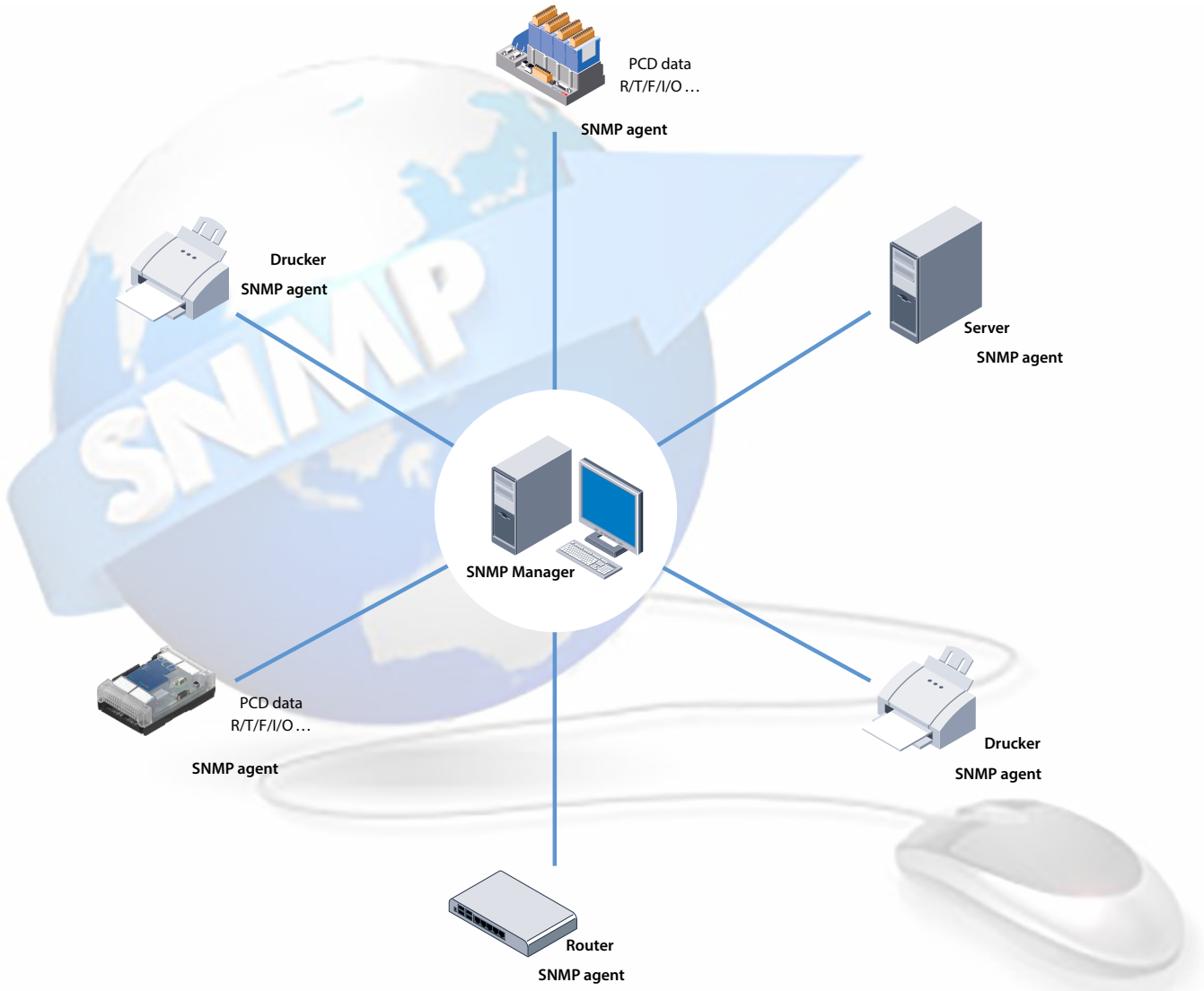


SNMP function of Saia PCD®

Document 27-639 | Edition ENG03 | 2017-09-14

0

Content

0

0

0.1 Document versions

Version	Changes	Published	Remarks
EN01	2014-06-13	2014-06-13	New document (from Word to InDesign)
ENG02	2017-07-24	2017-07-25	Ch.3.4.2 “OID” added
ENG03	2017-09-14	2017-09-14	Ch.3.3 Replaced the whole list of PCD’s with the term “PCDs with Ethernet”

0.2 About this manual

See the section in the appendix in relation to some of the terms, abbreviations and the references used in this manual.

0.3 Brands and trademarks

Saia PCD® and Saia PG5®
are registered trademarks of Saia-Burgess Controls AG.

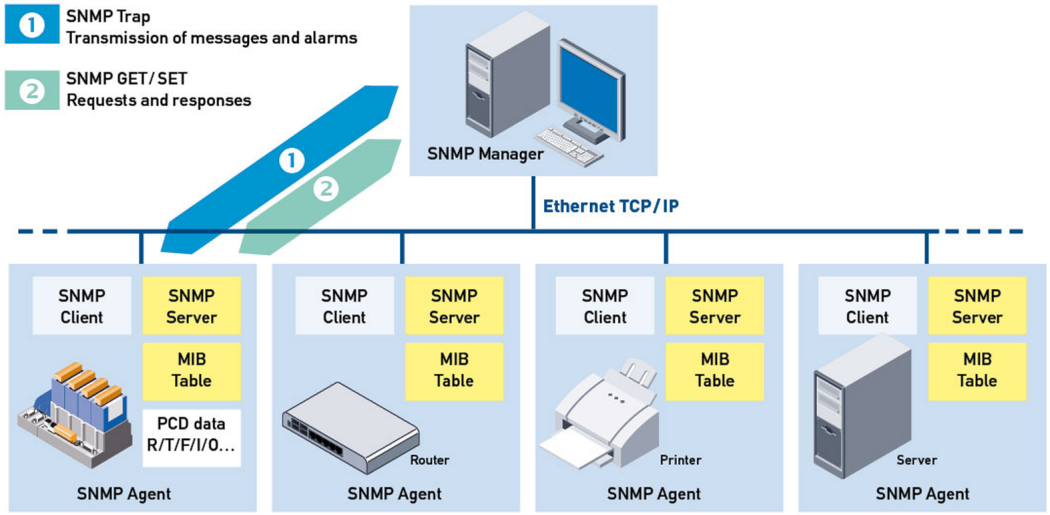
Technical modifications are based on the current state-of-the-art technology.

Published in Switzerland

1 Introduction

The SNMP manager software usually runs on a server. It monitors and controls SNMP agents. The SNMP manager reads and sends data from the agent using SET and GET commands. The SNMP agent can also send unrequested so-called 'trap' messages to the SNMP manager. This allows, for example, the direct reporting of faults.

Saia PCD®-MIB has been defined for Saia PCD® with SNMP support. Within it all the resources that can be queried and modified with SNMP are represented. Basically, all PCD media can be accessed (inputs/outputs, registers, flags, DBs, etc.). In the MIB file, the programmer can restrict access to selected areas only.



1.1 Application possibilities

With implementation of the SNMP protocol in Saia PCD, the customer can now monitor all network-enabled devices (routers, servers and Saia PCD) on the same management platform. SNMP is supported by very many modern devices and can be used as a common interface for administration and configuration. The management system can link up different items of information available throughout the network. With SNMP, you have access to all PCD resources. Parameters can be retrieved and modified. Using a configuration file, it is possible to define which resources can be read and written. When the controller changes the mode in which the user program is processed (change between RUN / STOP / HALT), the PCD can automatically send out traps. It is also possible to define whether a measured value (e.g. temperature) should spring a trap when a certain value is exceeded.

1.2 General Data

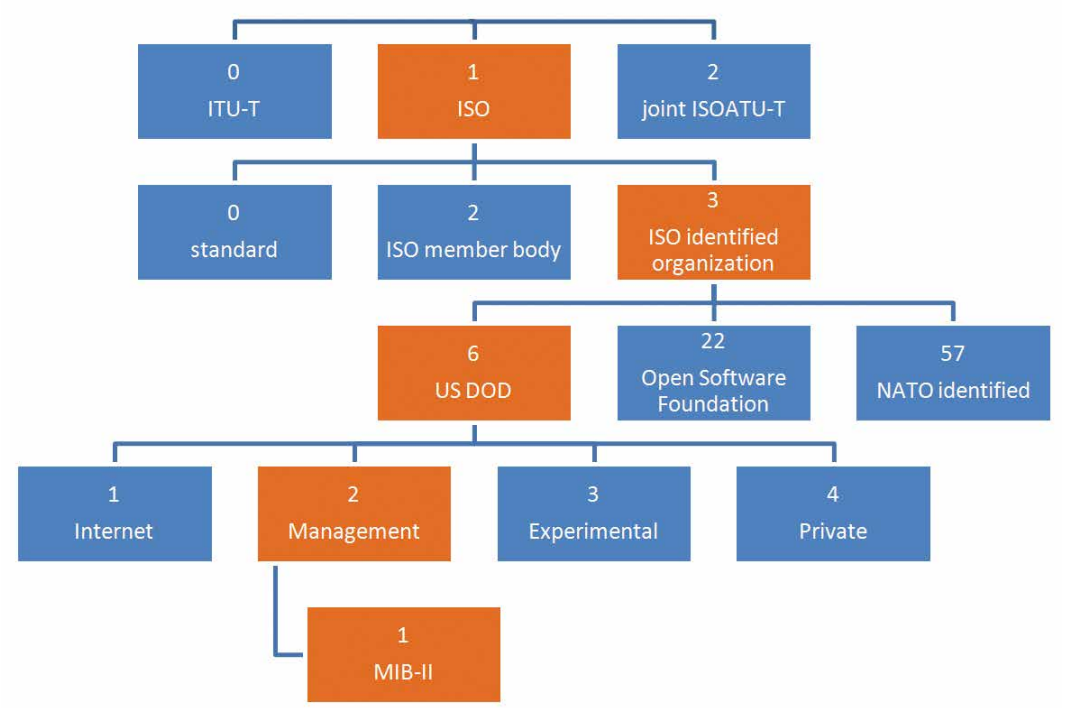
General Data

Type	Description
SNMP standard	v1, v2c
Port	161, 162
RFC	SNMP v1, v2: RFC 1155,1157, 1353, 1398, 1447 SNMP v2: RFC 1573, 1757, 1902 to 1908, 2011, 2012, 2013, 2096, 2863 MIB II: RFC 1213
Trap text length	128 characters
Connection to SNMP Manager	Only direct via fixed IP address
Standard MIB Files	MIB-II standard
Specific MIB Files	SaiaMIB_Classic

2 Description of the MIB-II standard

The SNMP protocol is based on requests which are sent by a network management station to a host and answered with a reply. All SNMP requests refer to a tree structured directory where all network settings, protocols and statistics of a device are saved under the name «MIB-II» (Management information base). The MIB-II has a standardised part for IP-networks. In addition a private part can be added as well. This makes it possible to add your own data with your own structure and to make them available via SNMP set and get commands.

