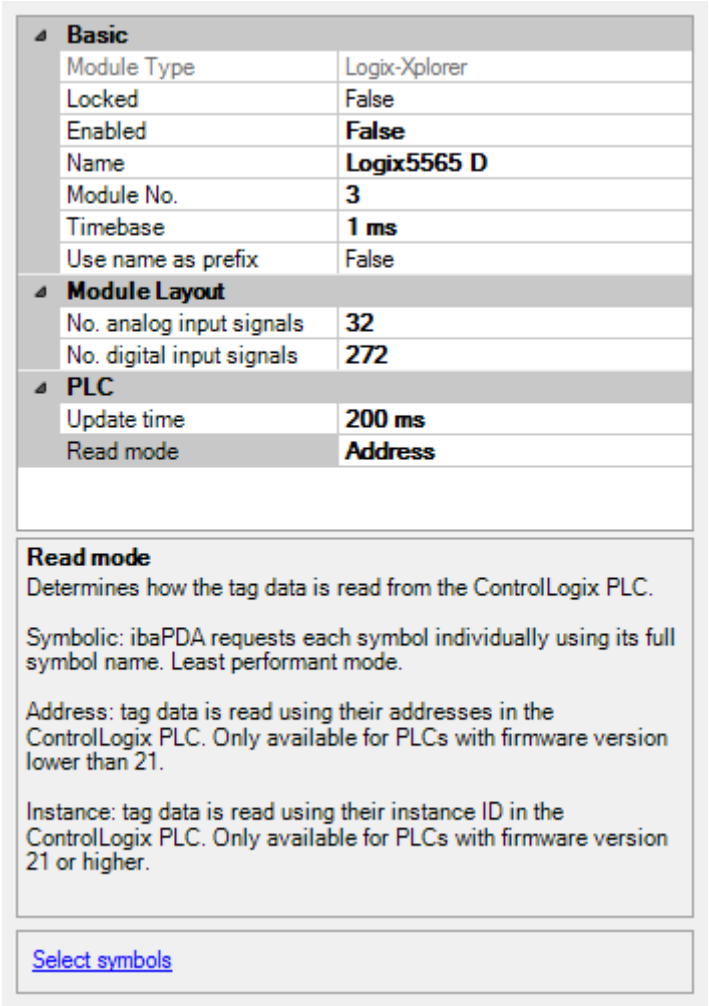
On the interface you can also decide how to handle some error conditions:

- When the connection to a PLC is lost during the acquisition then you can choose if the values stay at the last read value or if they are set to zero.
- When a PLC is not accessible during the start of the acquisition then you can choose if the acquisition starts without this PLC or if the acquisition is not started. When the acquisition is started without the PLC then ibaPDA will periodically (every 10s) try to connect to the PLC during the acquisition. As long as the PLC is disconnected the values will remain at zero.

- When ibaPDA tries to get the address of a symbol that is no longer available in the PLC then the PLC will return an error. If the option “Allow inaccessible symbols” is enabled then ibaPDA will ignore this signal and start the acquisition without this signal. If the option is not enabled then the acquisition will not start.

When ibaPDA detects a change in the configuration of the PLC, it will automatically restart the acquisition.

The General tab of a Logix-Xplorer module looks as follows:



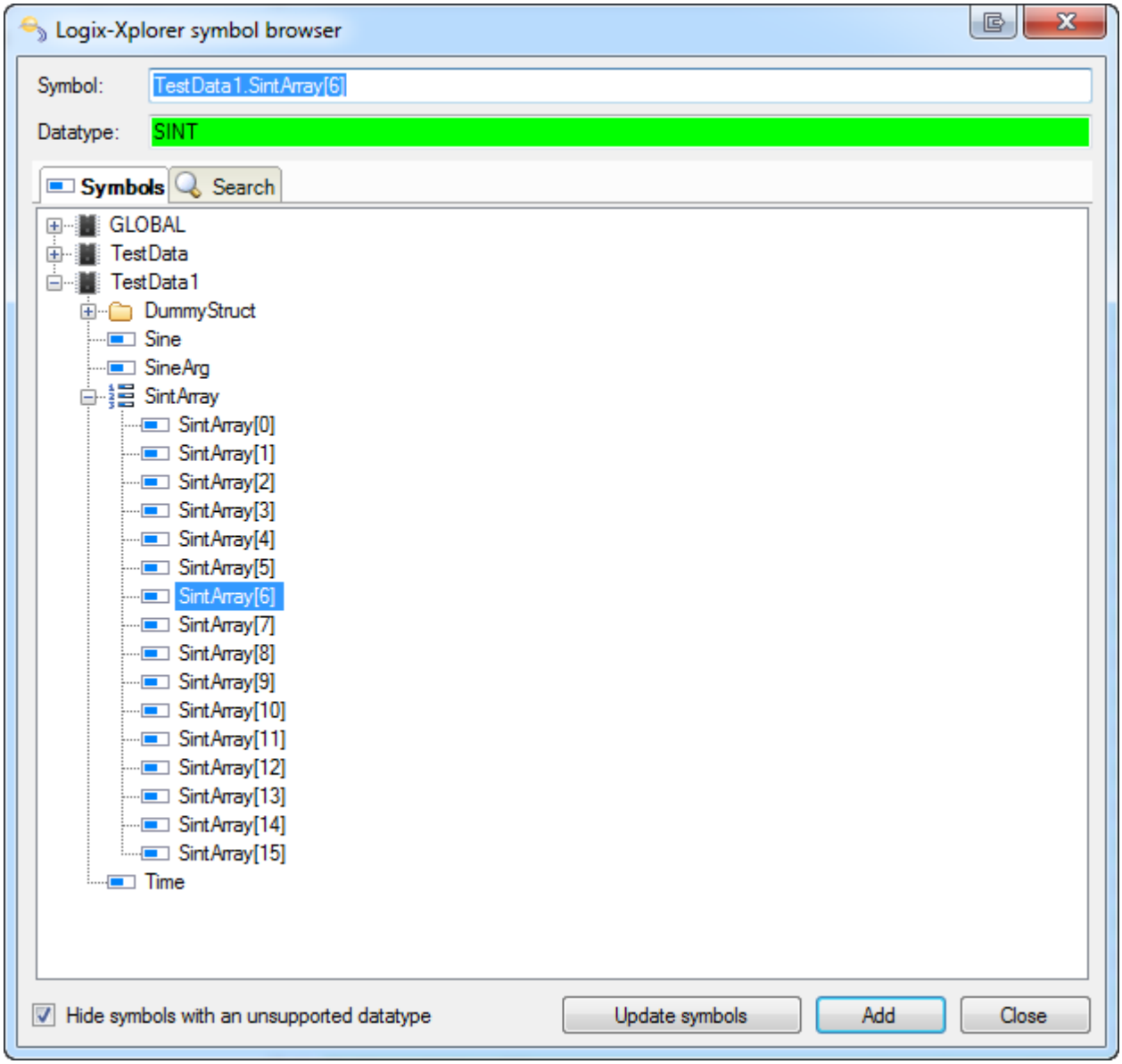
Apart from the standard options that can be found on most other modules, there are some specific settings:

- **Read mode**: there are several ways to request data from the Logix PLC. The most general though least performant method is **Symbolic** where the symbols are addressed using their full symbolic path for each read cycle. Since the PLC needs to resolve this name into a memory address for each symbol and for each read request, this will create a lot of overhead. ibaPDA will automatically try to optimize this by trying to group symbols (e.g. by requesting an entire structure instead of all its separate members). For PLCs with a firmware version below 21, the **Address** mode can be used: here, the addresses of all tags are retrieved during address book generation. This is the most performant method since the PLC no longer needs to resolve the symbol names for every read operation. Since this method is not officially documented by Rockwell, it might not function properly on certain types of PLC. For PLCs with firmware version 21

or higher, the alternative to **Symbolic** mode is **Instance** mode: instead of addressing symbols using their full symbolic path, their instance ID is used which is retrieved during address book generation.

