Saia Visi.Plus

Version 1.7

Saia Visi.Plus

User Manual

by Saia-Burgess Controls AG

Saia Visi.Plus is a process control and visualisation system for the areas of building, traffic and industry

Saia Visi.Plus

Printed: Juli 2017 in Belp, Switzerland

Publisher

MST Systemtechnik AG

Managing Editor

Christoph Müller

Technical Editors

Thomas Niklaus
Lukas Dillier
Zdenek Sulc
Gaby Heyde
Christoph Müller

Team Coordinator

Christoph Müller

Special thanks to:

All the people who contributed to this document.

Content

Preposition

Saia Visi.Plus

Introduction

Chapter 1

1 Introduction

Visi.Plus is a process control system which is successfully applied in the following areas:

- Building control
- Traffic engineering
- Machine construction visualisation systems

The system supports several different PLC systems, but is of course designed for the PLC series from SBC. Currently, other systems can only be connected through an OPC server. The engineering is designed for the PLC controllers of Saia Burgess Controls in Murten.

1.1 Summary

Tips for reading this manual are contained in this chapter. It explains where additional information can be found, how to use the manual and what symbols and conventions are applied within it.

If specific expressions in this manual are not understood, there is a glossary at the end of the manual for consultation.

1.2 Information

Information about Visi.Plusis found in:

- The **Visi.Plus Manual**
  It contains the most important answers and explanations to questions in connection with Visi.Plus.

 The **Help** menu can be accessed at any time with the **<F1>** key or via the **Help buttons**

- **Online**
  The following addresses provide information about: product descriptions, news, updates, links, support addresses, course data and much more.

www.sbc-support.com

Version: 1.7
Date: 11.07.2017

1.3 Using the manual

This manual is written for users who are already familiar with the basic functions of a personal computer. For those interested in being able to create Visi.Plus applications more

quickly, we recommend the course offered by your respective SBC representative. The course dates are posted on the internet and available from a Visi.Plus representative by telephone.

Experienced Visi.Plus users can use this manual as a reference book.

Visi.Plus is continually being modified to respond to logical requirements. Therefore we recommend staying up to date through the website.

1.4 Structure of the manual

The eleven chapters were designed so that the desired information can be found as efficiently as possible. These chapters explain the key capabilities, basic functions and the concept of Visi.Plus and clarify with the use of practical exercises.

Visi.Plus manual and Visi.Plus help
The introduction to the manual and to Visi.Plus help

Working with Visi.Plus
Overview of general functions and applications of Visi.Plus

Installation
Instructions for the installation of Visi.Plus

Engineering
A guide for engineering a project with the help of Visi.Plus

System start
Instructions for starting up Visi.Plus

Visi.Plus Engineering Tool (PET.exe)
Operating instructions for the PET

Visi.Plus databases
Instructions for the Visi.Plus databases:

- Data Management System (DMS.exe) incl. control functions
- Longterm database (PDBS.exe)

Visi.Plus utilities
Instructions for the Visi.Plus utilities:

- Alarm Manager (AlmMng.exe)
- Remote alarms (MalmMng.exe)
- Historical data (HDAMng.exe)
- Log manager (prtmng.exe)
- Time switch program (clkmng.exe)

- WebServer (PWEB.exe)
- FTP server (pFTP.exe)
- Cyclical data logging (Logger.exe)

Visi.Plus communications driver

Instructions for the communications drivers supported by Visi.Plus:

- Saia PCD® (SDriver.exe)
- OPC client (OPCDriver.exe)
- ESPA client ESPADriver
- SNMP client SNMPDriver

Visi.Plus operating programs

Instructions for the Visi.Plus operating programs:

- Graphical editor (GE.exe)
- Alarm Viewer (AlmView.exe)
- Log viewer (PrtvVew.exe)
- Listen viewer (pList.exe)
- Time switch program (ClkCfg.exe)
- Chart program (pChart.exe)
- User management (pUser.exe)
- Change password (changepwd.exe)
- Remote alarm program (MalmCfg.exe)
- ASCII export (AsciiExport.exe)
- Text editor (pEdit.exe)
- Login (plogin.exe)

Visi.Plus help programs

Instructions for the Visi.Plus help programs:

- Search for and open recipes (PMoSFilePicker.exe)
- Log formatting (prtformat.exe)
- Set DMS value (SetDMSVal.exe)
- Data backup (pBackup.exe)
- Drawing program (Ppaint.exe)
- Convert images (bmp2jpg.exe)
- Show version (ShowVersion.exe)
- Quit system (pStop.exe)

Additional information and glossary

General information that is not assigned to any chapter as well as a comprehensive glossary with the terms used in Visi.Plus can be found here.

1.5 Symbols and conventions used

Symbols indicate text segments that are either important, helpful or contain other additional information. The individual symbols are described in the following:

Information

A section of text that contains additional information for an alternative means of achieving a particular goal.
This is so you can work faster and more efficiently.

Attention!

A section of text with important information for data and system security. Heed this information in your own interest.

Along with the chapters and sections, the dialog and other control elements of Visi.Plus are printed in **bold**.

Commands and menu functions are accentuated by quotation marks **" "**.
Example: "**Save**"

Key combinations and buttons are visually accentuated by angle brackets **<>**.
Example: **<CTRL+E>**

1.6 Using the manual

In addition to the operator's manual, which is available in PDF format, Visi.Plus allows for the documentation of the respective module to be called up directly.

For example if you are in the DMS and press the **<F1>** key, the help file is opened and jumps to the chapter on the DMS.

This takes place either through the help menu, the help button or by pressing the **<F1>** key.

Saia Visi.Plus

Working with Visi.Plus

Chapter 2

2 Working with Visi.Plus

2.1 Summary

The following subchapters include information about the philosophy, function, fields of use, hardware requirements, network environment and method of operation of Visi.Plus.

2.2 What is Visi.Plus ?

Visi.Plus is intended to represent operational data in a descriptive, simple and directly manageable form. The widest range of industries across the world have already taken advantage of the benefits Visi.Plus provides.

It is developed by MST Systemtechnik AG in Belp, Switzerland, and finds practical application in the widest range of systems. In the process, it is ensured that the functions are also tested in the respective environment.
Visi.Plus is primarily designed for the PLC controllers of Saia-Burgess Controls AG, based in Murten, Switzerland. MST Systemtechnik AG has approximately 100 man years of experience with the Saia PCD® systems and programs several hundred Saia PCD® controllers each year, primarily with Visi.Plus.

The uses recommended in this manual and the concept, which differs somewhat compared to other visualisation systems on the market, make it possible to achieve efficient project planning and programming in the areas of industrial and building automation. The performance which it delivers, combined with an affordable price, helps little by little in the creation of more cost-effective systems. The modular concept provides additional help in the optimisation of costs and features.

List of some features:
- Acquisition of process data and its storage as historical data
- Representation of process data in graphical form
- Fault monitoring, remote alarms
- Logging
- Evaluations

Possible areas of application include:
- Monitoring and safety systems
- Heating, ventilation, air conditioning (HVAC)
- Water treatment systems
- Production systems
- Tunnel control systems
- Transport systems
- Railway control systems

With the subdivided system architecture, large-scale systems can also be realised (by combining several individual systems).

2.3 Which functions does Visi.Plus offer?

Visi.Plus provides completely interactive visualisation, operation and monitoring at an extremely attractive price. The system is expandable and open for applications as well as network-capable with one hundred percent data transparency.
The transparent network capability of the operating system is fully utilised by Visi.Plus. By integrating multiple systems, computing power can be multiplied. Additional operator stations can be integrated into the system at any time.

Visi.Plus is based on recognised standards and has the following principal features:

- Outstanding range of functionality
- Distributed client/server processing
- Object-oriented vector graphics / bitmap-oriented graphical editor
- Interface to other Windows applications
- Online help/forum
- Scalable, modular architecture
- System capable of multi-tasking
- Compatible with WINDOWS® 2000/XP/Vista operating systems
- Efficient database access
- Object-oriented graphical user interface
- Network-oriented architecture
- Innovative system function
- Object-oriented data structure
- Integration with PG5 (Saia-Burgess Controls AG)
- Efficient project creation
- Automatic generation of PLC code (Saia PCD®)
- Programming interface Visual C++, Visual Basic, Java or Delphi

The architecture used offers clear interfaces. Projects can be realised reliably and economically. Applications are simplified in their engineering with Visi.Plus.

2.4 Where can Visi.Plus be used?

Thanks to its modular design, Visi.Plus can be utilised for projects both large and small. Visi.Plus takes your particular requirements into consideration. The modularity additionally provides system engineers with optimal integration into system architecture.

Anywhere from only a few to tens of thousands of physical data points can be logged, visualised and monitored. Multiple PLC drivers can be connected to various PC locations and combined to form a entire network.

2.5 What types of communications are supported by Visi.Plus ?

Visi.Plus can be operated on just one PC or over a network. Communication between the individual modules takes place either within the same PC or over the network. The network can be an ethernet network or one with a complex structure, encompassing multiple modems/routers, etc.

2.6 Performance characteristics and technical data

Visi.Plus has the following **Performance characteristics**:

- Up to 2,000,000 data points can be managed in the DMS (Visi.Plus Data Management System)
- More than 30,000 value changes per second.
- More than 50,000 reads per second.
- Any number of process diagrams (depending on free hard disk space)
- Up to 100 control stations (GE) possible.
- Full web operability.
- Trend data logging and access capability for any length of time
- Up to 700'000 alarms on a month can be managed.
- Unlimited number of remote alarms can be defined
- Unlimited number of users can be defined, subdivided into user levels and system levels

Performance characteristics for the BAC driver:
" 100 BACnet devices
" Up to max. 4000 BACnet objects per device
DMS performance characteristics must also be taken into account here!
A BACnet object can easily require 50 DMS data points!

Access times in the Visi.Plus data management system (DMS) are not affected by the amount of data in the database. Depending on hardware, the DMS can receive between 800 and 120,000 accesses per second. The lower limit relates to accesses via a network and the upper limit to local accesses from the same PC.
The deciding factor for short access times is sufficient memory (RAM) in the PC on which the DMS is installed. All process data (process image and all configuration data) is stored in RAM for performance-related reasons. The outsourcing of individual memory areas to the hard drive must be avoided. Speed can decrease 100-fold when insufficient memory is available.

We recommend the following RAM configurations (in addition to what the operating system needs):

Number of DMS data points	Speicherbedarf RAM
Up to 10,000 data points	At least 256 MB
Up to 100,000 data points	At least 512 MB
Up to 300,000 data points	At least 1,024 MB
Up to 2,000,000 data points	At least 2 GB

Space for bitmap and JPG images must also be factored in.

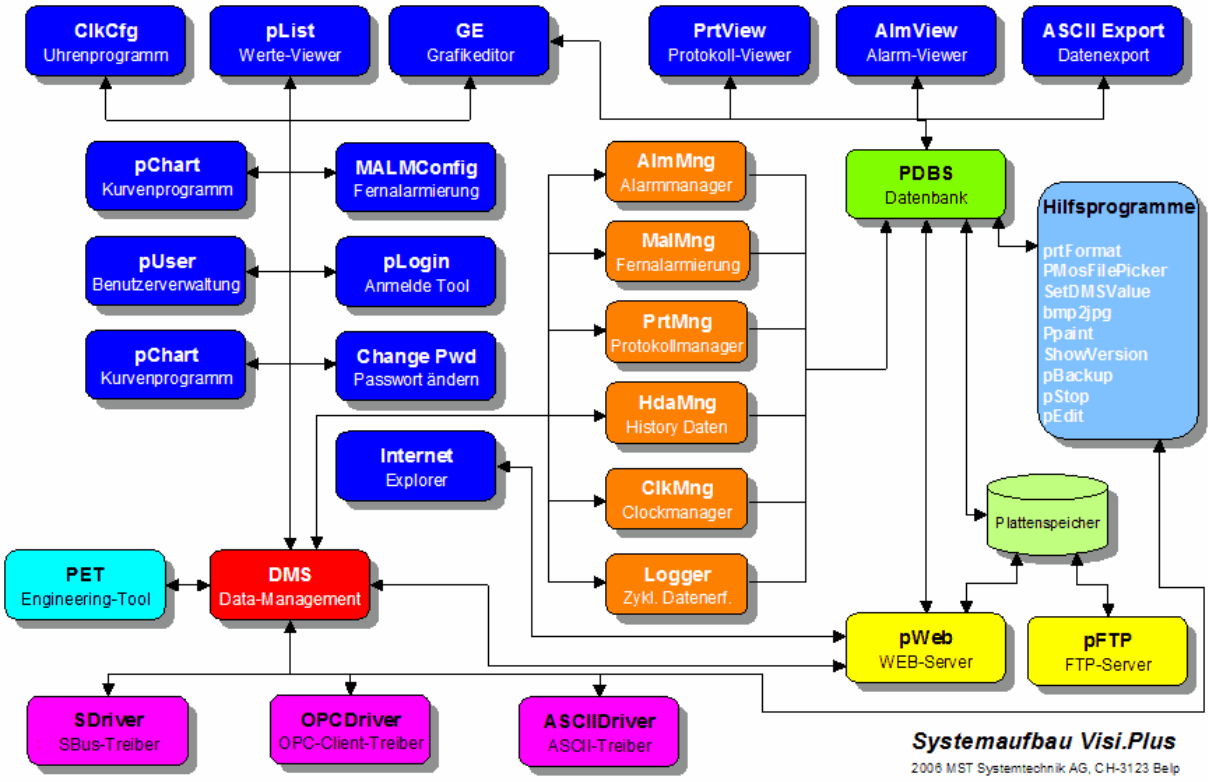
2.7 System architecture

Visi.Plus is not a single program, but a collection of many individual programs, each of which fulfils its assigned task. This provides a great deal of flexibility, resulting in significant benefits: Thanks to its individual programs, the stability of Visi.Plus is increased.

The system structure is distributed. Programs can be added or removed again at any time. The user determines the level of performance of Visi.Plus.
Like any other program or system, Visi.Plus also requires a core. For Visi.Plus, this is, as already mentioned, the database running in the computer's main memory.

Short, cursory description of operation
The smallest Visi.Plus system consists only of the above-mentioned **Data Management System (DMS)**. In order to be able to define this DMS at all, an additional program is required: the **Process Engineering Tool (PET)**. This module defines how and what should be communicated with the controllers.
It is precisely this information that the next tool, the **SDriver (interface driver)**, retrieves from the DMS in order to establish the connection with the PCD and permanently exchange the defined data.
With the help of the Graphical Editor, process diagrams are created for the display of data during runtime: the simple visualisation is complete.

But what about alarm handling, trend graphs, logging, remote messages, etc.? The other programs and functions that connect and communicate with the DMS are responsible for this.



Visi.Plus currently consists of the following programs and remains under continual development:

- **ALMMng.exe** (Alarm Manager)
  Manages alarms

- **ALMView.exe** (Alarm Viewer)
  Visualises alarms

- **ASCII Export**
  Exports ASCII files

- **bmp2jpeg.exe**
  Converts bitmaps to JPGs

- **changePwd.exe**
  Provides the option of changing passwords

- **CLKCfg.exe** (Clock Configuration)
  The Time Switch Program enables the switching of binary elements.

- **CLKMng.exe** (Clock Manager)
  Manages the system time

- **DMS.exe** (Data Management System)
  All programs access the DMS through a defined interface. All process data is saved in the DMS.

- **GE.exe** (Graphical Editor)
  Generates and displays the visualisation images during operation

- **HDAMng.exe** (Historical Data)
  HDA gathers all historical data and stores it in the PDBS database. The database is used in order to be able to display trend graphs and other statistical values.

- **Logger.exe (Cyclical Data Logging)**

- **MALMConfig.exe** (Mobile Alarm Configuration)
  Remote alarms are set here.

- **MALMMng.exe** (Mobile Alarm Manager)
  Manages alarms in case of malfunction.

- **pBackup.exe** (Data Backup)

- **pChart.exe** (Generate and edit statistics. Automatic forwarding of CSV files via e-mail)

- **PDBS.exe** (Database of Visi.Plus)
  Database on the computer's hard disk

- **pEdit.exe** (Text Editor)

- **PET.exe** (Engineering Tool)
  Generates the operating resource objects and defines the system objects. Data input, alarms, control functions

- **pFTP.exe** (File Transfer Protocol Server)
  FTP Server of Visi.Plus

- **pList.exe** (Data Viewer)
  Used for editing and visualising the system signals
- **pLogin.exe** (Log in to Visi.Plus)
- **PMosFilePicker.exe** (Find and open files)
- **pPaint.exe** (drawing program)
- **ProjectCfg.exe** (Project Configuration)
  Manages Visi.Plus projects
- **promos.exe** (starts the Visi.Plus system)
- **PrtFormat.exe** (defines log formatting)
- **PRTMng.exe** (Log Manager)
  Manages log formats
- **PRTView.exe** (Log viewer)
  Save and display any events in the Visi.Plus database
- **pStop.exe** (stop system, exit modules)
- **pUser.exe** (User Management)
- **PWEB.exe** (WebServer)
  With the help of the WEB Server you can display the visualisation images drawn in the GE directly in a browser.
- **SetDMSVal.exe** (read and write DMS values)
- **ShowVersion.exe** (display version of all installed modules)
- **SDriver.exe** (SBus Driver)
  Communications program between DMS and PLC (Saia PCD®)
- **DPDriver** (Profibus DP Driver – no longer in development)
  Connects a Profibus station to Visi.Plus (only slave)
- **OPCDriver** (OPC Client Driver)
  Connects an OPC station to Visi.Plus
- **ESPADriver** (in Version 1.5 or higher)
  Communication with ESPA 4.4.4 devices (master or slave)
- **SNMPDriver** (in Version 1.5 or higher)
  Communication with SNMP-capable devices (UPS, network hardware, PCs, etc.)

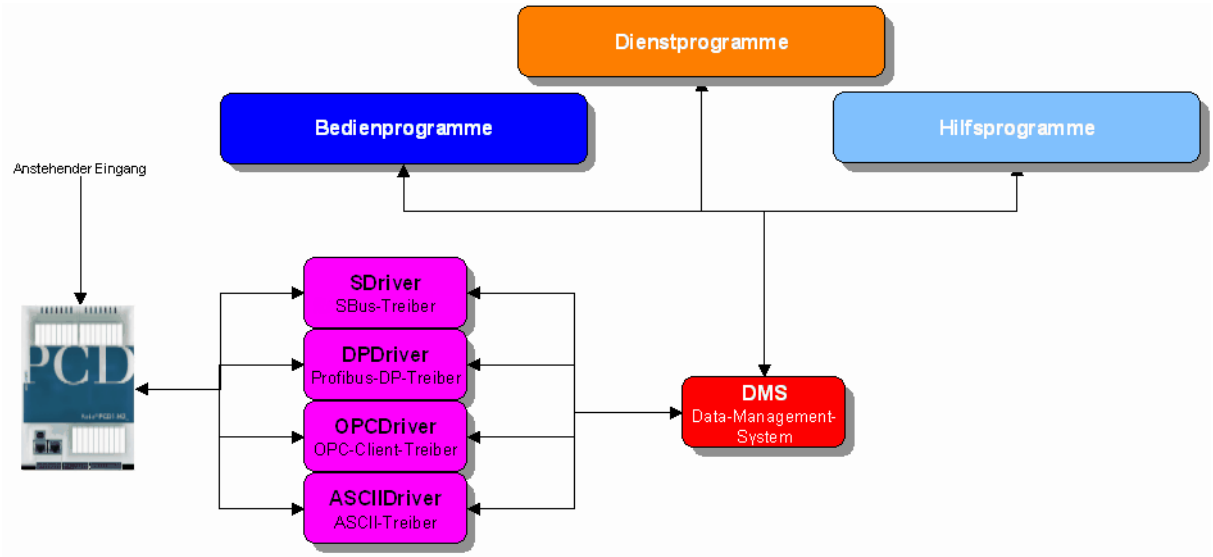
- **TAPIDriver** (in Version 1.5 or higher)
  The TAPI Driver accepts telephone calls and, after successful verification of the caller (code input control), can acknowledge all alarms that were sent via MalmMng (SMS, e-mail,

pager, ESPA, Tel.Voice).

- **ASCIIDriver** (ASCII Driver – no longer in development)

2.7.1 Signal processing

Data in Visi.Plus is processed as shown in the graphic below:



A signal is generated by the PLC and copied to communications resources (marker/register) (possibly also processed further by the PLC).

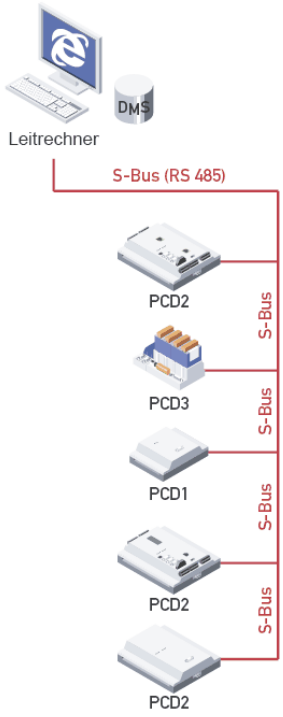
The exchange of data between PLC and DMS occurs through the respective driver which has been set (xxxDriver), preferably using a flag and register.

The DMS informs the drivers which data is exchanged. The contents and/or the data structure of the DMS itself are defined with the help of the PET (for further information, see Chapter 6).

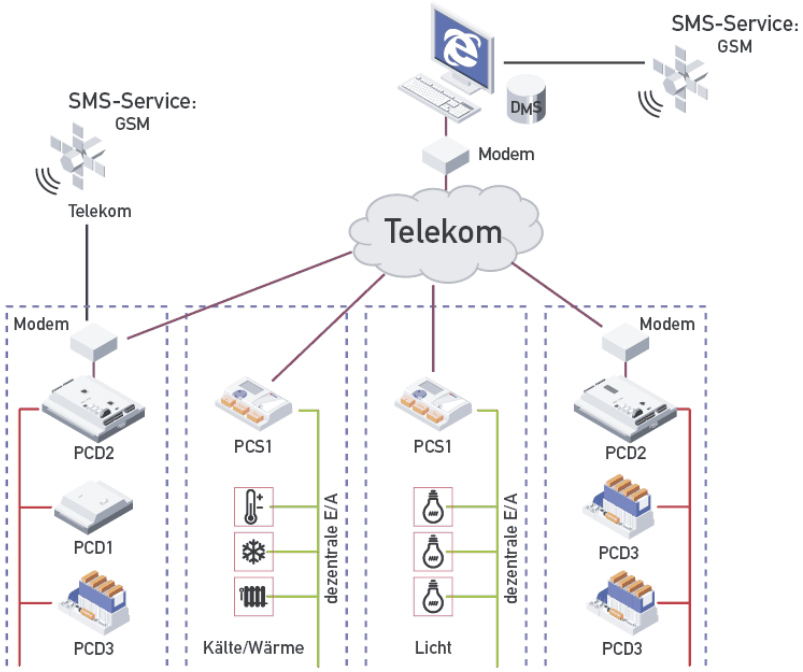
The DMS, in turn, provides the data to all the programs and also receives data from them. This makes the DMS the central hub for the exchange of data between the various Visi.Plus program modules (for further information, see the chapter DMS - Data Management System).

2.8 Overview of network structure and topology

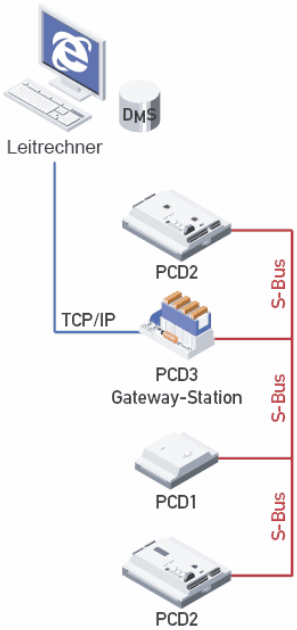
Visi.Plus enables fully free and variable configuration. Visi.Plus can be completely installed on a system computer. Various components can communicate with Visi.Plus through different protocols.



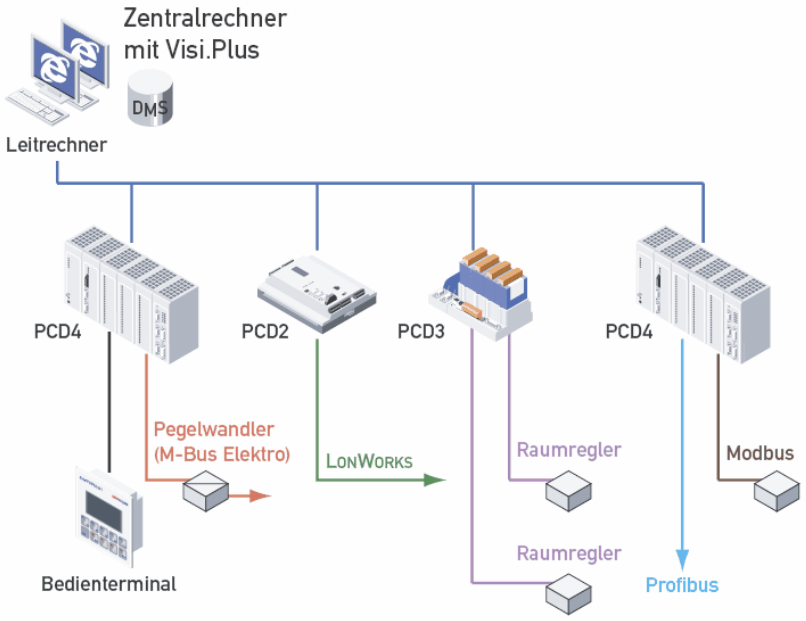
Single Mastersystem mit serieller Anbindung



System mit verteilten Liegenschaften, welche über Modem-Verbindung gekoppelt sind



Duo Mastersystem mit serieller Anbindung



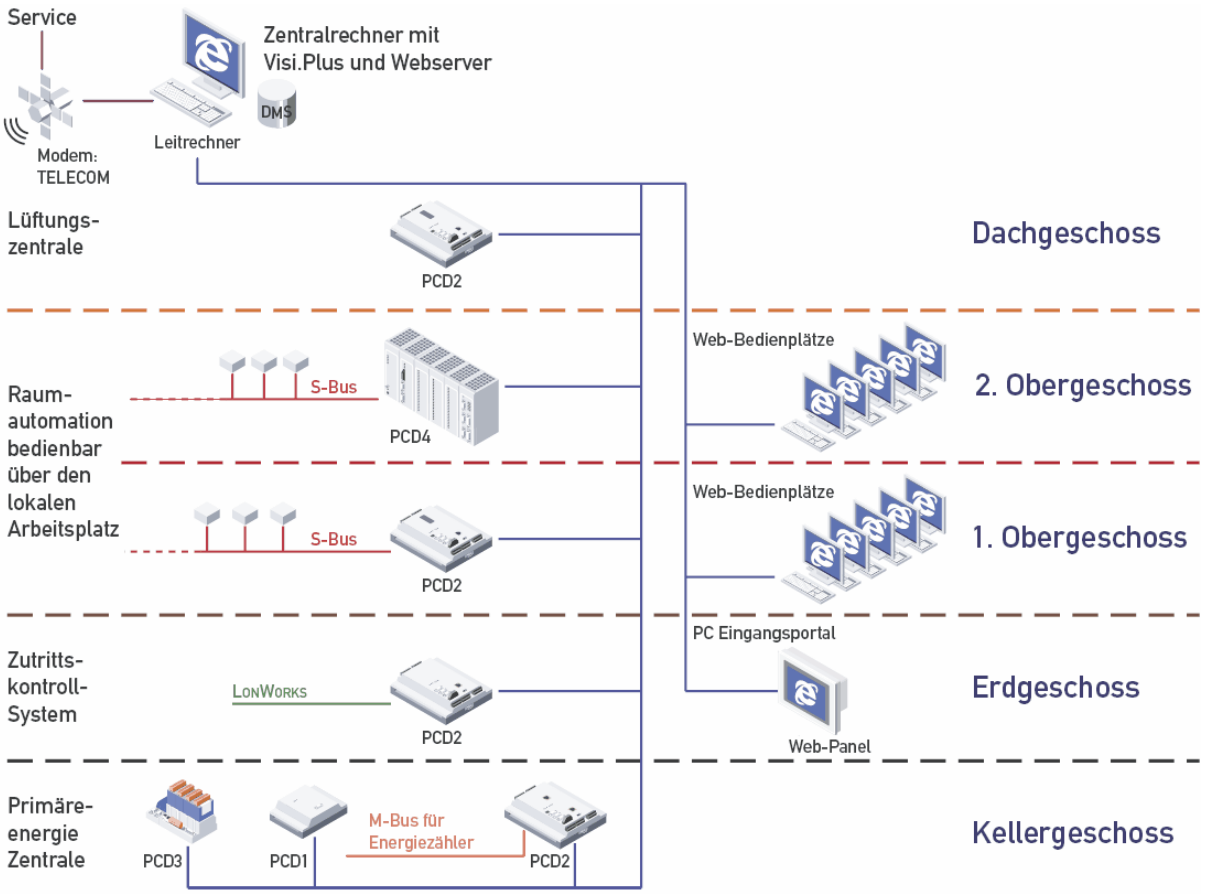
Multi Mastersystem mit einer Anbindung der gesamten Automationsebene über Ethernet TCP/IP

In addition, parts of Visi.Plus can be installed separately on different computers within a network. In this situation, data access takes place using the following network protocols:

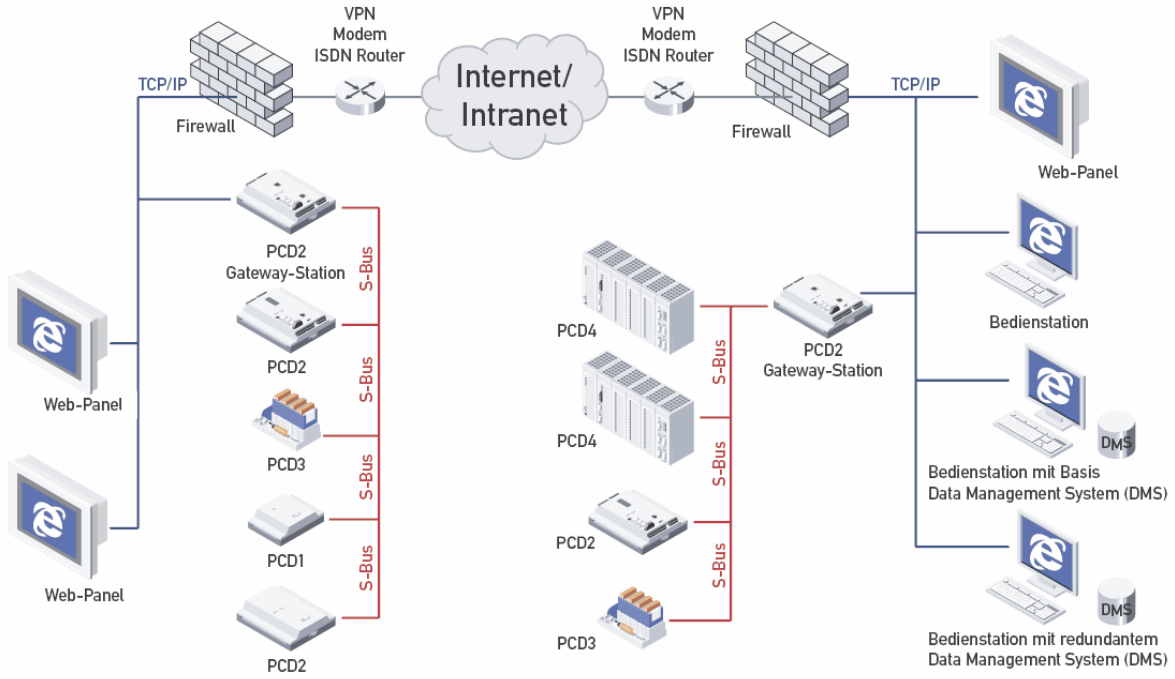
TCP/IP, IPX or NETBIOS

A connection to the individual PCs in the network must exist.

During commissioning, it may be beneficial if Visi.Plus is fully active on the control system PC, and a laptop computer is also connected over a network. For example, the Project Engineering Tool (PET) can be started on the notebook and access the DMS of the control system PC directly.
This means, for example, that adjustments to the images can be made without disrupting ongoing operations.



Theoretically, an unlimited number of controllers can be connected to each system (computer). These systems do not have to be connected to the same computer. Visi.Plus allows the use of various networks.



Redundantes Data Management System

2.9 Typical methods of working with Visi.Plus

2.9.1 Method 1

The project engineer begins with the integration of template objects (VLOs), such as monoblock, time switch, weekly programmes, etc.
VLOs consist of images, icons, PLC function blocks (Fupla/AWL/IL code) and even entire screens linked together.
Multiple VLOs are provided as libraries from Saia-Burgess Controls AG and optionally from MST Systemtechnik AG.
You can also create and manage VLOs on your own.

The project manager collects all system objects in tabular form with the help of VLOs through the Visi.Plus tool PET. In the process, performance data, order numbers, manufacturers, diagram numbers, threshold values of measurements, PLC I/O, etc. are assimilated into the Visi.Plus Data Management System (DMS).

Visi.Plus then generates the finished PLC code, which must be created, linked and compiled with the remaining program components in PG5.

Downloading to the PLC concludes the process. Now the system can be commissioned.

Advantages:

- The documentation of system objects is already taken care of.
- Monoblocks, etc. only have to be parameterised.
- Complex, predefined graphics/images/screens can be assigned to PCD elements/ Visi.Plus objects with just a few clicks of the mouse.
- Considerable time savings

Ideal for:

- Programmers who also want to use third-party libraries in order to save time on programming and troubleshooting.