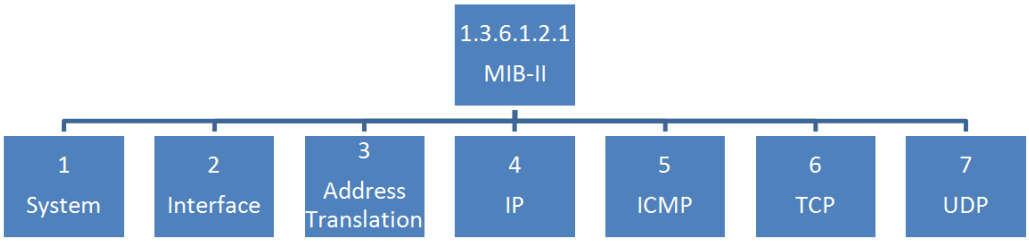
The structure of the **standard MIB** is shown below.



2.1: Standard MIB Tree

This graph shows the way to the MIB-II folder in the Standard MIB tree. In most implementations all folders except the MIB-II folder are empty. Furthermore, every folder has a specific number which allows the path to the MIB-II folder to be described. For this reason, all elements of the **MIB-II** start with:

1.3.6.2.1



2.2: The MIB-II

The MIB-II already contains a lot of information. In the chart only the first category is shown. To see all the specific data you have to load the SaiaMIB_Classic MIB as described in chapter 3.4.

The MIB-II tree is described in the RFC 1213 and works with every Saia PCD. The MIB-II is usually already installed with the SNMP Manager software (in iReasoning you have to load the RFC1213 MIB).

In addition to the MIB-II you can load a Saia PCD specific **SaiaMIBClassic.mib** which is located in the private folder and reachable with the number

1.3.6.2.4

This MIB contains SBC specific data and makes it also possible to get specific registers or other Saia PCD data. You can customise this MIB with the SBC MIB File generator and add specific variables to it. This powerful option enables you to change output flags directly via an SNMP command, for example.

3 Getting Started with the SNMP function

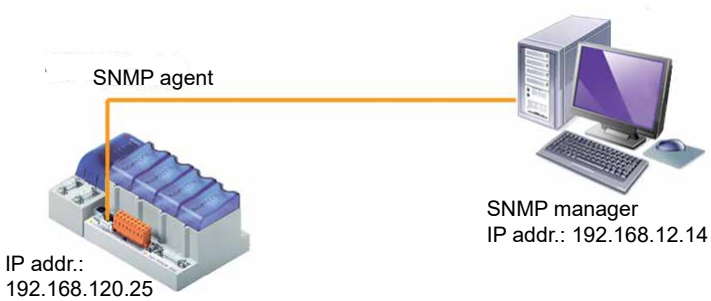
3.1 Description

The Simple Network Management Protocol (SNMP) was developed to allow network elements like routers, servers and switches to be monitored and controlled from a central station. The SNMP manager is usually software that runs on a server. It monitors and controls the SNMP agents. They may be any choice of device that can be reached via the network and that supports SNMP. With the new firmware, the Saia PCD supports SNMP agent functionality.

The following SNMP versions are available: v1, v2c, v3 (safety mechanism with MD5 authentication, encryption with DES 56 bits). The standard v3 has not yet achieved very wide circulation. Version v2c is, in principle, still the current standard. Saia PCD supports version v1 and v2c.

The following shows the possibilities of SNMP implementation in Saia PCD with the help of one PCD and a Windows PC with a Net-SNMP tool.

Diagram:



3.1: Getting started: hardware configuration

3.2 Possible applications

Implementation of the SNMP protocol in Saia PCDs has enabled customers to monitor all their network devices: (routers, servers and Saia PCDs) on the same management platform. SNMP is supported by many modern devices and can be used as a common interface for administration and configuration. The management system can link diverse pieces of information available throughout the network.

SNMP gives access to all PCD resources. Parameters can be polled and modified. A configuration file is used to define which resources can be read or written to. If the controller changes the mode for processing a user program (change between RUN / STOP / HALT) the PCD can automatically send a trap. It is also possible to define whether a measurement (e.g. a temperature) should spring a trap if a certain value is exceeded.

3.3 Material required

- 1 laptop / PC with PG5 utilities
- 1 PCD is required for this application.

The following types may be used:
- any PCD with Ethernet and the Operating System: Saia PCD® COSinus

The following material is used for the test:

- PCD3.M5540
- Laptop HP Compaq 6715b
- Net-SNMP client software

3.4 Configuration and start with SNMP

The examples always include a long sequence of numbers that are separated by points: this number always starts as follows: 1.3.6.1.4.1.31977… . If you want to use the SNMP protocol with a Saia PCD, the addresses must always start with these seven numbers. The first 6 numbers (1.3.6.1.4.1) stand for «iso.org.dod. internet.private.enterprise».

The number 31977 is the number registered with IANA for products from Saia Burgess Controls.

This number is a fixed setting in firmware and cannot be modified.

The SaiaMIB_Classic.mib file translates this hierarchy of numbers (starting with 1.3.6.1.4.1.31977) into a structure of text definitions.
They begin with: SaiaPCDClassic: … .

Text definitions and numbers may be mixed.

3.4.1 Configuration PCD

The following settings are required in the PG5 Device Configurator.

<table>
<tr><td>To use PCD as an SNMP Agent, enable this functionality first.</td><td>
Ethernet Protocols
<table>
<tr><th>Section</th><th>Description</th></tr>
<tr><td>IP Transfer Protocols</td><td>FTP, HTTP Direct Protocols, ODM.</td></tr>
<tr><td>IP Protocols</td><td>DNS, SNTP, SNMP protocols.</td></tr>
<tr><td>HTTP Portal</td><td>HTTP Portal Communication For PCD Over Private Network</td></tr>
</table>
</td></tr>
<tr><td>1) Enable SNMP
2) Define IP addresses where to send the SNMP traps (in our case your computer)</td><td>
⊟ SNMP (Simple Network Management Protocol)
<table>
<tr><td>SNMP Enable</td><td>Yes</td></tr>
<tr><td>sysContact Message</td><td>Saia Burgess Controls AG</td></tr>
<tr><td>sysLocation Message</td><td>CH-3280 Murten</td></tr>
<tr><td>Life Trap Interval [ms]</td><td>0</td></tr>
<tr><td>Trap 1 Port Number</td><td>162</td></tr>
<tr><td>Trap 1 IP Address</td><td>192.168.12.13</td></tr>
<tr><td>Trap 2 Port Number</td><td>0</td></tr>
<tr><td>Trap 2 IP Address</td><td>0.0.0.0</td></tr>
<tr><td>+ Advanced Parameters</td><td>Yes</td></tr>
<tr><td>Start Delay [s]</td><td>5</td></tr>
</table>
</td></tr>
<tr><td>3) Please note that usually these strings have to correspond to the entries in the SNMP manager.</td><td>
<table>
<tr><td>Read Command String</td><td>public</td></tr>
<tr><td>Write Command String</td><td>private</td></tr>
<tr><td>Trap Command String</td><td>public</td></tr>
</table>
</td></tr>
<tr><td>4) Memory allocation for data transfer
Only the defined areas can be used for read or write data transfer over SNMP (SET and GET commands)
The default all values are read only.
Supported Media: IO, F, R, T, C, DB</td><td>
⊟ SNMP (Simple Network Management Protocol)
...
<table>
<tr><td>First Readable Flag Address</td><td>0</td></tr>
<tr><td>Last Readable Flag Address</td><td>14336</td></tr>
<tr><td>First Writable Flag Address</td><td>0</td></tr>
<tr><td>Last Writable Flag Address</td><td>14336</td></tr>
<tr><td>First Readable Register Address</td><td>0</td></tr>
<tr><td>Last Readable Register Address</td><td>16384</td></tr>
<tr><td>First Writable Register Address</td><td>0</td></tr>
<tr><td>Last Writable Register Address</td><td>16384</td></tr>
</table>
</td></tr>
</table>

Download the configuration to the PCD after you have made the changes.

3.4.2 Using the MIB File Generator to handle project symbols (requires PG5 2.1 or later)

This tool allows a device specific MIB (Management Information Base) file to be created. The MIB files are the bases for SNMP managers to access information on SNMP configured devices.

This file can then be used by any MIB browser to get access directly to the symbol name related to the device.

Please Note: the project symbols OID (Object Identifier) will not appear when executing a walk command on the device.

Generated Symbol Names

The MIB file syntax (ASN.1 notation) imposes some restrictions concerning the names.

If the symbol name does not comply with this syntax, the MIB generator will transform the symbol names into ASN.1 compatible names.

This includes

- The first symbol letter should be written in lower case
- The '_' (underscore) is not allowed
- The ' ' (space) is not allowed
- Special characters (ä, ö, ü, é, à, è, etc.) are not allowed

The log file contains all changes made to the symbol names.