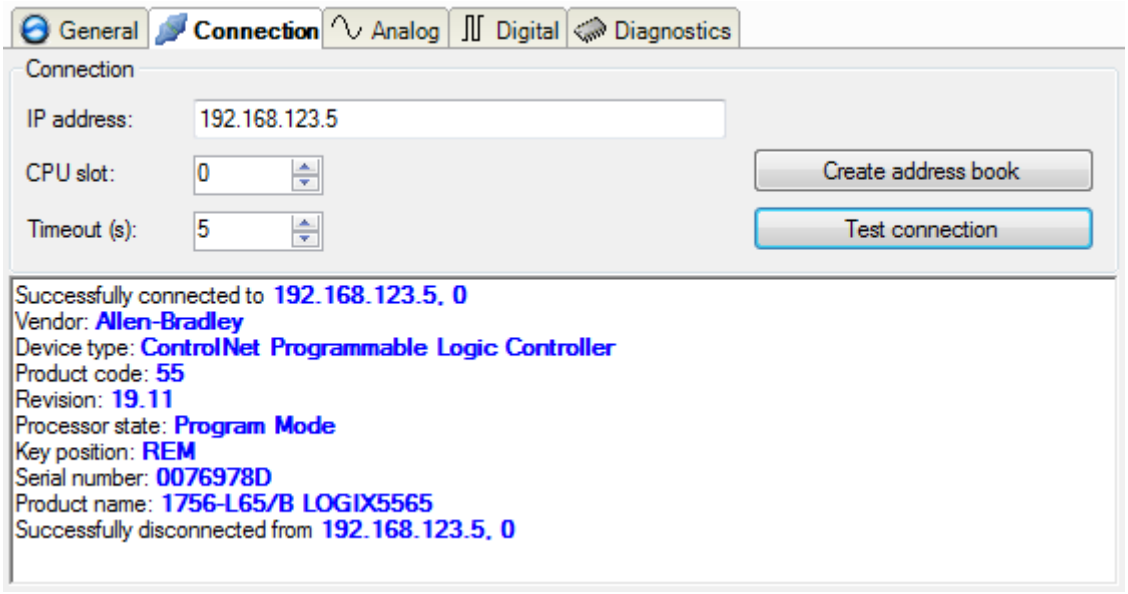
- **Update time**: the time in ms between two read operations.

At the bottom, the symbol browser can be opened by clicking **Select symbols**.



The symbol browser shows all the symbols that were loaded from the PLC. You can select single or multiple symbols in the tree. Click the *Add* button to add them to the corresponding analog or digital signal grid. If you selected a single symbol then the next symbol will be selected after you clicked the *Add* button. This allows you to hit *Add* multiple times in order to add consecutive symbols. You can also double click a symbol to add it to the signal grid. Use the *Update symbols* button to read the symbols again from the PLC.

On the search tab you can search symbols by name. The search result tree works in the same way as the complete symbol tree.



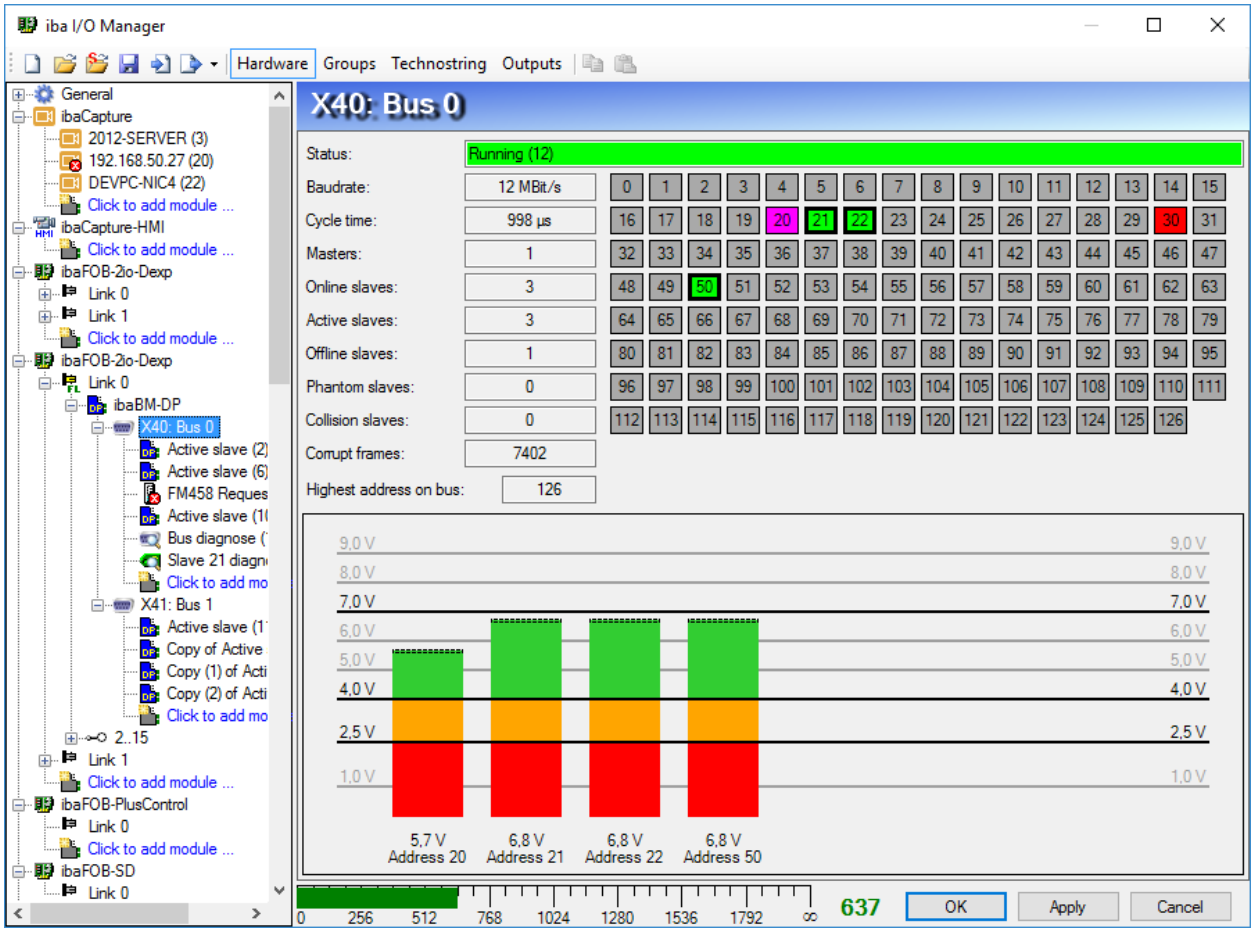
In the Connection tab, the IP address and CPU slot of the PLC can be configured. There is also a possibility to define a connection timeout (in seconds). By clicking the **Test connection** button ibaPDA will retrieve some basic information from the PLC and display it in the text box. Clicking **Create address book** will read the list of tags from the PLC and store them in an address book for later use in the Symbol browser.

12 ibaBM-DP: Diagnostic functions

Firmware version v01.04.001 of the ibaBM-DP introduces some new diagnostic features:

1. Profibus voltage diagnostics
2. Device event log
3. Extra message counters

In ibaPDA the voltage diagnostics are shown on the PROFIBUS bus node.