Existing libraries can be used:
- Saia PG5® DDC Suite (ideal for Fupla templates)
- MST VLO Library (ideal for code generation)

2.9.2 Method 2

Like Method 1, but no libraries are used. Everything else remains the same.

Advantage:
- No costs in addition to Visi.Plus

Disadvantage:
- The template objects must be created by the user (more time-intensive).
- Object information/data may be forgotten.
- The entire program, including monoblocks, must be created from scratch with Fupla or AWL/IL. Troubleshooting is more time consuming.

Ideal for:
- Engineering agencies that want to create and use their own VLOs.
- Programming/engineering agencies that want to maintain their independence.
- Simple systems

2.9.3 Method 3

The system has been programmed and possibly already commissioned using Saia PG5®.

Then the PG5 symbol resources are exported and imported into Visi.Plus. In the process, a Visi.Plus object is created for each PCD element. Two types of VLOs are used in the process: one digital and one analogue.

Advantage:
- Existing systems can be visualised without a great deal of expenditure.

Disadvantage:
- No clear overview of the structure as with Methods 1 and 2.
- No documentation for system objects, only symbol tables from PG5 including comments
- Graphical elements must be individually assigned.
- Work is repetitive
- The entire program (including monoblocks, etc.) must be created from scratch with Fupla or AWL/IL.

Ideal for:
- Smaller and/or older systems
- Casual programmers/visualisers

Saia Visi.Plus

Installation

Chapter 3

3 Installation

The product is installed automatically through a setup program. Please observe the minimum requirements for the PC and the supported operating systems.

3.1 Summary

This chapter describes how to install Visi.Plus as well as the minimum requirements for the computer. In addition, it shows how licences and template objects for a project should be installed. The structure of a Visi.Plus project and the porting of projects to other platforms are explained.

3.2 System requirements

Visi.Plus requires the following minimum configuration:

" Windows 7, Windows 8, Windows 10, Windows Server 2008 R1 & R2; Windows Server 2012 R1 & R2 *)
" Core 2 Duo
" 1024 MB RAM (see “Memory requirement” in Chapter 2)
" Hard drive with at least 1 GB of disk space available (more space is required for alarming and historic data)
" CD-ROM drive (only for installation and potential external backup of data (CD burner))

 *As a general rule, the better equipped a PC, the more efficient the work with Visi.Plus.*

Recommended system requirement:

" Industrial or server computer. Computer must be in operation 24/7. Using a home computer is not recommended for this reason.
" Windows 7 or Windows Server 2008 R2.
" 8 GB RAM.
" 500 GB disk space. Visi.Plus writes on the hard drive relatively often, which is why an SSD is not recommended
" Min. processor i5 from Intel
" USB 2 and USB 3 ports for data transfer and external devices
" LAN port
" Optional: RS232/RS485 interface for devices.

*) MalmVoice has been tested so far only with modem under WindowsXP and Windows server. The guarantee for the modem driver under actual operating systems we cannot insure.

3.2.1 Operation on systems without a hard disk

Operation on systems with RAM-based disks is possible, but we cannot offer any support. Various tests with SBC panel PCs have shown that Visi.Plus is capable of running on such systems (Win XP Embedded). Visi.Plus is not capable of running on Windows CE.

In particular, historical data will often automatically require a memory of several hundred megabytes. In addition, the operating system usually requires a memory capacity of one gigabyte.

Please note that the storage of historical data can lead to a large number of disk writing cycles. Older flash memory can only write a few thousand times.
The number of erase cycles of NAND flash memory cannot be precisely determined. According to the manufacturers it ranges from 100,000 to 1,000,000 (as of 2006). With NOR flash types the assured number is only approximately 10,000. However, manufacturers indicate that their memory can even achieve ten times these figures. In a test conducted by a computer magazine a USB stick was written to 16 million times (always the same file) without an error occurring. However this assumes very good fault management of the stick.

3.2.2 Virtual PCs

The control system can be installed on VMWare Workstation. It must be ensured that sufficient system resources are available.
There are no known problems on ESX systems.

Other VM-supplier are not tested.

3.3 Executing the setup program

Visi.Plus is installed by executing a setup program. This is either found on the supplied DVD or can be downloaded from the internet.

Before installation, check that your system meets the minimum requirements.

Before installing Visi.Plus, the PG5 programming tool should be installed. The Visi.Plus setup program will then automatically carry out the necessary adjustments in the PG5 installation.

Start the setup program and follow the prompts.

After Visi.Plus is installed, the program can be started via the Windows Start menu or the automatically created desktop shortcut. Visi.Plus should be completely installed on all computers on which the program is to be used. This is also the case if the intention is only to establish a visual display with connection to a remote Visi.Plus computer.

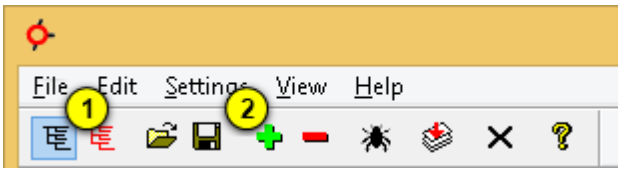
3.4 Software update

A software update must be performed by a specialist trained in Visi.Plus. The specialist requires administrator rights in Visi.Plus and the necessary rights on the computer to make system changes.

Software update to the version 1.7 is supported from version 1.5.

The following steps are obligatory when updating software:

- Save project data: Also open the DMS for a running project:



1 Update both subtrees before saving. Click the subtree icons to update the data

2 Now save the DMS.

- A data backup is recommended. Also open the Project Manager and create a backup copy of the project under the menu item Data backup. More information about a data backup and the pBackup module can be found in the chapter Data backup. The Project Manager can be started in the directory <Installation path>\bin\projektcfg.exe.
- End all modules of Visi.Plus via the Stop menu item in the Project Manager.
- A data backup of the installation is recommended. To do this, the entire installation folder of Visi.Plus must be copied or archived. Visi.Plus is installed under C:\Visi.Plus\ as standard.Visi.Plus
- Execute the new Visi.Plus setup.
- Open Project Manager and select the required project. The path for the projects are saved under <Installation path>\proj and under <Installation path>\backup for project backups after installation as standard. These can be changed in Project Manager at any time.
- Start project.

The data backups must be created with the existing modules of Visi.Plus, otherwise this may lead to compatibility problems.

After updating, it is recommended to save the GE and pWeb images again. Carry out the following steps to do this:

- Start GE and switch to the edit-mode.
- Use the function **"Save all GE images of the project"** under the menue ***File.***
- Use the function **"Save Web-Images"** under the menue ***File***.

A function control must be performed by the system integrator after the update. The function control will look different from project to project. A general checklist cannot be issued. However the following points generally apply:

- Communication control: Check the data exchange between the PCD/field device and the

control system with the drivers used.
- Visualisation control: Control of the change of image, visibilities and access controls in the graphic editor.
- Web access: Control of the web access, change of image, user login and access rights via the Internet browser.
- Alarming: Control of alarms. Triggering of an alarm on the controller must be visible in the control system via the viewer used (Web, GE, AlarmViewer).
- Historical data: Control whether the data points are saved and can be called up using the different viewers. (GE, Web, pChart).
- pChart: Check the existing templates.
- Remote alarms: Control of the protocols used for the remote alarm.
- Backup: Control of the backup and restoration functions.

 All changes to the new version are recorded in the ReadMe. The ReadMe is included in the setup and saved after installation under ..\ Visi.Plus\ReadMe.txt.

3.5 Licence packs

Visi.Plus Mini - Mini pack

Data management system, Database system,
Graphical editor, Engineering tool, Time switch program,
Trend and alarm logging and
Saia® S-Driver for 1,000 information points

Visi.Plus Basic - Basic pack

Data management system, Database system,
Graphical editor, Engineering tool, Time switch program,
Trend and alarm logging,
MALM - Remote alarms via pager/SMS/e-mail,
PRT - Access logging and
Saia® S-Driver for 10,000 information points

Visi.Plus Standard - Standard pack

Data management system, Database system,
Graphical editor, Engineering tool, Time switch program,
Trend and alarm logging,
MALM - Remote alarms via pager/SMS/e-mail,
PRT - Access logging,
pChart - Trend graph, Web server 2 and
Saia® S-Driver for 100,000 information points

Engineering Edition

Data management system, Database system,
Graphical editor, Engineering tool, Time switch program,
Trend and alarm logging,
MALM - Remote alarms via pager/SMS/e-mail,
PRT - Access logging,

pChart - Trend graph, *Web server (20 simultaneous connections)* and
Saia® S-Driver

Visi.Plus is provided to every owner of a PG5 licence in the Engineering Edition at no additional cost. This means that Visi.Plus can perform valuable services already during the engineering and commissioning phases.The Engineering Edition can be used at a company's location as a development environment (on any number of PCs). The Engineering Edition does not require a hardware dongle and is limited to 24 hours of runtime in runtime mode, which means that if the GE is running in the runtime mode, it switches to the edit mode after 24 hours.

DEMO Version

Data management system, Database system
Graphical editor, Engineering Tool, Time switch program
SDriver (communication with *10,000* data points)
Trend logging, Alarm logging,
Logging, Remote alarms via pager/SMS

The Demo Version does not need a dongle and runs for max. 4 hours (if no software licence is installed – with SW licence only one hour).
The DEMO version has the following limitations:

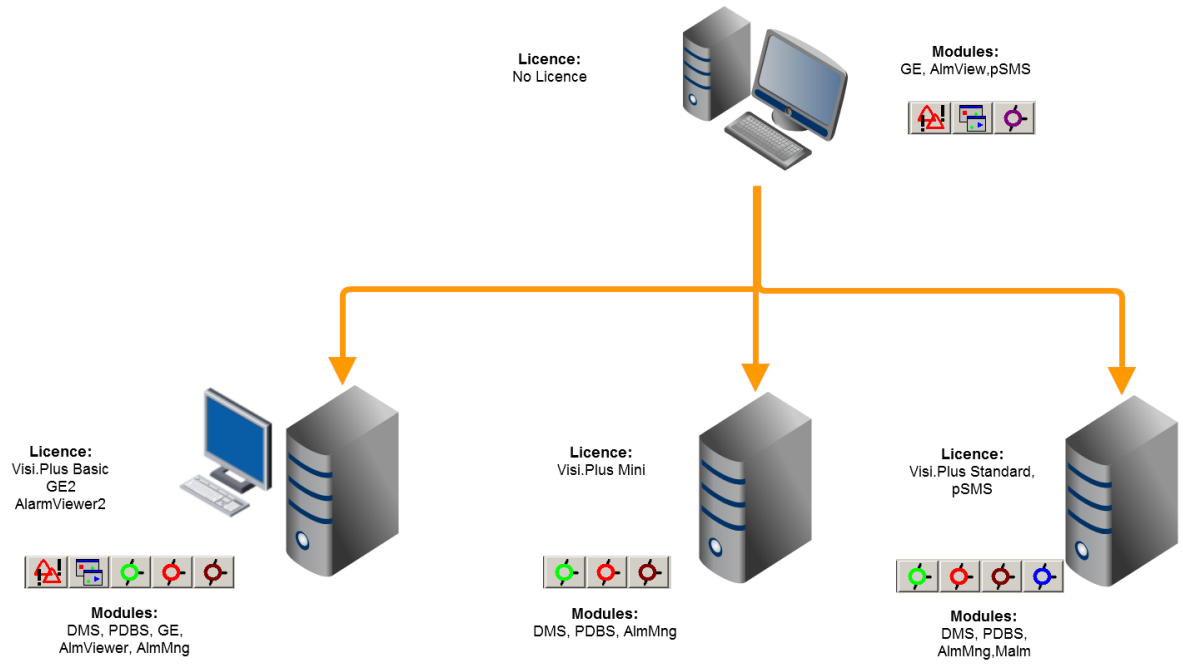
- *The start dialog must be acknowledged*
- *Code generation not possible (PET)*
- *Drivers are not automatically started*
- *Web Server runtime is 1 hour*
- *Alarms are not fully processed*
- *Logs are not fully processed*

3.6 Licensing Consept

The license will be checked by the DMS and compared with the dongle. When a module logs on, it is first checked for the existence of adequate licenses. If not found, a message appears on start-up. In the graphical editor (GE), this looks as follows:

If the graphical editor (GE) or alarm viewer (AlmViewer) is intended for connection with more than one DMS, no special licence is required. The decisive factor for the licence is the DMS. On the computer where the DMS is running, an appropriate license must also be present. Each module accessing that DMS will need the appropriate authorization. If more than one GE or AlmViewer needs to access the same DMS, the license must be authorized for more than one GE or AlmViewer! The following modules can be connected to more than one DMS: GE, AlmViewer, pSMS. The pChart module cannot connect to multiple modules in the current version. The alarm viewer currently runs without limit using any license package and can be called any number of times with no additional package.
The GE and AlmViewer log into the selected DMSs in multi-DMS mode. This means each Visi.Plus licence on the DMS computer must also include at least one GE. AlmViewer and GE are already present in each Visi.Plus basic package. If a further GE is to be added for visualizing one or more DMSs, it will need an additional GE extension on the licence! The managers all run on the same computer as the DMS.

This is illustrated by the following diagram:



When the GE is starts up, connections will be established to the DMS according to settings and the licence will be checked. If the licence is exceeded for any DMS, the following message will be displayed:

If the “No” key is pressed, the GE will start anyway, but will be unable to access the appropriate data and change the images according to the settings under ***Project Settings -> Definition in Case of Error***. See also Chapter: Error definitions

SDriver and PCDDriver:

Both drivers check the corresponding licence field.
The packages are designed to allow for projects of various sizes. In our experience, an increasing number of configuration points are written on the PLC. This increases the number of data points required. The limit has been doubled for this reason. This means that with an SDriver_1000 licence, up to 2,000 data points can be communicated between the control system and the PCD. A warning will occur in the driver if the limit is exceeded.


The communication of additional data points ceases. There is no rule or overview of the data points that are communicated or not if the limit is exceeded. In the PCDDriver, it is possible to deactivate a channel, which is then not counted when checking the communicated data points. This is not the case in the SDriver however. Here, all data points are counted, included those of the deactivated channels.

The DMS contains help data points that show the number of data points being communicated and the size of the licence limit.

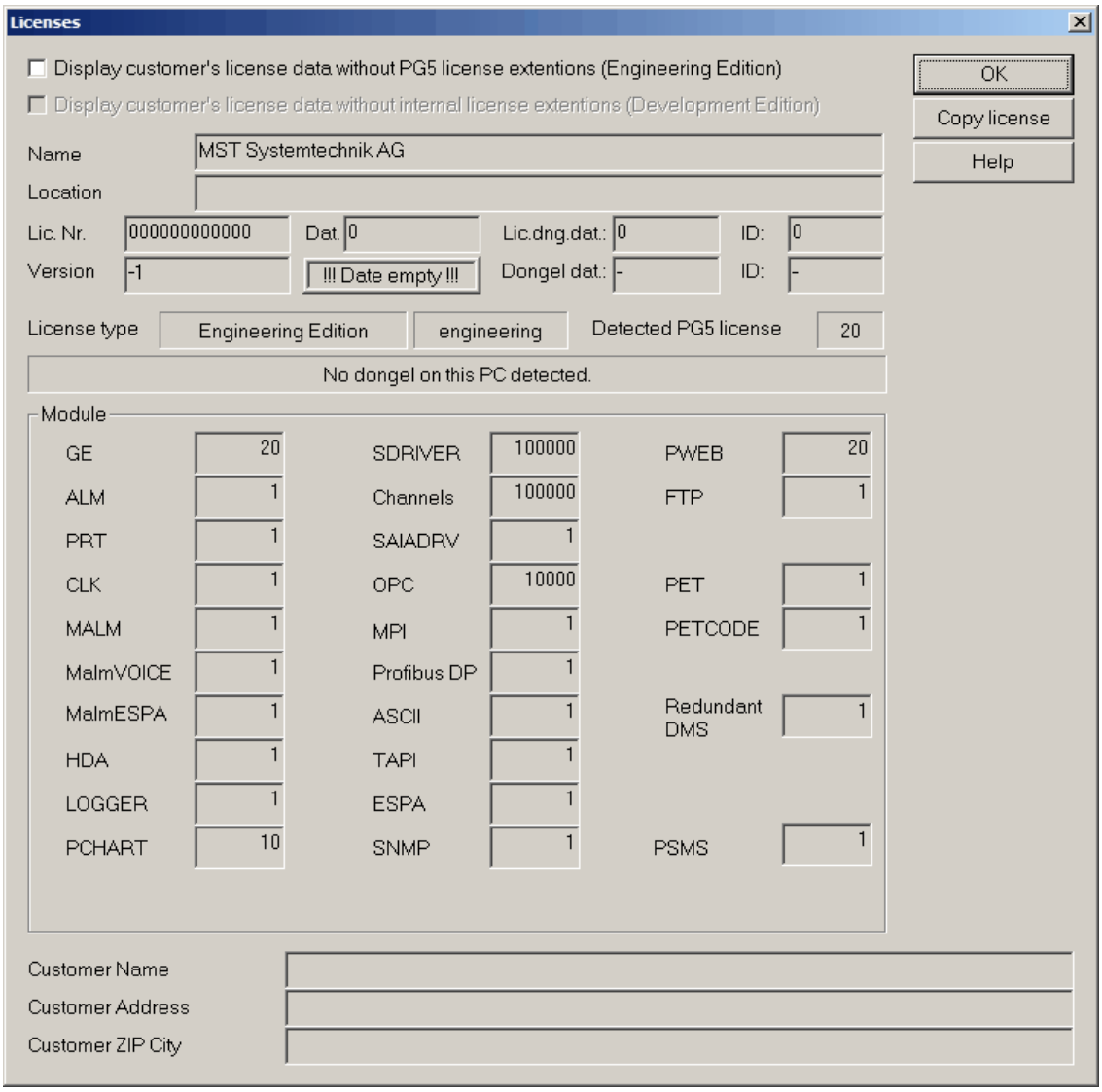
System:Driver:SDriver:Info:Licence :	Number of data points approved with the licence.
System:Driver:SDriver:Info:Total :	Number of used data points in the project.
System:Driver:SDriver:Info:Channels:	Number of available channels in the project.

Please contact your supplier if the limit has been exceeded and requires an extension.

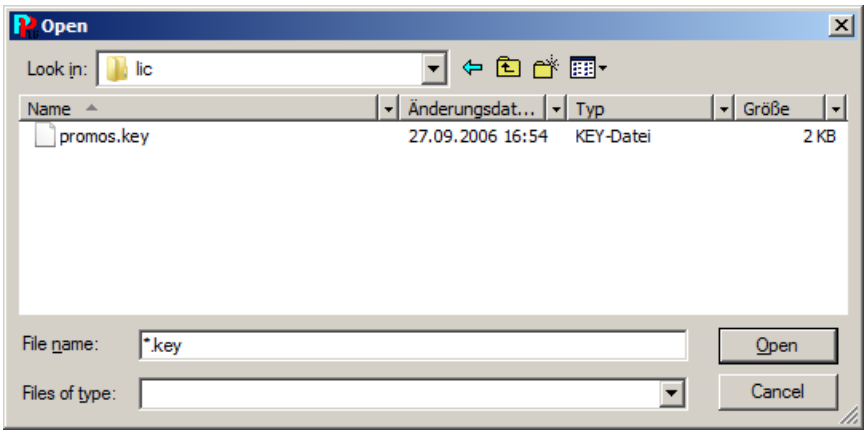
3.7 Licence key installation

The official Visi.Plus licence of the definitive owner is delivered via e-mail. To enable the relevant project, proceed as described in the following:

Start the **Project Manager (ProjectCfg.exe)** module in the Windows Start menu, mark the appropriate Visi.Plus project and press the **<Licences>** button. The following window opens:

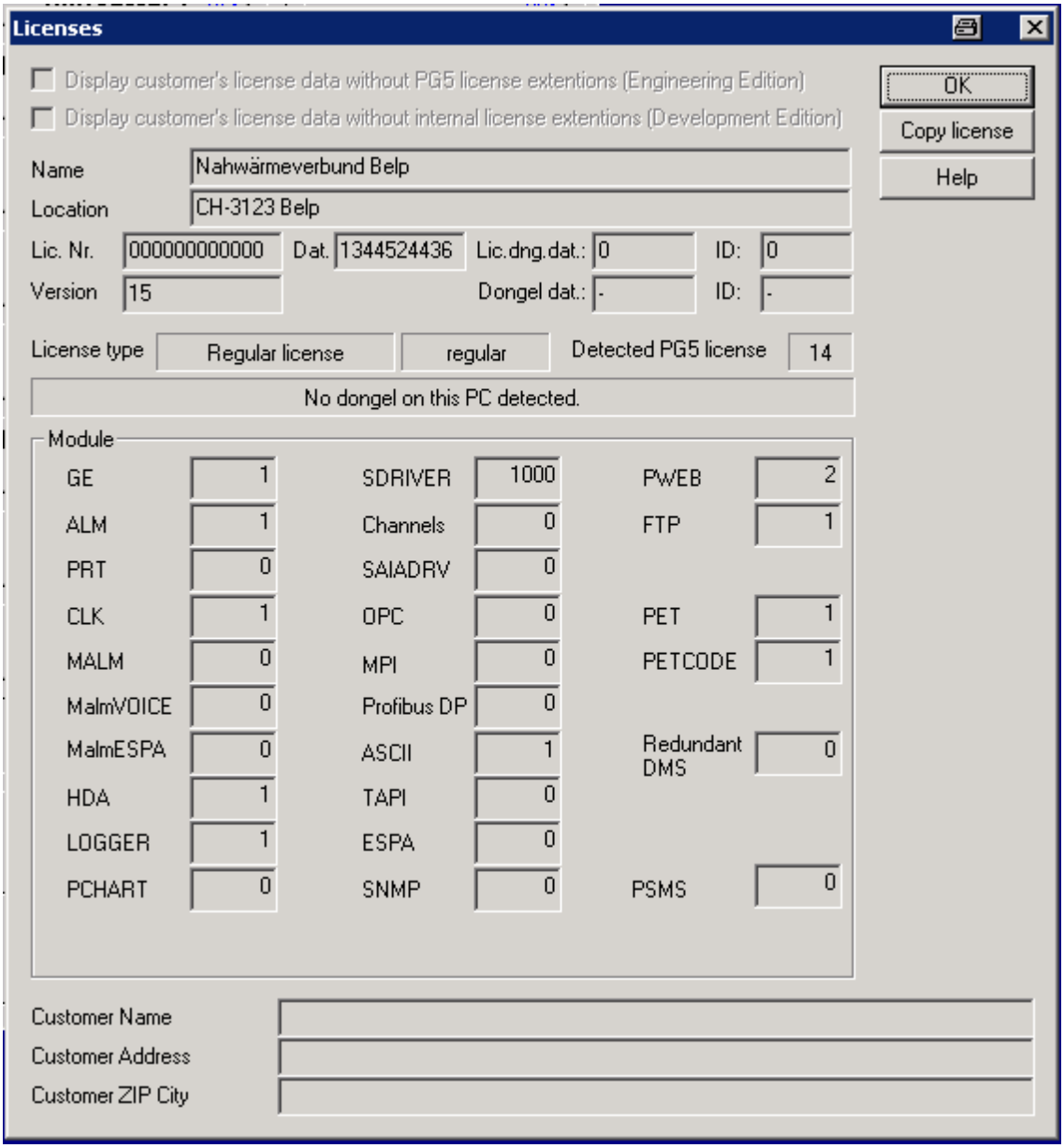


All attributes of the purchased licence are displayed in the **Licence Display** window. Click on the **<Copy Licence>** button in order to assign a Visi.Plus license to the current project. Then the following window opens:



Change to the directory or the drive where the Visi.Plus licence to be installed is located. Select the licence file and click the **<Open>** button. Now all the purchased licence data is shown in the **Licence Display** input window.

A licence file must be generated and installed for each project you create! The licence must match the supplied hardware dongle.

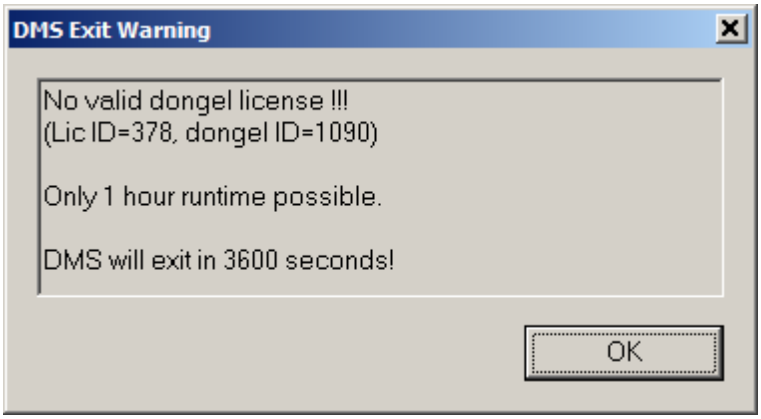


In the **Licence Display** window the licencee is shown in the upper part.
In the **Modules** text frame (programs of Visi.Plus), the number after the respective software module indicates how many licences have been purchased for the corresponding software module. In the lower part the customer data is shown.

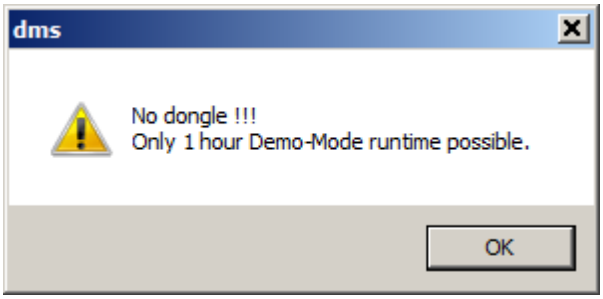
*If a demo licence is installed, the expiration date for the licence is shown to the right below the **<OK>** button. The enabled modules can be tested until this date. Afterwards, a customer licence must be installed.*

3.8 Hardware dongle

The supplied hardware dongle must match the software licence, otherwise Visi.Plus will only run for short time in demo mode with a continuous warning message:



If Visi.Plus is started with a valid license, but without dongle, the software can run for 1 hour in demo mode (warning box is displayed once):



The dongle can be plugged in to any USB interface on the computer.

If a dongle is pulled out while in operation, the program exits after a few minutes.

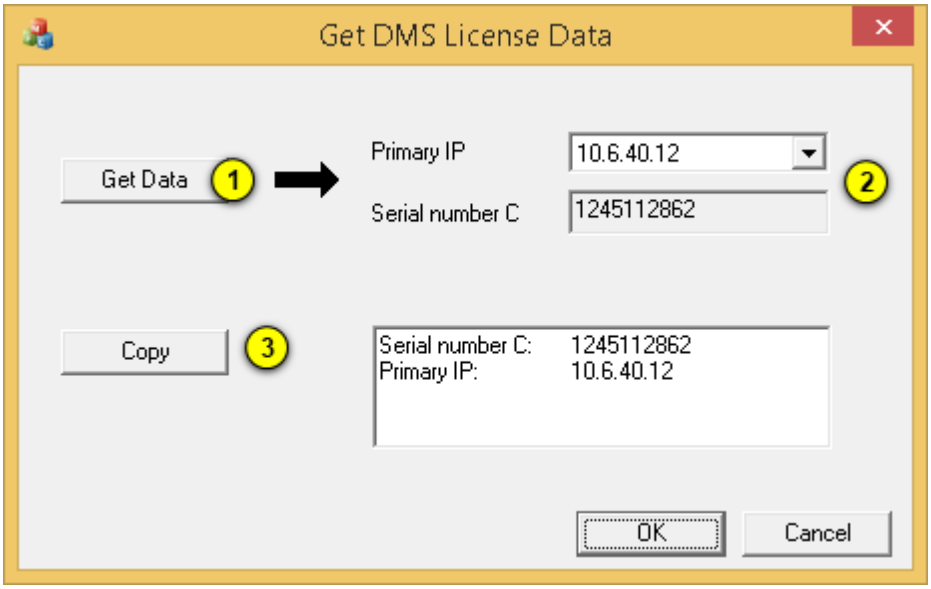
3.9 Software dongle

A software dongle can be used instead of a hardware dongle. A software dongle is required for virtual computers.
A software dongle is a file, promos2.ini, that can be ordered from SBC. The file must be copied into the bin folder of Visi.Plus. The bin folder is saved under the installation path. For Visi.Plus, this is C:\Visi.Plus as standard.

A software dongle is specific to a licence and computer, and replaces the hardware dongle.

Procedure when ordering a software dongle:
For a software dongle, the licence specifications, the IP address and C hard drive number of the computer are required.
This information is read with the GetIPSerHD.exe tool. GetIPSerHD.exe is included in the setup of Visi.Plus and can be called up after installation via *<Installation path>\bin.*
Alternatively the tool from the support page of SBC can be downloaded.
These specifications (IP address and hard drive number C) must be sent to SBC when ordering the licence.

GetIpSerHD tool:



1 Update both subtrees before saving. Click the subtree icons to update the data.

2 Now save the DMS.

3 Use the ***Copy*** button to copy the data to the buffer. Use the key combination <CTRL> + <v> to paste the data into an email.

The data backups must be created with the existing modules of Visi.Plus, otherwise this may lead to compatibility problems. After updating, it is recommended to save the GE and pWeb images again. Carry out the following steps to do this:

Required specifications for the licence:
- Type of licence.
- End user, SI and Reseller Information
- Visi.Plus packages
- Add-On to Visi.Plus packages
- Serial number C and primary IP of GetIPSerHD.exe

Procedure when replacing a hardware dongle with a software dongle:
If a licence (promos.key) is already provided with a hardware dongle, the change to a

software dongle can be made subsequently. The following steps are necessary for this:

- Use the GetIPSerHD.exe tool to read the IP addresses and the hard drive number C.
- Send the information to SBC Order Processing with the promos.key licence file. The licence file is saved under *<Project path>\<Project name>\lic*.
- The hardware dongle must be removed after receiving the software dongle. Now install the software dongle.
- The hardware dongle must be sent back to SBC.

Installation of the software dongle:

- Save and exit any project that may be running.
- Copy the **promos2.ini** file to the *<Installation path>\bin* directory.
- Copy the associated **promos.key** licence file to the project directory <Installation path>\proj\<Project name>\lic.
- Start the Visi.Plus project.

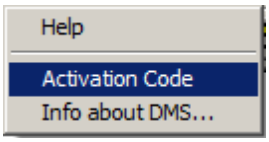
If the hardware dongle is not returned to SBC, clients will be invoiced the full price of a new licence.

If the IP address or the hard drive number C is subsequently changed, a new licence is required. Full prize for the licence will be invoiced.

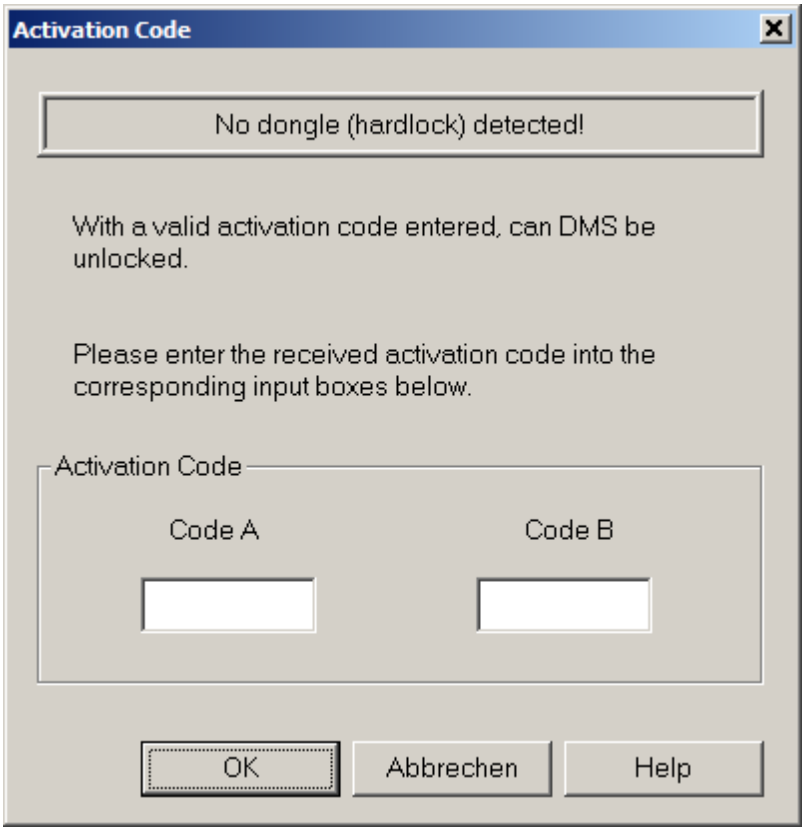
3.10 Installation activation code

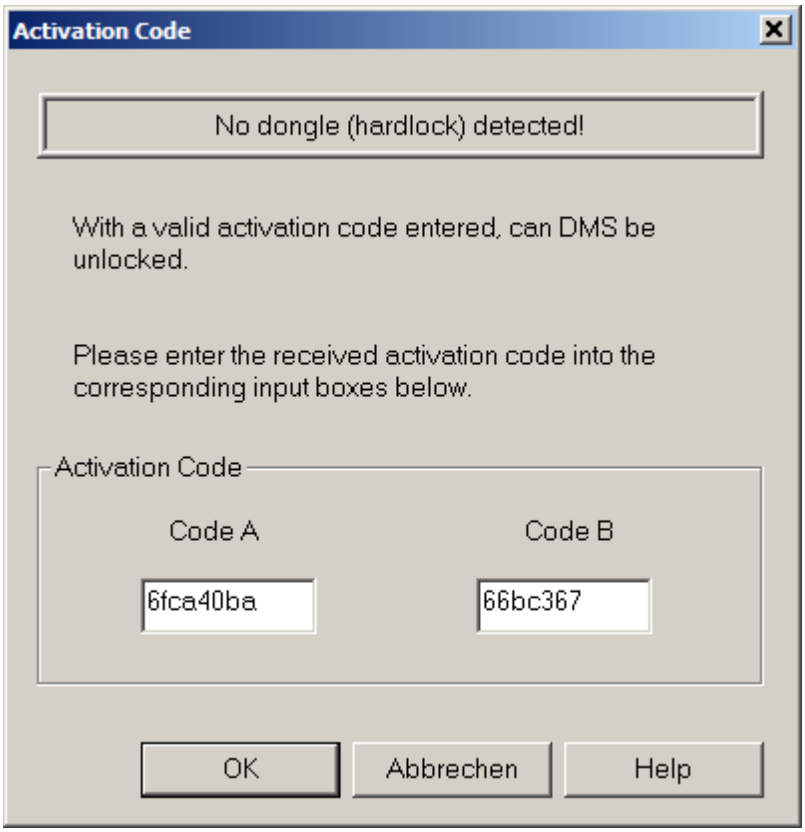
See also chapter DMS menu activation code.