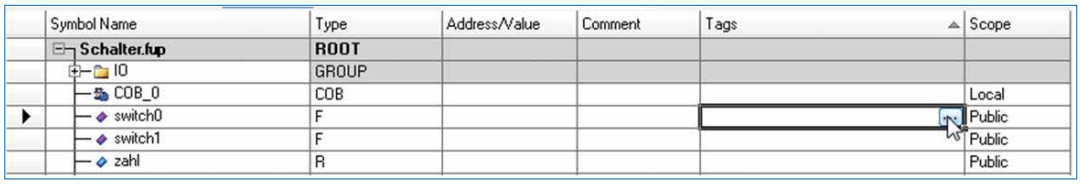
The tool has two modes:

- MIB file editor
- MIB file compiler

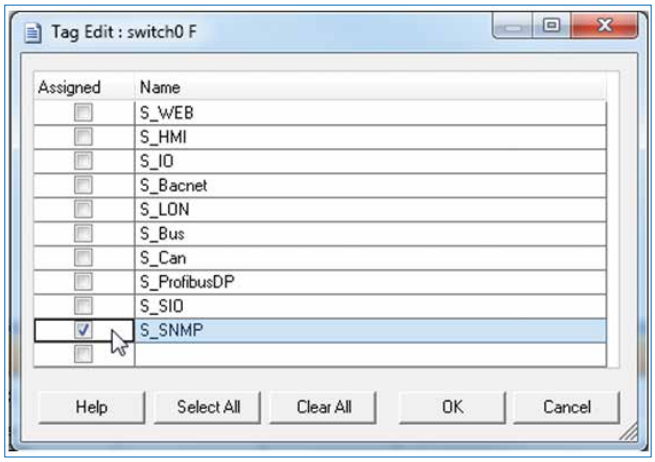
The purpose of this tool is to create an MIB project file to be used by SNMP to transmit PCD media, according to the ASN.1 standard

How to have symbols in the MIB file generated

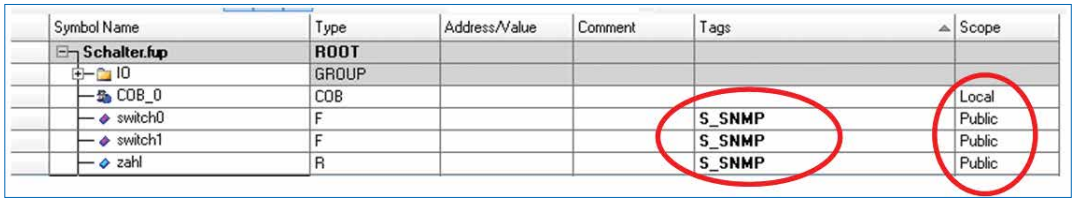
In the device symbol editor, select a symbol and click in the related «tags» column.



In the «Tag Edit» window, select an existing tag (new tags can be added as required). Best practice would be to add an S_SNMP tag. Individual tags contain only alphanumerics, underscores and single dots.



For all symbols that have to be present in the MIB file, assign the tag. Make sure that all variables for the SNMP are public.

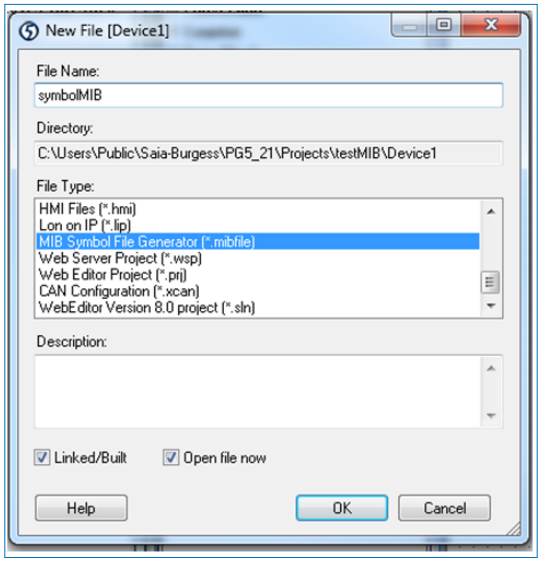


Save all files and compile the project.

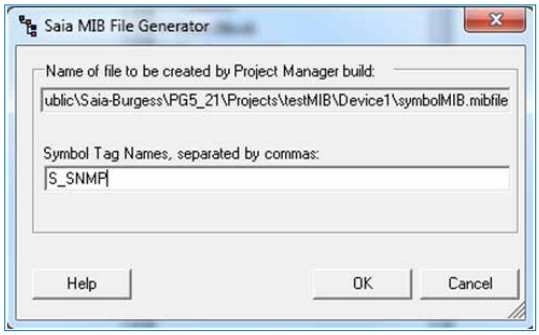
3

How to generate the device MIB file

By inserting a device mibfile with File/New in the «Program Files» folder and configuring the «Symbol Tag Names» (as used/defined in the symbol editor).

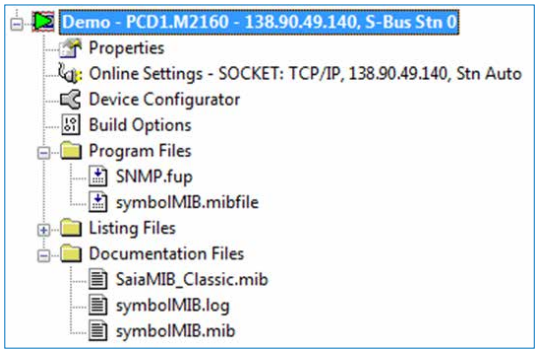


Open the symbolMIB.mibfile and add the tag from the symbol editor (here: S_SNMP)



After that the .mib Files will be created automatically with the next compilation. The generated files can be found in the device «Documentation Files» folder, as wells in the related log file.

The function will always generate the standard SaiaMIB_Classic.mib file as well. This file is needed in addition to the symbolMIB.mib file we generated.



3.4.3 PC configuration

Installation of the MIB Browser software from http://www.ireasoning.com/

Open the MIB Browser and select the MIB Files from your project folder (open the project folder in PG5 Project Manager with Tools/Explorer) with File/Load MIBs.

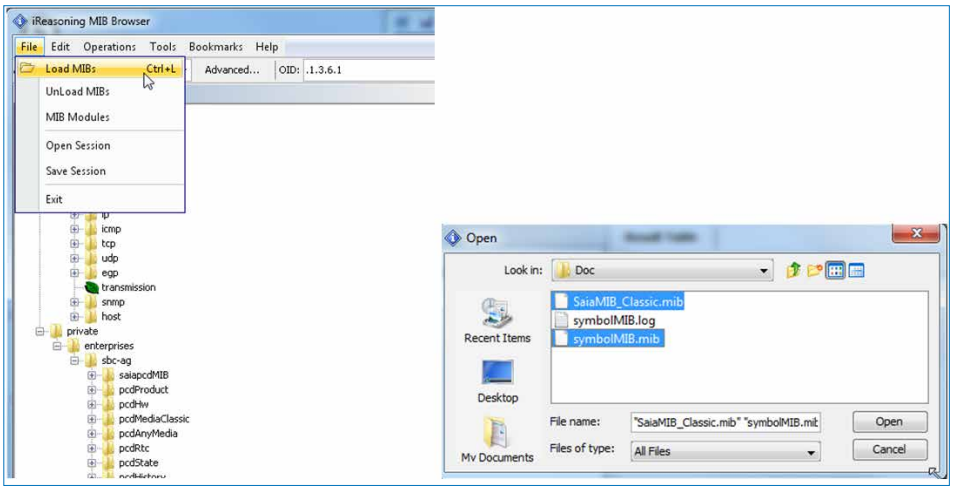


Figure 3.2: MIB Tree in iReasoning

Connect your Computer via the Ethernet cable to your PCD and ensure that the Windows firewall – or any other software firewall you are using – is correctly adjusted. It is easier to shut down the fire-wall service completely. The risk involved is small, because you are only using the LAN. At the end of the test, the service must be reactivated.

3.4.4 Examples of Snmpget function

To read data from your PCD you have to follow the instructions below.

1. Write the IP address of your PCD device (SNMP Agent) in the address window

2. Select the register reg1000-32s in the MIB tree
   The OID number of the register will appear automatically in the OID window

3. The last number of the OID is the register address. In our example register 1 is selected. You can change this directly in the OID window.

4. Set the Operations to Get
   This function reads the value from the register

5. Press Go to read the value
   In the example the register has the value 0

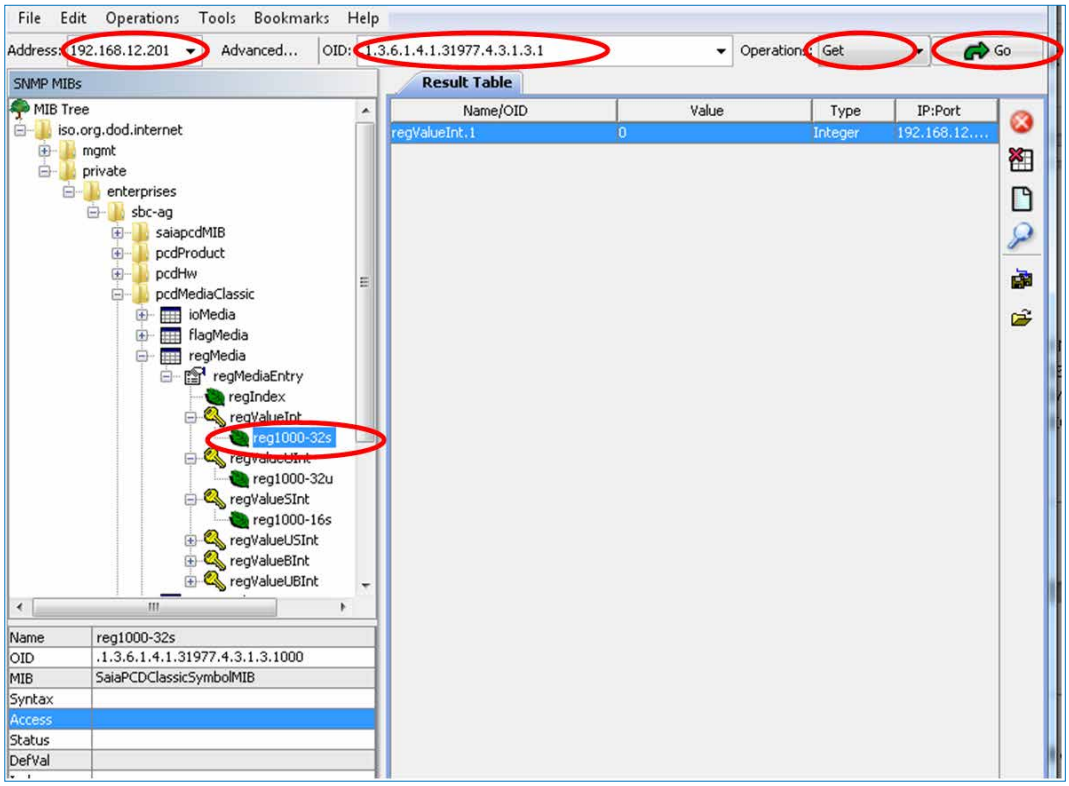


Figure 3.3: Read value from register 1 with iReasoning

The value of Register 1 can now be changed with the SBC Online Debugger and the snmpget command can be executed again.