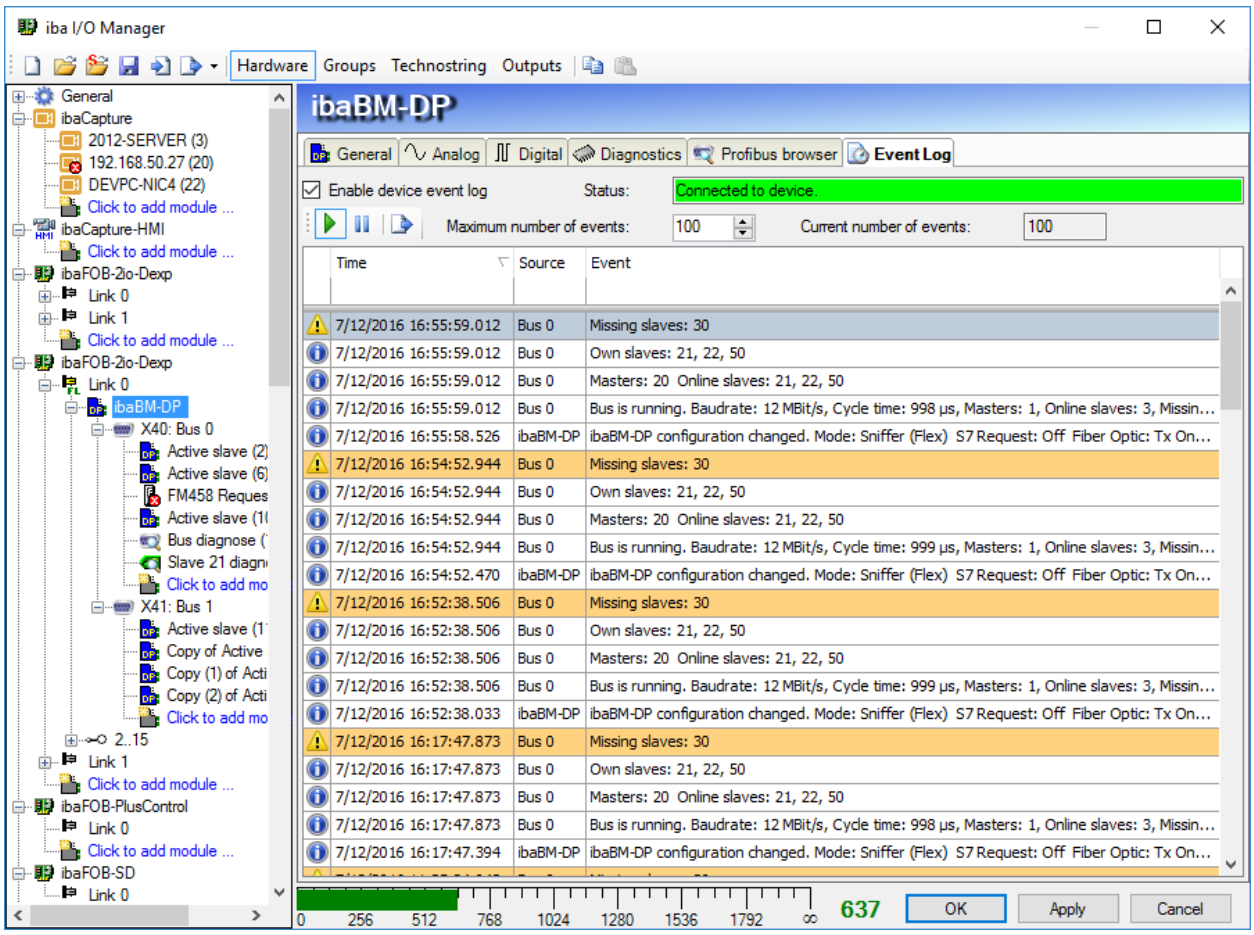


The voltage shown for each PROFIBUS address is the difference between the voltage level for 1 and for 0. It is a helpful utility to get an impression of the overall signal quality of the bus. There are 4 ranges:

- 0 – 2.5V: The voltage is too low. This could be caused by a low impedance or a short circuit in the bus cable. This range is colored red.
- 2.5V – 4V: The voltage is a bit low but the signal is still useable. This can happen when you have a longer bus. This range is colored orange.
- 4V – 7V: The voltage is ok. There is a strong signal available. This range is colored green.
- 7V – 10V: The voltage is too high. This typically points to missing termination or a wire break. The extensive reflections cause the high voltage. This range is colored red.

The actual voltage for an address can be seen under the bar. A rectangle drawn with a dashed line shows the minimum and maximum voltage that has been measured for an address while

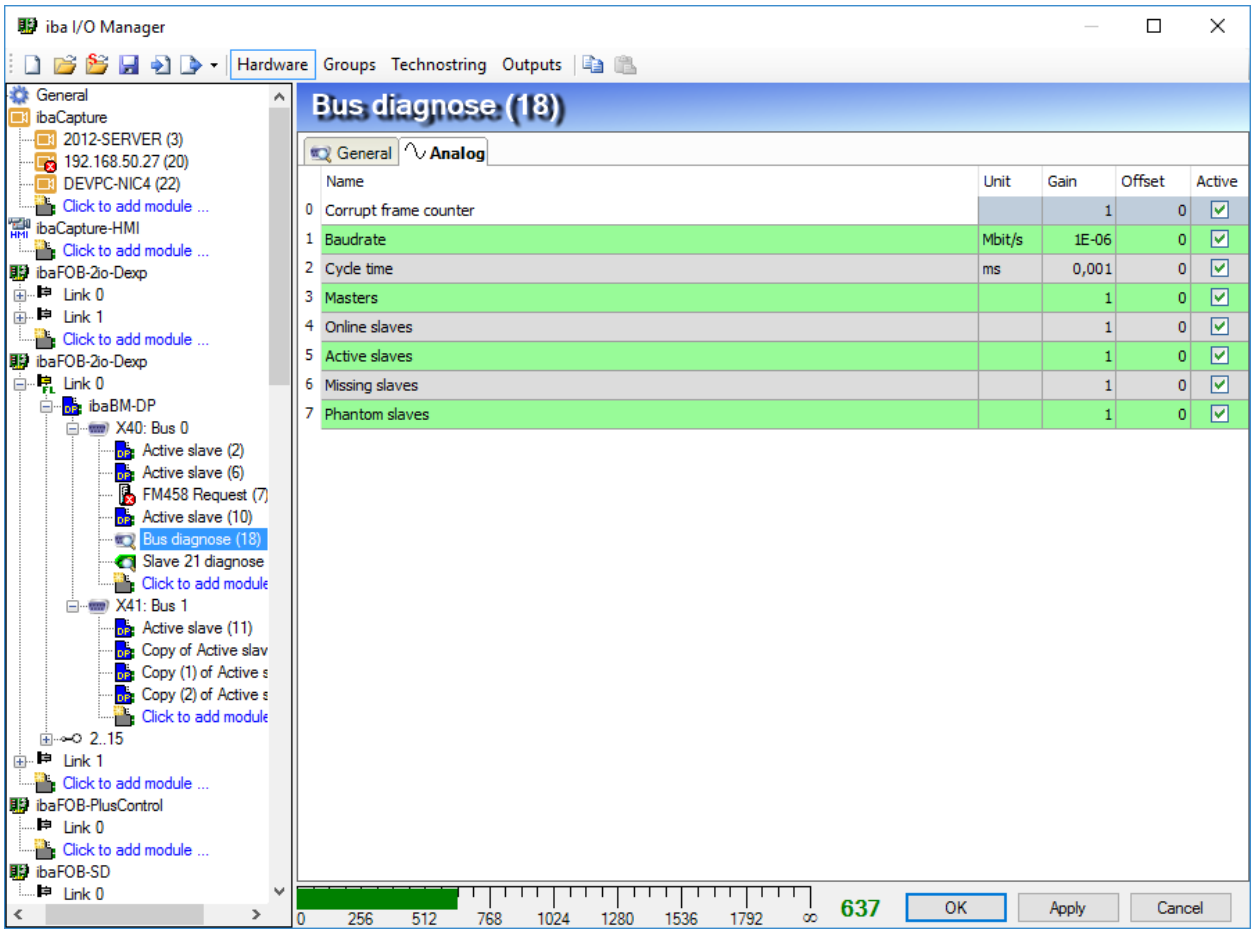
the graph is visible. For more details on how the voltage is measured you can read the ibaBM-DP manual.



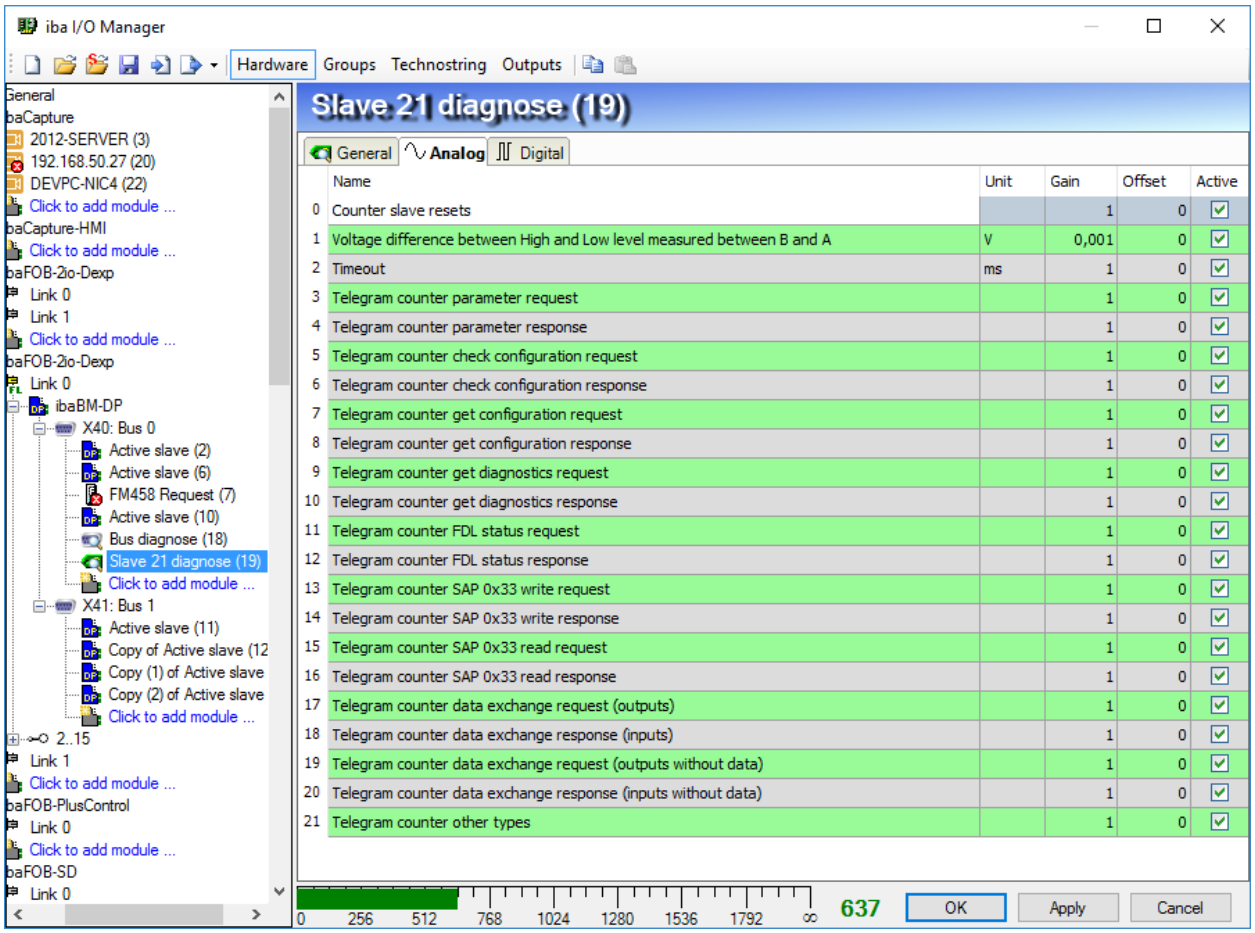
An ibaBM-DP device keeps an event log where it enters all events it encounters. These events can be PROFIBUS related like connects/disconnects/slave state changes/… They can also be related to the ibaBM-DP device itself like receiving a new configuration from ibaPDA. The event log in the device is cleared when the device is rebooted. In ibaPDA you can now enable the retrieval of this event log. When it is enabled ibaPDA periodically checks for new events and it stores it in its own ringbuffer that is kept on disk. In the I/O manager you can configure how many events are kept in the ringbuffer. A table shows the events. Each event has a type, timestamp, source and description. You can use the first row in the table to filter. Wildcards like * and ? are supported. The play and pause buttons allow you to control the update of the table. There is also an export button that allows you to save the visible events to a tab-separated file.