When Visi.Plus is running in demo mode (either no licence or no hardware dongle present), activation codes can be entered in the Help menu in the DMS. This enables the full functionality of Visi.Plus during a 90-day trial period:



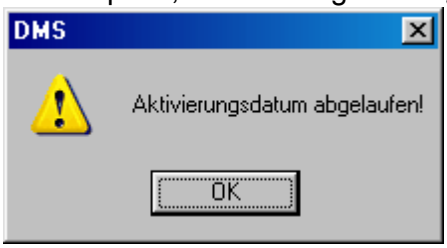
When the **Activation Code** menu entry is selected, the following dialog appears:





After entry a confirmation box appears:

If the trial period has expired, the following message appears when starting Visi.Plus:



The activation code can be requested at www.sbc-support.com.

3.11 Installing template objects

The Visi.Plus system comes as standard with four template objects (VLOs): **MOT01**, **MES01**, **ANA01** and **DIG01**. You can start working with these template objects right away. Template objects enable data points to be combined into groups of signals. The chapter **Process Engineering Tool (PET)** describes how some template objects can be created.

Additional template objects are available. The following objects are already available:

- ANA01 (in the scope of delivery)
- DIG01 (in the scope of delivery)

If you have all of the template objects indicated above (as an object library), no additional objects beyond these tried-and-tested objects should be necessary. Additional Information can be found on our homepage.

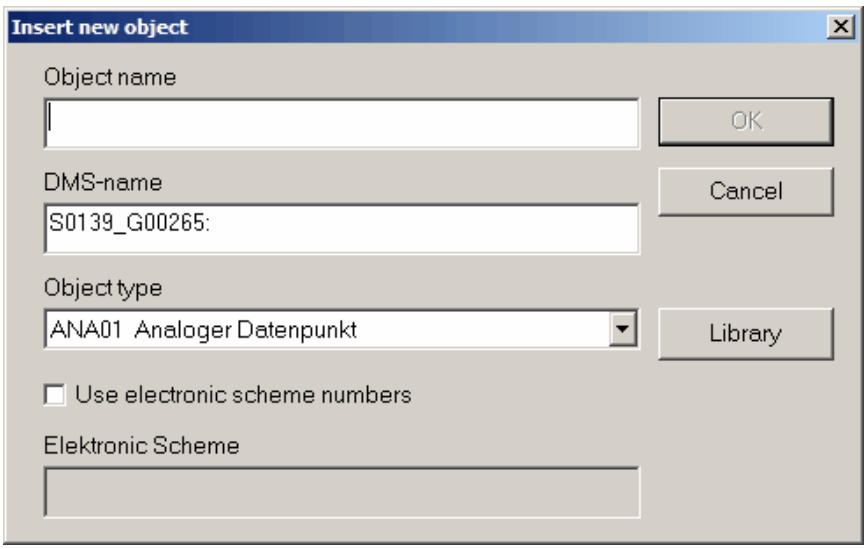
Before templates can be integrated into a Visi.Plus project, they must be installed. To do this, start the Visi.Plus Project Engineering Tool **PET**.

Start the PET with the Start button
Start > Programs > Engineering Tool.

There are two ways to install template objects:

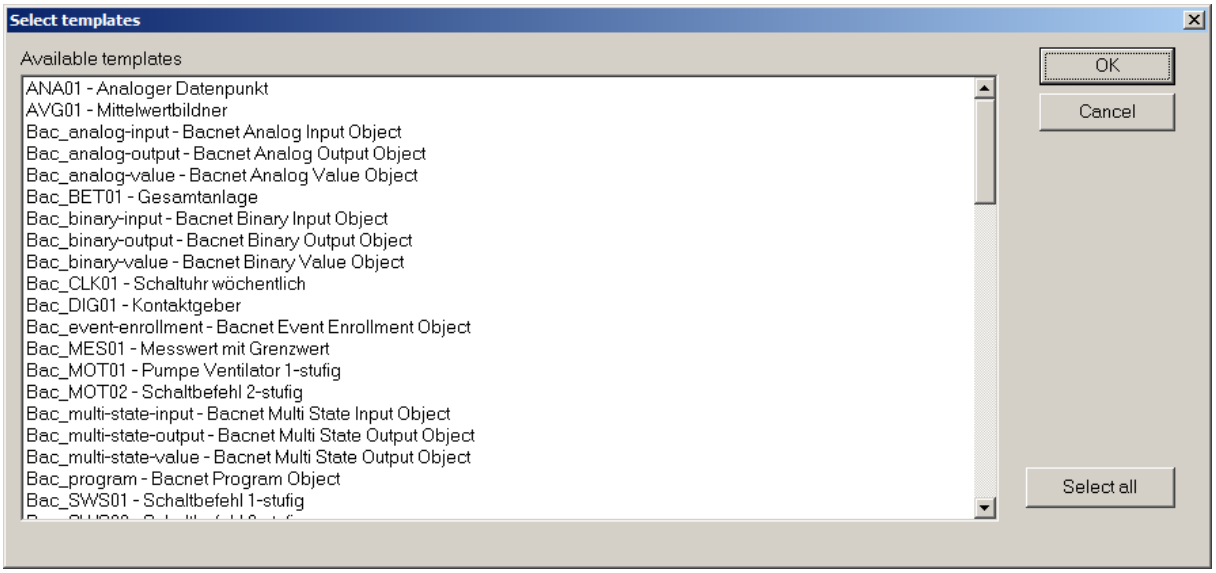
- Either
  , in the PET under the menu **Template Objects** select the command **Templates from Library**

- or 
  click on the **Insert Object** icon on the toolbar. Then press the **<+>** or **<Insert>** key. The following image appears.



By clicking on the button **Library** , the **Select Template Objects** dialog window appears.

All available template objects are listed in the window **Select Template Objects**:



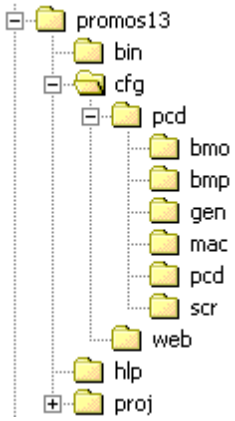
After selecting the template objects to be installed, they are integrated by clicking the **<OK>** button.

The DIG01 and ANA01 template objects are objects that can display the individual values (digital or analogue data point). The two objects are used when the data should be displayed 1:1 for a project (e.g. with Fupla programs).

Visi.Plus is object-oriented and should therefore be structured on complex objects. This makes it more productive in how it works. The MOT01 template object covers the possibilities of a complete motor, including registering hours of operation, manual/auto operation, confirmations, faults, revision switch, etc. The PLC code of the motor is automatically generated from Visi.Plus. The graphical elements and control windows are already pre-generated and can be used with a few mouse clicks. The programmer only has to worry about connecting the individual objects (so-called reinitialisation).

The MES01 template object covers the possibilities of a complete analogue measurement including trend acquisition and control screens.

*All template objects are stored in the Visi.Plus installation directory under **..\cfg\pcd**. If you create your own template objects, copy your objects to this directory so that they are included in the installation list.*

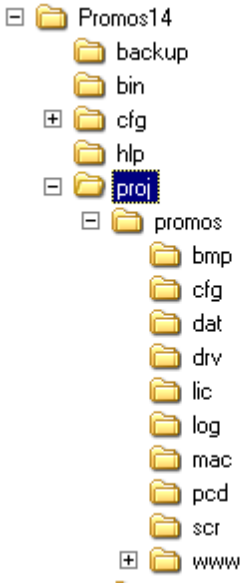


Each template consists of its structure, bitmaps, macros (groups of graphical elements), control screens and PLC code. The files of each template object are distributed over different directories under **..\cfg\pcd\..**. The following table shows where different types of files are saved:

Directory	Contents
bmo	Structure of an individual object
bmp	All images belonging to the template object (bitmaps or jpegs)
gen	Code generation templates for each object
mac	All template object macros
pcd	Pre-generated PLC code of the template object
scr	Contains the control or system screens

3.12 File structure / project structure

All projects created with Visi.Plus are stored in the installation directory under **proj** in a project-specific folder.
The name of the project is used as the folder name. **All** necessary data for the project is saved in the subdirectories within it.
In the following picture you can see the project structure of the **promos** project:



Each project contains the following subdirectories in which the described files are saved.

Directory	Contents
bmp	Contains all bitmap files required for the project. Visi.Plus also supports JPG files, which are also saved in this directory.
cfg	Contains all configuration files, such as object and DMS structures
dat	Saves all recorded data, such as alarm history, trend data and log data
drv	Configuration files for the communications driver
lic	Contains the Visi.Plus licence for the current project
log	All actions, such as login/logout and module start/exit, are saved here.
mac	Contains all macros and graphical library objects defined for the project
pcd	All data created by PG5 should be saved here. The PG5 project is called 'PCD' by default. The Visi.Plus PCD Code Generator, as well as all functions of PG5 that can be accessed from the PET are preset to this directory.
scr	Contains all process diagrams (screens)
www	All JAVA classes and process diagram data are located in the www directory. This allows you to use your process diagram via a web browser.

We also recommend saving the PLC code created by the PET and PG5 in this directory. By copying the complete directory structure, all data required by the project (Visi.Plus data as well as the complete PLC program) is backup up.

3.13 PG5 installation

The Saia PG5® programming tool must be installed prior to the installation of Visi.Plus. The Visi.Plus setup program will automatically carry out the necessary adjustments to the PG5 installation.
The documentation from Saia-Burgess Controls AG provides information on the installation of PG5.

The Visi.Plus communications driver accesses the SCOMM.DLL (driver) of SBC. If PG5 is not installed, the SCOMM DLLs are contained in a subdirectory of the bin directory. The path to these DLLs must be set in the SDriver.

3.14 Network installation

In principle, it would be possible to install Visi.Plus on a server. However, for resource and speed reasons, we recommend installing and operating Visi.Plus locally. In order to save project data, select the complete project with its subdirectories from the directory **proj** and save it to the server.

Exception: Remote systems like pChart Remote or GE Remote.

3.15 Installing the WEB Server

All data required for visualisation over the internet can be automatically installed via the standard installation. No additional programs are required.

*If you create a process diagram and save it, it is automatically stored in the **www** folder in the respective web format.*

More detailed information about the visualisation possibilities over the internet can be found in the chapter Utility programs, PWEB.exe.

Installing Java applets

Applets are installed in the directory c:\Visi.Plus\cfg\web\lib (possibly a different drive). When the Web Server is started, the jar files are automatically copied to the current project directory if they are more recent versions.

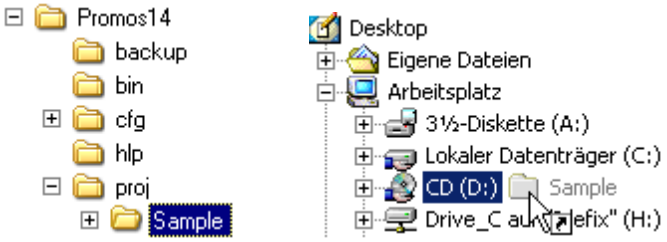
*PromosClas*Applet for displaying images
ses.jar
PromosAlar Applet for displaying the Alarm Viewer
m.jar
PromosProt Applet for displaying the Log Viewer
.jar

3.16 Porting existing projects to other platforms

To do this, the desired project under ...\Visi.Plus\proj\... should be copied with all of its subdirectories to a data storage device.

This means that all the required files for the project are backed up on the data storage device.

The reverse procedure should be carried out on the destination computer:



Alternatively, the **Backup** function, which creates a ZIP file with the respective date and project files in the folder **...Visi.Plus\backup\...** is available in the Project Manager.

This file can then first be selected by pressing the **Restore** button in the Project Manager, then restored and finally edited (see Data backup using pBackup).

3.17 Remote systems

Individual modules can be operated as remote systems:
- GE (Graphical Editor)
- AlmView (Alarm Viewer)
- pChart (evaluations)
- pLogin (log-in program)

Remote installations only function if the respective licences are installed. If you only have a single GE licence, the GE can be exited on the master PC and then started on the remote computer. Only a single GE system can access a DMS over the network.

3.18 Connect with multiple DMSs

From version 1.6, various modules are capable of multiple DMS connection:

- GE
- AlarmViewer
- pSMS
- pChart

The pWeb is not capable of multiple DMS connection.

In the GE, assignment takes place via layers. The layers can be linked to an IP address, allowing the GE to reference the right DMS.
The DMSs will continue to run independently and are autonomous. If it is still necessary for them to exchange data with each other, this is possible using, for example, the additional tool SyncDMS. Information about this module may be obtained from Saia-Burgess Controls AG.

The modules can be configured to up to 20 DMS connections. There are no additional limits for the GE. This also applies to the AlarmViewer, pChart and pSMS. The limits for single systems also apply to multi-DMS systems.

Saia Visi.Plus

Starting programs

Chapter 4

4 Starting programs

4.1 Summary

This chapter describes how Visi.Plus and its modules can be started individually or in combination. How to create new projects and install licences is also described. Backing up data is another important topic. This is explained with an example.

4.2 Starting Visi.Plus

After the standard installation has been successfully completed, Visi.Plus can be started as follows:
START > Programs > Saia Visi.Plus 1.6 > Visi.Plus

Visi.Plus starts different modules depending on the project. Apart from the Graphical Editor, no other windows of Visi.Plus are opened.

First the DMS (Data Management System) is started, followed by the PDBS module (database of Visi.Plus) if a licence has already been installed. Finally, the Graphical Editor is started, as long as nothing different was defined in the start options (see the chapter Project Manager (ProjectCfg.exe)).

In order to check whether modules that are not visible, like the DMS in this case, are running, they are represented on the Windows taskbar in the bottom right of the screen by dots of various colours.

The DMS (Data Management System) module is running.

The PDBS module (database of Visi.Plus) is running.

ALM module (Alarm Manager) is running.

The PRT module (Log Manager) is running.

The HDA (Historical Data Acquisition/trend data logging) module is running.

The Graphical Editor (GE) is an exception. It does not require a dot, because it is responsible for the on-screen display of the visualisation and therefore must be visible.

4.2.1 The first start-up of the Visi.Plus graphic editor

The first time the Visi.Plus **graphical editor** is opened, the following screen appears:

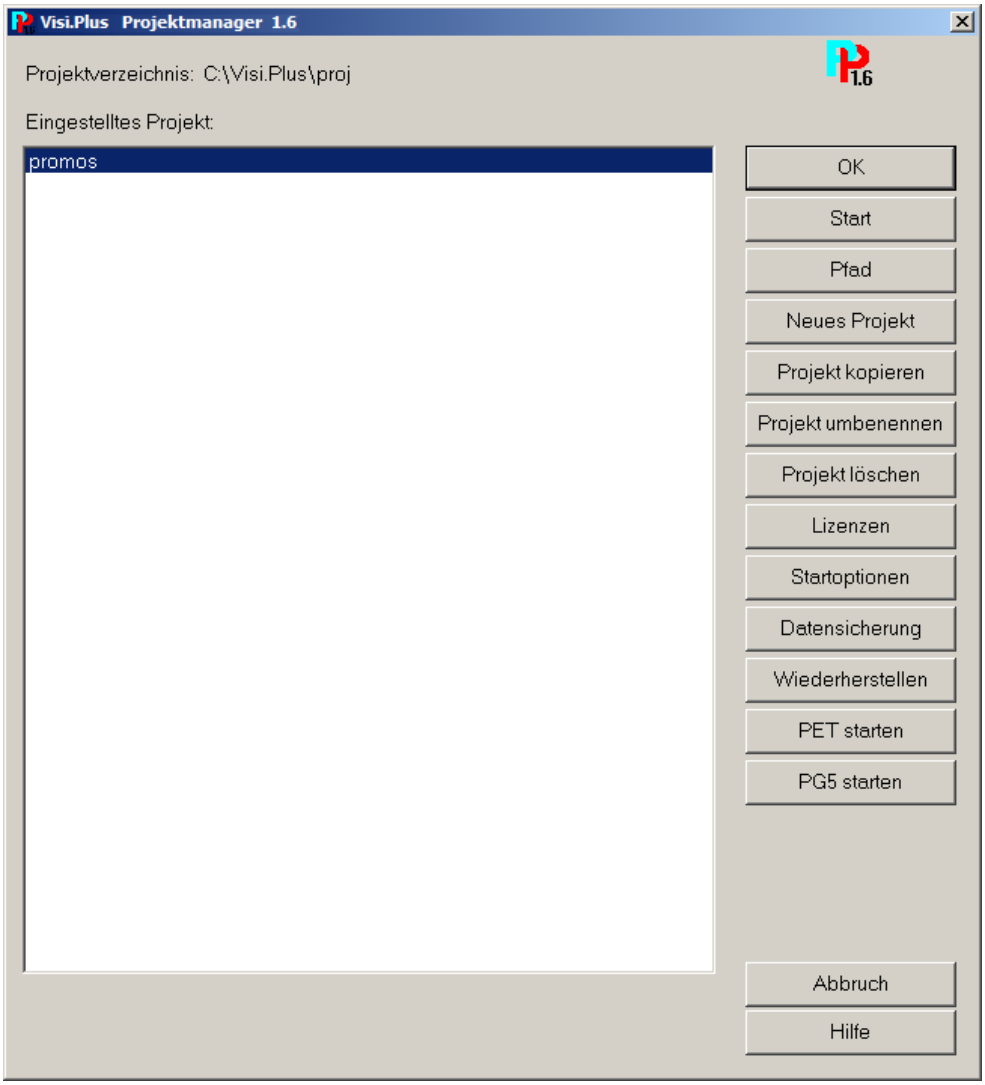
Visi.Plus is introduced using its own graphical resources. The introduction is only short, so it is recommended that you look through its couple of pages. To start the introduction, click on the **<Next>** button.

4.3 The Project Manager (Project.Cfg.exe)

The **Project Manager** module is used for opening or deleting projects, changing existing start options or adding licences:

START > Programs > Saia.Visi > Project Manager

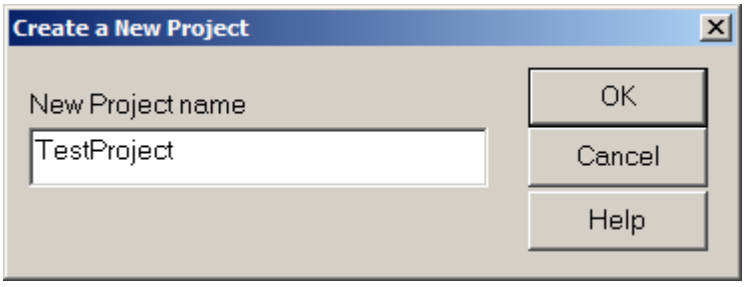
The following control window is displayed on startup of the Project Settings module:



All projects located in the **\proj** folder in the installation directory of Visi.Plus are shown in the white display window.
The current project is displayed next to the text **Set Project**. The set project serves as a default for the next startup of the graphical system, the DMS or the PET.

4.3.1 Creating a new project

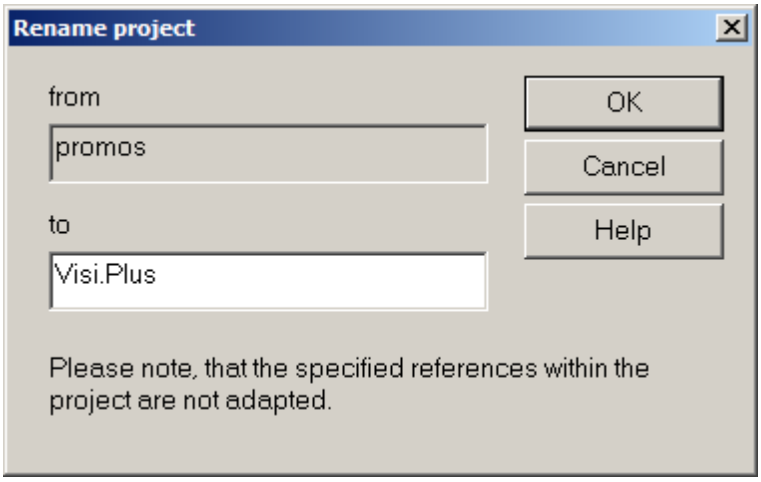
This purpose is served by the **<New Project>** button.



The **Create New Project** dialog window is displayed. Enter a project name in the input field under **Name of the New Project**. It is wise to only enter names without blank spaces or special characters. By clicking on the **<OK>** button, text entry is concluded. The "TestProject" is listed in the Project Settings control window.

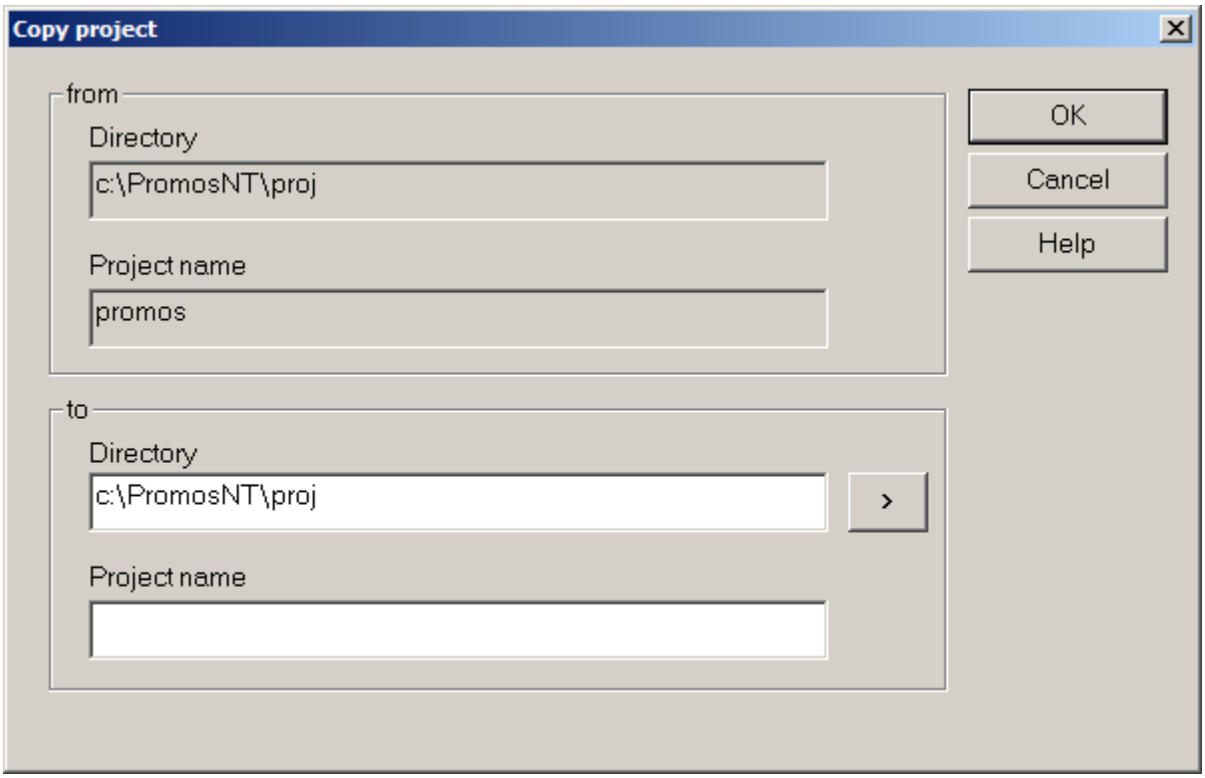
4.3.2 Renaming a project

This purpose is served by the **<Rename Project>** button.



4.3.3 Copying a project

This purpose is served by the **<Copy Project>** button.



4.3.4 Changing project

There are two ways to change project:

1. Select the desired project in the displayed list and click on **<OK>**. The project has now been noted and the **Project Manager** module is exited. At a later point in time the Graphical Editor, DMS or the PET can be called up right from the current project.

2. As described under **1.**, but this time instead of clicking **<OK>** click on **<START>**. The project with its settings is started immediately in the modules (including the DMS) defined under the **<Start Options>** button.

If a valid licence is installed, the PDBS module (the database of Visi.Plus) is also started automatically. A green dot appears on the Windows taskbar.
By double-clicking on a project, it can be started immediately.

4.3.5 Editing a network project

A Visi.Plus project is not always located on the local computer. Therefore it is possible to enter the directory path using the **<Path>** button.

For resource reasons, however, we recommend editing current projects locally! If large projects are edited over a computer network, it places a very heavy burden on the network!

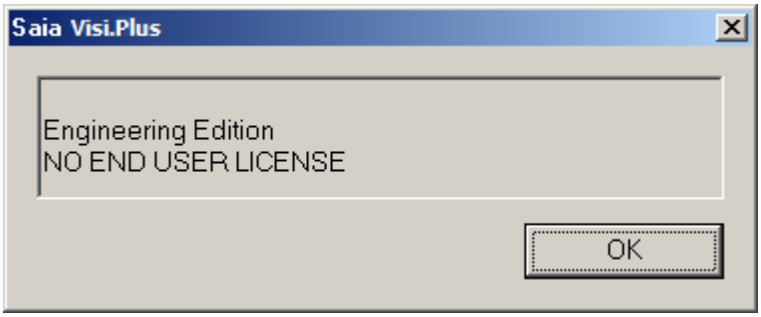
By default, all Visi.Plus projects are located in the directory "**C:\Visi.Plus\proj\...**". Therefore it is logical to create the project on the local computer and then install it on the network.

4.3.6 Installing a project licence

Every Visi.Plus project requires a licence file. This licence file can be purchased from local representatives or the supplier of this software.

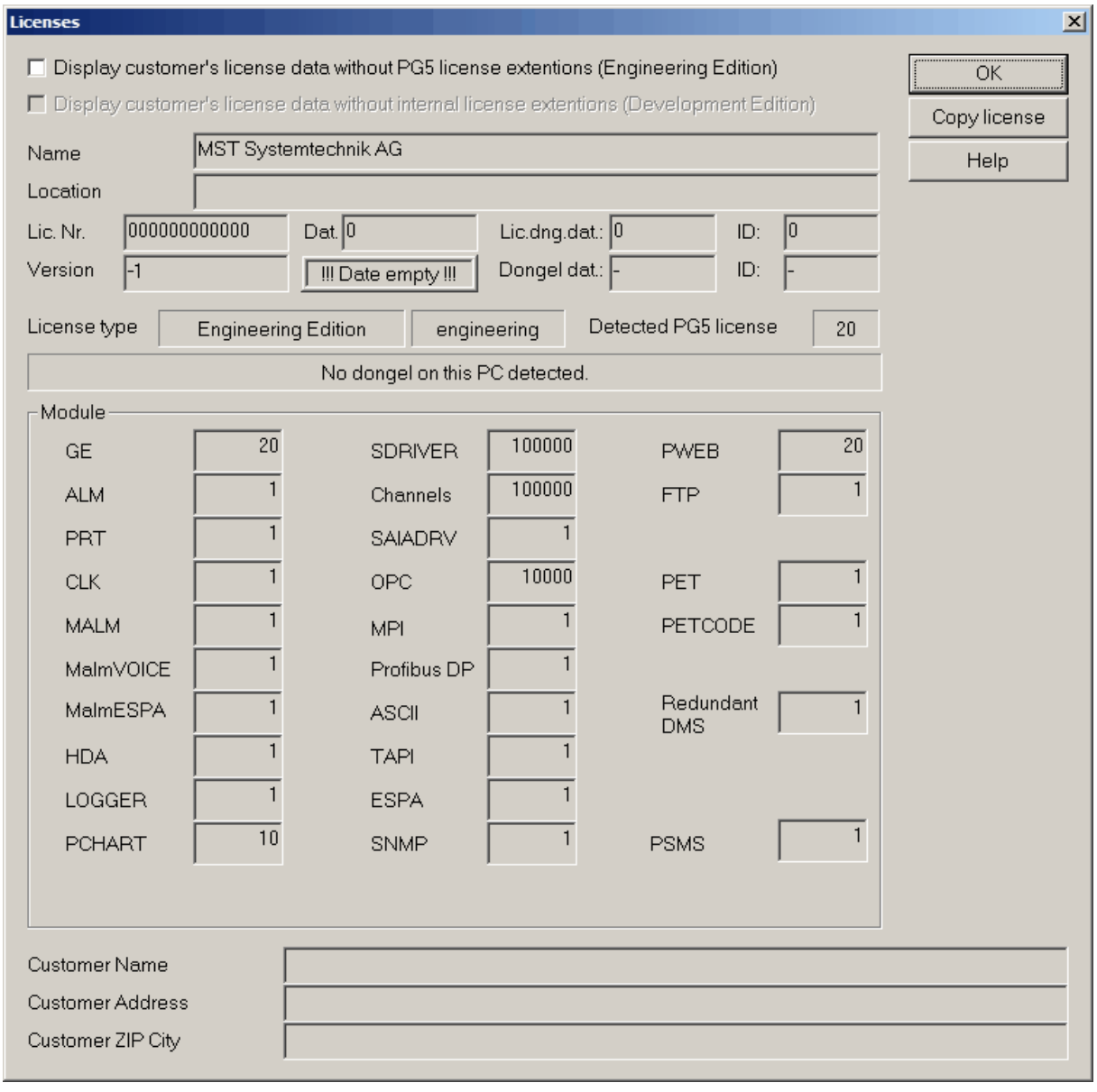
If no license file is installed for the respective project, Visi.Plus runs in demo mode until a predefined deadline.

***Exception**: If a valid PG5 V. 2.0 (or higher) licence is installed on the PC, this also applies for Visi.Plus automatically. Visi.Plus then runs in the **Engineering Edition** mode: All module options are activated, no USB dongle is necessary and there is no Visi.Plus license file. A splash window appears on the DMS server every 24 hours:*

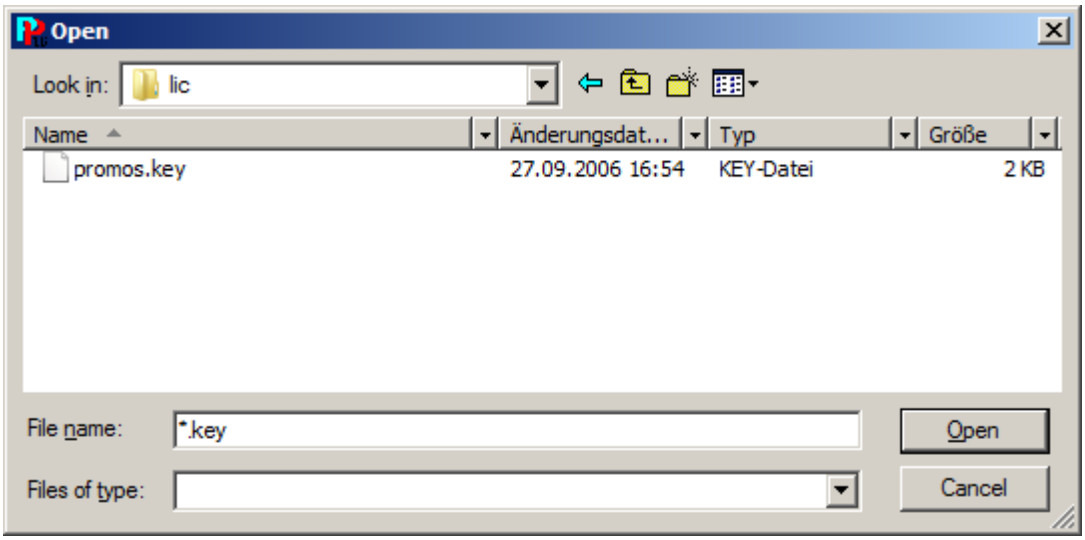


The same splash window appears for every login and GE.