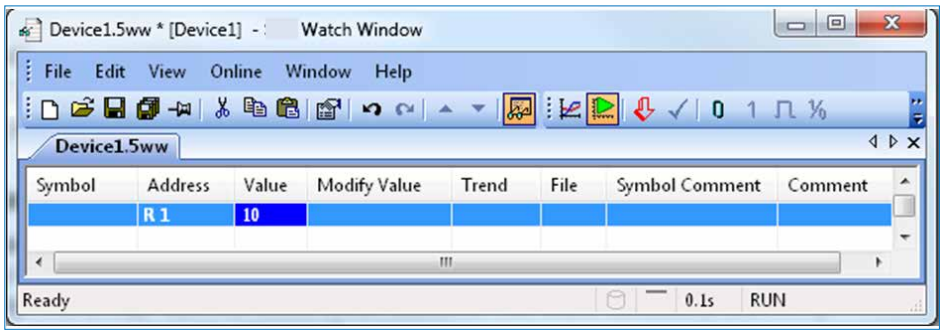


Figure 3.4: Online Debugger in PG5 to write 10 to register 1

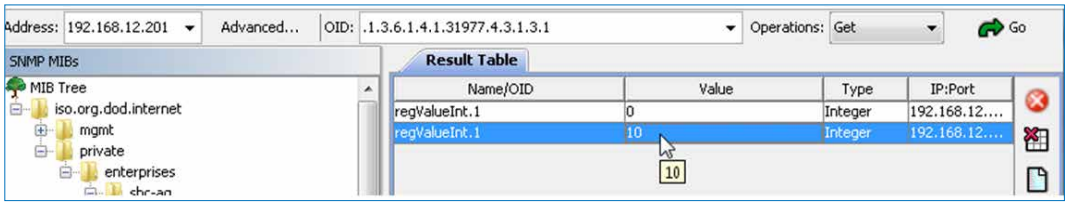


Figure 3.5: Changed register value in iReasoning

Further examples for reading values:

PCD Run / Stop switch:

MIB Tree: Private/enterprises/saiapcdMIB/pcdSwitchState

→ pcdSwitchState.0 = 1 (PCD in run)

MIB Tree: Private/enterprises/saiapcdMIB/pcdSwitchState

→ pcdSwitchState.0 = 0 (PCD in stop)

3.4.5 Examples of Snmpset function

In this example, the value sbc123 has been set for the parameter Write community. This is to protect against unauthorized write access. However, it must be remembered that these passwords are sent over the network without encryption.

SNMP (Simple Network Management Protocol)	
SNMP Enable	No
sysContact Message	Saia Burgess Controls AG
sysLocation Message	CH-3280 Murten
Life Trap Interval [ms]	0
Trap 1 Port Number	0
Trap 1 IP Address	**192.168.12.14**
Trap 2 Port Number	0
Trap 2 IP Address	0.0.0.0
+ Advanced Parameters	**Show**
Start Delay [s]	5
Read Command String	public
Write Command String	**sbc123**
Trap Command String	public
Trap version	V2c

Figure 3.6: Settings in the device configurator

The settings you made in the device configurator have to be made in the advanced settings of the MIB browser as well:

1. Write community = sbc123
2. SNMP version= 2 (in the device configurator V2c)

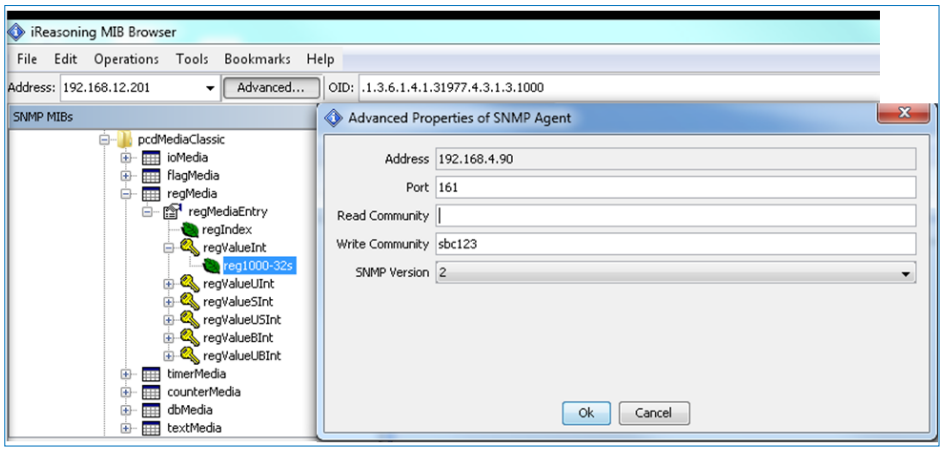


Figure 3.7: Settings for write access in the MIB Browser

After that change the operations parameter to <set> and fill in any value you want to write to register 1. Press OK to execute the write command.

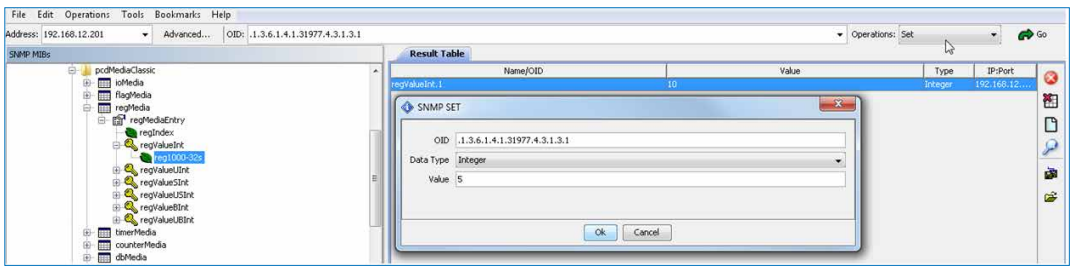


Figure 3.8: Set command in iReasoning

After setting the value, you can read it again to make sure it has worked.

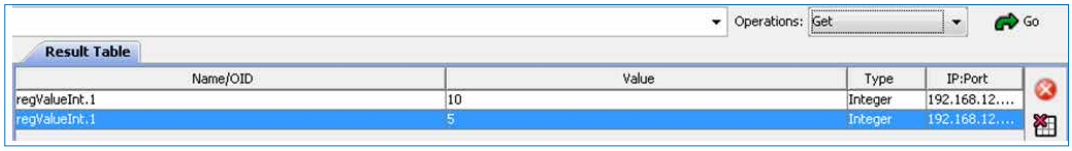
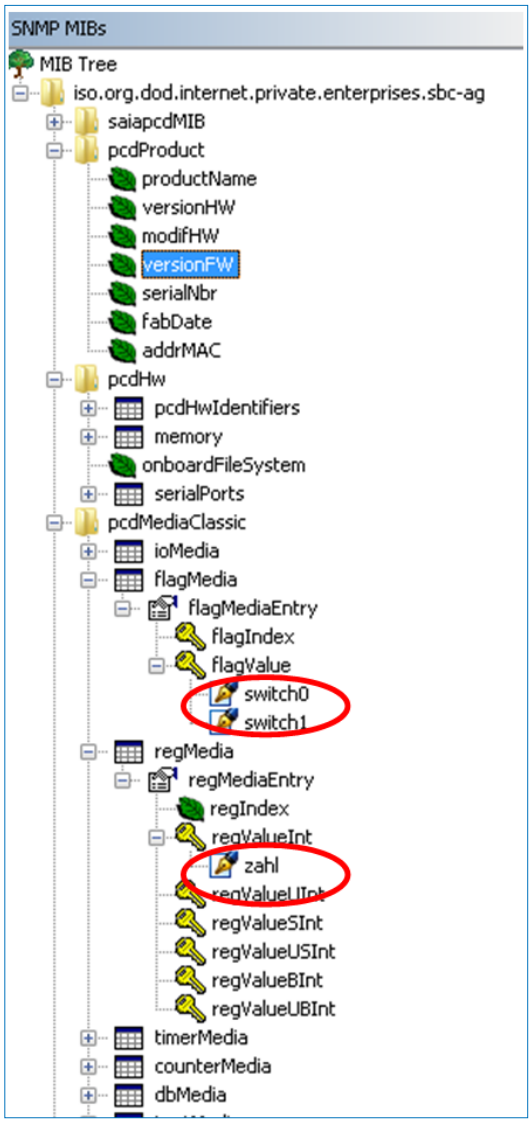


Figure 3.9: Check if the set command worked

3.4.6 Switching Symbols via SNMP

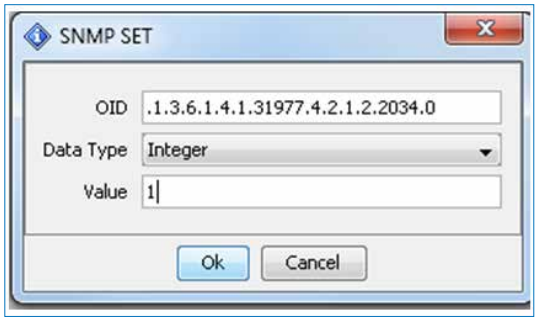
Open the MIB Browser and select the MIB file from your project folder. The Saia-MIB_Classic.mib must have been loaded in advance!

You will find the symbols in the related media table. In our example the flags switch0 and switch1 are flag values and the register number is a register value.



You can change the variables directly with the set command. Remember to change the write community in the advanced functions to the same settings as in

the device configurator. (Description in chapter 3.4.1)



After you have set the variable you can see the reaction directly in PG5.

Note: Not all data types in the symbol editor can be added to the MIB file. The data types will also be different to the types in PG5. Refer to the table below:

PG5 Data Type	SNMP Data Type
IO (Input/Output)	Integer32
F (Flag)	Integer32
C (Counter)	Integer32
R (Register)	Integer32
R FLOAT (Register float)	not supported by SNMP
TEXT RAM	not available
T (Timer)	Integer32
DB	Integer32 (For every Index of the DB a new Integer will be generated)

3.5 Comment

A large number of software products are on the market with an SNMP manager function. There are command-line oriented tools and small test tools with GUI. There are also some comprehensive and usually very costly software products. Many IT departments with quite large networks, servers and PCDs use powerful management software. This software includes the SNMP function together with other functions that serve to increase the availability of the systems. A large number of these products can be imported by an MIB file like the SaiaMIB_classic.mib file. Our tests to date have shown that the SNMP agent function has been implemented in accordance with the standard. Unfortunately, we do not have the capability to carry out tests with a large number of the SNMP management tools that are available. Tests have been conducted with the Windows version of the Net-Snmp tool and the free version of the iReasoning MIB browsing software.

3.6 Using SNMP Traps

3.6.1 Examples of the SNMP Trap function

Some important traps are sent by firmware even without the user program. If the Run/Stop switch changes over, a message is sent to the trap IP addresses (1-3) set in the Device Configurator. The automatically sent traps have a fixed ID from 1 to 5. User specific traps (see chapter 3.6.2) are sent with the ID 6.