There are 2 new module types in ibaPDA for the ibaBM-DP:

- Bus diagnose module
- Slave diagnose module



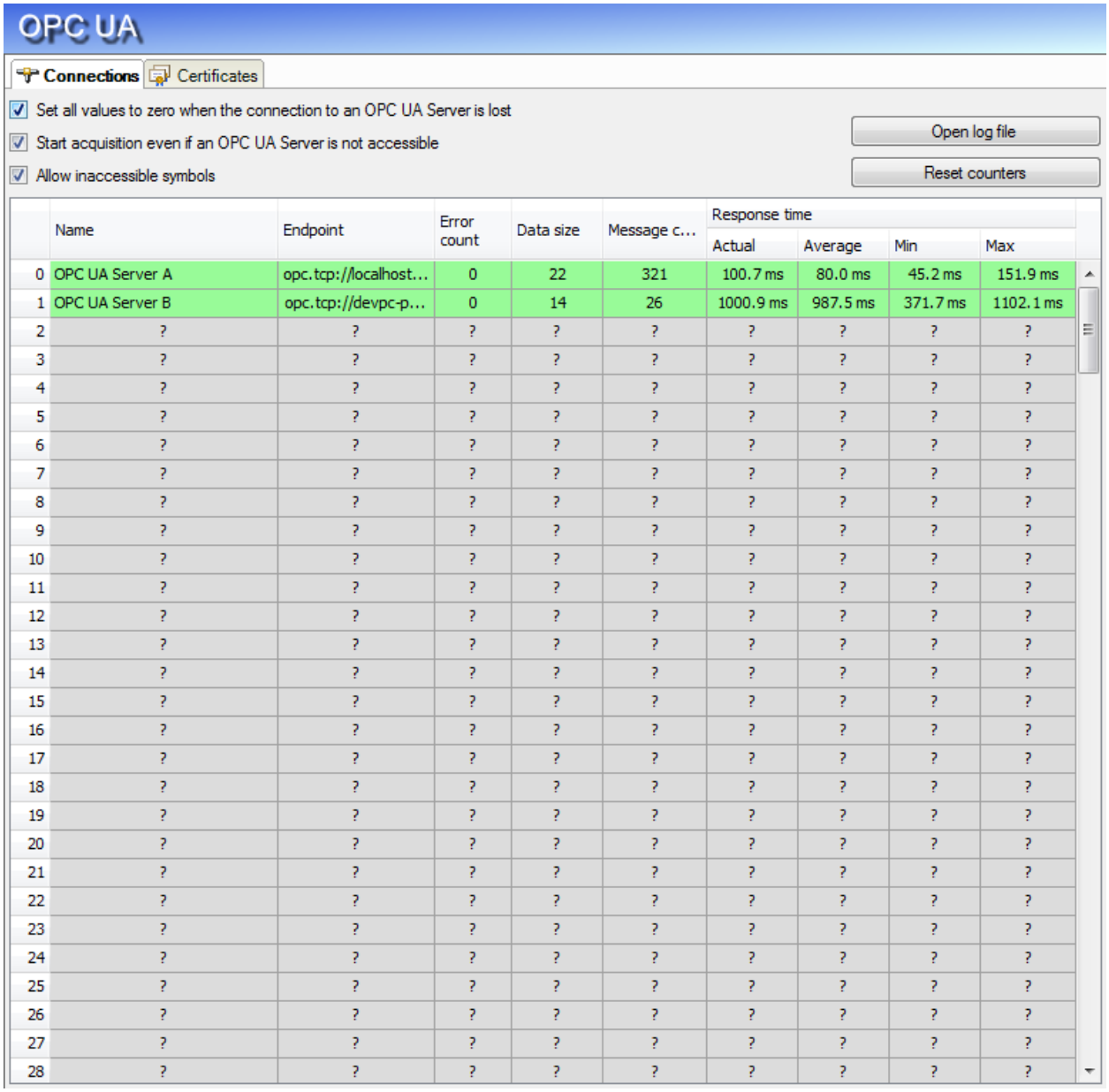
The bus diagnose module can be added once to a bus. With this module you can measure all the counters shown on the bus node.



The slave diagnose module can be added multiple times on a bus. Each slave diagnose module is coupled to a single PROFIBUS slave. With this module you can measure all message counters, voltage level and status info of a PROFIBUS slave.

13 OPC UA Client interface

The OPC UA Client interface in ibaPDA is used to measure data from OPC UA Servers. Instead of cyclically polling for new data, ibaPDA gets notified by the OPC UA Server when one of the requested variables has changed.