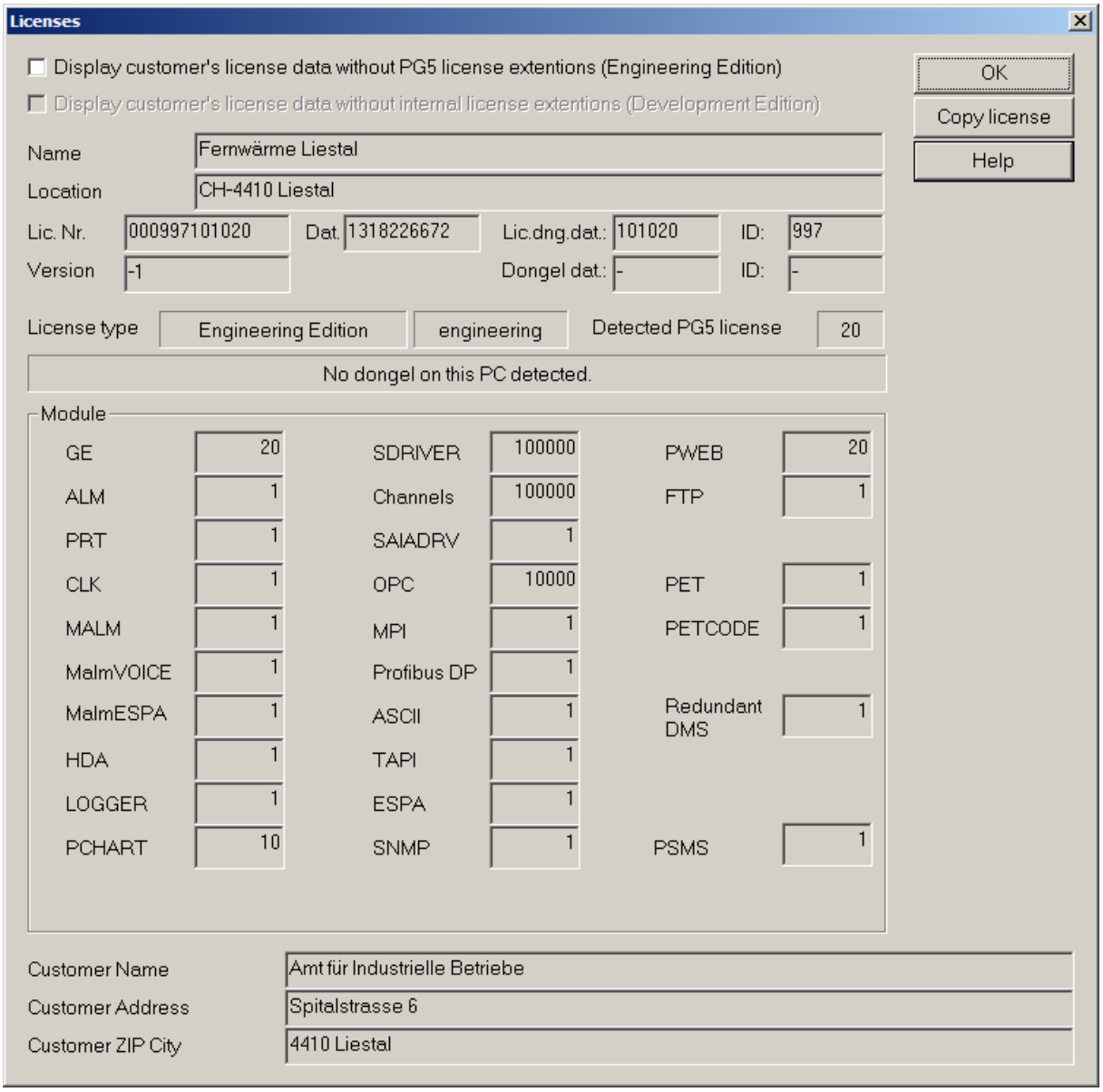
The licence file is installed as follows:
Starting the **Project Manager module:**
START > Programs > Visi.Plus > Project Manager
Select the respective Visi.Plus project and click on the **<Licences>** button. The fields of the **Licence Display** window are still empty if a licence has not already been installed.



In order to install the purchased licence file, click on the **<Copy Licence>** button. The window **Open Licence File** appears:



Navigate to the directory where the Visi.Plus licence to be installed is located. Select the licence file and click the **<Open>** button. Now all purchased licence data is shown in the **Licence Display** window.



In the **Licence Display** window the licencee is shown in the upper part.

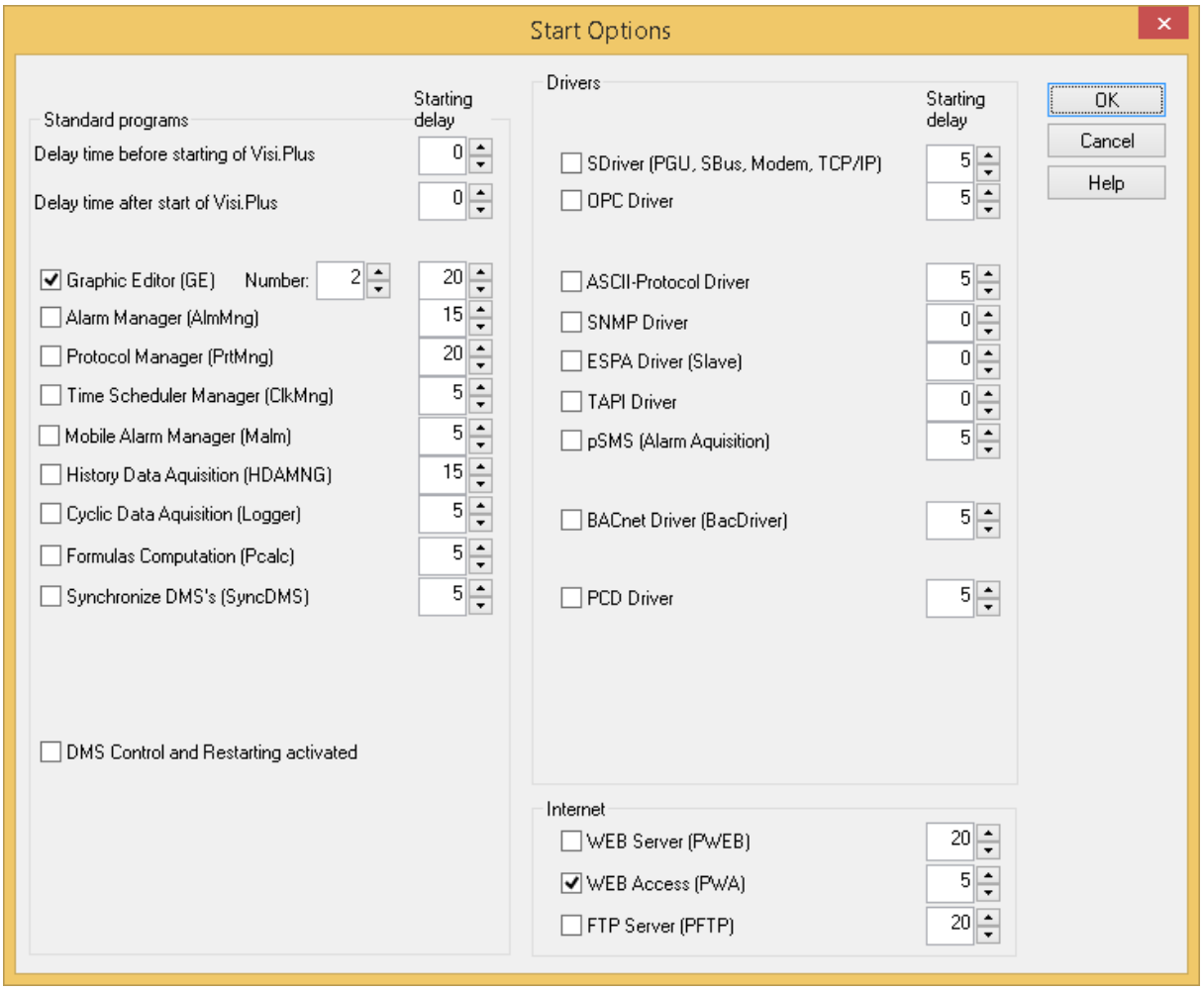
In the **Module** text frame, the number after the respective software module indicates how many licences were purchased for the corresponding software module.
In the lower part, the customer data is shown.
The two checkboxes can be used to switch off both the PG5 licence and the engineering licence. This makes it possible to use the USB dongle to check whether the project functions properly with the purchased licence.

*If a demo licence is installed, the expiry date for the licence is shown to the right below the **<OK>** button. The enabled modules can be tested until this date. Afterwards, a new licence must be installed.*

4.3.7 Defining the programs to be started

Visi.Plus is a collection of many individual modules (programs). To determine which modules should also be opened when Visi.Plus starts up, use the **<Start Options>** button in the **Project Manager** module.

The **Start Options window opens:**



All software modules which have been purchased under the terms of the **licence** have been enabled by installing the licence file. All modules displayed in grey cannot be used with the licence currently installed.
By clicking on the checkbox next to a particular program, it will be automatically started the next time a project is opened. It is also possible to specify for the graphical editor how many GEs should be started.

If the checkbox "DMS Control and Restarting activated" is set the TrayIcon of promos.exe is still existing after the start up process:

This application is controlling the DMS after the startup process. In case DMS isn't answering (internal Watchdog for one minute) the project will be automatically restarted. All other module (except pWA, PCDDriver who are connecting automatically new if the DMS is running again) will be terminated first. All other modules are controlled by the DMS itself.

In the previous image, you can see that, the next time the program is started the Graphical Editor (GE), Alarm Manager (AlmMng), Log Manager (PrtMng), and Remote Alarm Manager (Malm) programs, the SDriver and the two internet modules (Web Server and FTP Server) will also be started.
By clicking on the dropdown menu for the respective program module, the start delay period can be set. If a project contains a great deal of data, you can set a delay period to ensure that, for instance, the Alarm Manager is only started once the data from the SDriver has been loaded into the DMS. In this example, the Alarm Manager is started 5 seconds later than the other modules and the Remote Alarm Manager 10 seconds later.

When the Visi.Plus Graphical Editor is started, the DMS (Data Management System, which is the process database) and the GE (Graphical Editor) modules are started.

4.3.7.1 Starting additional (external) programs

You can integrate any programs into the startup sequence:

Open the configuration file **start.cfg** (in the \cfg project directory) in an editor (e.g. Notepad) and add the following lines to the end of the file:

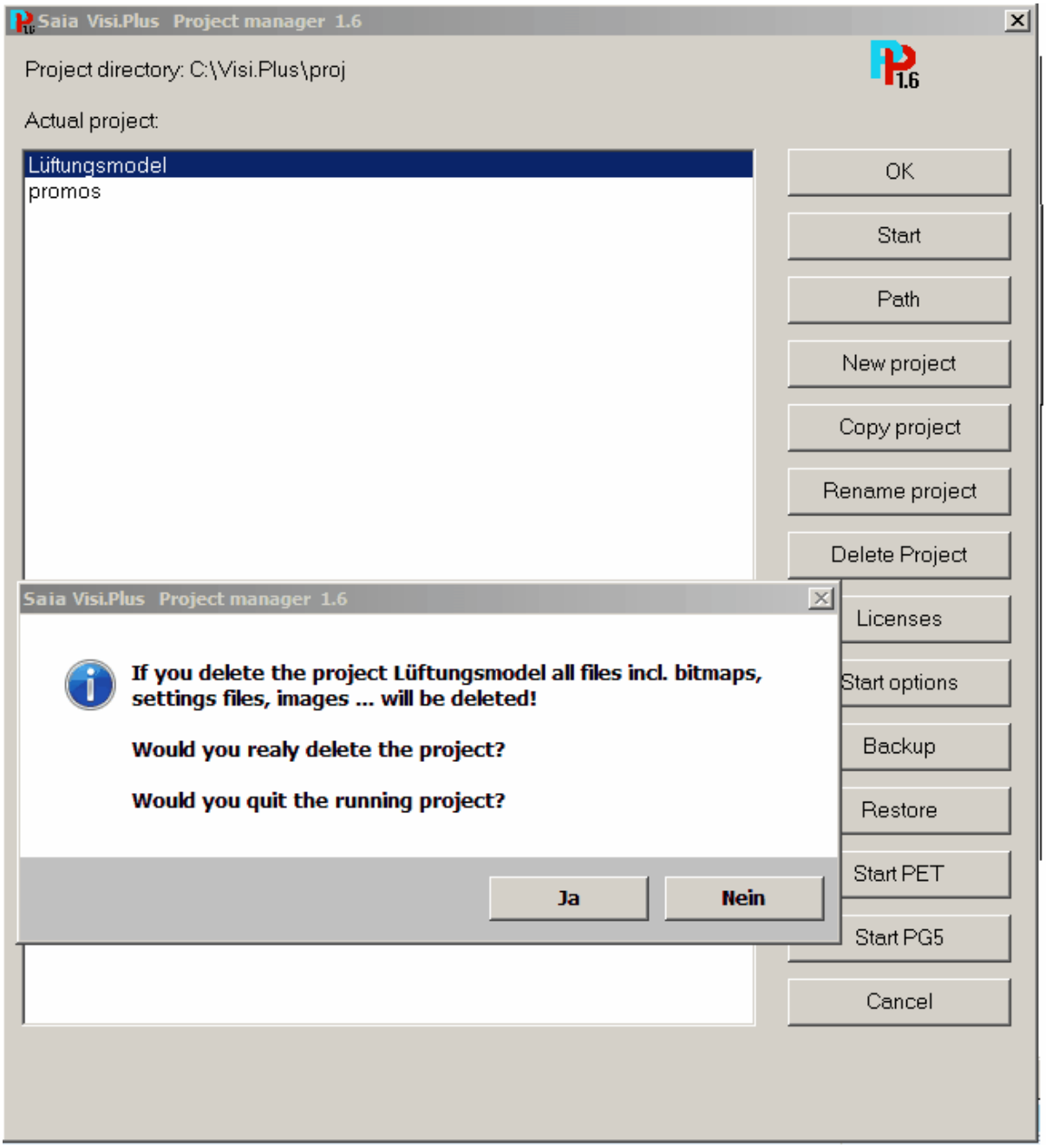
- New section:
  [START]
  (if it does not already exist)
- Corresponding program name with prefixed number (example):
  1=C:\Windows\system32\notepad.exe

Any other program is listed with a sequential number (example):

- **2=C:\AppletServer\NtRegEdit.exe**

4.3.8 Delete project

In order to delete a project, select it in the **Project Manager** module and click on the **<Delete Project>** button.

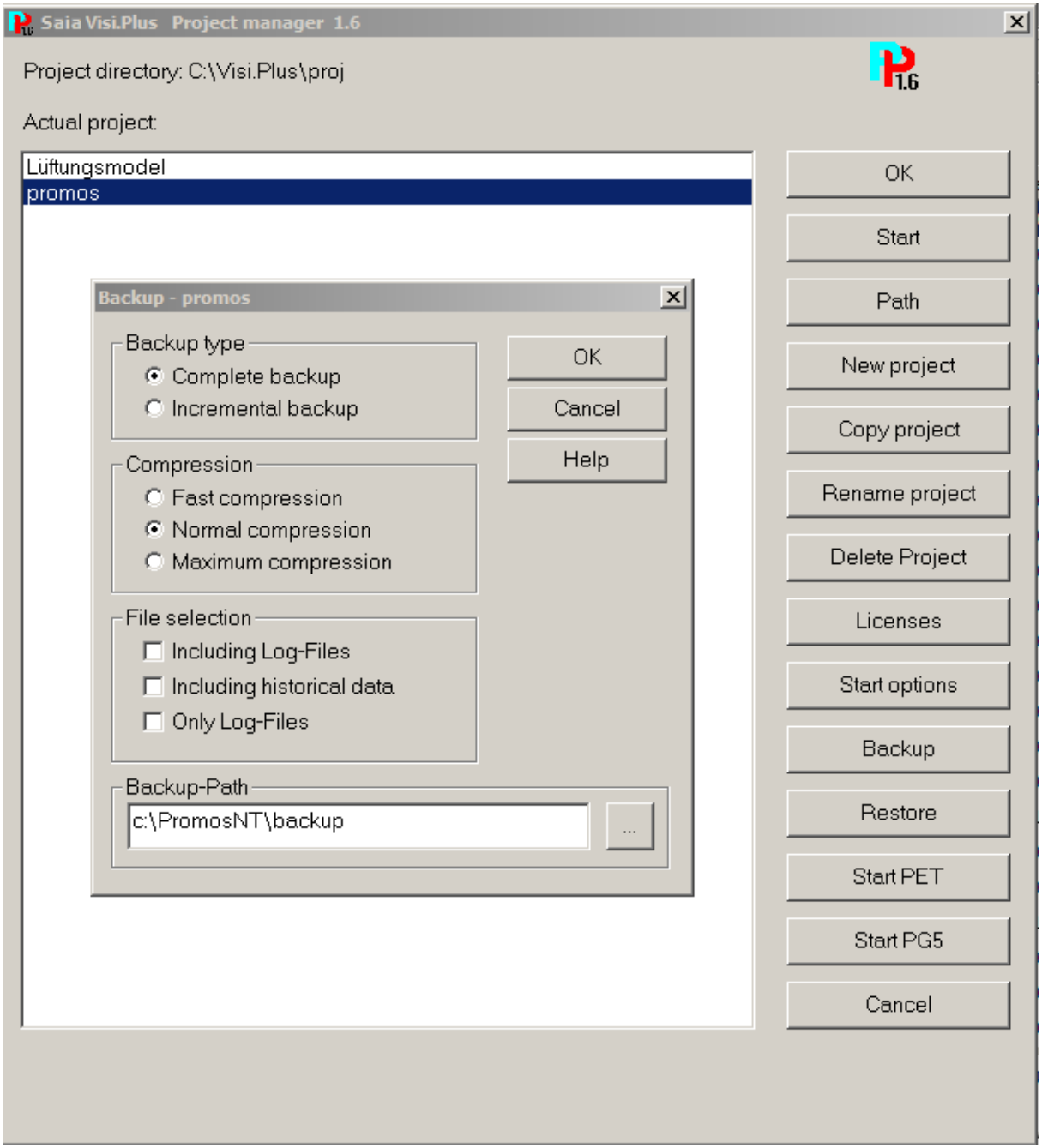


The confirmation message that appears should be answered accordingly.

4.3.9 Creating a project backup

Visi.Plus offers the possibility of saving all data from a project directly to a ZIP file.

Select the project to be saved in the Project Manager and click on the **<Backup>** button. The following window is displayed:



In this example a backup of the project **promos** is generated.

You can select whether a complete save or only an incremental save (only changes) should be generated under **Backup Type**.

The degree of compression of the ZIP file can be selected under **Compression**.

Under **File selection**, a selection can be made as to whether the project is backed up with the log files and the trend files or only the log files.

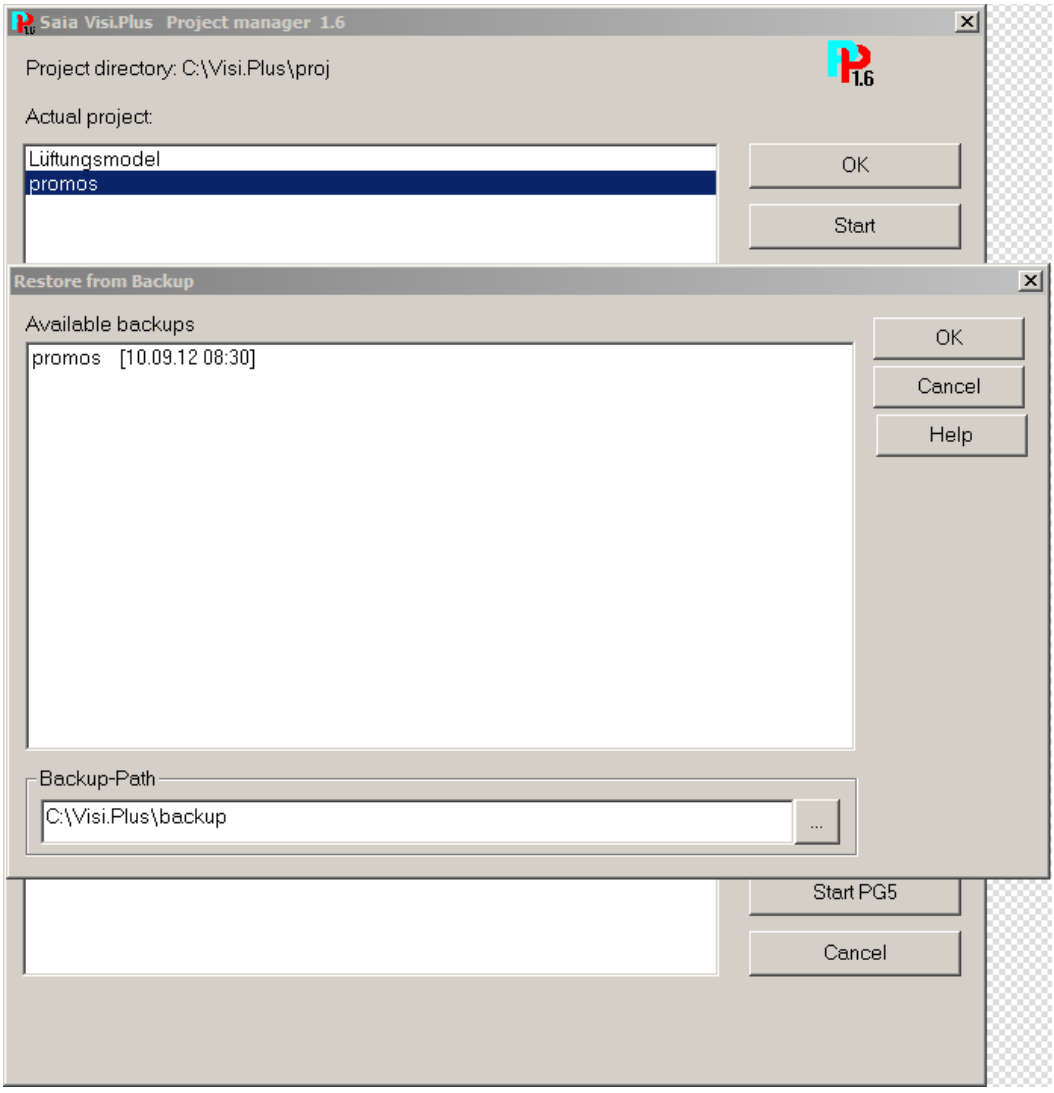
Under **Backup path**, the button next to the entry field can be used to conveniently indicate the path where backup files are to be stored.

By clicking on the **<OK>** button, a ZIP file with the following name structure is generated in the indicated path:

Projectname_VERNumber_Date Time.zip
Example: **promos_VER_06.08.03 14.34.zip**

4.3.10 Restoring a backup

To restore a backup, the **<Restore>** button should be used.



All backups from the directory specified under "**Backup Path**" are displayed for selection. After selecting the path, press the **<OK>** button to perform the restore operation.

4.4 Windows autostart

Windows can be set up to boot the entire system (computer) on start-up, without the user having to log on. Three registry entries must be adjusted to do this.

The following entries should, of course, only be made by experienced persons, and

after the registry has been backed up.

4.4.1 Autostart setup in the registry

- Start up Regedit (in the Windows Start menu under **Run**, enter "Regedit" and click on **<OK>**).
  Make modifications under:

- "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion \Winlogon\AutoAdminLogon"
  set value to logical 1.

- "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\WindowsNT\CurrentVersion \Winlogon\DefaultUserName"
  enter the default user name.

ATTENTION: The user **MUST** have administrator rights, otherwise logging on as an administrator will no longer be possible.

- "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\WindowsNT\CurrentVersion \Winlogon"

- Create a new character sequence of the type REG_SZ (right mouse button > **New**) with the name **DefaultPassword**.
  The value should be the password of the default user.

On many computers it is possible to make bios settings to ensure that the computer will start up again automatically after a power failure.
This function should be switched on for all computers in the system.

4.4.2 Autostart setup under Win XP (GUI)

In Windows XP (depending on the service pack) setup of the autostart can also take place through a graphical interface. In order to do so, proceed as follows:
In the **Start > *Run*** dialog, enter the following command:

control userpasswords2

The following window is displayed:

This hidden user dialog contains useful functions.

To set up autostart, deactivate the checkbox **Users must enter a user name and password to use this computer**.
Confirm by clicking **<*OK*>** or **<Apply>**.

Afterwards follow the steps on the screen:

You will be prompted to enter the respective password for the user name twice.
After entering the password, confirm by clicking **<*OK*>**.
Then exit the user management module.
After the computer has been restarted, the user is automatically

logged on with the previously entered user name.

As an alternative, there are freeware and shareware applications that can select these settings in a simpler way, e.g. ***TweakUi.exe*** *from Microsoft.*

.

Saia Visi.Plus

Engineering

Chapter 5

5 Engineering

5.1 Summary

The following topics are described in this chapter:

- What is an AKS system and how is it used in Visi.Plus?
- Logic, utilisation and creation of template objects
- Use of existing template objects
- What process diagrams are and how they are created
- How the code generators of Visi.Plus are used

5.2 AKS system

Every data point must have a unique name. In Visi.Plus this name is called the **DMS name** and it originates from the **AKS facility designation system (in German: Anlagekennzeichnungssystem).**

Example:
B10:H01:MT:511

Meaning:
Building B10, Heating 1, Measurement Temperature, Sensor 511.

The **AKS** is advantageous because of its conciseness. It simplifies and facilitates the engineering as well as the documentation and the electrical diagram of the application. As a result, the desired data point can be accessed quickly and with certainty at any time in case of alarm or fault.

*In order to avoid extensive searching for a valve or a pump or to provide more information about a device, affixing the **AKS as a name tag on the device,** etc. is recommended.*

The **AKS** should be used throughout the entire company and not just on a single installation.

Tip: End customers probably already have their own AKS in place. It is best to enquire with the customer.

5.2.1 Structure of an AKS system

In Visi.Plus an **AKS** (**DMS**-) designation has the following typical structure:

Location:System:Type:Number:Addendum

Example:
B10:H01:MT:511:Addendum

The **AKS** key can have whatever structure you wish.

Important to note is that:

- The key must bear a clear reference to the system parts.
- The designations must be separated by a **colon**.

*Important: In Visi.Plus, AKS (DMS) designations **must** be separated by a **colon**!*

If an AKS designation table has never been developed, the following AKS designation table will likely help. The AKS designations used there have been successfully applied in many systems. Of course they can be adapted to your requirements.

5.2.1.1 Other delimiters

In some particular systems, other delimiters are required for differentiation. By default, Visi.Plus only supports a colon as a delimiter.

However, it is possible to define other delimiters for the outputs. For this purpose, a file named DELIMITER.CFG must be created (using a text editor) in the CFG directory of the project with the following content:

```
[Delimiter]
1=!
2=?
3=;
4= " "
5=+
6=-
10=@->
7=%%%
```

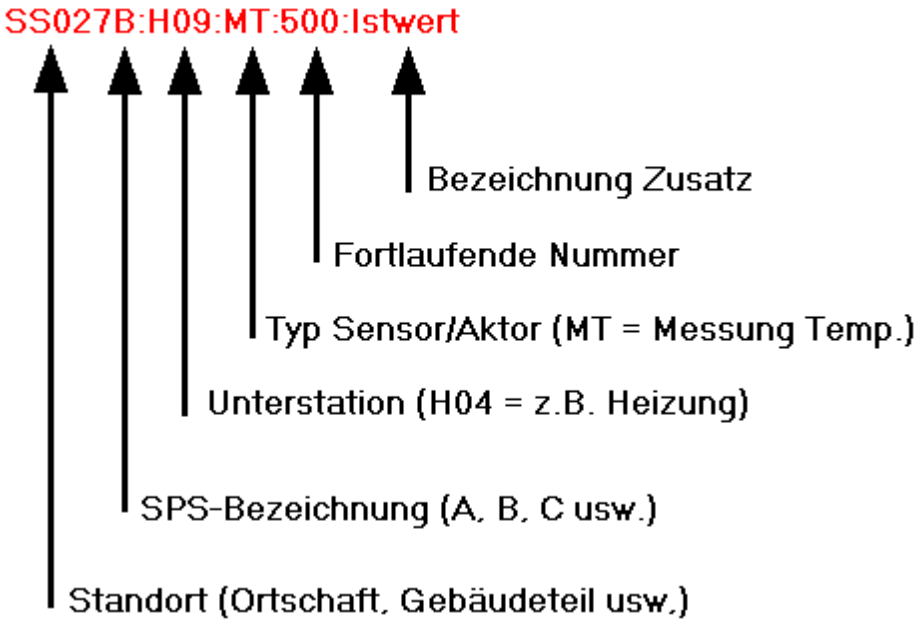
By doing so, a DMS name will be represented differently: The first delimiter is replaced with the symbol that was defined at 1=. The second delimiter with the sign that is defined at 2=, etc.
Levels that are not defined are not converted (colon remains).

Example: `WL027B:H02:MT:500:ActualValue:Comment`

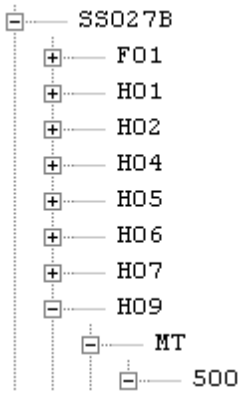
is represented as `WL027!M02?MT;500 ActualValue+Comment`

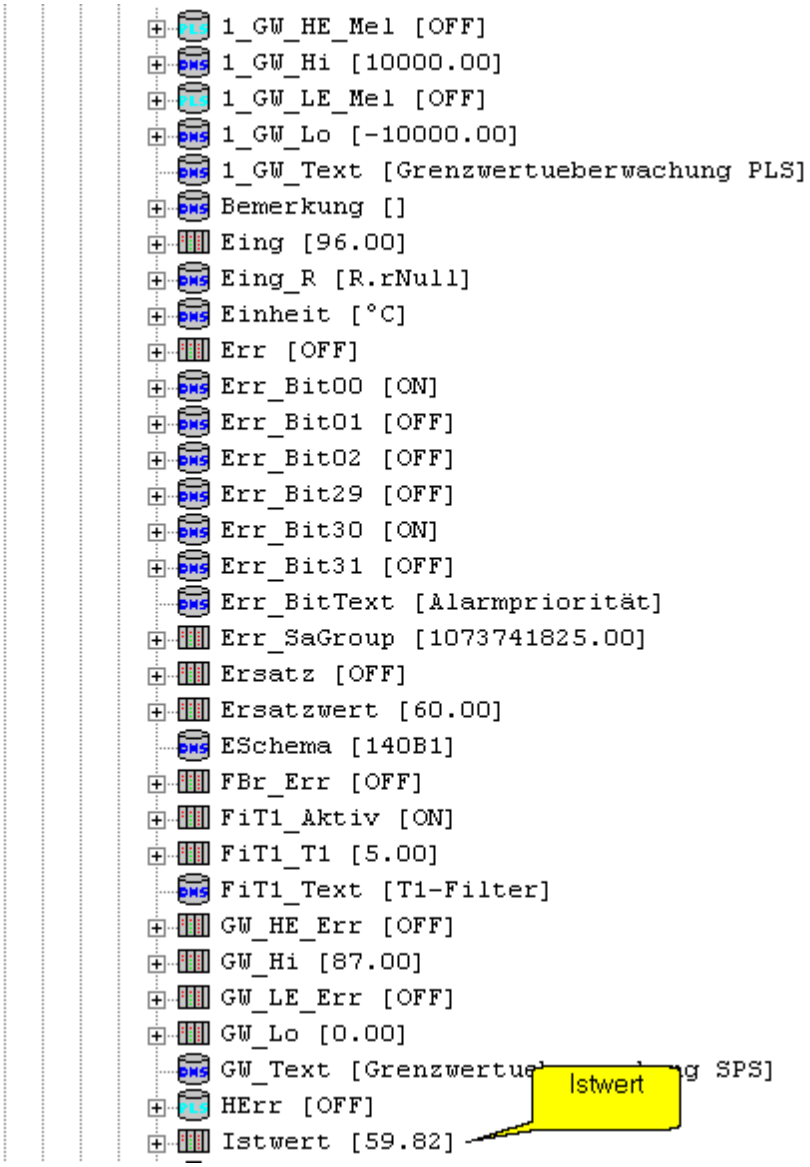
5.2.2 AKS key

AKS facility designation system



The structure is integrated 1:1 into the DMS (Data Management System).





Location
Stands for location, e.g. a building or the PLC station number.

Example:
B10 (Building10)
C01 (C stands for controller or PLC)

Attention: The first character should not be a number! (PG5 has problems with numbers)
Important: Select the shortest possible designations.

Further examples for local designations are:
EW East Wing
WL027 Wipkingen, Lettenstrasse 27
PA03 Hall PA, third floor
PA00 Hall PA, ground floor

PA99 Hall PA, 1st basement floor
PA98 Hall PA, 2nd basement floor
etc.

The AKS key "location" should be unique for each PLC!
(if code generation will be used)

Depending on the system size, the AKS key may have to be structured differently. If many buildings in different locations must be described, defining the locations in the AKS key is recommended as well.

Examples:

CO022:H02:MT:500:ActualValue	Chur, Ottostrasse 22 = CO022
ZL027:H02:MT:500:ActualValue	Zürich, Lettenstrasse 27 = ZL027
ZLP027:H02:MT500:ActualValue	Zürich, Läuferplatz 22 = TLP027
BKFD354:H02:MT:500:ActualValue	Berlin, Kurfürstendamm 354 = BKFD354

If multiple PLC controllers are present in each building, this can be expanded e.g. with letters:

Example:

WL027A:H02:MT:500:ActualValue	Zürich, Lettenstrasse 27, first PLC
WL027B:H05:MT:500:ActualValue	Zürich, Lettenstrasse 27, second PLC
WL027C:H09:MT:500:ActualValue	Zürich, Lettenstrasse 27, third PLC
etc.	

It is **important** for code generation purposes that the AKS key differs uniquely at the beginning.