To see these messages you can open the Trap Receiver in iReasoning.

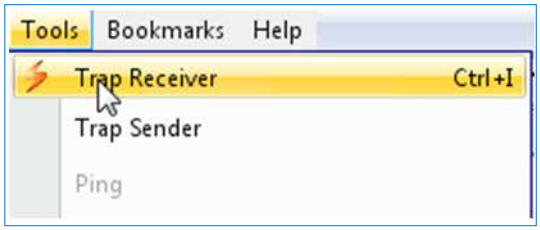
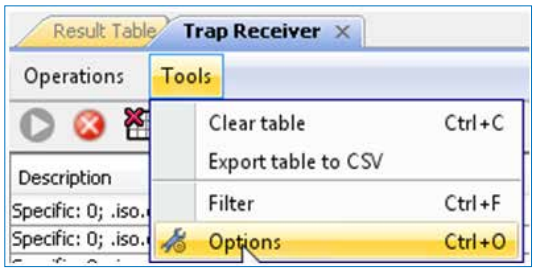


Figure 3.10: Trap receiver

Make sure that you have set the same port for the trap receiver as set in the device configurator. The standard is port 161.



3.11: Port Settings in Trap Receiver

An automatic trap will be generated when you change the state of the PCD with the Run/Stop Switch, for example. You receive a message in the trap window every time this is done. There's a big difference between the traps of the two SNMP versions v1 and v2c. You can see the difference in the following examples.

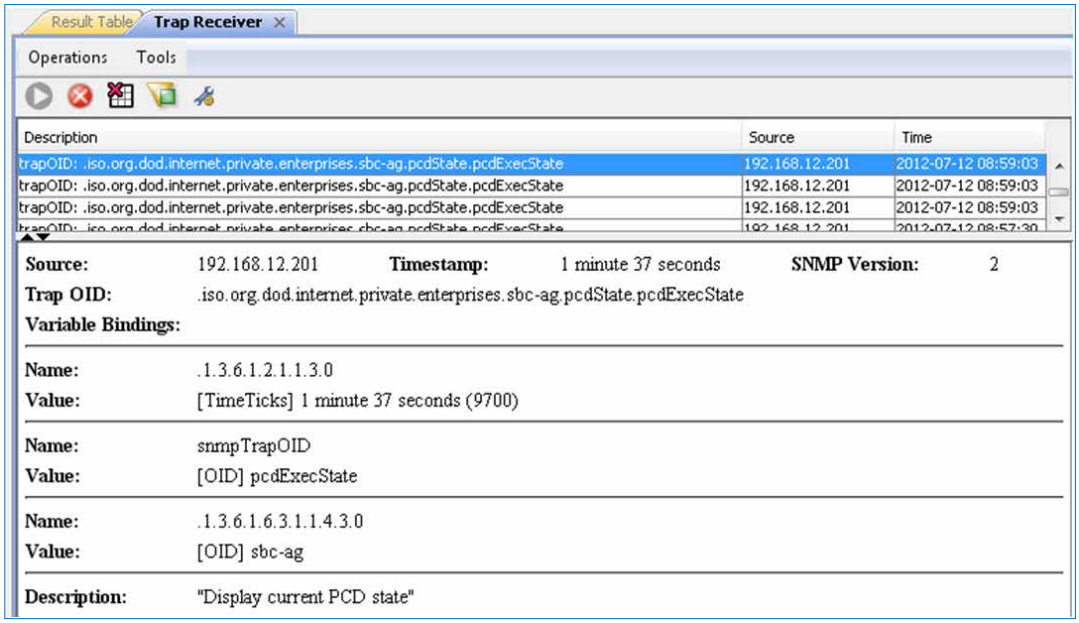
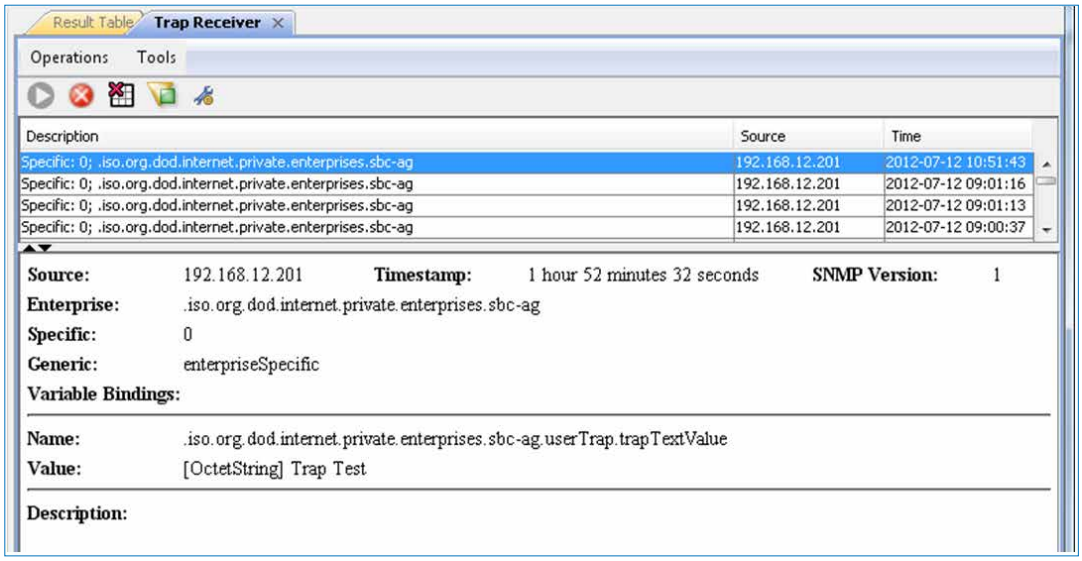


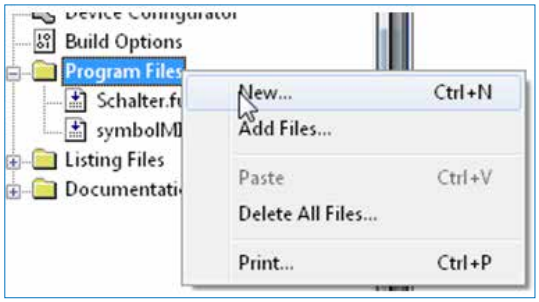
Figure 3.12: Automatically generated traps with SNMP v2c when the PCD state changes



3.13: Automatically generated traps with SNMP v1 when the PCD state changes

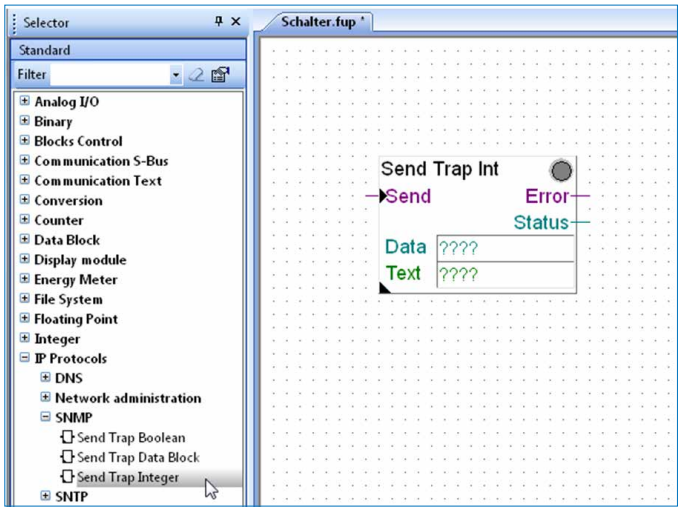
3.6.2 Creation of user specific traps with F-Boxes

There are three F-Boxes in PG5 which are available to create user specific traps. The user specific traps are always sent with the ID 6. Create or open a new fupla file in the project tree.



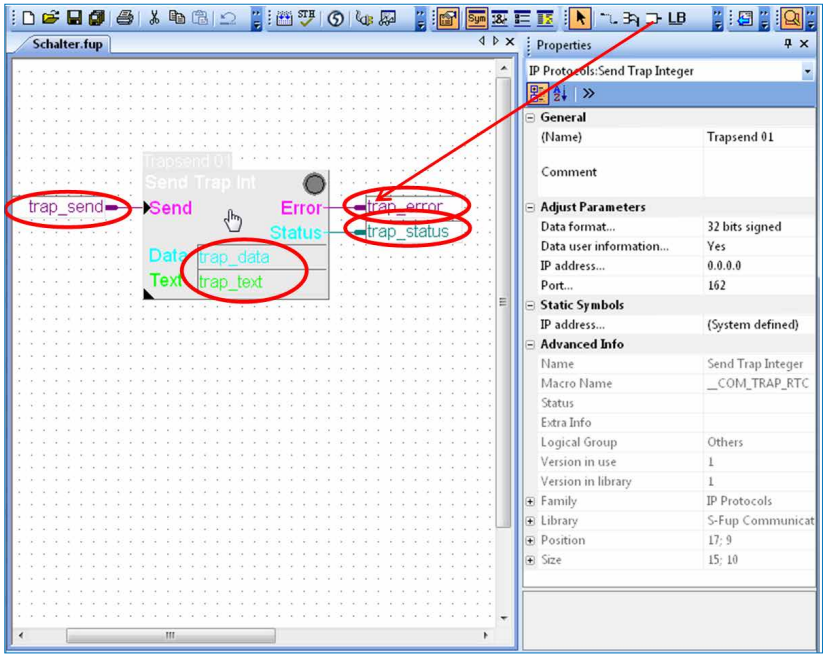
3.14: New fupla file for Send Trap F-Box

Open the fupla file and place the «Send Trap Integer» F-Box in the workspace. You can also send DB and boolean data. In this case you have to use the related F-Box.



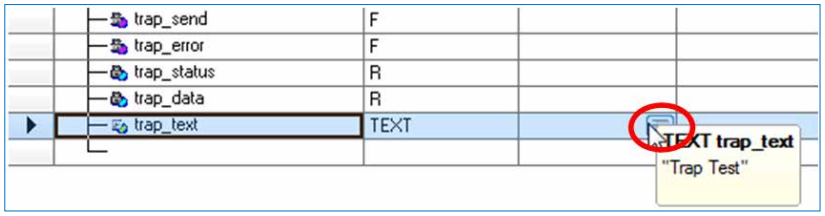
3.15: Insert the Send Trap F-Box

To adjust the parameters of the Send Trap Int F-Box you have to select it. When you set the IP ad-dress to 0.0.0.0 the trap will be sent automatically to the IP ad-dresses set in the device configurator. Add new variables to the inputs and outputs of the F-Box. Give the data and text field a variable name as well.