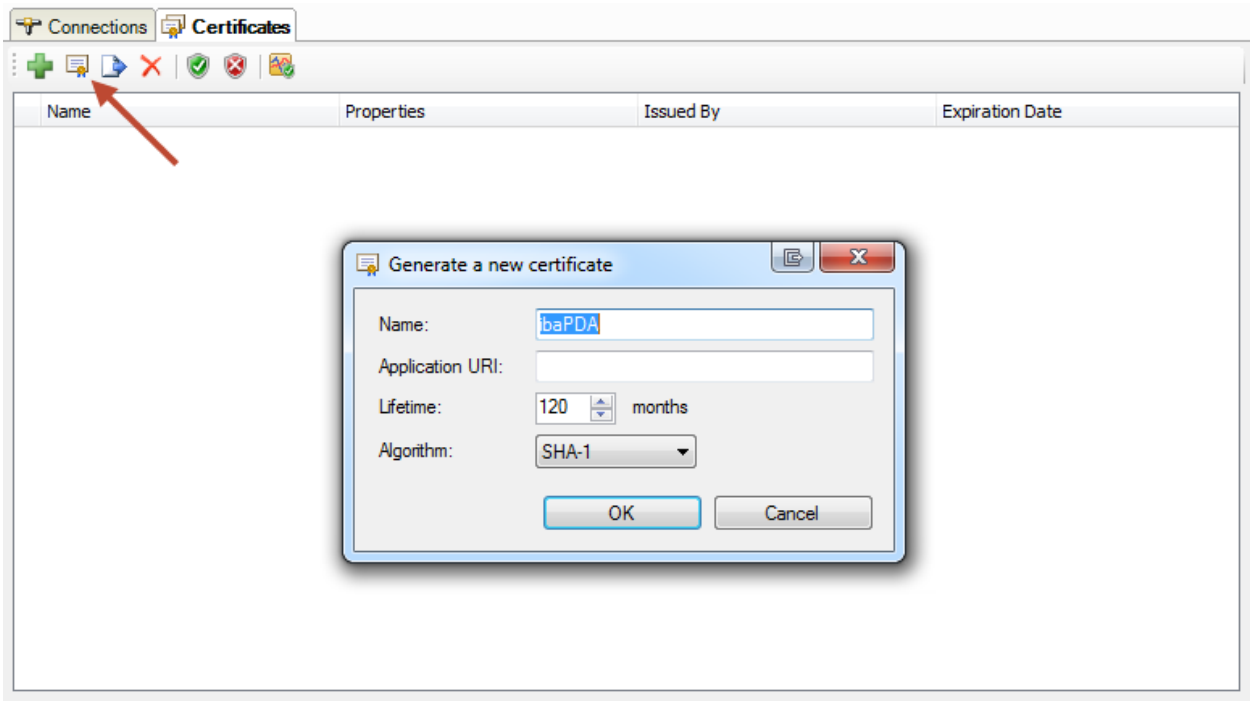


	Name	Endpoint	Error count	Data size	Message c...	Response time			
						Actual	Average	Min	Max
0	OPC UA Server A	opc.tcp://localhost...	0	22	321	100.7 ms	80.0 ms	45.2 ms	151.9 ms
1	OPC UA Server B	opc.tcp://devpc-p...	0	14	26	1000.9 ms	987.5 ms	371.7 ms	1102.1 ms
2	?	?	?	?	?	?	?	?	?
3	?	?	?	?	?	?	?	?	?
4	?	?	?	?	?	?	?	?	?
5	?	?	?	?	?	?	?	?	?
6	?	?	?	?	?	?	?	?	?
7	?	?	?	?	?	?	?	?	?
8	?	?	?	?	?	?	?	?	?
9	?	?	?	?	?	?	?	?	?
10	?	?	?	?	?	?	?	?	?
11	?	?	?	?	?	?	?	?	?
12	?	?	?	?	?	?	?	?	?
13	?	?	?	?	?	?	?	?	?
14	?	?	?	?	?	?	?	?	?
15	?	?	?	?	?	?	?	?	?
16	?	?	?	?	?	?	?	?	?
17	?	?	?	?	?	?	?	?	?
18	?	?	?	?	?	?	?	?	?
19	?	?	?	?	?	?	?	?	?
20	?	?	?	?	?	?	?	?	?
21	?	?	?	?	?	?	?	?	?
22	?	?	?	?	?	?	?	?	?
23	?	?	?	?	?	?	?	?	?
24	?	?	?	?	?	?	?	?	?
25	?	?	?	?	?	?	?	?	?
26	?	?	?	?	?	?	?	?	?
27	?	?	?	?	?	?	?	?	?
28	?	?	?	?	?	?	?	?	?

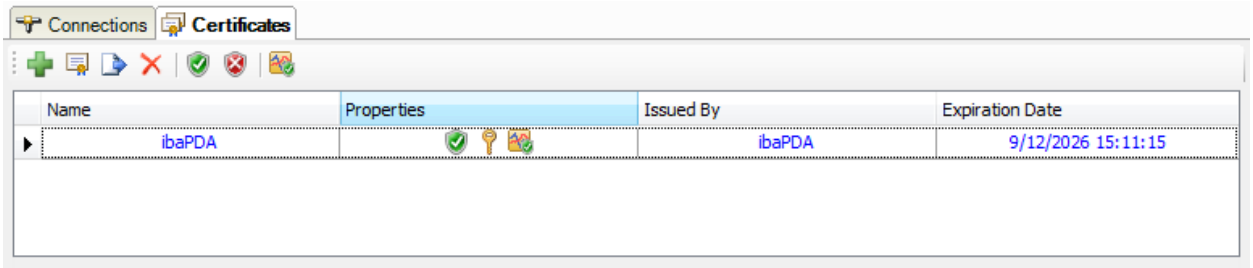
The Connections tab of the OPC UA Client interface shows a table of the available connections. Per OPC UA Client license you get 16 connections. A maximum of 256 connections is allowed. This means that maximum 16 licenses can be used. Each connection corresponds with a row in the table. The row is green when the connection is ok and data is being read. The row is red when the connection could not be established. The row is grey when there is no connection configured. The response time is the time between two notifications from the OPC UA Server that the data of one of the requested variables has changed. The table shows the actual, average, minimum and maximum of the response time. The data size shows how much data is read per read. You can use the “Reset counters” button to clear the counters for all connections.

On the interface you can also decide how to handle some error conditions:

- When the connection to a OPC UA Server is lost during the acquisition then you can choose if the values stay at the last read value or if they are set to zero.
- When a OPC UA Server is not accessible during the start of the acquisition then you can choose if the acquisition starts without this OPC UA Server or if the acquisition is not started. When the acquisition is started without the OPC UA Server then ibaPDA will periodically (every 10s) try to connect to the OPC UA Server during the acquisition. As long as the OPC UA Server is disconnected the values will remain at zero.
- When ibaPDA tries to get the address of a symbol that is no longer available in the OPC UA Server then the OPC UA Server will return an error. If the option “Allow inaccessible symbols” is enabled then ibaPDA will ignore this signal and start the acquisition without this signal. If the option is not enabled then the acquisition will not start.



The Certificates tab shows a list of certificates used for setting up communication between the OPC UA Client (i.e. ibaPDA) and an OPC UA Server. Before ibaPDA can connect to an OPC UA Server, an application certificate needs to be configured. This can be done by either importing an existing certificate file (by clicking the button) or by generating a certificate: when clicking the button, a dialog will appear where you can choose the name of the certificate, an optional application URI, the certificate lifetime (in months) and the hashing algorithm. After clicking OK, ibaPDA will ask whether you want to use the newly created certificate as the OPC UA Client certificate for communicating with OPC UA Servers. When selecting Yes, the certificate list should look as follows:



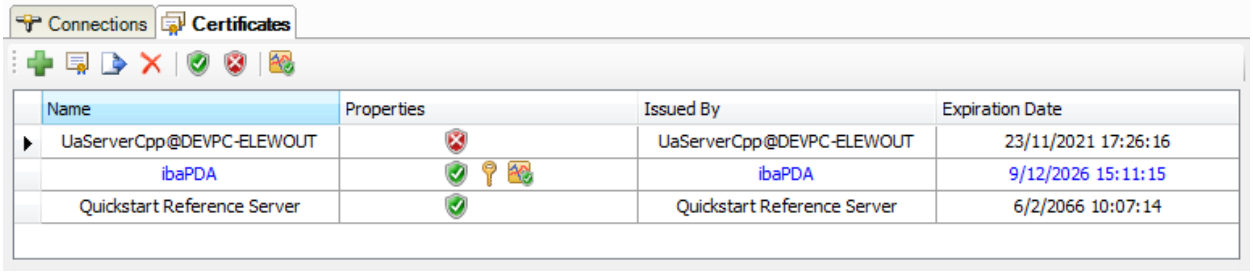
The icon indicates that the certificate is currently trusted, that a private key is present and that this certificate is the OPC UA Client certificate (i.e. the certificate representing ibaPDA).

In the Certificates tab, it is also possible to export a certificate to a file using the button. This way, you can export ibaPDA's generated certificate and import it on an OPC UA Server.

Certificates can be removed from ibaPDA's certificate store by using the button.

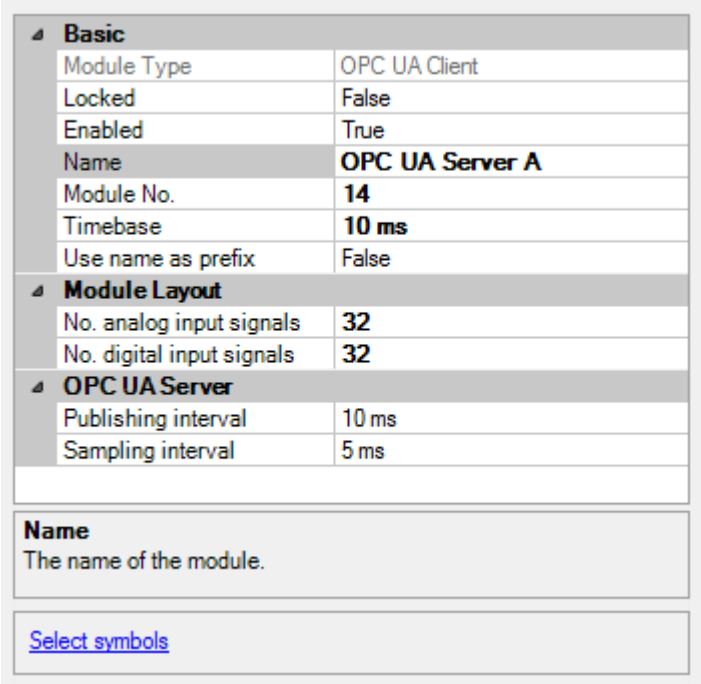
Using the button, a certificate can be trusted; using the button a certificate can be rejected without being removed from the certificate store. When ibaPDA encounters a new certificate that is not yet present in the certificate store, it will prompt you (except during the start of the acquisition) whether the certificate should be trusted or rejected. Once the certificate has been added to the store (either trusted or rejected), ibaPDA will no longer prompt you on subsequent encounters with this certificate.