Reserved word for the System
There exist some word for the System. Please do not use this word in the AKS:

eschema Reserved word for Eschema (German version)
scheme Reserved word for escheme (English version)

5.2.2.1 Facility: designation of the facility as a code

Example:
C001:**H**01
H for heating and hot water facility. The facility type is followed by numbers which uniquely identify each part of the facility.

A Security
B Lighting
C Communication
D Miscellaneous
E Electrical
F Fire alarm components
G Gas
H Heating and hot water

I	Installation bus (EIB)
J	Transport equipment (lifts)
K	Cooling
L	Ventilation/air conditioning
M	Medium-voltage system
N	Emergency power
O	Oil

The following conventions have proved useful in numbering:

01	Source
02	Heat pump
03	Buffer store
04	Boiler
05-08	free for other facility parts
09	General
10	Expansion
11-14	free for other facility parts
15	Recharge hot water
16-19	free for other facility parts
20-99	Heating groups

5.2.2.2 Type of actuator/sensor

Example:
S001:H01:**MT**
M stands for measurement and **T** stands for temperature.

1st place		**2nd place**	
A	Drive incl. motor	B	Burner
		G	Bellows
		G	Gear unit
		H	Lifting device/crane
		K	Compressor
		L	Lift
		V	Fan
		W	Roller drive
V	Valves	R	Electrical control valve
		S	Electrical adjustment valve
		P	Pneumatic adjustment valve
P	Pump incl. motor	O	Oil pump
		W	Water pump

1st place		2nd place	
		T	Immersion pump
		G	Group pump
G	Device without motor	B	Humidifier
		D	Pressure tank
		E	Softener
		F	Filter
		F	Air filter
		H	Horn/bell
		I	Injector
		K	Condenser
		O	Oil tank
		P	Sampler
		T	Dryer
		V	Safety valve
		W	Heat exchanger
R	Reactor	B	Reservoir
		M	Vat
H	Heating	B	Boiler
		L	Pipe, channel, passage
		O	Furnace
		R	Room
		S	Radiator
		W	Heat pump
M	Measurements	A	Energy measurement
		B	Output/flow
		B	Temperatures
		D	Tension (web tension)
		D	Density
		F	Flow rate
		H	Brightness
		I	Current
		K	Time
		L	Level
		M	Humidity

1st place		2nd place	
		N	Output
		P	Pressure
		Q	Concentration, content
		S	Rotational speed
		S	Speed
		T	Temperature
		U	Voltage
		V	Viscosity
		Y	Diffusion
		Z	Status
Y	Switch, sensor	D	Density
		F	Flow rate
		I	Current
		K	Time
		L	Level
		M	Humidity
		N	Output
		P	Pressure
		Q	Concentration, content
		S	Rotational speed
		S	Speed
		T	Temperature
		U	Voltage
		V	Viscosity
		Y	Diffusion
		Z	Status
Z	Counter	F	Flow rate
		I	Current
		K	Time
		N	Output
		S	Running metre
		W	Heat measurement
		Z	Status
S	Controller (not accessible)	C	Heating, cooling circuit controller

1st place	2nd place	
	P	PID controller
	S	Target value
	F	Flag
	R	Register
	Z	Status (PLC01)
	B	Bus components (M-bus master)
L	G	Components of the logical controller

5.2.2.3 Example:

Example:
C001: or
C002: stands for controller or PLC station 1 or 2
- Tank pump 1 C001A:PO:001
- Tank pump 2 C002A:PO:001
- Discharge valve T1 C001A:VS:001
- Discharge valve T2 C002A:VS:001
- Ventilation valve C001A:VS:002
- Temperature Tank 1 C001A:MT:500
- Temperature Tank 2 C002A:MT:500

AKS number:
Each sensor and actuator is provided with a successive number (e.g. three-digit), whereas actuators (motors, etc.) begin with the number 001. The sensors are provided with numbers starting from 500. This allows for differentiation on the basis of the AKS numbers.

Example:
S001:H01:PW:**001**

Attribute:
The attribute corresponds to the attribute of the objects and is usually written out completely. The individual signals are defined with the help of the attributes. The attribute describes a physical or virtual signal..

Example:
S001:H01:PW:001:**HOp**

Possible table for this:
- HOp (Hours of operation)
- On (Confirmation of the motor)
- Fault (PLC fault input, no confirmation)
- Start (Motor is started, if manual operation)
- Manual (Manual operation)

- Release (Motor release)
- Inspection (Motor inspection)
- Ack. (Fault acknowledgement)

5.2.3 Examples of AKS designations

Example 1
A temperature sensor located at 27 Lettenstrasse in the city of Wipkingen, belonging to the plant component H04 (e.g. the boiler), would have the following AKS designation:

WL027:H04:MT:500

The sensor has a series of addendums or attributes that refer to the individual signals:

Actual value	WL027A:H04:MT:500:Current
Limit value HI	WL027A:H04:MT:500:Hi
Limit value LO	WL027A:H04:MT:500:Lo
LV HI reached	WL027A:H04:MT:500:HiErr
LV LO reached	WL027A:H04:MT:500:LoErr
Fault (sensor break)	WL027A:H04:MT:500:Error
Default value	WL027A:H04:MT:500:Default
Default value On	WL027A:H04:MT:500:Default On

Example 2
For an actuation command of a 2-stage motor in ventilation system 01, located at 27 Lettenstrasse, Wipkingen:

Stage 1 Off/On	WL027A:L01:AG:001:Start_1
Stage 2 Off/On	WL027A:L01:AG:001:Start_2

Example 3
For a set point command of a frequency converter for a pump in distribution control plant H03 located 27 Lettenstrasse, Wipkingen:

0 - 100 %	WL027A:H03:PW:004:Target

5.3 Template objects

5.3.1 What is a template object?

A template object is a parent object that can be completely copied so that it does not have to be redrawn each time. Each system consists of different object types, such as pumps, valves, analogue measurement points, etc., which always have the same attributes as objects. A pump object, for example, always has a Fault attribute. All actuators/sensors for a system should be defined using template objects (parent objects) insofar as possible. Entire groups of signals can be combined, e.g. heating groups, motors, pumps, measurements, etc. The engineering expenditure is thereby significantly reduced. As a result it is possible to

automatically generate a large portion of the PLC code.

5.3.2 How are template objects defined in a project?

First, a list of all sensors and actuators which are available in your project must be created. This list should be complete, because any later addition always means greater effort. The list is used to combine the signals for different template objects, to great advantage. A large system has a maximum of 15 to 25 different template object types. The goal is to have as few template object types per project as possible.

Both dynamic and static values, which can be used for the documentation of the system or as notes for the operator/programmer, can be added to each template object.

If more than 20 objects are required in order to realise the project, the list should be revised for reasons of clarity. This may mean, for example, that a fan can also be treated as a motor.

If the list is to be prepared in a spreadsheet, the rules of the PET must absolutely be adhered to so that the project data can be directly imported. How the table must appear and how the rules are defined is described in the ***chapter PET****.*

5.3.3 Example template object: Motor

A user-provided object may appear as follows:

- Actual value — Status display: On or Off
- HOp — Registers hours of operation
- Circuits — Registers the number of circuits
- Faults — Faults (coil, LS triggered, overheating, jamming, etc.)
- Manual operation — Display of manual or automatic operation
- Inspection — Motor inspection
- Release — Motor release
- Acknowledgement — Motor fault acknowledgement
- Diagram number — Electrical diagram number
- PLC_inputs — Absolute address of PLC inputs (important for code generation)
- PLC_outputs — Absolute address of PLC outputs (important for code generation)
- Designations — AKS designation
- Comments — Special attributes
- Slow-down time — Input of a slow-down time during which the motor cannot be switched on again

A template object does not consist of its data points alone. If a template object has been defined and generated in the PET, the corresponding control and information screens must be drawn in the graphics editor and initialised to your template object (see chapter Graphical editor - GE).

5.4 Working with template objects

The Visi.Plus system has four standard template objects (ANA01, DIG01, MOT01 and MES01) which it can work with directly once they have been integrated.
A description of how users can define their own template objects is found in this manual in the **chapter** **Project Engineering Tool (PET)**.

*As mentioned in the **chapter** **Installation of template objects** complete template objects for the Visi.Plus system can be purchased from a template object library. The engineering costs are thereby considerably reduced, because experience has shown that additional objects no longer have to be defined. Information can be found on the Visi.Plus Internet site www.sbc-support.ch under the heading **Template objects**.*

5.4.1 Using existing template objects

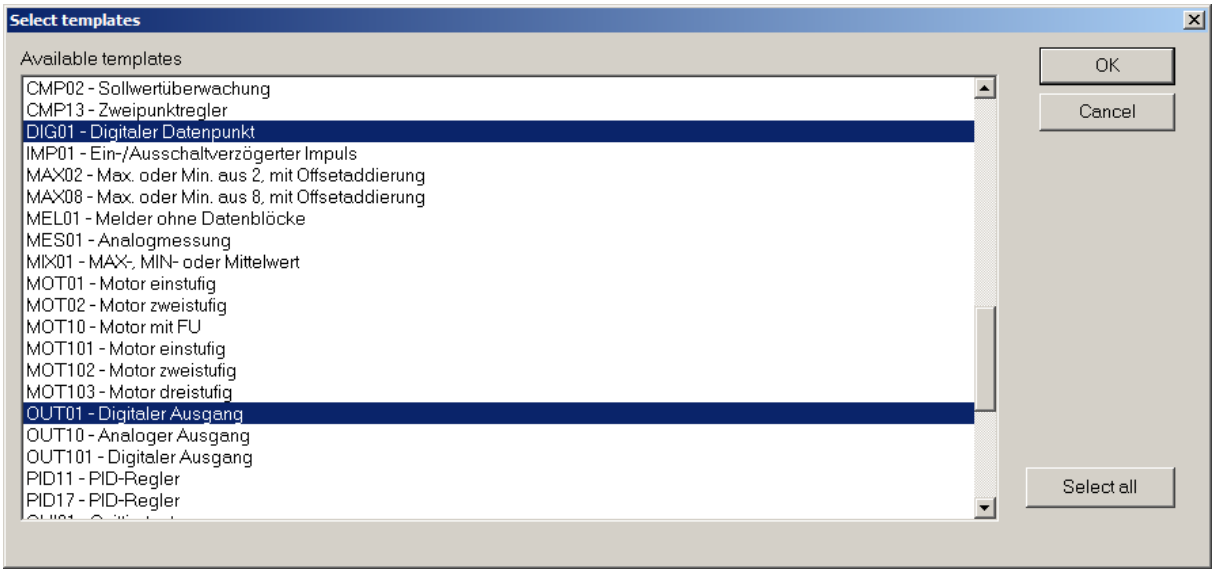
Template objects are inserted into a Visi.Plus project in the module **PET (Process Engineering Tool)**. The PET module offers the possibility of managing all data belonging to a Visi.Plus project. The module registers all existing physical data points (sensors/actuators, etc.).

Before the PET is started, you must check whether the DMS is already located in the main memory of the computer. This is recognisable by a red spot on the taskbar. Those who are already familiar with Visi.Plus and its various modules can carry out the necessary steps in the correct sequence in Windows Explorer. Otherwise, you can proceed as follows:

The simplest method is to call up the **Project Manager module (projectcfg.exe).**Here you can open an existing project or create a new one. Under **Start Options** the **GE** (graphical editor) box should be checked so that it automatically starts up. After clicking on Start, Visi.Plus is started with the DMS and GE.

 In the graphical editor (GE), the PET can now be started using the **PET** icon on the toolbar. Alternatively, the PET can also be started from the Windows start menu.
In order to integrate template objects from existing libraries, proceed as follows:

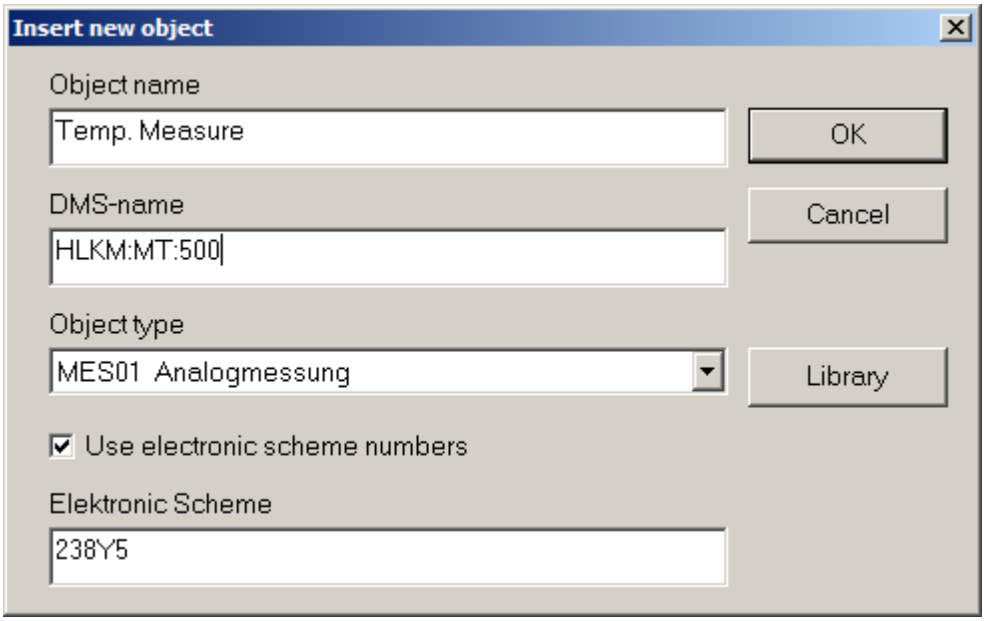
Menu **Template Objects > Templates from Library**. Then the following window appears:



If template object libraries are available, all installable objects will be listed here. The templates **ANA01** and **DIG01** are supplied as standard and can be selected.
After the desired selection has been made, the template objects are installed by pressing **<OK>**.

After installation is complete, the PET window becomes active again.

 To register new system objects in the PET, you must click on the **Insert Object** icon in the toolbar. Alternatively, the **<+>** key or **<Insert>** can be used.
The following image then appears.



 *Before an object type can be added, it must have been installed from the library or created by the user (see* ***Chapter 6*** *Project Engineering Tool (PET)).*

If template objects from existing libraries must also be integrated now, this is possible via the

<Library> button.

System objects (pumps, valves, etc.) can now be allocated to object types with the window shown in the previous image. All objects/data present in the project can be successively registered in this way.

*By pressing the <**Up arrow**> key, the last object designation in each case is shown again in the input field.*

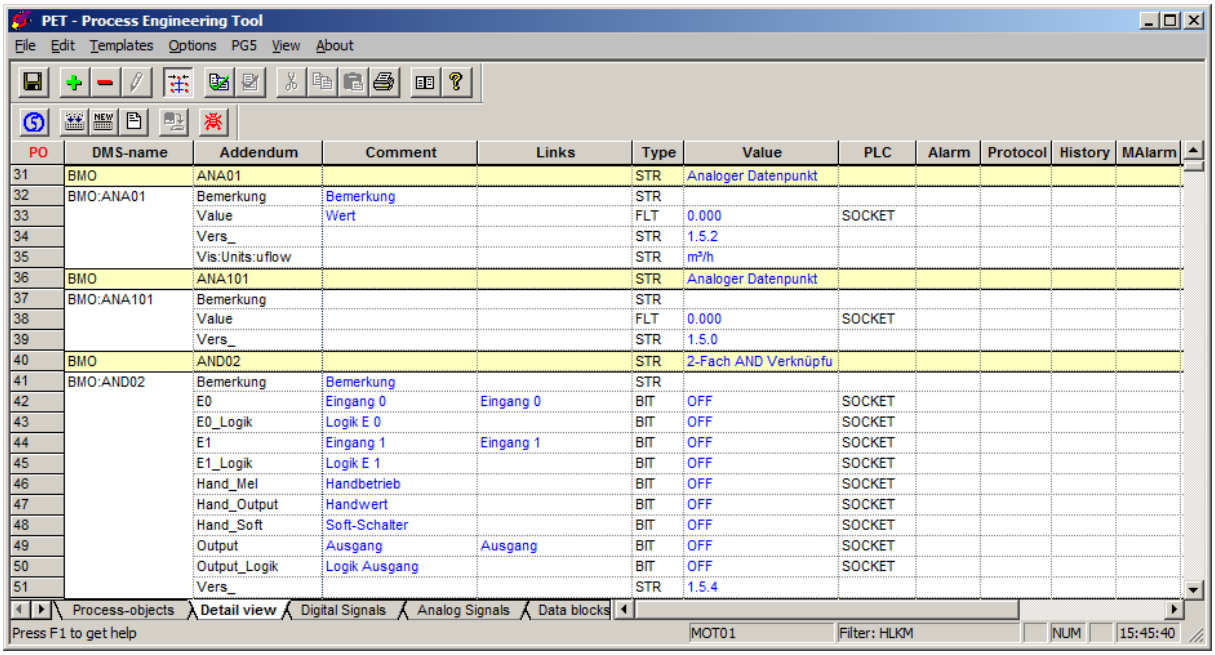
5.4.2 Creating your own template objects

New template objects are created in the **Process Engineering Tool (PET)**.

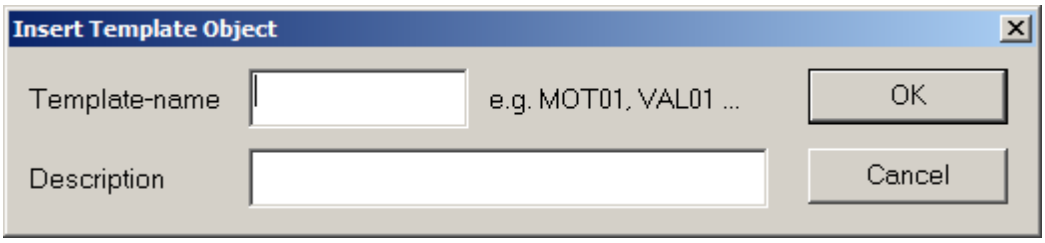
Starting the PET module

The start-up procedure is described in the chapter Starting the PET.

In order to access the creation mode for template objects, select the "**Template Objects**" button.



For each new template object, you must select **Template Objects > New Template** in the menu. The following dialog window appears:



Enter the name of the template object in the **VLO name** input field. Enter a description of the

template object in the **Description** field. In the preceding image, template object "Pump" was created. Complete the entry with <OK>. The pump object is now inserted in the PET in the template view.

If the template that has just been inserted is not visible, you can scroll down using the scroll bar on the right side. As can be seen, only a yellow line was inserted. The object now has a name, but still has no data points (Actuator, Sensors, etc.).

The following names may not be used in Visi.Plus. These are reserved system names!

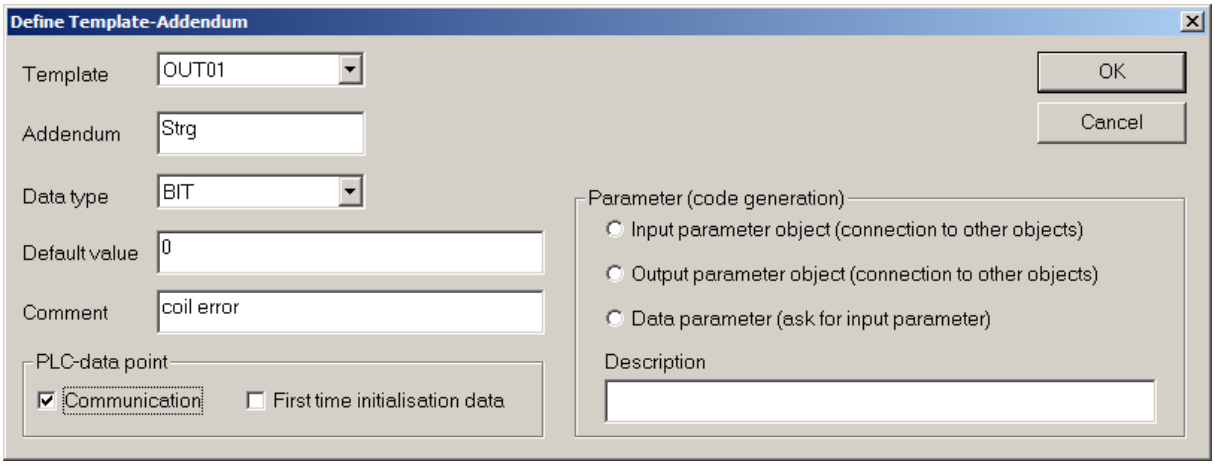
- OBJECT
- PLC
- ALM
- TRD
- PRT
- CLK
- PAG
- Comment
- PAR_OUT
- PAR_IN
- PAR_DATA

In addition, all PCD commands must also be avoided (e.g. "Actual").

The data points (Actuators/Sensors/Data) can be added to the template object in different ways:

- By pressing the green plus icon button
- Via the menu under **Template Objects > New Template Object Attribute**
- Or using the **<+> key on the keyboard**

The input window **Define Template Object Attribute** opens.



Template Object

The object to which an object attribute should be added can be selected by clicking on the dropdown menu.

Attribute
Enter the name of the new data point here, e.g. Status, TargetValue, ActualValue, DiagramNo, etc.

Data Type
The type of data point can be defined by clicking on the dropdown menu.
defined. The data types have the following meanings:

BIT	Binary signals	= marker (flag), inputs or outputs
FLT	Flow point values	= register contents converted by Visi.Plus
STR	Strings	= simple, free text
DWU	Data blocks	= PCD data block (data blocks are treated as **D**ouble**W**ord**U**nsigned)

Default Value
In this input field you can enter a default value that is adopted by the data point as a default value and passed on.

Comment
This field serves for a more precise, free description of the data point. This is shown in the PET in the column with the same name.

PLC Data Point
With the **"Communication to PLC"** checkbox marked, the PLC driver is informed that this data point should be communicated to the PLC.

If the "**Initialise value in PLC**" checkbox is marked, the default value entered in the input field is written to the PLC.

All data can be viewed at a later time.

Parameters (Code Generation)
With the radio button in this section, you can define whether the input or output parameters should be linked to another object. Through these links, associations such as a temperature to a control cycle or the output of the control cycle to a valve, etc. can be made at a later time. Only 1:1 links can be created. These links are generated later by the code generator (and then run on the PLC).

Object input parameters:	PAR_IN
Object output parameters:	PAR_OUT
Input parameters:	PAR_DATA

Description
In this input field a parameter description can be entered. This description is shown with the definition of the parameters of a system object.

Now click the **<OK>** button in order to adopt the settings.

*The function **Parameter Code Generation** should only be used by advanced users.*

A template object does not consist of its data points alone. After a template object has been defined and created in the PET, the corresponding control windows must still be drawn in the Graphical Editor and initialised to the template object. The icon for the template object is linked to the control window and saved as a complete template object.

5.5 Inheritance of template objects

If templates were created according to the philosophy of Visi.Plus (meaning that they include graphical user interfaces), they can be used and inherited in the current project as needed.

5.5.1 What does inheritance mean?

If, for example, a square is drawn, its "colour" attribute can normally be defined. However, if the square should change from red to green or the opposite, depending on a binary signal (e.g. flag, input or output), the "colour" attribute must be initialised with a DMS element (e.g. Flag5).

If several graphical and text elements must be combined into a group (e.g. for a motor object), it would be logical to save this group under a separate name in order to possibly use it again for the current project or a later one. Such a group is called a **template object** or **VLO** in Visi.Plus.

If such a VLO is now used frequently in a process diagram, each attribute with which a PCD element is initialised must now be painstakingly reassigned. Why? Because otherwise, for example, flag 500 would also simultaneously change colours for each motor that is represented.

In order to avoid all this work, Visi.Plus makes it possible to reinitialise such VLOs with a few mouse clicks. The following sections explain how this is done.

The idea of the inheritance can be applied to all graphical elements. Special register contents can be used as value displays, for text selection or position coordinates.

5.5.2 Procedure for reinitialisation

Start the Graphical Editor (GE) of Visi.Plus.

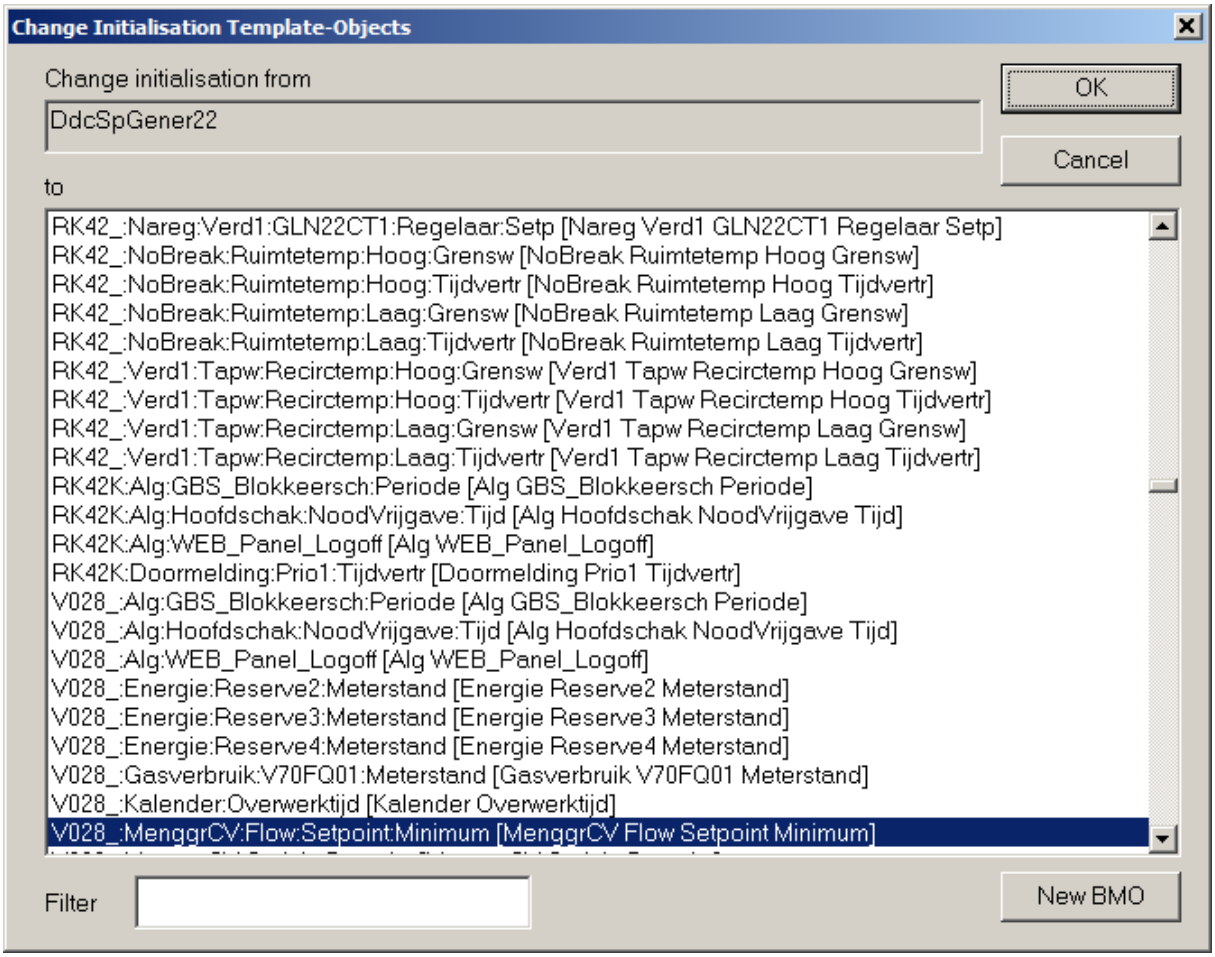
Click on the **Open Template Object** icon or choose the command **Load VLO Objects** in the **File** menu.

*In order to copy template objects from an existing project to a new project, the respective bitmaps must also be copied. These are normally located in the directory **C:\Visi.Plus\proj\(ProjectName)\bmp***
*The data structures of the template objects are saved in the directory **...\mac**.*

User-created template objects can also be copied to the template save location and are then available for all projects (see chapter 3.5 Installing template objects).

The selected template object is integrated into the process diagram at the top left corner of the screen. The template object can now be placed anywhere in the process diagram by dragging (with the mouse).

In order to inherit the template object (reinitialise), select the template object, press the right mouse button and choose the **Reinitialise** command or press the **<Spacebar>**. The **Reinitialise** popup window appears.



Here you can select the DMS name to which the template object should be reinitialised and then click **<OK>**.

5.6 Process diagrams

5.6.1 What is a process diagram?

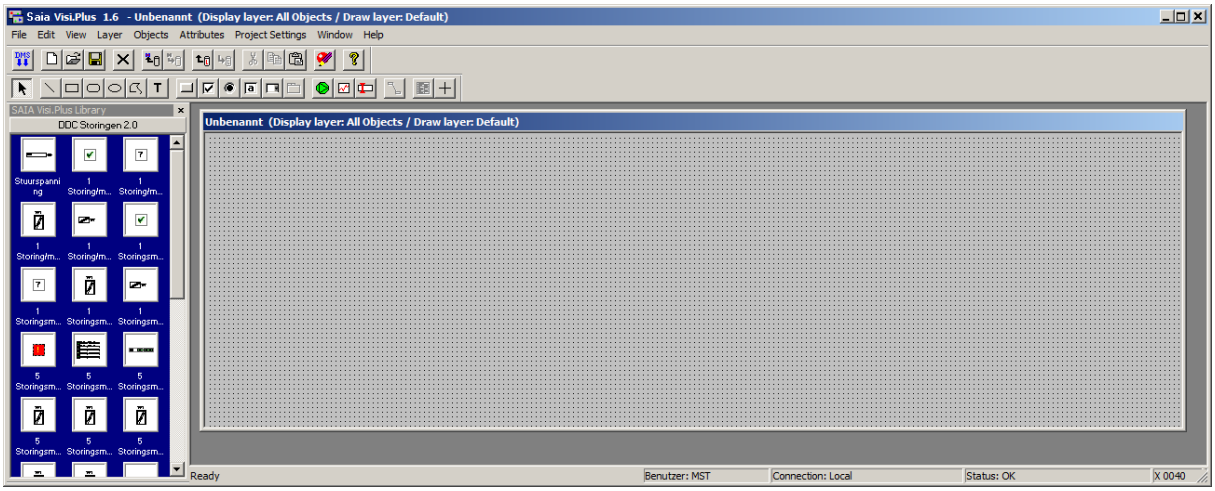
A process diagram is a graphical control screen that comprises a schematic representation of the system or of its components. Such a process diagram can be used to monitor and control the system.

5.6.2 Creating process diagrams

All process diagrams in a Visi.Plus project are created by the Graphical Editor GE. The GE module can be started directly from the Windows Start menu of from the default installation directory.

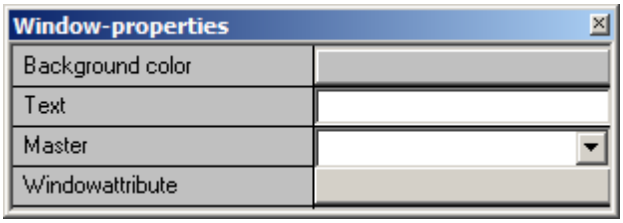
Start button > Programs > **Visi.Plus** > **Graphical Editor**
C:/Visi.Plus/bin/ge.exe
The Graphical Editor opens.

 A new process diagram is created by clicking on this icon button. Alternatively, you can also select the key combination
<CTRL+N> or **File > New** in the menu bar.



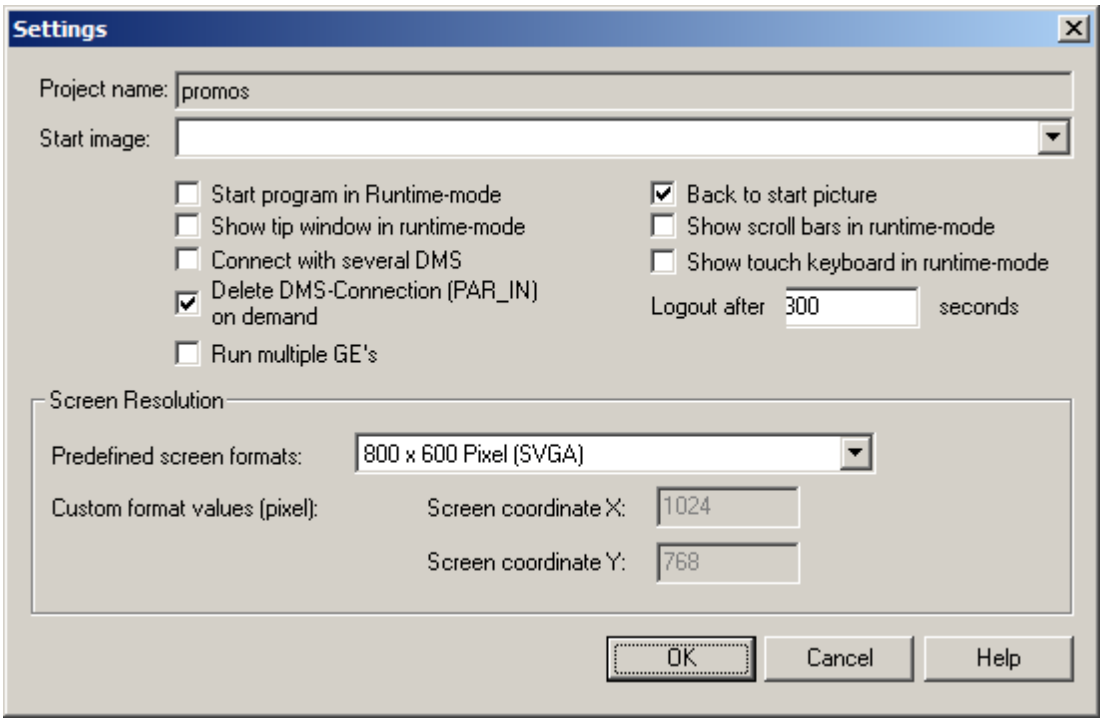
The image attributes of the new process diagram can be adjusted via the object attributes. To do so, click on the blank process diagram (not on the title bar). Then open the Attributes window.