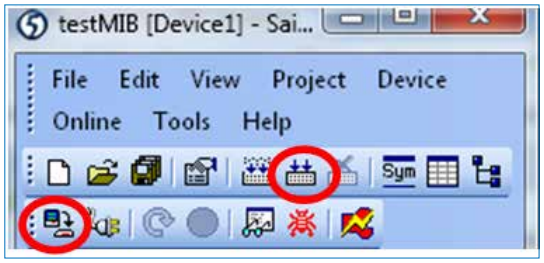
3.16: Send Trap Int F-Box

After you have made the new symbols, they will appear in the symbol editor (open with F5). Press the edit button of the trap_text symbol and edit the text you want to send with your trap (here: «Trap Test»).



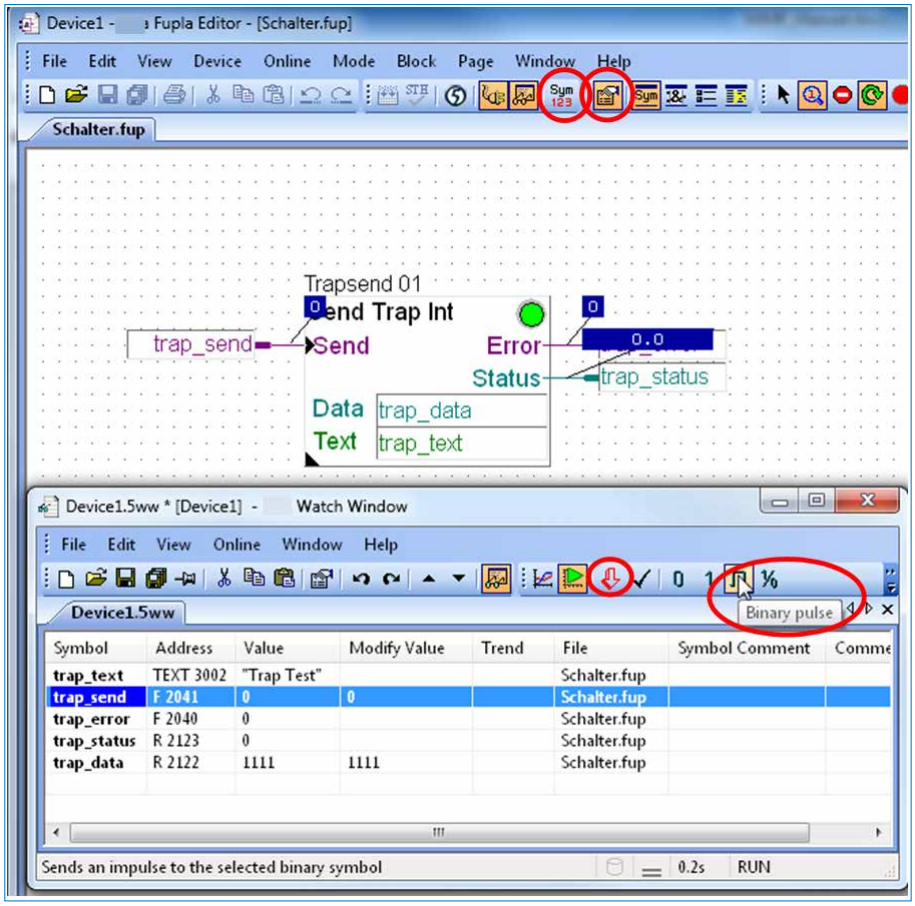
3.17: Edit the trap text

Compile and download the program in the project manager.



3.18: Compilation and download

Go online  and open the watch window  . Drag and drop the symbols from the symbol editor to the watch window.

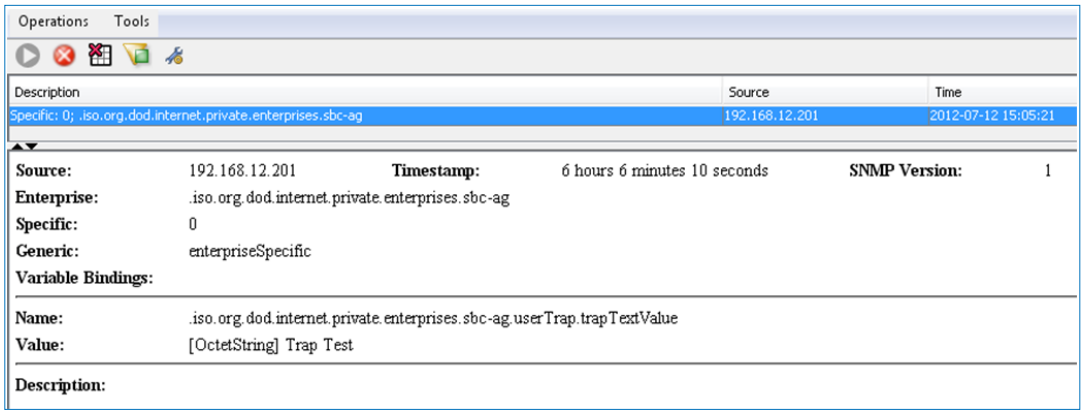


3.19: Trigger a trap with the watch window

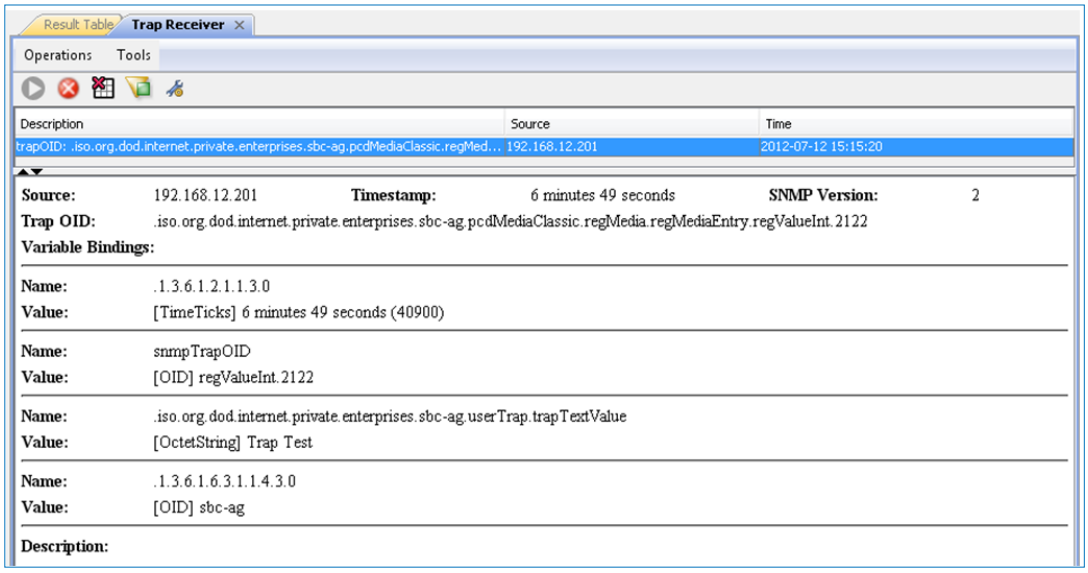
After you have done that you can modify the trap_data values in the «Modify Value» column. To download the new value you have to press the red download arrow. To trigger the trap select the strap_send row and give a binary pulse with the pulse button.

The maximum text size is 128 characters.

You will see the trap in your iReasoning MIB Browser. Depending on the SNMP version you will see a different message.



3.20: User trap with SNMP v1



3.21: User trap with SNMP v2c

4 Management Information Base MIB

4.1 Working with MIB

To standardise the data structures that SNMP compatible devices are intended to provide, MIB (management information base) tables have been introduced. All the resources that can be managed with SNMP are stored in one file. For a PCD these are all media (inputs/outputs, registers, flags, DBs, etc.) that can be accessed. In the MIB file, the programmer can limit the access to selected areas only. Within the SNMP implementation, the Saia PCD supports the default MIB-II structure.

→ The standard Saia PCD MIB file does not support symbol names.
→ The SNMP manager needs to know the fixed address range used.

The support of Symbol Names is in preparation.
The following standard MIB are included by default:

- ID = 1.3.6.1.2.1.1, system group
- ID = 1.3.6.1.2.1.2, interfaces group
- ID = 1.3.6.1.2.1.3, address translation group
- ID = 1.3.6.1.2.1.4, IP group
- ID = 1.3.6.1.2.1.5, ICMP group
- ID = 1.3.6.1.2.1.6, TCP group
- ID = 1.3.6.1.2.1.7, UDP group
- ID = 1.3.6.1.2.1.11, SNMP group

4.2 MIB-II Standard description

In preparation!

4.3 Net-SNMP installation with the command line

You can read and write the values as well from the command line. Therefore you first have to install pearl http://www.activestate.com/activeperl.

After that it is possible to install the command line tool Net-SNMP. Do not change the default installation folder while installing it! http://net-snmp.sourceforge.net/

The last step is to copy the SBC specific MIB files

- SaiaMIB_Classic.mib

into the folder C:\usr\share\snmp\mibs and rename the file extension from .mib to .txt

You can open the windows command line (cmd.exe) and read/write the values directly there.

```
C:\>snmpget -v2c -c public -m SaiaPCDClassic 192.168.12.201 companyName
SaiaPCDClassic::companyName = STRING: "Saia Burgess Controls AG"
```


4.4 Saia PCD MIB file structure

Using SNMP Management tools, e.g. NET-SNMP, the Saia PCD MIB can be accessed as «sbc-ag» which corresponds to the following ID

sbc-ag => 1.3.6.1.4.1.31977

The numbers of the ID has the following meaning:

iso(1) identified-organization(3) dod(6) internet(1) private(4) enterprise(1)

Further on the MIB contains the following items which will be described in the following chapters. Please note that you can either use the complete ID or the corresponding alias name <alias_name>.

ID	Alias
sbc-ag.1	<saiapcdMIB>
sbc-ag.2	<pcdProduct>
sbc-ag.3	<pcdHW>
sbc-ag.4	<pcdMediaClassic>
sbc-ag.5	<pcdAnyMedia>
sbc-ag.6	<pcdRtc >
sbc-ag.7	<pcdState>
sbc-ag.8	<pcdHistory>
sbc-ag.9	<userTrap>

4.4.1 saiapcdMIB

ID	Alias	Description
saiapcdMIB.3	<companyName>	Details of the company name
saiapcdMIB.4	<companyDescription>	Details of the company
saiapcdMIB.5	<companyURL>	URL of the company

All these fields are read-only fields.