Finally, using the button, a different certificate can be selected as OPC UA Client certificate. Note that in order for a certificate to be able to become OPC UA Client certificate, it needs to have its private key in the certificate store.



In the picture above, the certificate ibaPDA is the OPC UA Client certificate (marked in blue) while there is one certificate from an OPC UA Server that is trusted and one from another OPC UA Server that is rejected.

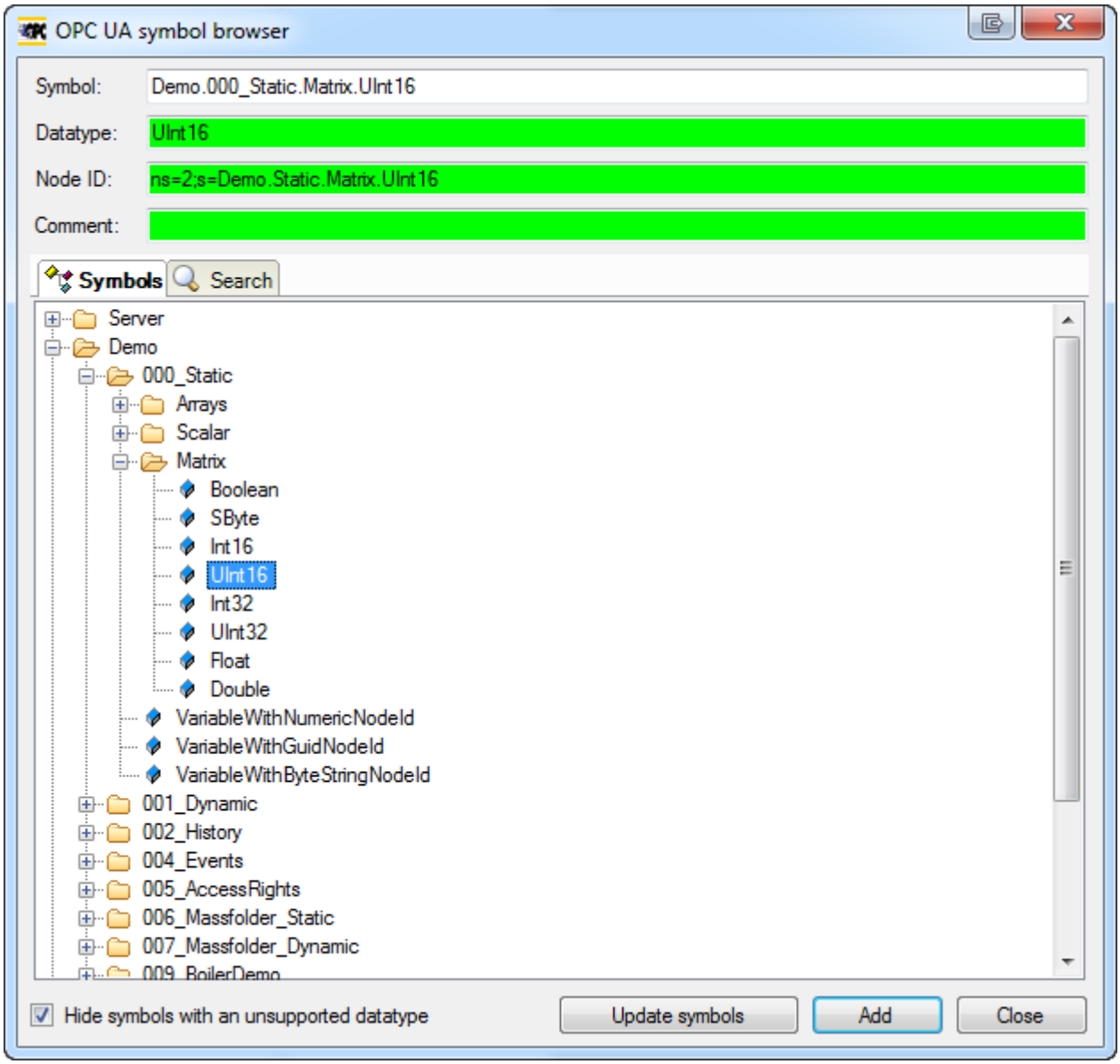
The General tab of an OPC UA Client module looks as follows:



Apart from the standard options that can be found on most other modules, there are some specific settings:

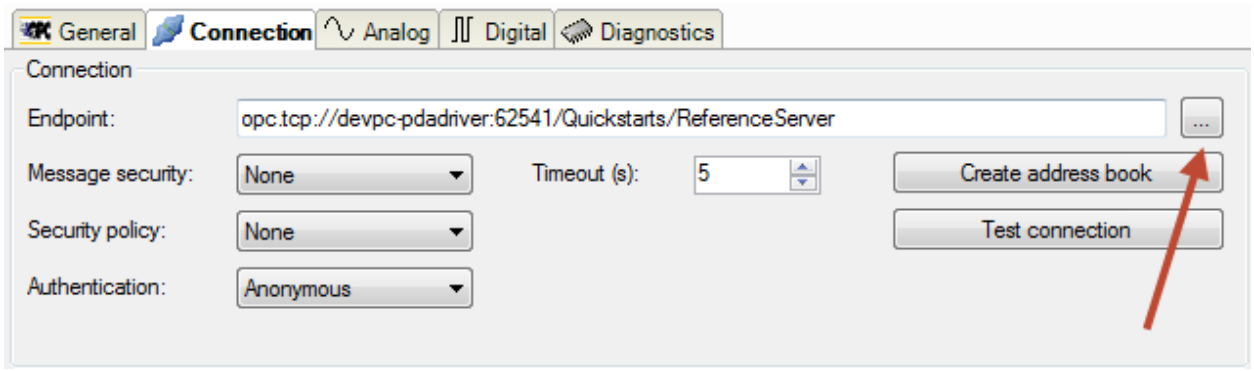
- **Publishing interval**: the minimum amount of time the OPC UA Server waits to notify ibaPDA that the data of one of the requested variables has changed.
- **Sampling interval**: the timebase used to sample the underlying data source in the OPC UA Server. In theory, this should be equal to the publishing interval but it is advised to set this to half of the publishing interval in order to compensate for small internal delays in the OPC UA Server.

At the bottom, the symbol browser can be opened by clicking **Select symbols**.

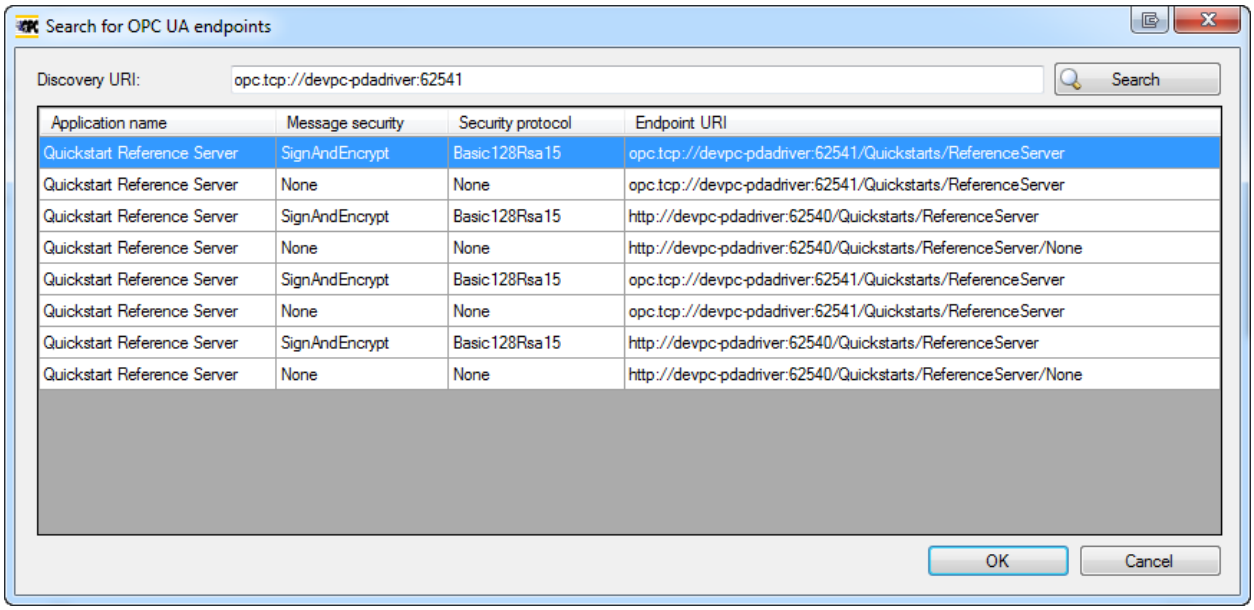


The symbol browser shows all the symbols that were loaded from the OPC UA Server. You can select single or multiple symbols in the tree. Click the *Add* button to add them to the corresponding analog or digital signal grid. If you selected a single symbol then the next symbol will be selected after you clicked the *Add* button. This allows you to hit *Add* multiple times in order to add consecutive symbols. You can also double click a symbol to add it to the signal grid. Use the *Update symbols* button to read the symbols again from the OPC UA Server.