 This icon button opens the **Window attributes** dialog window. Alternatively, you can also click on the menu item **Edit > Attributes**, the keys **<ALT+ENTER>**, or right click on the dialog.



 *Before commencing with drawing, you **must** first adjust the process diagrams to the same resolution as the destination computer.*
It is not really possible to change the resolution afterwards.
If the wrong resolution is selected, the process diagrams will either not be completely shown or the screen will not be completely filled.
To adjust the resolution of the process diagrams, use the respective command in the menu

Graphics Settings.



5.7 Interaction with PG5

The Engineering Tool and the PLC Driver directly access functions of the Saia PG5® development environment. In doing so, communications channels between the computer and PCD of both tools can also be used simultaneously. Therefore, the PCD can be simultaneously reprogrammed or changed with the PG5 debugger variables over the same physical connection during data logging and visualisation.
PG5 projects can also be created directly in the PET. Variables from PG5 can be adopted into the PET.

5.7.1 Generate PLC Code (general function)

By working in an object-oriented manner (each object –e.g. pump, fan, etc. – of the same type looks the same), the PLC-AWL code or Fupla boxes can also be organised on an object-oriented fashion. If many pumps are used in a project, each pump should be processed identically on the PLC and displayed on screen in the same way. Differences in the behaviour of pumps are defined once only via parameters. The goal is to only have to configure each pump (no longer having to program and draw).

If all pumps can be processed by means of the same PLC code, the PLC code should also be generated automatically. A function block call can be generated automatically for each object, using the PET code generator. The corresponding function blocks (written by the user or from a library) are saved on the data carrier and are then integrated by the PG5 compiler. In doing so, between 50 and 100% of the PLC code is automatically generated, depending on the system type.

The automatically generated code is written in AWL and fully documented. The user

does not have to worry about the AWL code, because it only has to be linked to the project. The generated code should not be changed by the user either, because the code will be overwritten the next time code is generated. The Fupla function library is also based on AWL code. The user need not worry about the AWL code generated with the code generation of Visi.Plus either.

5.7.2 Linking objects (generally held possibility)

Visi.Plus offers the possibility of directly linking objects. A compressor object can be directly linked to a temperature measurement object in the Graphical Editor. The compressor is then controlled dependent on the temperature measurement. It is no longer necessary to create PLC programs for simple links.

Object input parameters (PAR_IN)
Up to six attributes can be defined as input parameters for each object. Input parameters include temperature, pressure and other statuses.

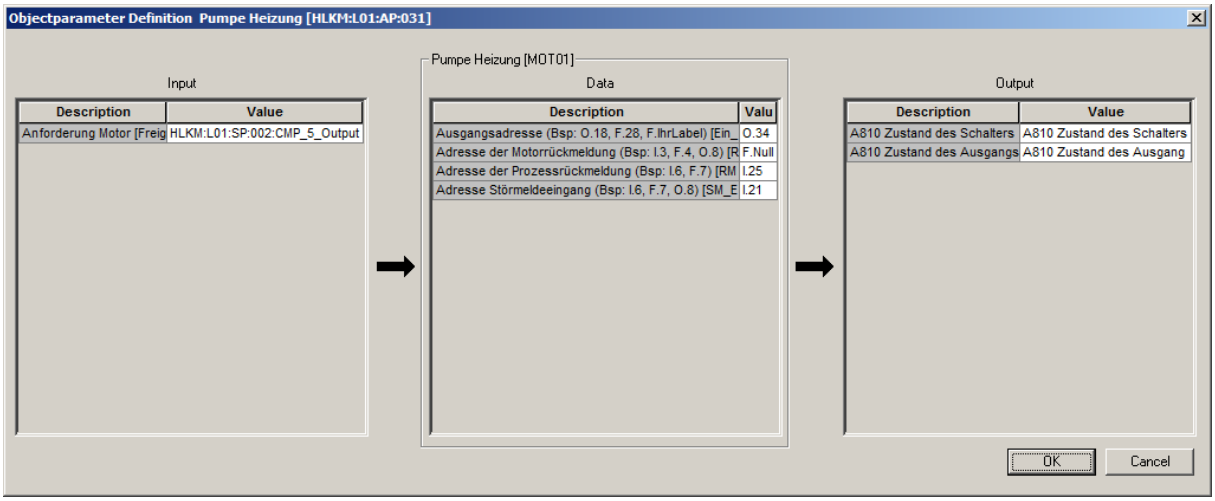
Object output parameters (PAR_OUT)
Up to six attributes can be defined as output parameters for each object. An output parameter can, for example, control a pump or a valve.

Input parameters (data parameters) (PAR_DATA)
Up to six attributes can be defined. These parameters are internal in the object and are not externally influenced. Examples are: addresses of PLC inputs or outputs, card types (analogue card), factors, etc.

Example parameter function
A control circuit object (PID02) has the following parameters:

Input parameters	- Outside temperature
	- Supply temperature
Output parameters	- Heating group pump switch command
	- Heat valve actuating variable
Data parameters	- Controller trigger (10 or 12 bit)
	- Cycle time
	- PID factors

All parameters can now be entered while creating the object in the Graphical Editor. The Code Generator then creates all necessary commands so that the control circuit maintains the correct temperatures and the valve is automatically controlled. PLC programs no longer have to be generated for simple links.



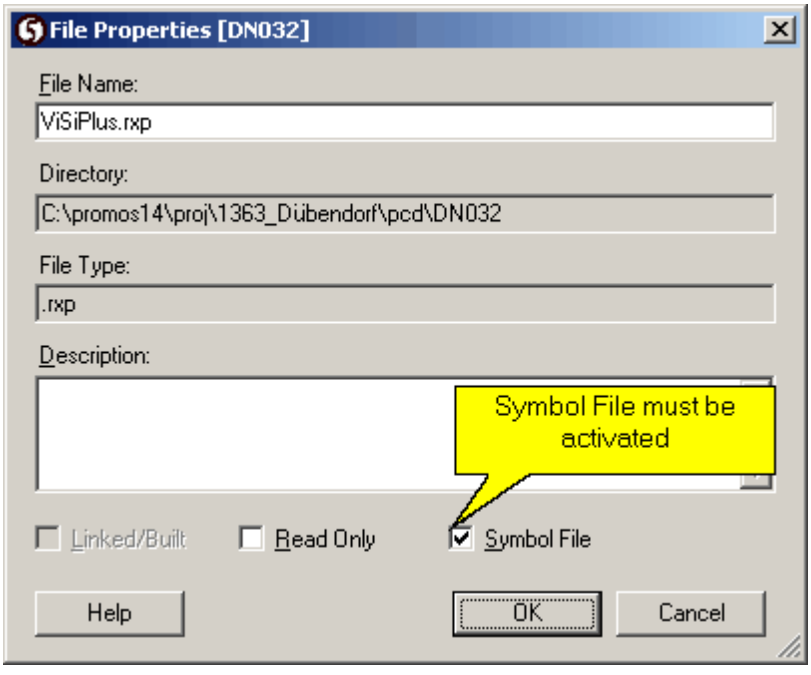
5.7.3 FUPLA integration

The majority of programmers work with Fupla. The method of operation is fundamentally different, because Fupla does not work directly in an object-oriented manner (apart from the actual Fupla boxes).
In Fupla, all variables that should be applied to Visi.Plus must be provided with absolute addresses.
Disadvantage of this process: all signals must be individually visualised (time-consuming) and are not easily inheritable.

The resources from Visi.Plus can be applied directly in PG5. For this purpose, a resource list should be generated from the PET (RXP file).

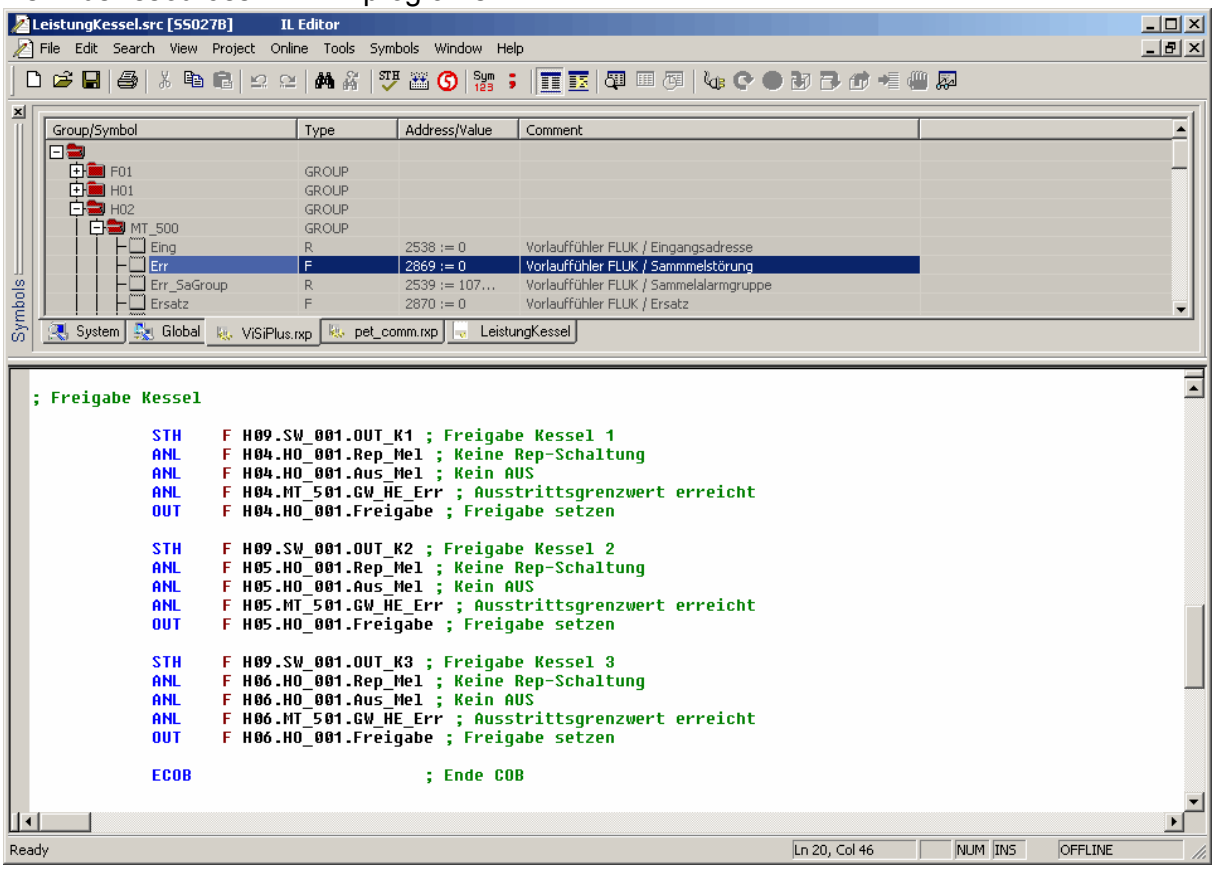
Integration into PG5 takes place via "Program Files" (right mouse button) - "Add Files".

The file must be declared as a 'Symbol file'.

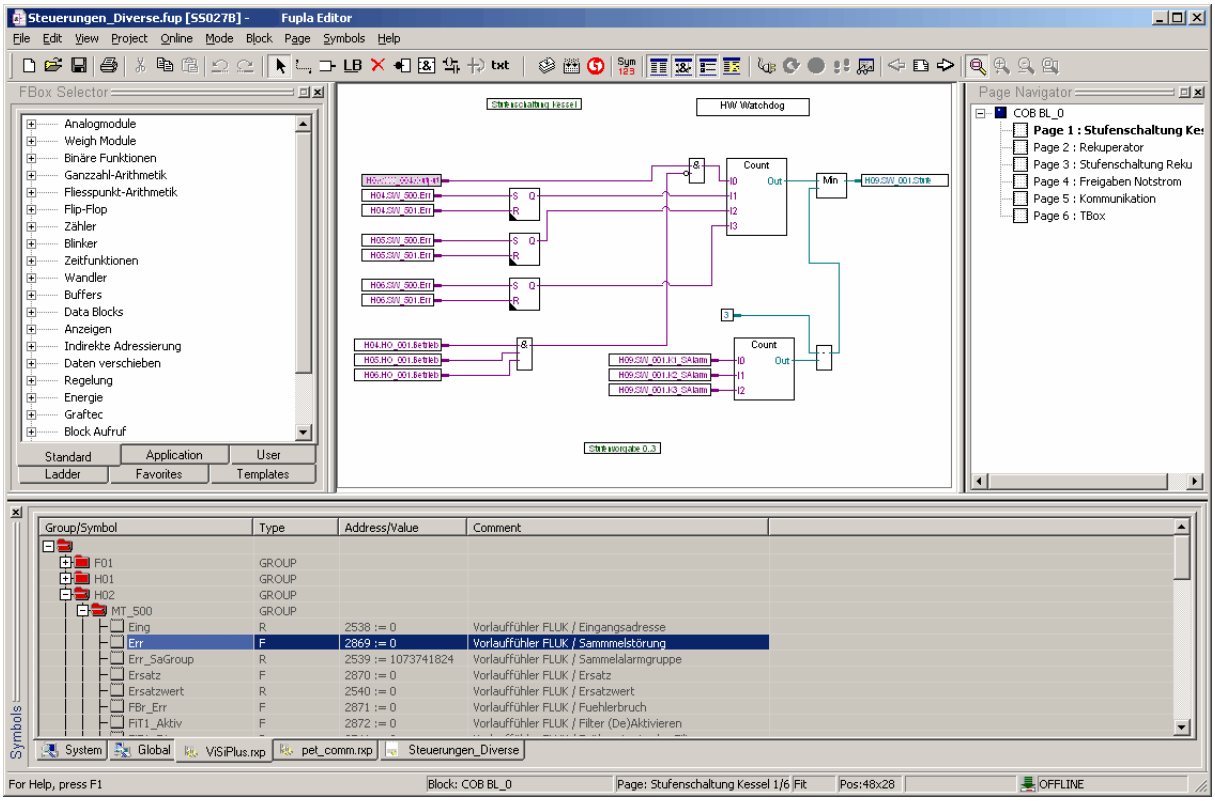


The resources are then freely available in AWL and in Fupla.

Visi.Plus resources in AWL programs:



Visi.Plus resources in Fupla:



5.8 WEB Interface

All images that are created with the Graphical Editor (GE) are automatically WEB capable. The Visi.Plus WebServer (pWEB.exe) modifies the graphical image information for the WEB applets.

The WEB Interface consists of a WebServer and Java applets, which the WebServer transmits to the browser on startup. The process diagrams can be displayed in Microsoft Internet Explorer (with Sun or Microsoft Java VM).

For more detailed information, please refer to the chapter Utility programs, pWeb.exe.

The Microsoft VM (java virtual machine) is no longer supported (by Microsoft).

5.9 Backing up project data

Project data is backed up with the program pBackup or with the program ProjectCfg.

5.9.1 Storage space for historical data

For each logged data point, 1 KB of hard disk space is required. An additional 12 bytes of data is saved for each measurement.

Example of the calculation:

100 values with 1000 daily measurements require:

(100 values * 1024 bytes) + (100 * 1000 * 12 bytes)
results in a daily memory requirement of approx. 1.3 MB.

Experience has shown, for example, that in the case of an extensive HVAC system (heating/ ventilation/air conditioning) with 1000 logged data points (analogue and digital), between 40 MB (summer) and 100 MB (winter) must be allowed for. Such a system should have around 1 GB of hard disk space available per year. With current hard disk sizes, 20 to 50 years can be saved to each hard disk without a problem.

5.10 Data backup

All historical data as well as all logs (incl. alarms) can be automatically swapped out on a daily basis. In the process, older data (normally older than 31 days) in the current data base is deleted. Access to older data is automatically carried out by the system.

Whether and where the data should be saved is dealt with in the PDBS menu **Settings > Data storage**.

Local hard disks or network drives can be used for the data swapping. The use of two hard disks in the same PC is also recommended. In doing so, the data can be swapped out on the standard drive and saved to the second disk. If a hard disk crashes, all of the data always remains available (at least since the last daily swap). The use of redundant hard disks (e.g. RAID 1 or RAID 5) is also possible (and recommended with failsafe systems).
The storage of data (backup) may take place on hard disks, CDs (burner), ZIP disks or a tape drive.

 The setting for the lifetime of the historical data is 31 days by default. Do not change this value. Data that is older than 31 days is managed in the backup and will remain available.

Saia Visi.Plus

Project Engineering Tool (PET)

Chapter 6

6 Project Engineering Tool (PET)

The PET program displays the DMS data in tabular form. This enables a system to be more easily configured. No data is saved in the PET. Saving merely sends an instruction to the DMS to save the current databases.

➔ The PET is the front-end program of the DMS (Data Management System).

6.1 Summary

All system data belonging to a Visi.Plus project is managed by the **PET (Project Engineering Tool)**. All existing physical data points (sensors/actuators, etc.) are logged with the PET and the structure of the DMS is thereby determined.

In addition to the data necessary for control and visualisation, the manufacturer, order number, electrical diagram number, service technician telephone number and much more can be logged if desired. In other words, this includes all data that may be necessary to access for the system at some point in time.

Data points should not only display their values on the screen with the help of the GE, but should also issue alarm messages, supply measurement values for trend graphs, logging and much more. All these actions are defined under the respective columns in the **PET**.

The **PET** supports the possibility of combining multiple signals/data points into groups or so-called template objects (VLOs) (see the chapter Template Objects (VLOs)).

The **PET** imports and exports data to other applications, such as PG5, the development environment of Saia PCD®.

The **PET** generates data required for the Code Generator.

*Important: The **PET** reads and saves all data to the Visi.Plus database, DMS. Therefore: before the **PET** can be used or started up, the DMS must be started (through the **Start Options** in the module **"Project Settings"** this can be automated).*

For additional information on this topic, see also **the chapters "DMS" and "PDBS"**.

6.1.1 Restrictions

The following names may not be used for the definition of object names, attributes, etc. These are reserved Visi.Plus system names!

OBJECT	**PLC**	**ALM**
PAG	**PRT**	**CLK**
PAR_IN	**Comment**	**TRD**
PAR_OUT	**PAR_DATA**	

Moreover, names used by SBC may not be used either (e.g. STH, ANL, IST, etc.)

6.1.2 What do the Visi.Plus data points contain?

In order to explain this properly, we need to think about what a data point might look like:

Designation	Type	Address	Comment
Measurement1	R	1000	Outside wall, north side, ground floor, entrance
Pump1	F	50	Pump controller

The following question occurs:
What happens if the sensor line of the outside temperature sensor "Measurement 1" is broken? The measurement assumes the maximum value (e.g. 150□). This inevitably leads to the heating being switched off.
A remedy is achieved by a new data point that has a suitable fixed default value with which the heating can work in case of a sensor break.

For a sensor this could then appear as follows:

Designation	Type	Address	Comment
Measurement1	R	1000	Outside wall, north side, ground floor, entrance
DefaultValue1	R	1001	Sensor break outside wall, north side, ground floor, entrance

How does the list appear with multiple sensors (including default value)?
Example:

Designation	Type	Address	Comment
Measurement1	R	1000	Outside wall, north side, ground floor, entrance
DefaultValue1	R	1001	Sensor break outside wall, north side, ground floor,
Measurement2	R	1002	entrance
DefaultValue2	R	1003	Return flow
Measurement3	R	1004	Sensor break return flow
DefaultValue3	R	1005	Supply
			Sensor break supply

This is somewhat confusing!

An additional classification would be better:

Group	Attribute	Type	Address	Comment
Sensor1	Measureme	R	1000	Outside wall, ...
	nt	R	1001	Sensor break outside wall ...
	Default value			
Sensor2		R	1002	Return flow
	Measureme	R	1003	Sensor break return flow
	nt			
	Default value			

Therefore groups have been created. Each group, which is called an object, can be expanded as desired with so-called attributes.

Attributes describe the properties of an object (e.g. motor, sensor, valve, etc.). Such details as manufacturer name, order number, date of commissioning, hours of operation, star-delta starting time, electrical diagram number, etc. can also be added to the object. This would not only provide more information for troubleshooting and replacement parts procurement, but also more flexibility.

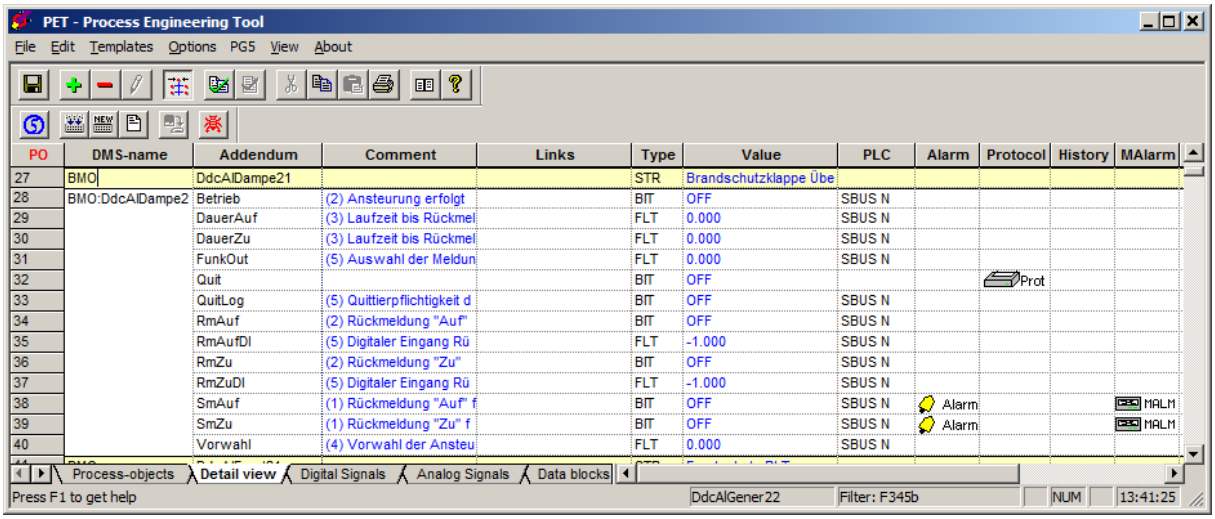
What if there were 20 sensors which had to be created in the PET with even more attributes?

A great deal of typing!

Why not give an object and its attributes a name and then use it as a template for the rest? A name such as "**MES01**", for instance.

In Visi.Plus such an object is called a **BMO (Betriebsmittelobjekt = operating resource object)** or **VLO (template object)**. Suitable graphical elements are available to the VLOs in the GE.

A **BMO** or **VLO** in Visi.Plus appears approximately as follows:



Now let us proceed with our example:

Two sensors should be created in the PET so that the DMS can enter them as data points.

When entering them in the PET under "**System Object**" we are first asked for the VLO name, in this case "MES01", so that the computer can provide the attributes as lines to fill in. However, it also wants a DMS name for each sensor!

A DMS name is nothing but an identifier for each system object (motor, sensor, etc.) in Visi.Plus. Since a DMS name may only occur once, it prevents mistakes and duplication. The DMS name not only helps the programmer, it also facilitates work. If the DMS name for each object is attached to each object in the system, entered in the electrical diagram and in a layout plan, the reference and documentation is already completed.

All modules (programs) in Visi.Plus are oriented to the DMS name.

The following are two examples of a DMS name for our Sensor1 and Sensor2:

Sensor1= S01:MT:500
Sensor2= S01:MT:501

S01 stands for Controller1
M indicates a measurement
T indicates a temperature

500 simple numbering

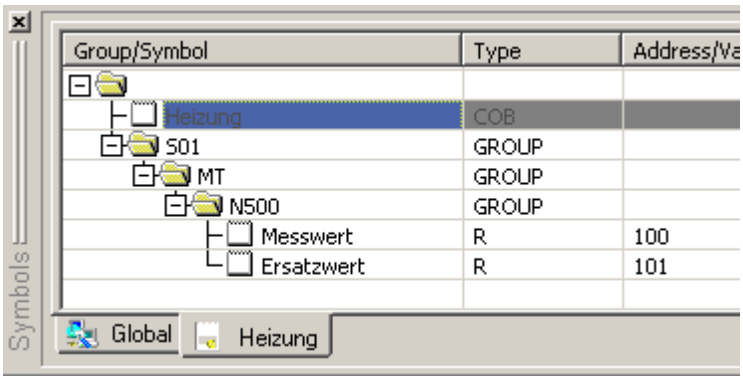
The colons are extremely important (for details, see AKS system).

***ATTENTION**: If data points are exported from Visi.Plus and then imported into Saia PG5®, there **MUST** be a letter before numbers in the DMS name after each colon!*

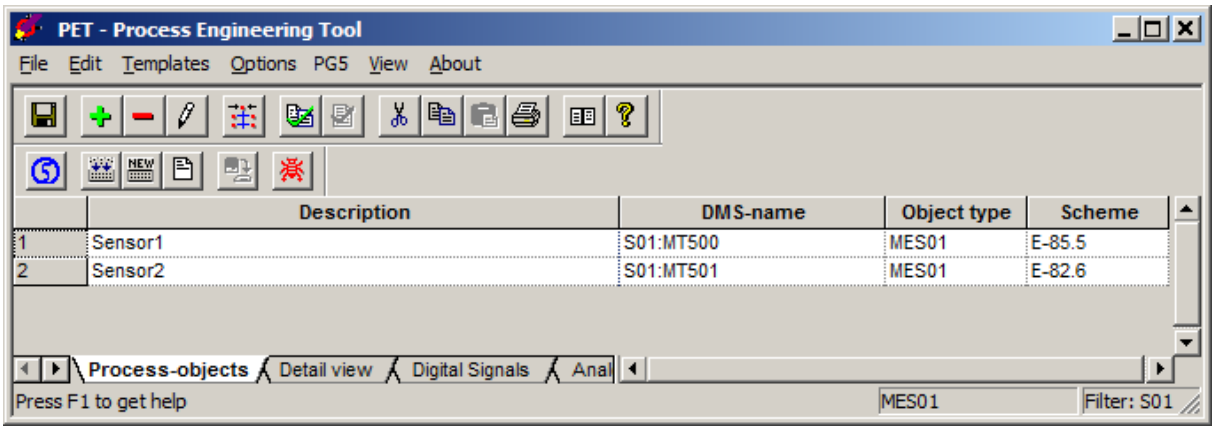
Incorrect** S01:MT:500* ***Correct: *"S01:MT:N500"*
:

During export as an RXP file, an underscore is automatically inserted before numbers: S01:MT_500

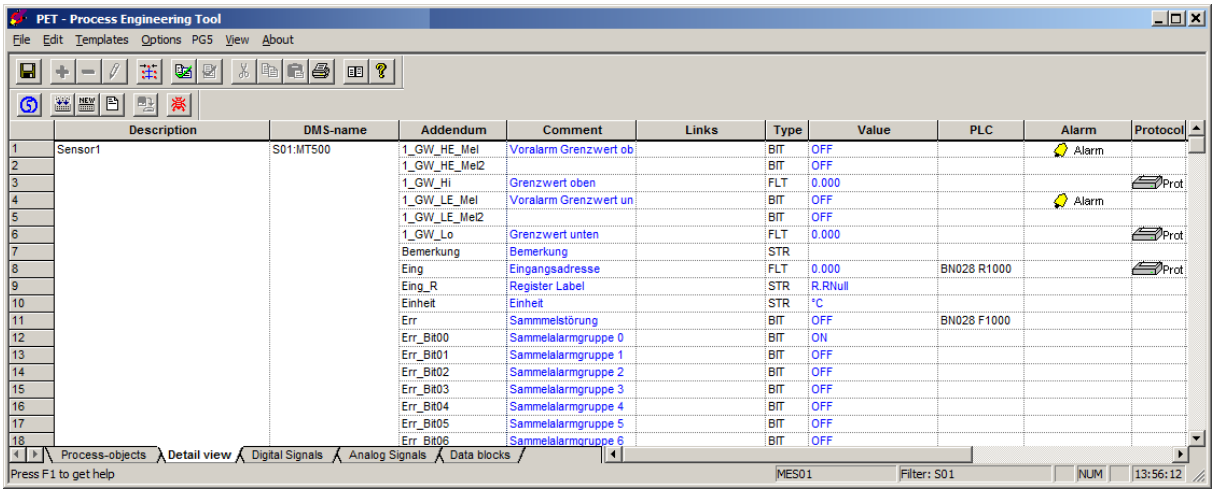
When importing Saia PG5® creates a sub-category for each colon in the DMS name. PG5 symbol table view:



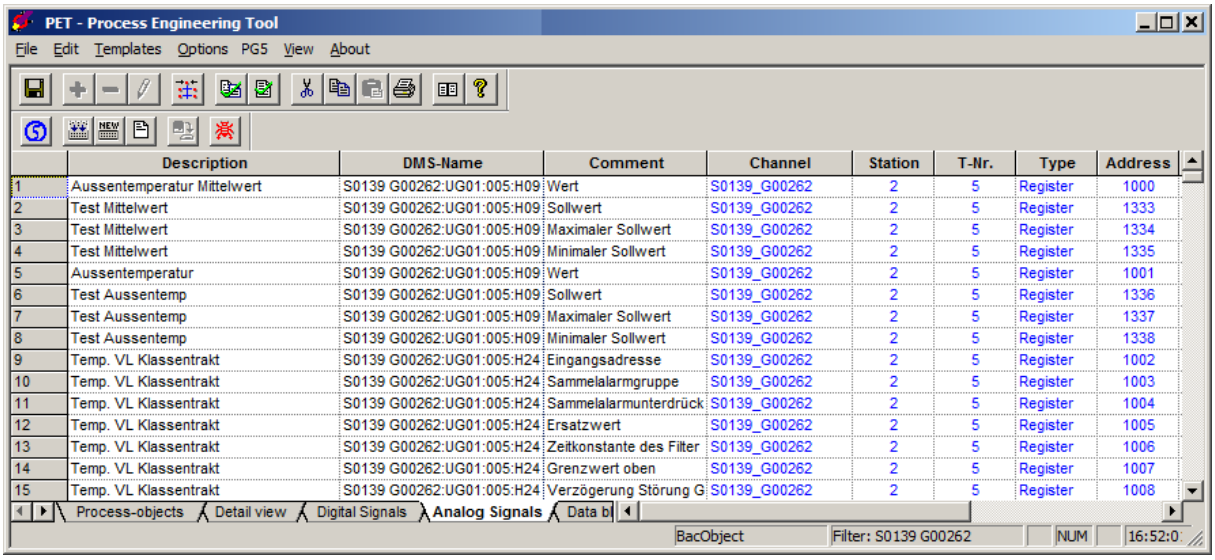
In the "**System object**" view in the Visi.Plus PET the Sensors1 and 2 of the system would then appear as follows:



In the "**Detail View**" ...



In the "**Analogue Signals**" view ...



Back to the exercise
The above illustrations showing how, where and what must be entered are described in the following chapters as well as in Chapter 4.

6.2 Starting the PET

Startup condition
The DMS **(see the chapter DMS)** must already be running on the computer and/or started (visible by on the taskbar). If Visi.Plus was started up with a project by default, this is already the case.

Normally, Visi.Plus is started with a project for which the GE is also started.

As a reminder: Visi.Plus can be informed through the module/program "Project Manger" as to which modules/programs should also be started up (see the chapter Starting programs).

Startup from the Graphical Editor

In edit mode in the GE (which can be switched to with the **<e>** key, if need be), the button  is found in the toolbar. By clicking on it, the PET opens.

Alternative startup

In order to be able to access the data points with the help of the PET (even without the GE) as quickly and easily as possible, and without having to start the whole of Visi.Plus, you can proceed as follows:

As soon as the DMS is loaded (!), the PET can be started via **Start > Programs >Saia Visi.Plus 1.7 > Engineering Tool**

or

by double-clicking on its filename, "PET.EXE", in the default installation directory "C:\Visi.Plus \bin\..." using a **file management program** (e.g. Windows Explorer), like any other program.

The process goes faster and more smoothly if the computer user has made shortcuts for the most important modules of Visi.Plus on the desktop.

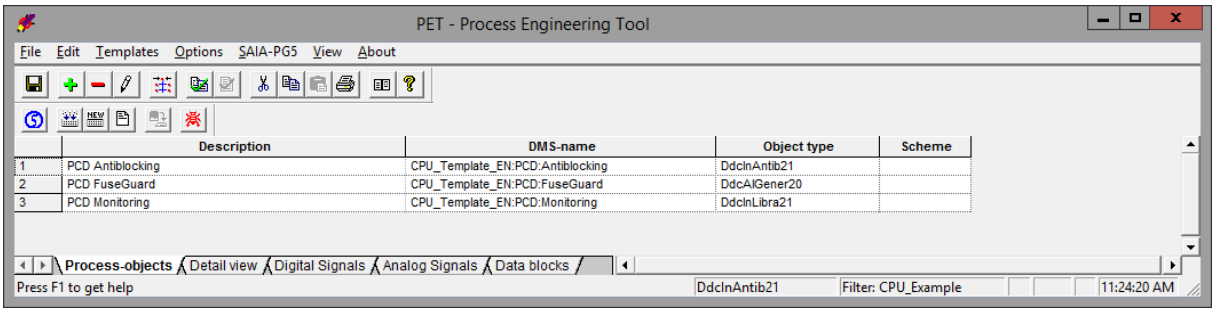
If no valid licences were purchased, the following message appears on startup of the PET:

The functions of the PET can also be utilised without a valid licence.