Examples:

`snmpget -v2c -c public -m SaiaPCDClassic <ip> 1.3.6.1.4.1.31977.`**`1.3`**
will return the coded company name.

By using the aliases as defined in the SaiaMIB.txt, the above example can be reduced to
`snmpget -v2c -c public -m SaiaPCDClassic <ip> companyName`

4.4.2 pcdProduct

ID	Alias	Description
pcdProduct.1	<productName>	PCD product name
pcdProduct.2	<versionHW>	Platform HW version
pcdProduct.3	<modifHW>	Platform HW modification
pcdProduct.4	<versionFW>	Loaded FW version
pcdProduct.5	<serialNbr>	Platform serial number
pcdProduct.6	<fabDate>	Platform fabrication date [week/year]

All these fields are read-only fields.

Examples:
`snmpget -v2c -c public <ip> 1.3.6.1.4.1.31977.2.1`
will return the product name as retrieved from the PCD platform.

4.4.3 pcdHW

ID	Alias	Description
pcdHW.1	<identifiers>	This is a table of all found HW identifiers (base board, extension, FLASH devices and intelligent boards). The number of displayed items depends on the HW currently found
pcdHW.2	<memory>	This ID is NOT available anymore
pcdHW.3	<onboardFileSystem>	This indicates whether the on board FLASH has a file system or not
pcdHW.4	<serialPorts>	This is table of all the possible serial ports, with their names and their availability.

All these fields are read-only fields.

Examples:
`snmpwalk -v2c -c public <ip> 1.3.6.1.4.1.31977.3.1`
will return the list of attached known PCD parts, with the corresponding identifier, name and location on the PCD device.

4.4.4 pcdMediaClassic

ID	Alias	Description
pcdMediaClassic.1 (or sbc-ag.4.1)	<ioMedia>	This is a table allowing access to the input/output values at a specified index. The access is carried out with the following identifiers: .1 (table) .2 <ioValue> .x (IO address): returns accessed IO value
pcdMediaClassic.2 (or sbc-ag.4.2)	<flagMedia>	This is a table allowing access to the flag values at a specified index. The access is carried out with the following identifiers: .1 (table) .2 <flagValue> .x (flag ad-dress): returns accessed flag value

Examples:
`snmpget -v2c -c public <ip> 1.3.6.1.4.1.31977.4.2.1.2.0`
will return the integer value of flag 0.

By using the aliases as defined in the SaiaPCDMIB.txt, the above example can be reduced to
`snmpget -v2c -c public -m SaiaPCDClassic <ip> flagValue.0`

ID	Alias	Description
pcdMediaClassic.3	<regMedia>	This is a table allowing access to the register values at a specified index. The access is carried out with the following identifiers .1 (table) .3 <regValueInt> .x (register address): returns accessed register value as 32 bits signed integer .1 (table) .4 <regValueUInt> .x (register address): returns accessed register value as 32 bits unsigned integer .1 (table) .5 <regValueSInt> .x (register address): returns accessed register value as 16 bits signed integer .1 (table) .6 <regValueUSInt> .x (register address): returns accessed register value as 16 bits unsigned integer .1 (table) .7 <regValueBInt> .x (register address): returns accessed register value as 8 bits signed integer .1 (table) .8 <regValueUBInt> .x (register address): returns accessed register value as 8 bits unsigned integer.

Examples:

```
snmpget -v2c -c public <ip> 1.3.6.1.4.1.31977.4.3.1.2.0
```
will return the integer value of register 0. By using the aliases as defined in the SaiaMIB.txt, the above example can be reduced to
```
snmpget -v2c -c public -m SaiaPCDClassic <ip> regValueInt.0
```

ID	Alias	Description
pcdMediaClassic.4	<timerMedia>	This is a table allowing access to the timer values at a specified index. The access is carried out with the following identifiers: .1 (table) .2 <timerValue> .x (timer address): returns accessed timer value
pcdMediaClassic.5	<counterMedia>	1 (table) .2 <counterValue> .x (counter address): returns accessed counter value
pcdMediaClassic.6	<dbMedia>	This is a table allowing access to the DB values at a specified offset. The access is carried out with the following identifiers: 1 (table) .3 <dbValueInt> .x (DB Nbr) .y (DB offset): returns the value of accessed DB(x) at given offset(y) as 32 bits signed integer. 1 (table) .4 <dbValueUInt> .x (DB Nbr) .y (DB offset): returns the value of accessed DB(x) at given offset(y) as 32 bits unsigned integer 1 (table) .5 <dbValueSInt> .x (DB Nbr) .y (DB offset): returns the value of accessed DB(x) at given offset(y) as 16 bits signed integer 1 (table) .6 <dbValueUSInt> .x (DB Nbr) .y (DB offset): returns the value of accessed DB(x) at given offset(y) as 16 bits unsigned integer 1 (table) .7 <dbValueBInt> .x (DB Nbr) .y (DB offset): returns the value of accessed DB(x) at given offset(y) as 8 bits signed integer 1 (table) .8 <dbValueUBInt> .x (DB Nbr) .y (DB offset): returns the value of accessed DB(x) at given offset(y) as 8 bits unsigned integer.

Examples:

```
snmpget -v2c -c public <ip> 1.3.6.1.4.1.31977.4.6.1.3.1.0
```
will return the signed integer value of DB 1 at offset 0.

By using the aliases as defined in the SaiaMIB.txt, the above example can be reduced to
```
snmpget -v2c -c public -m SaiaPCDClassic <ip> dbValueInt.1.0
```

4.4.5 pcdAnyMedia

With this, it is possible to access any media available on the PCD.

ID	Alias	Description
pcdAnyMedia.1	<mediaType>	Writing this allows the selection of which media area is accessed Writing a 1 allows access to the IO media area Writing a 2 allows access to the flag media area Writing a 3 allows access to the register media area Writing a 4 allows access to the timer media area Writing a 5 allows access to the counter media area Writing a 6 allows access to the db media area
pcdAnyMedia.2	<mediaIdentifier>	Writing this allows the media address to be defined
pcdAnyMedia.3	<mediaIndex>	For DB, writing this allows the offset within the media number to be defined
pcdAnyMedia.4	<mediaValueInt>	After defining the media type, the media identifier and eventually the media index, this allows access to the related value as 32 bits signed
pcdAnyMedia.5	<mediaValueUInt>	After defining the media type, the media identifier and eventually the media index, this allows access to the related value as 32 bits unsigned.
pcdAnyMedia.6	<mediaValueSInt>	After defining the media type, the media identifier and eventually the media index, this allows access to the related value as 16 bits signed
pcdAnyMedia.7	<mediaValueUSInt>	After defining the media type, the media identifier and eventually the media index, this allows access to the related value as 16 bits unsigned
pcdAnyMedia.8	<mediaValueBInt>	After defining the media type, the media identifier and eventually the media index, this allows access to the related value as 8 bits signed
pcdAnyMedia.9	<mediaValueUBInt>	After defining the media type, the media identifier and eventually the media index, this allows access to the related value as 8 bits unsigned.

The same limitation exists concerning the read and the write media range as for individual media.

Examples:

```
snmpset -v2c -c private -m SaiaPCDClassic <ip> mediaType i 1 # IO range
snmpset -v2c -c private -m SaiaPCDClassic <ip> mediaIdentifer i 0 # IO 0
snmpset -v2c -c private -m SaiaPCDClassic <ip> mediaIndex i 0
snmpset -v2c -c private -m SaiaPCDClassic <ip> mediaValueInt i 1 # Write 1
snmpget -v2c -c public -m SaiaPCDClassic <ip> mediaValueInt # Read
```

Remark: In order to be able to write output 1, it is necessary to configure to write access to IO range. Read is by default allowed over the whole IO range.

4.4.6 pcdRtc

With this, it is possible to access the RTC (Real Time Clock) of the system. All fields are read only.

ID	Alias	Description
pcdRtc.4	<rtcWeekNumber>	Displayed as an integer, it gives the current week number
pcdRtc.5	<rtcDayOfWeek>	Displayed as an integer, it gives the current day of the week
pcdRtc.6	<rtctimeTick>	Defined as «timeticks», it returns the number of 100th of seconds since the beginning of the current year
pcdRtc.7	<rtcDT>	Display the current data and time as string in the form «YY- MM-DD hh:mm:ss:ms». All these fields are read-only fields.

Examples:

`snmpget -v2c -c public <ip> 1.3.6.1.4.1.31977.6.7`
will return the date and time as a readable string.

By using the aliases as defined in the SaiaMIB.txt, the above example can be reduced to
`snmpget -v2c -c public -m SaiaPCDClassic <ip> rtcDT`

4.4.7 pcdState

With this, it is possible to access the current PCD states. All fields are read only.

ID	Alias	Description
pcdState.1	<pcdExecState>	Returns the current execution state of the PCD. The following values are returned: ● 1: PCD is in RUN ● 2: PCD is in conditional RUN ● 3: PCD is in STOP ● 4: PCD is in HALT ● 5: PCD is in HALT error ● Other states can be sent in specific cases. This MIB variable is also used when sending «Life Trap» and/or «State Trap»
pcdState.2	<pcdBattState>	The current state of the battery. The following values are returned: ● -1: Battery is present but fails ● 0: Battery present and OK ● 1: Battery module not present
pcdState.3	<pcdSwitchState>	Returns the current position of the PCD switch. The following values are returned: ● 0: switch is in Stop position ● 1: switch is in Run position

Examples:

```
snmpget -v2c -c public <ip> 1.3.6.1.4.1.31977.7.1
```
will return the current PCD state.

By using the aliases as defined in the SaiaMIB.txt, the above example can be reduced to
```
snmpget -v2c -c public -m SaiaPCDClassic <ip> pcdExecState
```

4.4.8 pcdHistory

With this, it is possible to access the PCD history. All fields are read only. It is accessed in a table; each entry has the following information:

ID	Alias	Description
pcdHistory.1 (table).1	<pcdHistoryIndex>	.x (line id): This returns the line id
pcdHistory.1 (table).2	<pcdHistoryRTC>	.x (line id): This returns the RTC (YY-MM-DD hh:mm:ss:ms) entry of the line id
pcdHistory.1 (table).3	<pcdHistoryLine>	.x (line id): This returns the history entry text of the line id.

Examples:

```
snmpwalk -v2c -c public <ip> 1.3.6.1.4.1.31977.8
```
will return all history entry lines, listing first all indexes, then RTC as string then history text.