On the search tab you can search symbols by name. The search result tree works in the same way as the complete symbol tree.

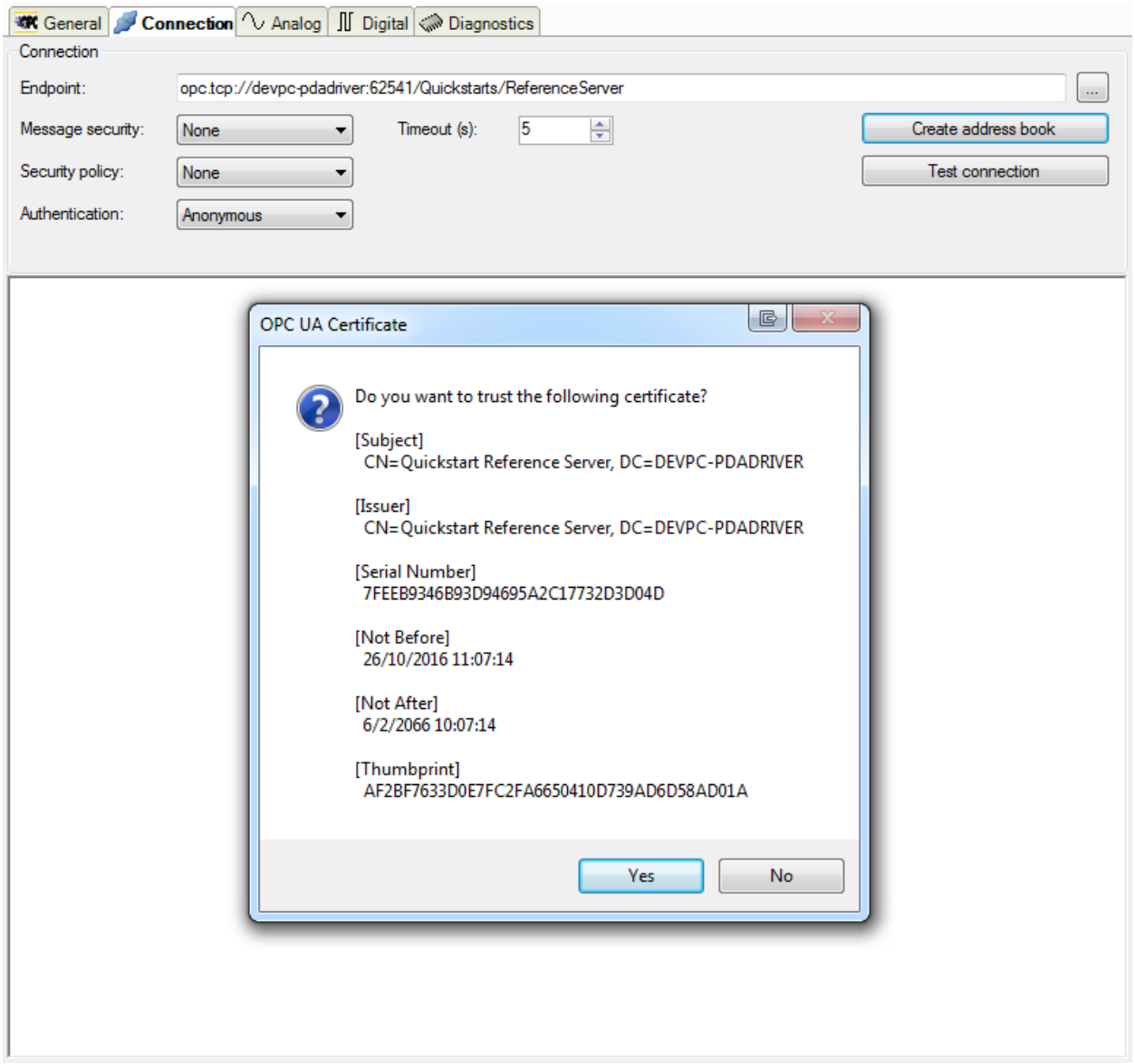


In the Connection tab, it's possible to search for OPC UA endpoints given an OPC UA Server's discovery address. By clicking the … button next to the Endpoint field, a discovery form will be opened.

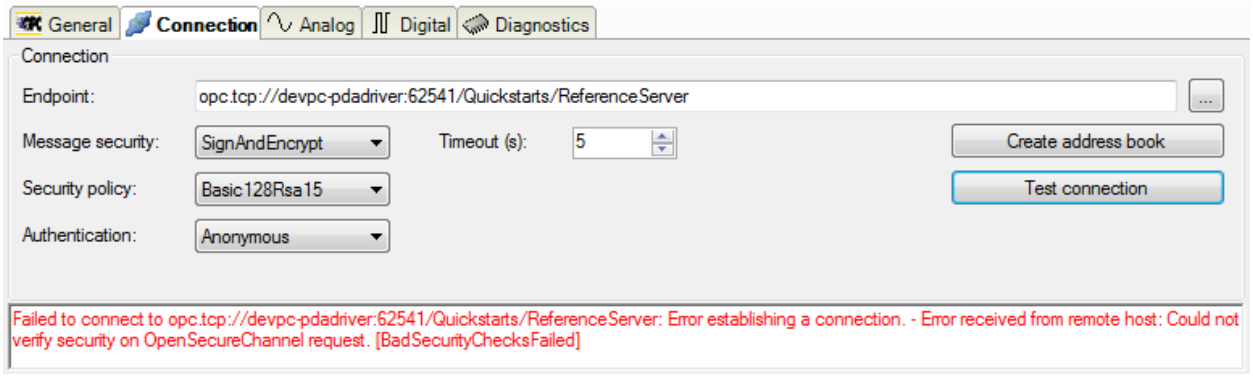


After entering the Discovery URI and clicking Search, a list of available endpoints should appear. Select the requested endpoint and click OK (or double click on the endpoint) to copy its settings to the OPC UA Client module's Connection tab.

Apart from the endpoint's parameters, the Authentication method should also be set according to the OPC UA Server's configuration. Currently Anonymous authentication and User/password based authentication are supported by ibaPDA.

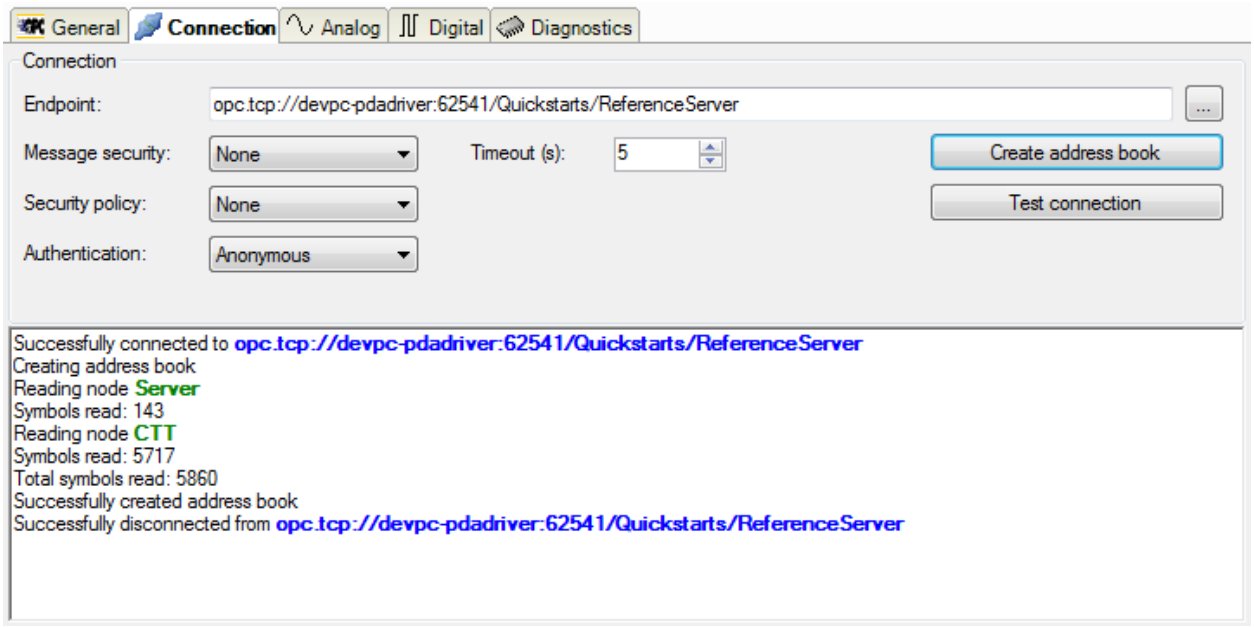


Once all parameters have been properly configured, it's possible to test the connection (by clicking Test Connection) or generating an address book (Create address book). In both cases, ibaPDA will set up a connection to the OPC UA Server. In case ibaPDA connects to the server for the first time, the server's certificate will not be present in ibaPDA's certificate store. You will be prompted to trust or reject the OPC UA Server's certificate. If you decide to reject the certificate, communication can not proceed.