*The following functions are available, **however they are limited** without a licence:*

- *Number of data points that can be entered*
- *Remote alarms*
- *Code generators*

6.3 The PET user interface



Title bar
Contains the name of the current module.

Menu bar
The menus are arranged according to basic functions and are explained in the following.

Toolbar
Displays icon buttons for some important commands and functions available in the **PET**.

Column labelling
Changes according to the **thumb index**.

Thumb index
The individual views are selected by clicking on them (details available under Thumb index).

Status bar
Shows the meaning of the buttons on the toolbar, the selected object type under **Object Type** and any filters set with **Filter** ("*" means show all).

6.4 Template objects (VLOs)

Template objects combine multiple data points into groups. VLOs allow system objects to be registered both quickly and efficiently.

How to create user-defined template objects is described in this manual in the chapter "Creating new template objects".

Two template objects, **ANA01** and **DIG01**, are included in the scope of delivery in the basic package of Visi.Plus. You can start working with these template objects immediately.

A complete template object library can be purchased from the developer of the Visi.Plus system.

***MST Systemtechnik AG (in Belp near Bern, Switzerland)** has developed and used these template objects successfully in numerous systems. Experience has shown that users do not have to create their own template objects when using a template object library. For more of the latest information about template object libraries, visit the Visi.Plus website.*

6.4.1 An example template object

Motor template object
A user-defined VLO could appear as follows:

- Actual value — Status display: On or Off
- HOp — Registers hours of operation
- Circuits — Registers the number of circuits
- Faults — Faults (coil, LS triggered, overheating, jamming, etc.)
- Manual operation — Display of manual or automatic operation
- Inspection — Motor inspection
- Release — Motor release
- Acknowledgement — Motor fault acknowledgement
- Diagram number — Electrical diagram number
- PLC inputs — Absolute address of PLC inputs (important for code generation)
- PLC outputs — Absolute address of PLC outputs (important for code generation)
- Designations — AKS designation (see the chapter AKS system)
- Comments — Special attributes
- Slow-down time — Input of a slow-down time during which the motor cannot be switched on again

A VLO does not consist of its data points alone. After a VLO has been defined and created in the PET, the associated control screens must still be drawn in the Graphical Editor and initialised to the object. Important information about creating graphical objects is found in the chapter "Graphical Editor (GE)".

6.5 AKS system

Every data point must have a unique name. In Visi.Plus the name is called the **DMS name** and it originates from the **AKS facility designation system (in German: Anlagekennzeichnungssystem).**

Example: **B10:H01:MT:511**
Meaning: Building 10, Heating 1, Measurement Temperature, Sensor 511

Important, please refer!
*The description for the **AKS** or **DMS names** can be found in the chapter AKS system.*

6.6 Working with existing template objects

After the work preparation (creation of the system object list or PG5 symbol list and determination of the AKS designation system) the system objects can be inserted into the PET.

6.6.1 Installing existing template objects

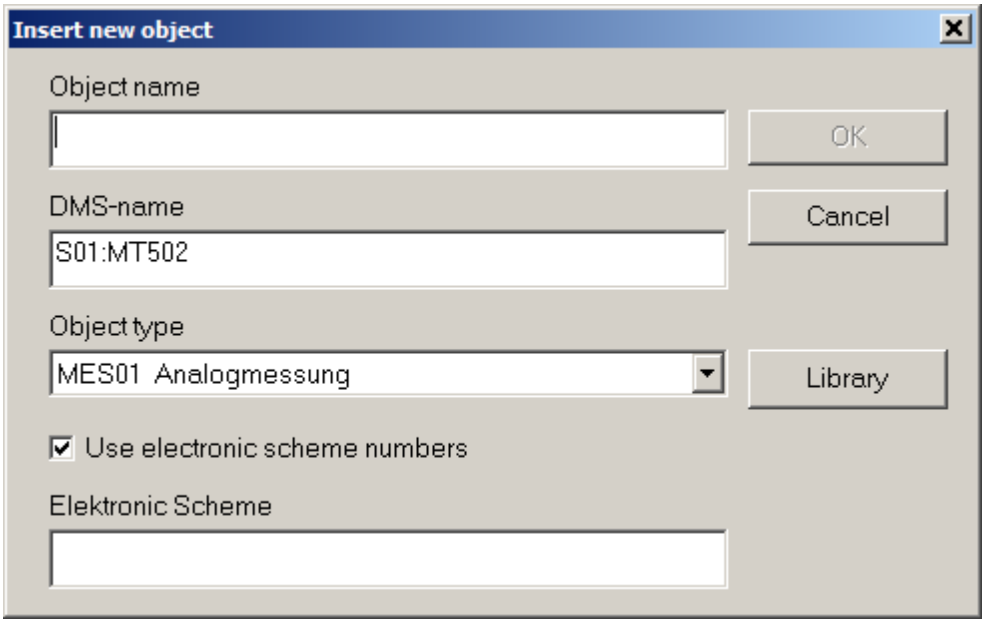
Before the data point is created or imported by PG5, the necessary **VLO**s (template objects) **must** be prepared.

Importing can also be carried out from a ".csv" file created with a spreadsheet. It must be have been created according to specific rules (see chapter Importing data points).

Two methods are available in the **PET**for installing **VLOs**:

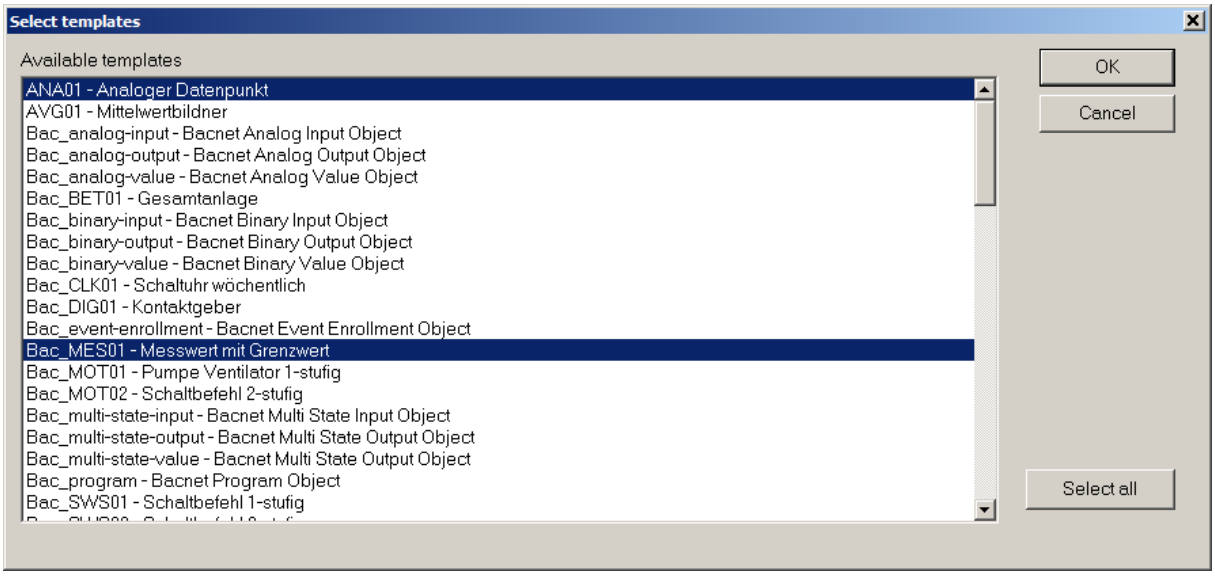
- Either select "**Template Objects > Template from Library**" in the menu
- or press the **Insert Object** **icon** and then press the space bar, **<+>** or **<Insert>.**

The following dialog window opens:



Click the mouse on the **<Library>** button and the dialog window **Select Template Objects** opens.

All available template objects are listed in the window **Select Template Objects**:



As soon as a VLO has been selected with the mouse, additional VLOs can be marked or deselected with the **<CTRL>** key.

***Important: Saia PG5® Fupla** programmers **only require** the VLOs **DIG01** and **ANA01**. These VLOs are responsible for the import of digital (binary) and analogue (integer) data points from the PG5 symbol list.*

After selecting the template objects to be installed and confirming with the **<OK>** button, the selected VLOs will be inserted.

6.6.2 Description in the scope of delivery

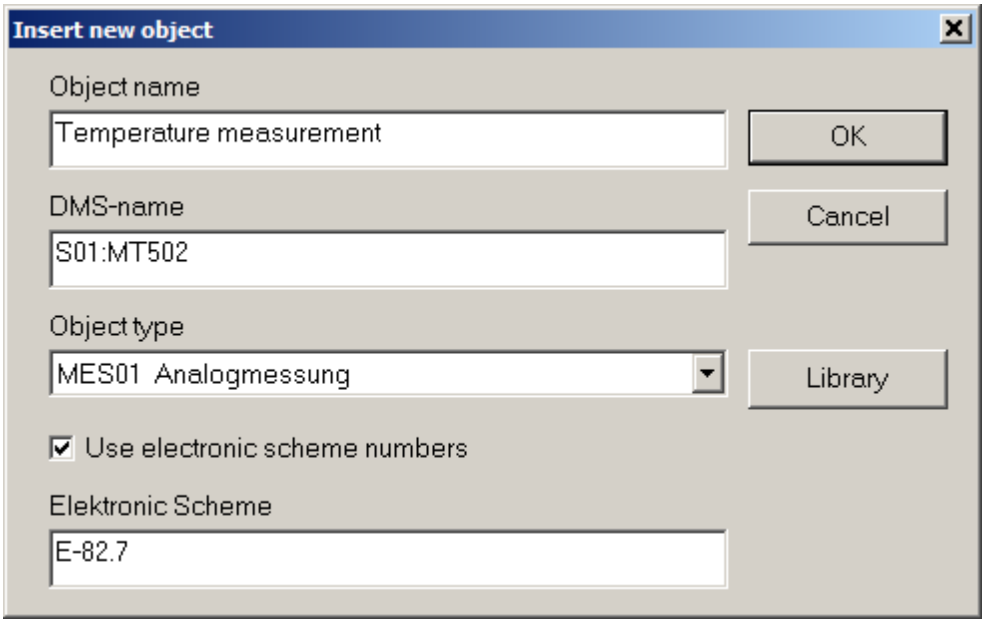
DIG01 is responsible in particular for the import of digital (binary) data points from the PG5 symbol list.

ANA01 is responsible in particular for the import of analogue (integer) data points from the PG5 symbol list.

6.7 Registering system objects

The **System object** thumb index must be activated!

The "Insert New Object**" input window is opened with the symbol**  , **<+>** key or **<Insert>** key.



Object designation
Name of the object.

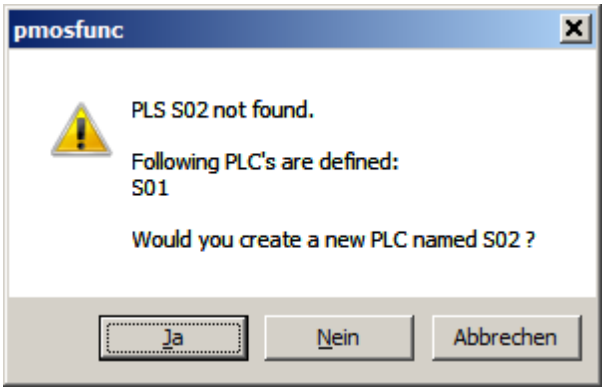
TIP:
*When creating an additional object, the previously entered object name is inserted again by double clicking above the **Object Designation** input field by pressing the **<Cursor Up>** key.*

DMS name (facility identification system)
Name allocated corresponding to the AKS designation. In the example the DMS name S02:MT:500 was assigned on the basis of the AKS list in **Chapter 4.2**.

Object type
Select a template by clicking on the dropdown menu.

Insert Electrical Diagram Name
If the checkbox is activated, the system object is supplemented with this data point. The electrical diagram designation can therefore be entered in the input field.

After confirmation with **<OK>** the following window appears:



If PG5 is installed on your computer, this message appears, as long as there is not a control with this name already. Visi.Plus indicates that a system object is inserted for which no CPU or PLC has been created.

In the preceding image a CPU has been created for the system **S01,** but not for the system **S02**.

The buttons have the following effects:

<Yes> creates a new PLC/CPU in the PG5.

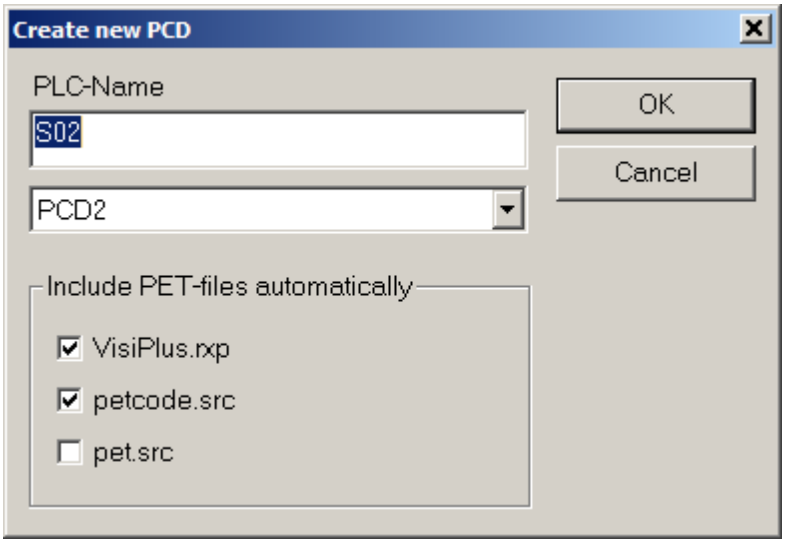
<No> does not create a new PLC. However, the system object is inserted into the project.

<Cancel> Neither a new PLC is created, nor is the system object inserted into the project.

The PLC name must always correspond to the initial identification of the DMS name! The PLC name may not begin with numbers!

If a new PLC should be created, Saia PG5® now automatically starts up and brings itself into the foreground on the screen. In the process, the current display is hidden.

By minimising the PG5 project manager or by switching with the key combination **<ALT +TAB>** (standard operating system shortcut) the following window appears:



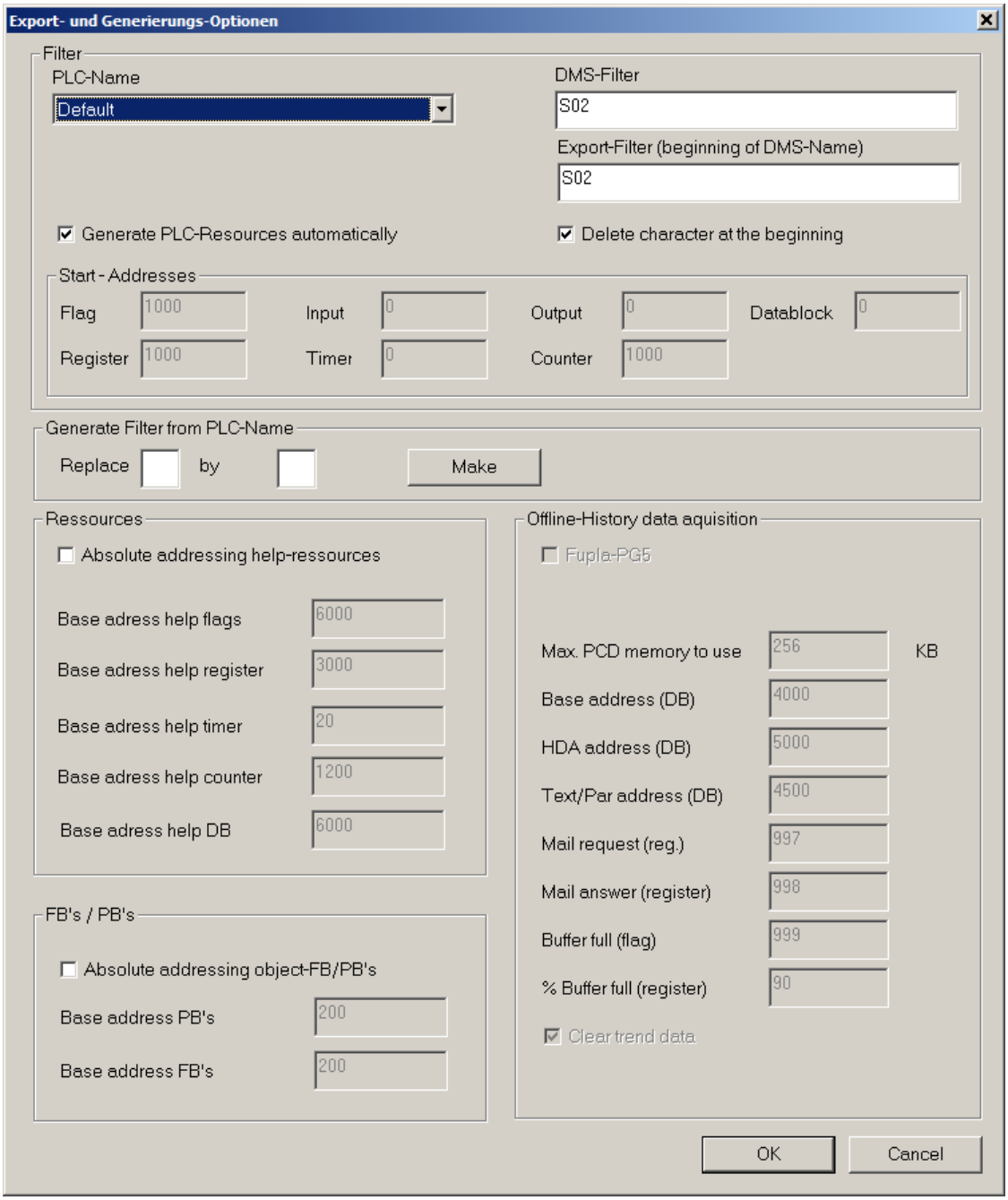
Name of the PLC
The input field carries over the first part of the AKS designation.

Dropdown menu
Enables the selection of the desired PLC type. In the example PCD2.M170.

Automatically Insert PET Files
When checked, it creates the PLC code files **VisiPlus.rxp, petcode.src** and **pet.src** in the new PLC.

<OK>
Creates the PLC in the PG5 "**Project Manager**"

When inserting system objects, it is possible to automatically generate the PLC resources (channel, station, addresses, telegram no.) with the PET (see chapter Export options). If the option **PLC Address Automatically Assigned** is activated in the **Export and Generation Options** dialog box, the following input dialog is subsequently opened:



DMS filter / Export filter
A corresponding DMS filter can be entered here for the PLC (first characters of the DMS name, normally PLC name).

PLC name
The desired channel is specified via the dropdown menu. In the example BN028.

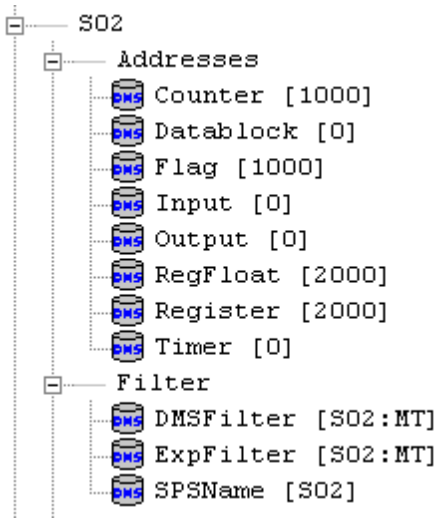
Automatically Assign PLC Resources
Normally the PLC resources (flag and register addresses, etc.) are automatically assigned by the system. With existing systems, all basic addresses of the resources can be specified (incl. function blocks and program blocks).

Start Addresses
In this section the respective start addresses of the PLC resources are defined. These

addresses must not conflict with the automatically assigned resources in PG5. The addresses are based on the assignment of resources by the system.

Offline Historical Data Acquisition
These parameters are based on offline acquisition (HDA offline). See corresponding chapter.

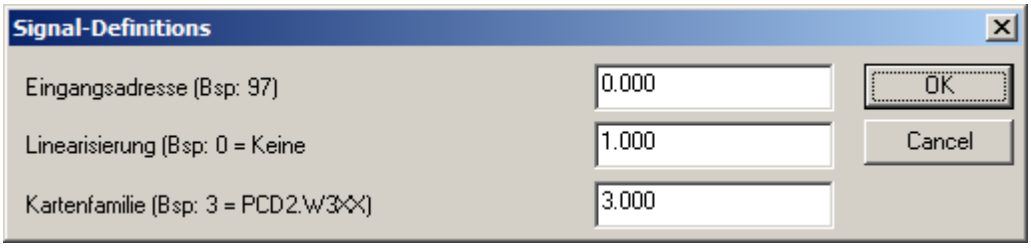
<OK>
Generates the PLC resources in the DMS and saves the settings in the DMS under System:Prg:PCD:<PLC-name>:Addresses or System:Prg:PCD<PLC-name>:Filter.



<Cancel>
The PLC resources are not automatically assigned.

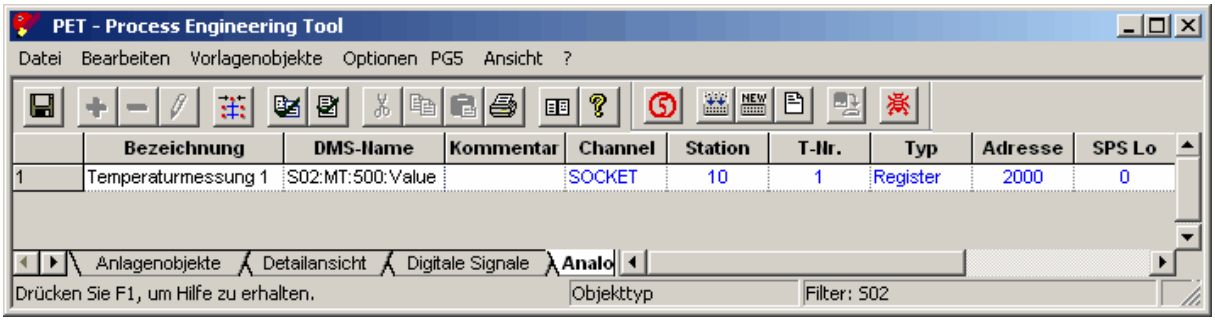
If attributes were entered in the template object with the option "**Input Parameters (parameter query for inheriting)**", finally the **Signal Definition input window appears.**

Image of the MES01 template object:



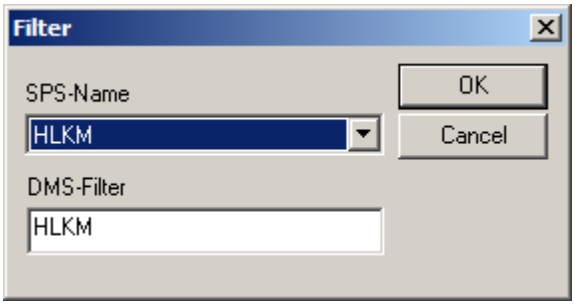
Additional information about the MES01 template object can be obtained from the MES01 template objects documentation (see appendix).

If the inputs in the PLC resources dialog box were confirmed with OK, when the **Analogue Signals Register** is called up, the screen shows that the inputs made in the PLC resources dialog box were applied:

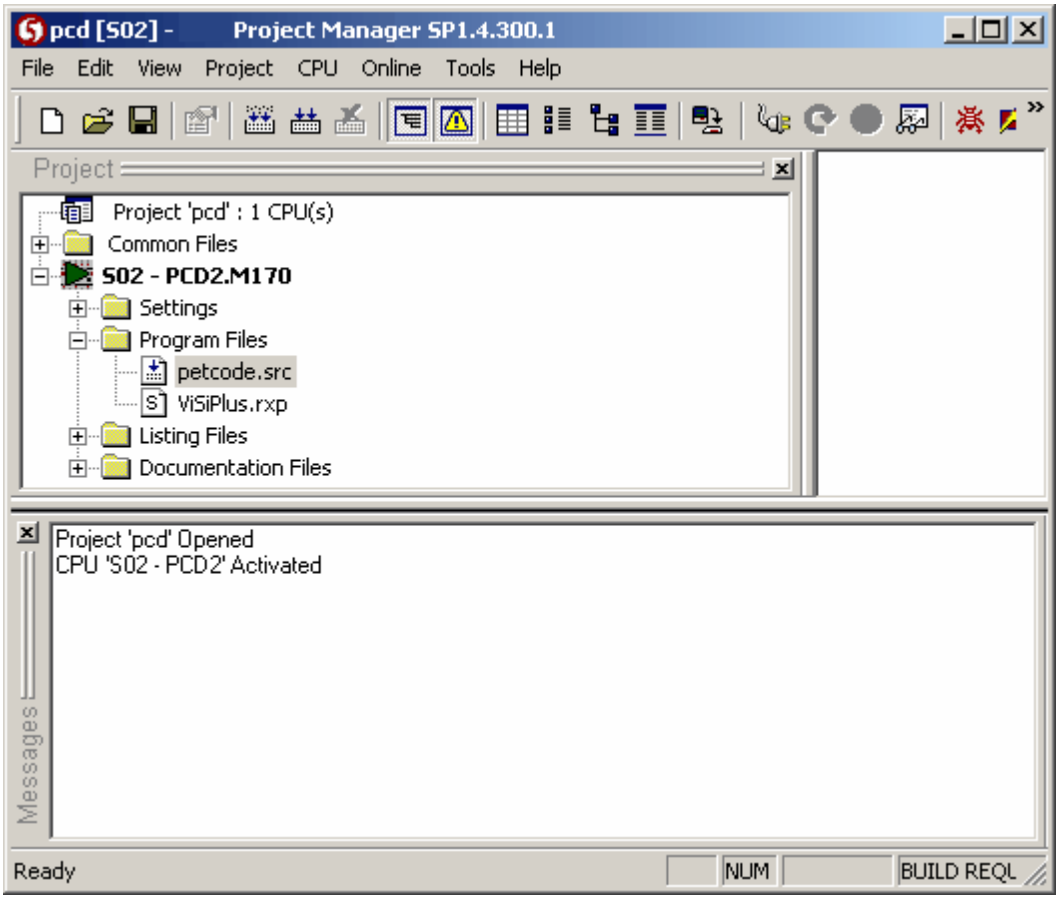


If an additional system object is inserted for the same PLC (S02), the address is increased by one accordingly (address of the new object = 2001).

After the generation of a new PLC, the filter is automatically adjusted in the PET and the views are updated accordingly. The filter can be called up with **<Alt+F>**. In the preceding example, the PLC name is set to S02 and the DMS filter is set to S02:MT (see input in the PLC resources dialog box).



The results in the PG5 appear as follows:

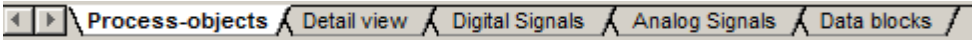


As can be seen above, a new project was created and the CPU **S02** was inserted in the **PG5 Project Manager**.
The files **petcode.src** and **VisiPlus.rxp** were automatically added to the **Program Files**.

If no PLC should be generated during the insertion of a system object, this must be indicated in the file <proj>/cfg/promos.cfg. In the process, the entry CreatePLC=0 must be added to the [Settings] section.

6.8 Delete an inserted system object

The **System object** thumb index must be activated!



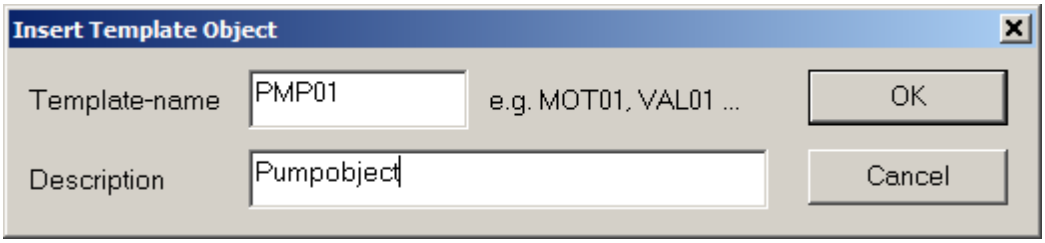
Select the system object and then click  or press the **<DEL>** key for deletion.

6.9 Creating new template objects

In order to create a **New Template Object (VLO)**, the **Template View** must first be called up.

This can be accessed via  or with the **<F5>** key.

New Template Object (VLO)
Access the menu command **"Template Objects >New Template Object"**:



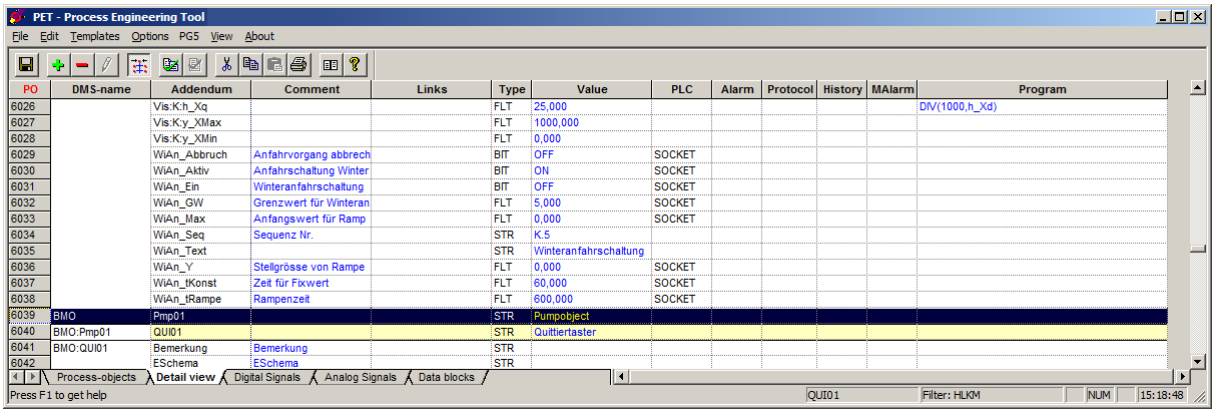
Resource object (BMO) name
Enter the name of the template object (VLO).

In earlier versions of Visi.Plus, VLOs were still called BMOs. For compatibility reasons this name cannot be changed to VLO).

Description
Enter template object description

<OK>
The settings are adopted.

The **template object** is now inserted at the end of the existing list of templates (if any). Simply scroll all the way to the bottom to check.



Only one yellow line with the corresponding data was inserted.

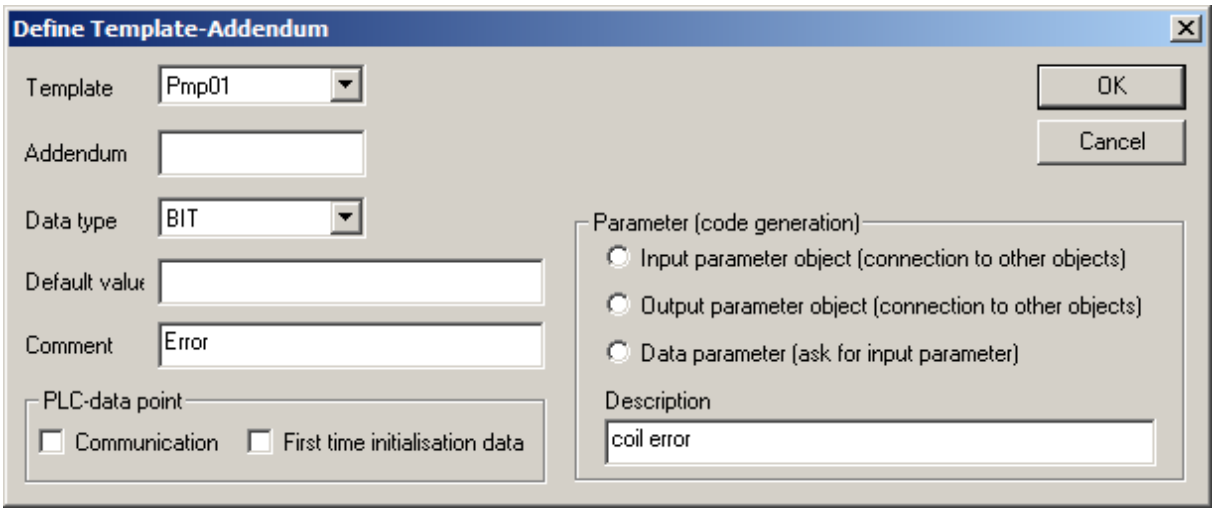
The template object now has a name, but still has no attributes (Actuator/Sensor signals, etc.).

6.9.1 New Template Object Attribute <+>

In order to insert a **new template object attribute**, click on the  or **<+>** button. Alternatively, select the menu command "**TemplateObjects > New Template Object Attribute**".

In order to delete a template object attribute, click the respective line or the  *icon or use the* ***<DEL>*** *key.*

The control screen **Define Template Object Attribute** opens:

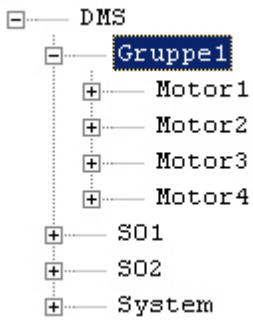


Template Object
Select the VLO to which an attribute should be added.

Attribute
Enter the name. Attention: there are restrictions!

Data Type
NONE
Has **no value**.
It is used to create a superordinate data point for additional subordinate objects.

Group 1
A data point of data type NONE, which has no additional objects (groups), is shown in the following image.