By using the aliases as defined in the SaiaMIB.txt, the above example can be reduced to
```
snmpwalk -v2c -c public -m SaiaPCDClassic <ip> pcdHistory
```

5 SNMP diagnosis through Web CGI

5.1 Access syntax

All SNMP configuration tags can be accessed through the Web CGI interface. The access has the following syntax

Read values:

http://hostname/cgi-bin/readVal.exe?<ConfigRegistry>,<TagName>

ConfigRegistry	CFG-SNMP, SYS-SNMP
TagName	Corresponds to the configuration tag in the tag table.

Example:

http://192.168.12.201/cgi-bin/readVal.exe?CFG-SNMP,StartDelay

→ gives the start delay value

http://192.168.12.201/cgi-bin/writeVal.exe?CFG-SNMP,StartDelay+3

→ writes the value 3 to StartDelay

5.2 SNMP Tag list

Web CGI Bin					
Tag Name	**Acc.**	**Type**	**Def.**	**Min/ Max**	**Description**
Enable	RW	CFG	0	NA	Enable (1) or disable (0) SNMP functionalities
UseV3	RW	CFG	0	NA	Enable (1) or disable (0) SNMP V3 functionality Actually, only V2 is supported. Setting this flag has no influence.
StartDelay	RW	CFG	5	0/60	Defines the time when the SNMP agent is started at power ON. Time is required in order to allow the PCD to set the IP configuration before the SNMP agent is started. If the start delay it too short, it is possible that the cold start trap event cannot be sent. Setting it to 0 allows SNMP to be started immediately when the IP configuration has been parsed.
IOReadFirst	RW	CFG	0	2^{31} - 1	Defines the first input/output address that can be accessed with an SNMP GET/ GETNEXT request. Below the given value, the request returns with an error.
IOReadLast	RW	CFG	1024	2^{31} - 1	Defines the first input/output address that can NOT be accessed with an SNMP GET/ GETNEXT request. Below the given value, but higher or equal to the IOReadFirst, requests will be answered without error. If Last and First are equal, no access is granted.
IOWriteFirst	RW	CFG	0	2^{31} - 1	Defines the first input/output address that can be accessed with an SNMP SET/ CHECK request. Below the given value, the request returns with an error.
IOWriteLast	RW	CFG	0	2^{31} - 1	Defines the first input/output address that can NOT be accessed with an SNMP SET/CHECK request. Below the given value, but higher or equal to the IOWrite-First, requests will be answered without error. If Last and First are equal, no access is granted.
FlagRead-First	RW	CFG	0	2^{31} - 1	Defines the first flag address that can be ac- cessed with an SNMP GET/GET-NEXT request. Below the given value, the request returns with an error.

Web CGI Bin					
Tag Name	**Acc.**	**Type**	**Def.**	**Min/ Max**	**Description**
FlagRead-Last	RW	CFG	8192	2^{31} - 1	Defines the first flag address that can NOT be accessed with an SNMP GET/ GETNEXT request. Below the given value, but higher or equal to the FlagReadFirst, requests will be answered without error. If Last and First are equal, no access is granted.
FlagWrite-First	RW	CFG	0	2^{31} - 1	Defines the first flag address that can be accessed with an SNMP SET/CHECK request. Below the given value, the request returns with an error.
FlagWrite-Last	RW	CFG	0	2^{31} - 1	Defines the first flag address that can NOT be accessed with an SNMP SET/ CHECK request. Below the given value, but higher or equal to the FlagWriteFirst, requests will be answered without error. If Last and First are equal, no access is granted.
RegRead-First	RW	CFG	0	2^{31} - 1	Defines the first register address that can be accessed with an SNMP GET/GET-NEXT request. Below the given value, the request returns with an error.
RegRead-Last	RW	CFG	16364	2^{31} - 1	Defines the first register address that can NOT be accessed with an SNMP GET/ GETNEXT request. Below the given value, but higher or equal to the RegReadFirst, requests will be answered without error. If Last and First are equal, no access is granted.
RegWrite-First	RW	CFG	0	2^{31} - 1	Defines the first register address that can be accessed with an SNMP SET/CHECK request. Below the given value, the request returns with an error.
RegWrite-Last	RW	CFG	0	2^{31} - 1	Defines the first register address that can NOT be accessed with an SNMP SET/ CHECK request. Below the given value, but higher or equal to the RegWriteFirst, requests will be answered without error. If Last and First are equal, no access is granted.
TimerRead-First	RW	CFG	0	2^{31} - 1	Defines the first timer address that can be ac- cessed with an SNMP GET/GET-NEXT request. Below the given value, the request returns with an error.

Web CGI Bin					
Tag Name	**Acc.**	**Type**	**Def.**	**Min/ Max**	**Description**
TimerRead-Last	RW	CFG	32	2^{31} - 1	Defines the first timer address that can NOT be accessed with an SNMP GET/ GETNEXT request. Below the given value, but higher or equal to the Timer-ReadFirst, requests will be answered without error. If Last and First are equal, no access is granted.
TimerWrite-First	RW	CFG	0	2^{31} - 1	Defines the first timer address that can be accessed with an SNMP SET/CHECK request. Below the given value, the request returns with an error.
TimerWrite-Last	RW	CFG	0	2^{31} - 1	Defines the first timer address that can NOT be accessed with an SNMP SET/ CHECK request. Below the given value, but higher or equal to the TimerWrite-First, requests will be answered without error. If Last and First are equal, no access is granted.
Counter-ReadFirst	RW	CFG	32	2^{31} - 1	Defines the first counter address that can be accessed with an SNMP GET/GET-NEXT request. Below the given value, the request returns with an error.
Counter-ReadLast	RW	CFG	1600	2^{31} - 1	Defines the first counter address that can NOT be accessed with an SNMP GET/ GETNEXT request. Below the given value, but higher or equal to the CounterReadFirst, requests will be answered without error. If Last and First are equal, no access is granted.
Counter-WriteFirst	RW	CFG	0	2^{31} - 1	Defines the first counter address that can be accessed with an SNMP SET/CHECK request. Below the given value, the request returns with an error.
Counter-WriteLast	RW	CFG	0	2^{31} - 1	Defines the first counter address that can NOT be accessed with an SNMP SET/ CHECK request. Below the given value, but higher or equal to the CounterWrite-First, requests will be answered without error. If Last and First are equal, no access is granted.
DBReadFirst	RW	CFG	0	2^{31} - 1	Defines the first DB number that can be accessed with an SNMP GET/GETNEXT request. Below the given value, the request returns with an error. When a DB is accessible, all elements within the DB can be accessed.

Web CGI Bin					
Tag Name	**Acc.**	**Type**	**Def.**	**Min/ Max**	**Description**
DBReadLast	RW	CFG	8192	2^{31} - 1	Defines the first DB number that can NOT be accessed with an SNMP GET/ GETNEXT request. Below the given value, but higher or equal to the DBRead-First, requests will be answered without error. If Last and First are equal, no access is granted. Once a DB has been defined for read access, the complete DB can be read.
DBWriteFirst	RW	CFG	0	2^{31} - 1	Defines the first DB number that can be ac- cessed with an SNMP SET/CHECK request. Below the given value, the request returns with an error. When a DB is accessible, all elements within the DB can be accessed.
DBWriteLast	RW	CFG	0	2^{31} - 1	Defines the first DB number that can NOT be accessed with an SNMP SET/ CHECK request. Below the given value, but higher or equal to the DBWriteFirst, requests will be answered without error. If Last and First are equal, no access is granted. Once a DB has been defined for write access, the complete DB can be written.
ReadCom-munity	RW	CFG	«public»	Max. 24 char.	Defines the string used in SNMP V2 to access (read commands e.g. GET/GET-NEXT) on board objects.
WriteCom-munity	RW	CFG	«private»	Max. 24 char.	Defines the string used in SNMP V2 to access (write commands e.g. SET) on board objects.
TrapCommu-nity	RW	CFG	«public»	Max. 24 char.	Defines the string used when Trap is sent to the SNMP Manager by the agent.
sysContact	RW	CFG	«Saia Burgess Controls AG»	Max. 100 char.	Defines the string displayed when ac-cessing the default SNMP object sys-Contact (defined in SNMPv2-MIB)
sysLocation	RW	CFG	«CH-3280 Murten»	Max. 100 char.	Defines the string displayed when ac-cessing the default SNMP object sysLo-cation (defined in SNMPv1-MIB)
TrapxPort	RW	CFG	0	65535	Up to three SNMP trap receivers can be defined. The x is replaced by a, b or c. The port defines the IP port defined by the receiver. Setting 0 implies the use of the default port, normally 162.

Web CGI Bin					
Tag Name	**Acc.**	**Type**	**Def.**	**Min/ Max**	**Description**
TrapxIPAddr	RW	CFG	0.0.0.0	NA	Up to three SNMP trap receivers can be defined. The x is replaced by a, b or c. The IP address defines the IP address of the receiver. Setting 0 implies that no receiver is defined for this trap entry.
LifeTimeout	RW	CFG	0	1 hrs	Expressed in millisecond value, it defines the time between two «Life Traps» sent to the configured managers. Setting this variable to 0 disables the sending of «life trap» message.

6 Special remarks

SNMP is using the UDP protocol for sending traps. This is NOT reliable; message can be discarded without network acknowledgment. Returning success in the CSF or F-Box calls does NOT mean that the trap has been received / processed by the manager. It just means that the request has been issued on the network by the PCD.

This has to be taken into account while preparing a user program. Repetitions / acknowledgment mechanism shall be planned between the user program and the manager, e.g. the manager writes a value to the trap provided OID.

A Appendix

A.1 Icons

This symbol refers to additional information, which is available in this or another manual or in technical documentation on this subject. There are not direct references to such documents.

This symbol designates instructions, which need to be strictly followed.

A.2 Contact

Saia-Burgess Controls AG
Bahnhofstrasse 18
3280 Murten, Switzerland

Phone central ..... +41 26 580 30 00
Phone Saia-PCD Support ..... +41 26 580 31 00
Fax ..... +41 26 580 34 99

Email support: ..... support@saia-pcd.com
Supportsite: ..... www.sbc-support.com
SBC site: ..... www.saia-pcd.com

International Represetatives &
SBC Sales Companies: ..... www.saia-pcd.com/contact

Postal address for returns from customers of the Swiss Sales office

Saia-Burgess Controls AG
Service Après-Vente
Bahnhofstrasse 18
3280 Murten, Switzerland