Unless ibaPDA's OPC UA Client certificate is already trusted by the OPC UA Server or no security is being used, the above error message might appear. In several OPC UA Servers

ibaPDA's certificate will probably automically end up in the server's list of rejected certificates. By accepting ibaPDA's certificate in the OPC UA Server, the next connection attempt should occur without problems.



Once OPC UA Client and OPC UA Server trust each other's certificate, ibaPDA can proceed to establish a valid connection to the OPC UA Server. By clicking Create address book, all symbols will be read from the OPC UA Server and stored locally in ibaPDA so they can easily be accessed offline (i.e. without a connection the OPC UA Server) using the symbol browser.

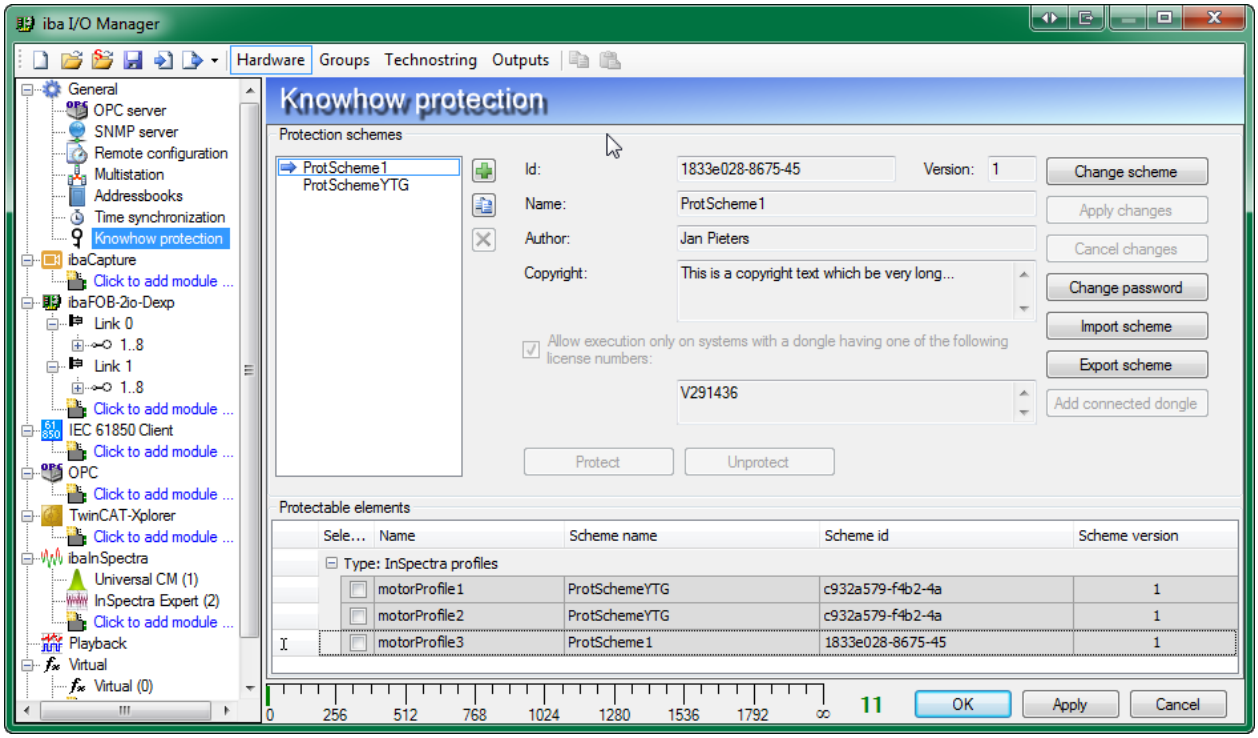
14 Knowhow protection

This new version of ibaPDA includes a framework to protect intellectual property. In the first version of this framework, only InSpectra profiles can be protected.

The framework has the following features:

- Change protection: the protected elements can not be changed without knowing the password
- Reading protection: the configuration of the protected elements is never displayed without knowing the password
- Dongle protection: possible to configure a set of dongle license numbers; the protected element will only work in combination with these dongles

The protection is done with reusable protection schemes. In a first step, one has to create a protection scheme. In a second step, one can apply the protection scheme to an element. The protection can be removed using the password of the protection scheme.



14.1 Creation of a protection scheme

Protection schemes have the following parameters:

- Identifier + version
- Name
- Password, used to:
  - Consult the configuration of the protected element
  - Change the configuration of the protected element
  - Change or remove the protection scheme

- (optional) Dongle license numbers: possible to configure a set of dongle license numbers; the protected elements will only work in combination with these dongles
- (optional) Author
- (optional) Copyright text

The password has a minimum length of 8 characters. Spaces are not allowed.

14.2 Applying a protection scheme

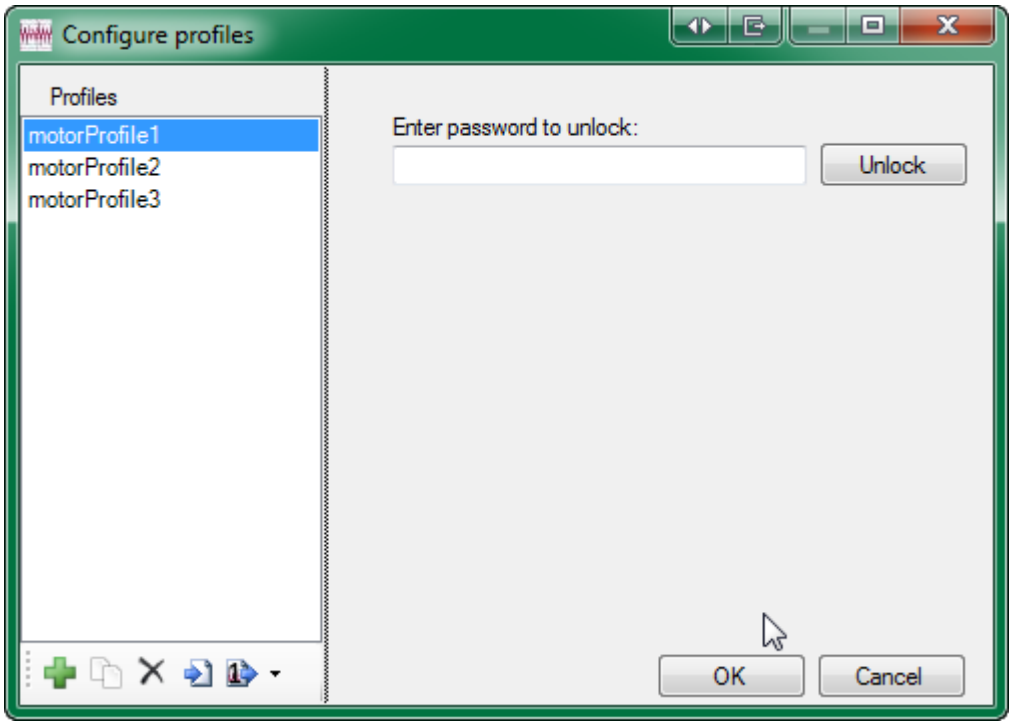
In the bottom table, one can select one or more elements using the checkboxes on the left. With the *Protect* and *Unprotected* buttons, one can protect or unprotect the selected elements. For each operation, you will be prompted to enter the password of the scheme(s) involved.

14.3 Importing and exporting a protection scheme

Protected elements can be exported to and imported from a *.protectionScheme file. This file is encrypted and makes it easy to deploy your intellectual property to an existing ibaPDA configuration.

14.4 Unlocking protected elements

The configuration settings of a protected element are hidden. In stead of the settings, one will see the following dialog:



By entering the password and pressing the unlock button, you can visualize the settings (the intellectual property becomes visible). The information will stay visible until you close the I/O manager.