Data Type BIT
Used for digital signals. This data type can only accept the two **statuses logic 0** or **logic 1** .

Data Type FLT
Used for **floating-point numbers**. **Integer values** from the PLC are automatically converted by the Visi.Plus driver **into FLT**.

Data Type STR
A string of max. **80 characters** can be entered.

Data Type DWU (DB)
Used in order to process PCD-specific **data blocks**. **DWU** stands for Double Word Unsigned. Each bit in a data block can be individually addressed with the help of the Visi.Plus control functions (see chapter DMS control functions).

Default Value
The template object attribute assumes this value by default and passes it on.

Comment
Text shown in the **Comment** column in the PET can also be displayed in the GE.

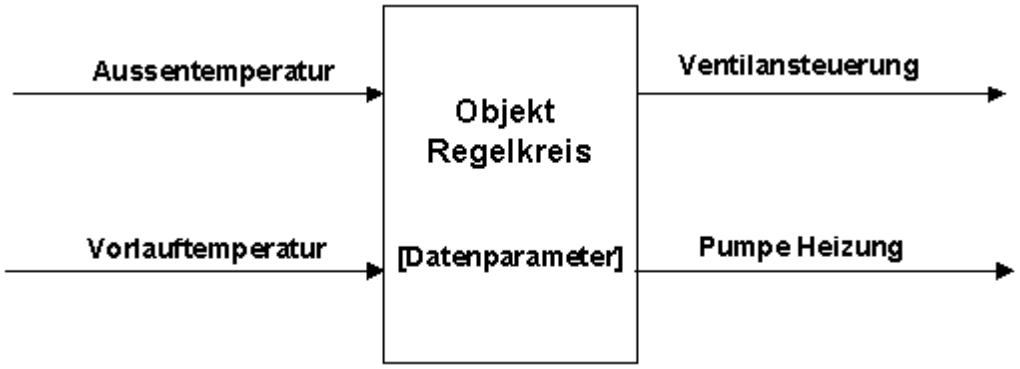
PLC Data Point
Communication with PLC
Only with the checkbox activated in the PET Detail View, **PLC** column, is a communications channel reserved. Therefore the data point is communicated to the PLC according to its definition.

Initialise Value in PLC
With this checkbox activated, the value entered under **Default Value** is assigned to the data point during code generation and is written to the PLC during startup of the Visi.Plus driver - regardless of whether the checkbox **Write DMS Values to PLC on Startup** is activated (XYZ EQU F 123 **:=45**).

*The more data points communicating with the PLC, the slower the communication. In the Template Object View, you can see that the data points **Designation** and **Comment** are not communicated for the object MOT01. The two data points only contain values that are purely informative and need not be communicated. Therefore only data points that are really necessary should be communicated to the PLC.*

Parameters (Code Generation)
Here, by selecting the checkboxes, you can decide whether the input or output parameters should be linked to another object.

These links enable the Code Generator, for example, to automatically assign an outside temperature to the "Control cycle" template object. The output of the control cycle is also connected to the target value of the valve.



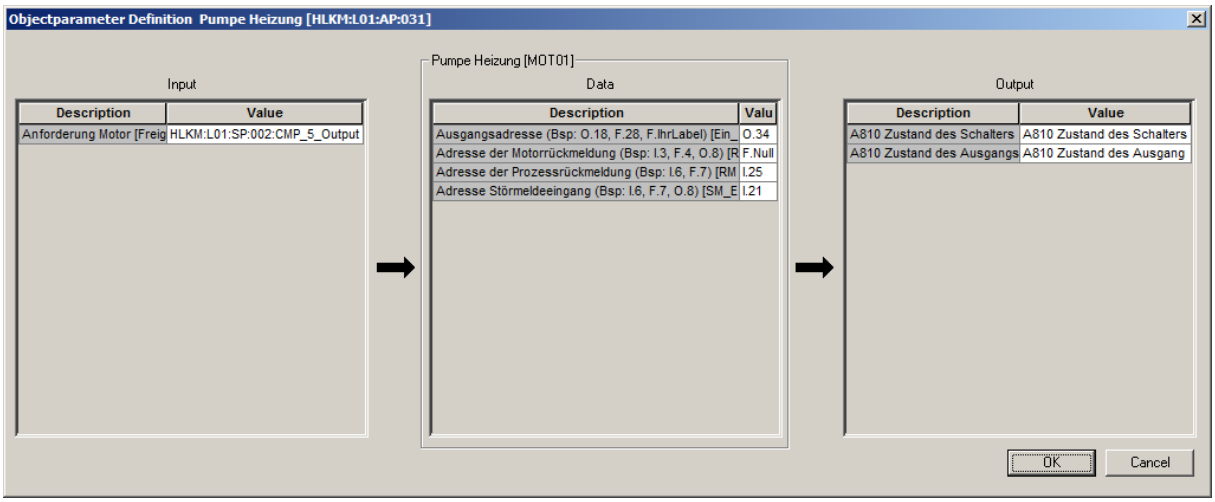
A value that has been defined as an input parameter is automatically copied to the object during code generation (only 1:1 relationships are possible, meaning no links can be made).

A value that has been defined as an output parameter is automatically copied after the object code is called up.

The assignments can be made in the Graphical Editor (right-click with the mouse and select the **Parameters** command).

PAR_IN	Input parameters that are copied to the object
PAR_OUT	Output parameters that are copied from the object
PAR_DATA	Data parameters (configuration data)

The template object (VLO) must be marked in the Graphical Editor by right-clicking with the mouse and selecting the **Parameters** command. Alternately, the list of parameters can be called up by double-clicking (left mouse button). The following entry screen appears:



The number of parameters that can be linked depends on the defined **PAR_IN** (input), **PAR_OUT** (output) and **PAR_DATA parameters**, which were created in the template object.

Input parameters	AKS designation of the system object to be linked
Output parameters	AKS designation of the system object to be linked
Data parameters	A constant is entered by prepending the letter "**K.**"

The text entered, **Parameters (Code Generation)** field in the **Description** input field, is

displayed in the respective input field.

The Code Generator generates the following AWL code for the entry screen:

```
; Control Heating Transmission Line [WI017:H20:RK:001]

COPY   R H09.MT_500.Actual  ; IN-PAR Outside temperature
       R H20.RK_001.AT

COPY   R H20.MT_500.Actual  ; IN-PAR Temperature VL transmission line
       R H20.RK_001.Actual

CFB    PID02                ; PID-Control cycle incl. heating curve
       R 1008               ; [=01] AT1
       R 1011               ; [=02] AT2
        ...                 ; ...
       F 1457               ; [=53] Ctrl
       R 3919               ; [=54] Register

       R H20.RK_001.Res     ; OUT-PAR
COPY   R H20.VS_001.Target  ; Control Valve Heating Transmission Line

       F H20.RK_001.Winter  ; OUT-PAR
STH    F H20.PW_001.Release ; Transmission line pump heating
OUT
```

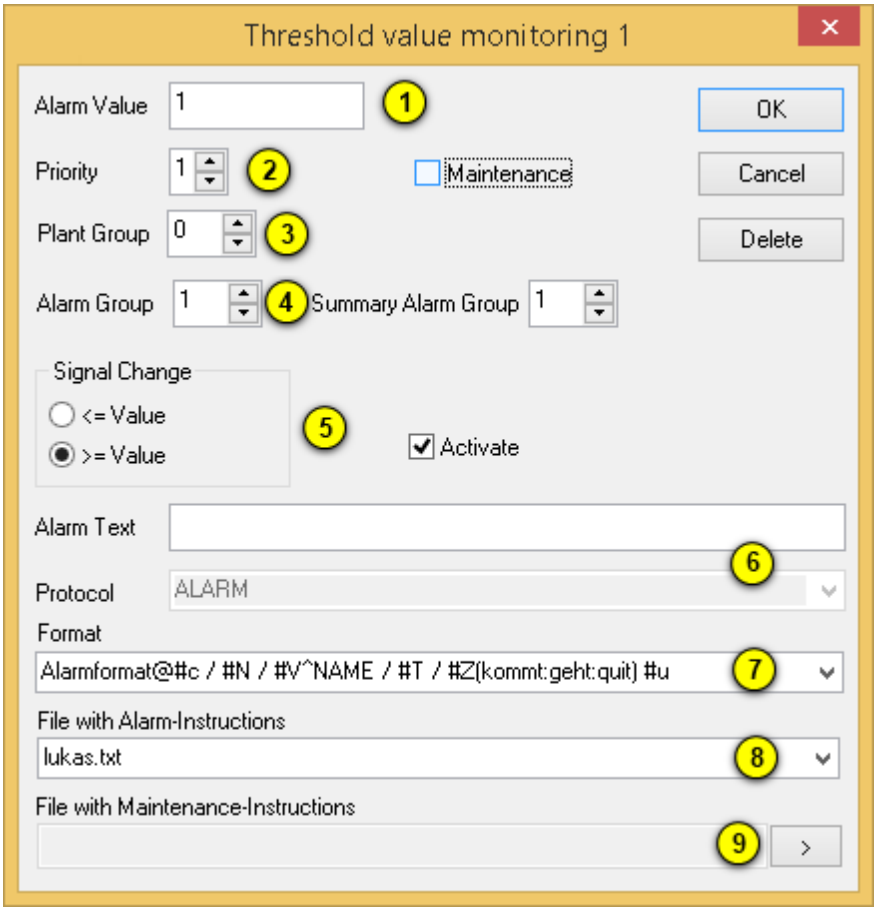
This is how the links between the indicated objects are created.

The data parameters (PAR_DATA) can be entered as function parameters (e.g. constants, etc.).

*Visi.Plus works in an **object-oriented** manner and should thus be based on objects. This allows the system to function in a very productive manner.*

6.9.2 Adding an alarm

If the corresponding attribute is to contain an alarm, an alarm can be added by clicking on the Alarm column in the detail view of the PET. The following window opens:



1): Alarm value
The limit value at which an alarm is triggered must be entered here. Depending on the setting, the signal transition sets the alarm when it either overshoots or undershoots the value.

2): Priority
Alarms can have priorities 1-5. If the Maintenance checkbox is activated, the priority is automatically set to 6.

3): Plant group
The alarm is given the plant group here. This subdivision of the alarm makes it possible for certain users to receive only certain alarms.

4): Alarm Group and Summary Alarm Group
An alarm can assign to a group and summary alarm group (From 0 to 999). Each group and summary group can be deactivated and acknowledged. In this case all alarms belonging to this group or summary group are deactivated or acknowledged.

5): Signal transition
Specify here whether the alarm is issued when the limit value is overshot or undershot. Furthermore, the alarm can be switched on or off using the Activated checkbox.

6): Alarm text
A note or message can be entered in this text field. This text is displayed in the Alarm Viewer and can be visualised in the GE if the corresponding alarm occurs.

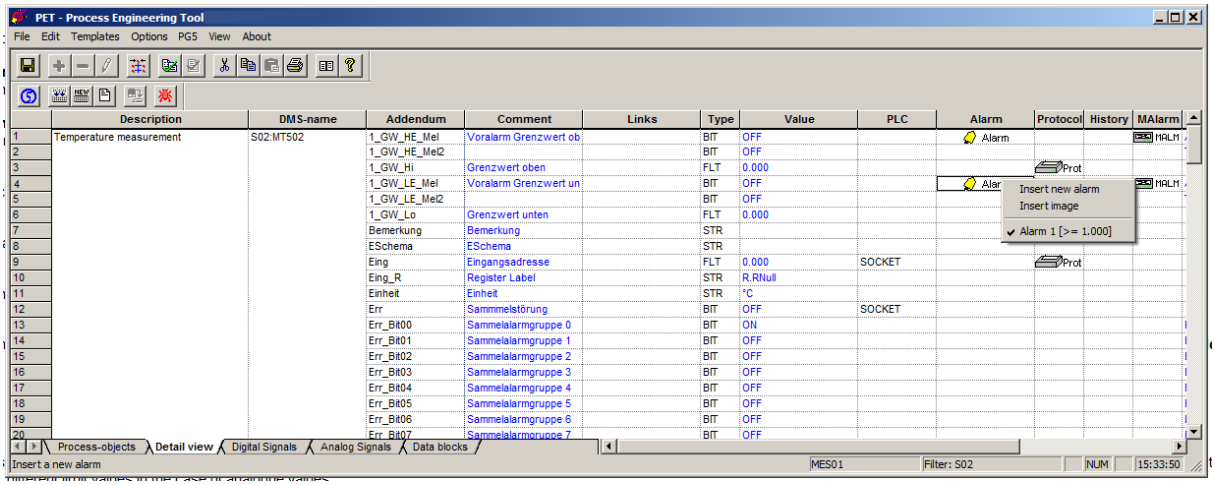
7): Format

The format for the alarm can specified here. A more detailed description can be found in the chapter Format definitions (PRTFormat.exe).

8): Alarm-Instructions
A text-file can be assigned here to the alarm. The text-file must be defined in the rtf-Folder of the project .\<proj>\<projekt>\rft. For more information see chapter alarm instructions from the alarm viewer.

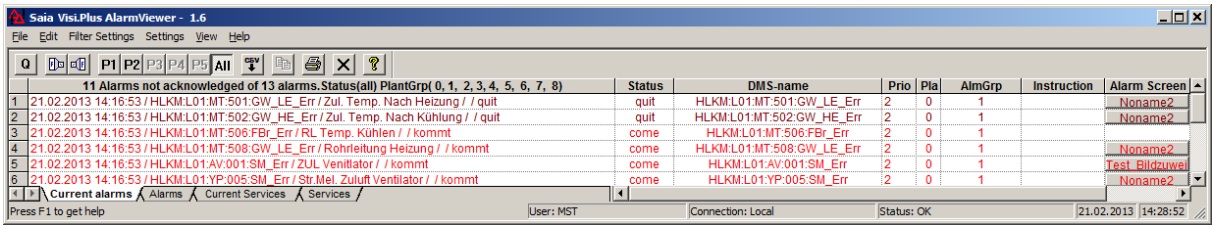
9): File
The path for a pdf-file which can be opened in the Alarm Viewer in the Info column can be specified here. This function is only possible for services.

Once confirmed with the <OK> button, the alarm is then present in the PET and the DMS. With another left click on the alarm, it is subsequently possible to switch between the setting windows **"Add alarm"**, **"Add diagram"** or **"Alarm 1"**.



Add alarm:
When this window is opened, a new alarm configuration screen opens (see the figure above). This makes it possible to set a second, third, etc. alarm for the same data point. For example, this makes it possible to set different priorities for different limit values in the case of analogue values.

Add image:
A GE page can be assigned to the alarm. This activates a button below the Alarm Screen column in the Alarm Viewer. When the button is actuated, the page specified opens in the GE. This way, the object can be called up in the GE. There exist a function in PET to automatically assign the specific GE screen to the Alarm. This is under the menu *option->image assignment of alarms*.



Alarm 1:

The configuration window for alarm 1 can be opened here. Here, changes can be made for alarm 1.

6.9.3 Deleting template object attributes

To delete template object attributes, mark the template object attribute and then click on  or press the **<DEL>** key.

6.10 The thumb index

Brief description of the thumb index:

System objects
Displays all system objects

Detail view
Displas all data points of the system objects.Alarms, trend charts, logs, etc. can be defined here.

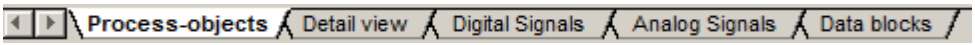
Digital Signals
Displays all digital signals. Telegram numbers, station numbers, flag addresses, etc. can be set here.

Analogue Signals
All analogue signals are displayed.Telegram numbers, station numbers, register addresses, conversion factors, etc. can be set.

Data Blocks
Displays all data blocks used.

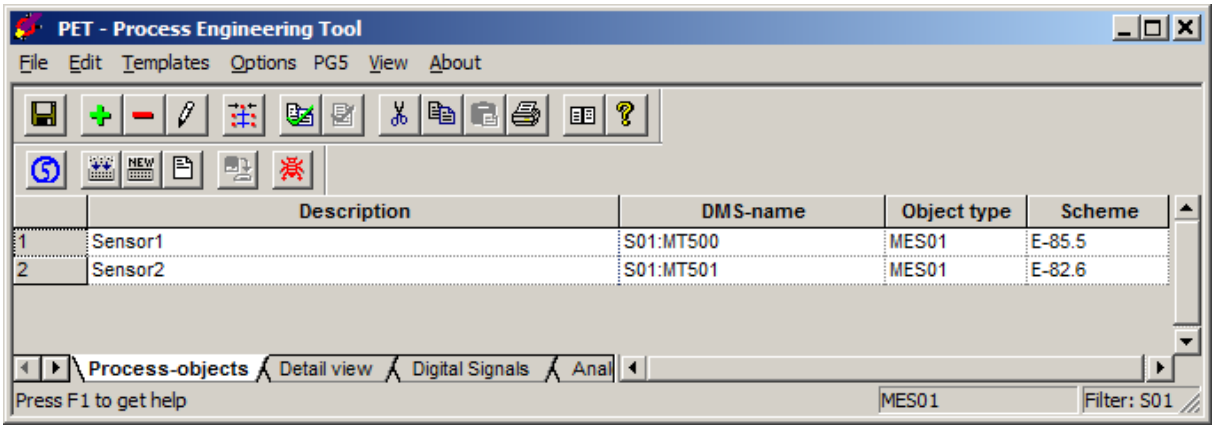
The thumb index is described in detail in the following pages.

6.10.1 View: System objects



The view "**System objects**" shows all previously defined objects of the system.

In the following image a sensor is shown with its data, which is connected to the station "**S02**:".



The columns include the following information:

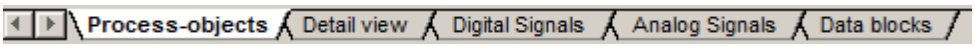
Designation
Name of the system object

DMS name
AKS designation of the system object

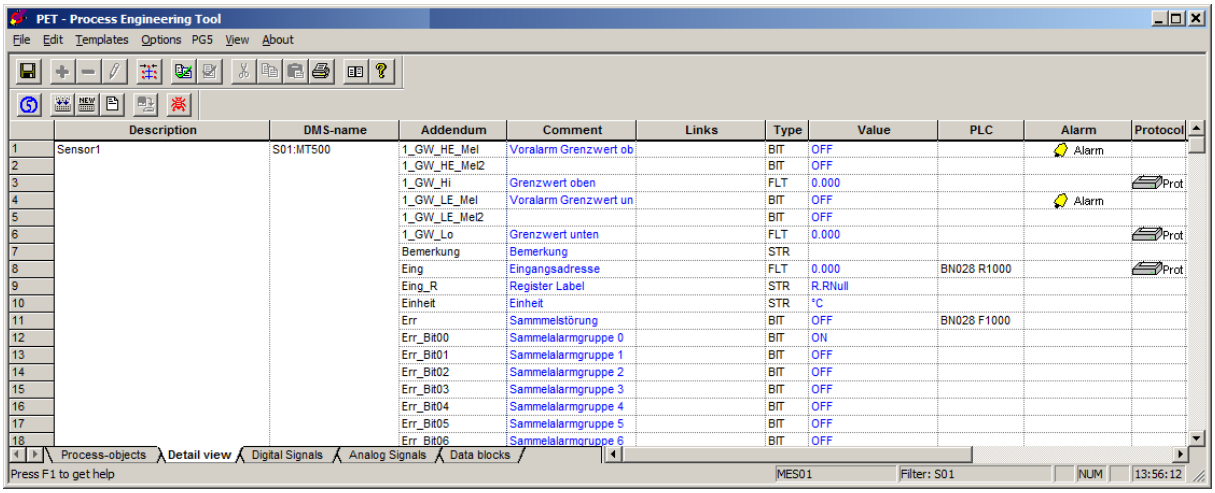
Object type
The sensor **"Temperature measurement 1"** is based, e.g. on the template object "MES01".

EDiagram
Is a useful entry if a sensor has to be found in the electrical diagram.

6.10.2 View: Detail view



All object attributes (data records) are shown in the detail view:



The columns include the following information:

Designation
Name of the system object

DMS name

AKS designation of the system object

Attribute
All template object attributes (data points) of the respective system object are shown.

Comment
Displays the text that was entered in the **Comment** input field when the template object was created.

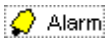
Links
Displays data defined as input and output parameters (PAR_IN, PAR_OUT) (links for code generator).

Type
The data type of the respective template object attribute is displayed.

Actual value
The actual value of the respective template object attribute is displayed and can be edited here.

PLC
Only template object attributes with an entry in this column, e.g. **SBUS R1234**, will be communicated **to the PLC.** Whether a template object attribute should be communicated to the PLC is defined during its creation!

The information displayed under this heading must match the information in the Visi.Plus driver! Otherwise Visi.Plus will never be able to communicate with the PLC!!!

Alarm
All template object attributes with the alarm symbol 

will trigger an alarm in the **ALM** module if their value is changed.

Whether a template object attribute should have an alarm or not has already been defined during creation of the template object!
By inserting the alarm into the template object, the alarm is inherited by all objects with the same name.
If the alarm is not inserted into the template object, but directly in a system object in the detail view, it is only inserted in the respective system object.
There is the risk that the alarm entry may be deleted in the system object if the corresponding template object is modified.

Before an alarm is assigned to the data point, an alarm format must be defined in advance.
*The **ALM** module must already be started in order for the alarm to be registered (this can be automated via **Start Options** in the "**Project Settings**" module).*

Attention:
*The need to define an alarm format and start the **ALMMng** module is often overlooked.*

When these requirements are not fulfilled, Visi.Plus cannot set off an alarm!

Log
All template object attributes with the **symbol** 

are logged with their value changes.

Trend
All template object attributes with the **symbol** 

are registered after each configured trend acquisition.

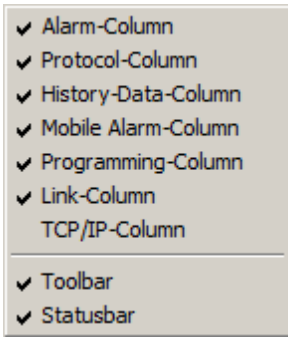
MAlarm
All template object attributes with the **symbol** 

set off a remote alarm with a change in their values on the basis of the data defined in the MALM module.

Control functions
All template object attributes can be linked to control functions. The possible control functions are described in the chapter DMS control functions .

Inserting control functions in the PET and not in the DMS is recommended, since the compilation of the control functions is automated by the PET. In the DMS compilation is carried out by the programmer using the respective button.

*In the detail view in the "**View**" menu columns can be displayed or hidden. All columns that are displayed are marked with a √.*



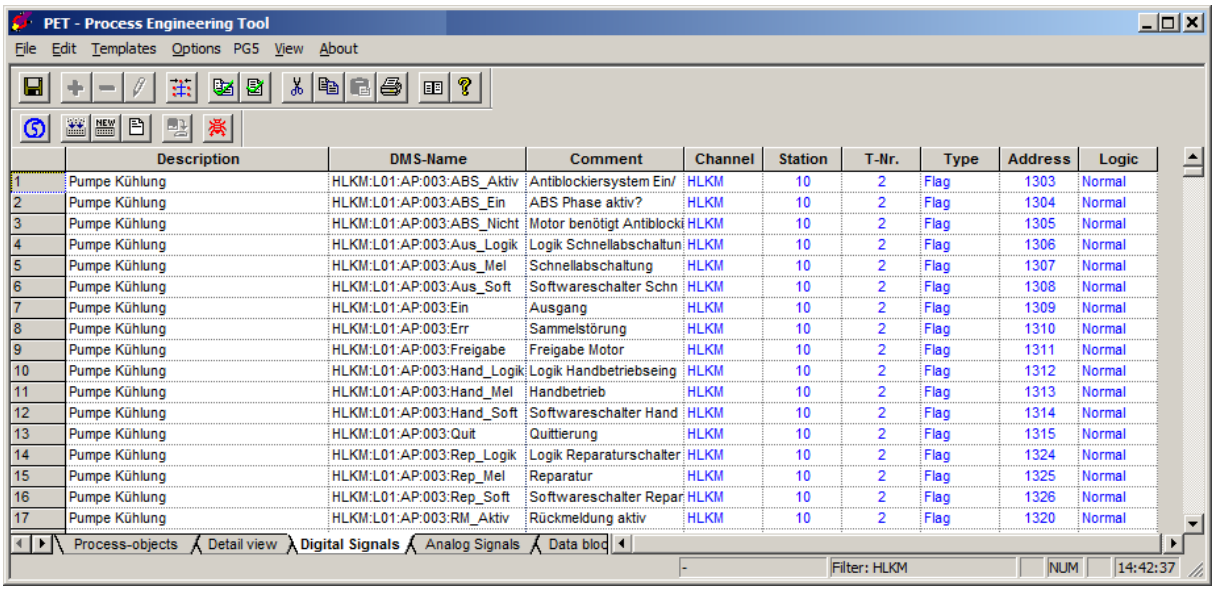
*All entries highlighted in blue in this view can be edited. Finalise any modifications by pressing **<ENTER>**. The value is then written into the DMS and then, if it is a PLC element, sent to the PLC.*

6.10.3 View: Digital signals

In this view, all digital signals of the system objects are visible.

The columns of the list depend on the PLC driver, see menu command "**Options > PLC Driver**". The following descriptions refer to the SDriver (Saia PCD® driver).

By double-clicking on a column name, the content of the list is sorted in ascending order and by double-clicking again the sequence is reversed.
*Depending on the PET filter set, only the signals that correspond to the filter criterion are shown (menu command "**Options > Filter Functions**").*



	Description	DMS-Name	Comment	Channel	Station	T-Nr.	Type	Address	Logic
1	Pumpe Kühlung	HLKM:L01:AP:003:ABS_Aktiv	Antiblockiersystem Ein/	HLKM	10	2	Flag	1303	Normal
2	Pumpe Kühlung	HLKM:L01:AP:003:ABS_Ein	ABS Phase aktiv?	HLKM	10	2	Flag	1304	Normal
3	Pumpe Kühlung	HLKM:L01:AP:003:ABS_Nicht	Motor benötigt Antiblocki	HLKM	10	2	Flag	1305	Normal
4	Pumpe Kühlung	HLKM:L01:AP:003:Aus_Logik	Logik Schnellabschaltun	HLKM	10	2	Flag	1306	Normal
5	Pumpe Kühlung	HLKM:L01:AP:003:Aus_Mel	Schnellabschaltung	HLKM	10	2	Flag	1307	Normal
6	Pumpe Kühlung	HLKM:L01:AP:003:Aus_Soft	Softwareschalter Schn	HLKM	10	2	Flag	1308	Normal
7	Pumpe Kühlung	HLKM:L01:AP:003:Ein	Ausgang	HLKM	10	2	Flag	1309	Normal
8	Pumpe Kühlung	HLKM:L01:AP:003:Err	Sammelstörung	HLKM	10	2	Flag	1310	Normal
9	Pumpe Kühlung	HLKM:L01:AP:003:Freigabe	Freigabe Motor	HLKM	10	2	Flag	1311	Normal
10	Pumpe Kühlung	HLKM:L01:AP:003:Hand_Logik	Logik Handbetriebseing	HLKM	10	2	Flag	1312	Normal
11	Pumpe Kühlung	HLKM:L01:AP:003:Hand_Mel	Handbetrieb	HLKM	10	2	Flag	1313	Normal
12	Pumpe Kühlung	HLKM:L01:AP:003:Hand_Soft	Softwareschalter Hand	HLKM	10	2	Flag	1314	Normal
13	Pumpe Kühlung	HLKM:L01:AP:003:Quit	Quittierung	HLKM	10	2	Flag	1315	Normal
14	Pumpe Kühlung	HLKM:L01:AP:003:Rep_Logik	Logik Reparaturschalter	HLKM	10	2	Flag	1324	Normal
15	Pumpe Kühlung	HLKM:L01:AP:003:Rep_Mel	Reparatur	HLKM	10	2	Flag	1325	Normal
16	Pumpe Kühlung	HLKM:L01:AP:003:Rep_Soft	Softwareschalter Repar	HLKM	10	2	Flag	1326	Normal
17	Pumpe Kühlung	HLKM:L01:AP:003:RM_Aktiv	Rückmeldung aktiv	HLKM	10	2	Flag	1320	Normal

The columns include the following information:

Designation
Name of the system object

DMS name
AKS designation of the system object (see AKS system for the description of AKS).

Comment
Displays the text that was entered in the **Comment** input field when the template object was created.

Before we address the subsequent columns, the following information is relevant for editing columns:

*Column content with **black type** is protected from modifications.*

*Column content with **blue type** can be edited.*
*Visi.Plus "**PET**" offers the possibility of numbering marked lines sequentially or filling them with the same value within an editable column.*

Procedure:
1. *Sort according to an existing column (already contains values/names, etc.) (double-click on column name)*
2. *Enter the first value (the type of value depends on the column) and press <ENTER> or click on another field with the mouse.*
3. *Click and hold the left mouse button on the value that was just entered, pulling down with*

the mouse pointer as far as desired.
4. Release the mouse button
5. Leave the mouse in the marked area (do not click!)
6. Press the right mouse button
7. Finished

Channel
Contains the log names (drivers) for communicating with the PLC. The setup of the PLC driver is described in **Chapter 9.2 Selection of the PLC driver**.

Station
The station number of the PLC where the data point is located is indicated here. If multiple PLCs are connected to Visi.Plus, the controllers **must** have different station numbers. The station numbers are freely definable, within the framework of the manufacturer's specifications.

T no.
This column contains the telegram number under which the data point is communicated. (The drivers from Visi.Plus use the SBUS telegram format.)


*At least one telegram **must** be assigned for each PCD resource type (flag, inputs, outputs, timer/counter, register).*

Type
The **digital signal** data type can be modified here. Just click on the desired line in the **Type** column and in the subsequent dropdown list, select from the data types **Flag, Input** and **Output**.


There is generally no reason to change the type of a data point. If the template objects are created properly from the start, the data type never has to be modified. Therefore it is worth checking the data type for correctness when the template object is created. Inputs and outputs should not be communicated insofar as possible, because the security of the system cannot otherwise be guaranteed.

Recommendation:
Only communicate flags, registers, timers and counters. In the PLC, flags should be assigned to the corresponding PLC inputs and outputs.

Example of the risk from directly controlled outputs:
If a PLC output is controlled directly by Visi.Plus, it cannot be monitored by the PLC (absence of links to other signals, e.g. limit switches, protective motor switch, etc.). This can have serious consequences.

Address
The PLC element addresses of the data types in use are listed here.


Caution: The PET numbers independently of the data type. With differing data types this can cause problems with timer/counter addresses.

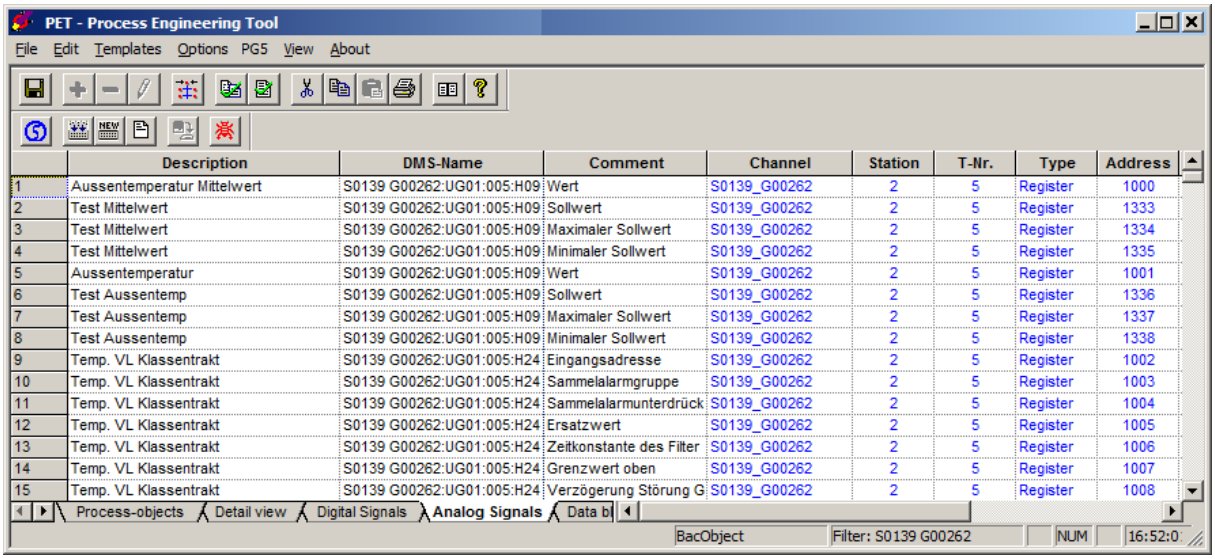
*When programming the **PLC with Fupla**, the addresses do not always have to be assigned by the programmer (except for inputs and outputs). The addresses are indicated during creation of the symbol and determination of the media code.*

Logic
The logic of the signals can be modified here. To do so, click in the column and select between **Normal** and **Inverted** in the dropdown list.

6.10.4 View: Analogue signals

All analogue signals of the system objects are displayed in the **Analogue signals view**.

This chapter describes the differences from the chapter View: Digital signals in order to dispense with reiterating the description of details and operation. Before reading further, we strongly recommend reading the preceding chapter first.



	Description	DMS-Name	Comment	Channel	Station	T-Nr.	Type	Address
1	Aussentemperatur Mittelwert	S0139 G00262:UG01:005:H09	Wert	S0139_G00262	2	5	Register	1000
2	Test Mittelwert	S0139 G00262:UG01:005:H09	Sollwert	S0139_G00262	2	5	Register	1333
3	Test Mittelwert	S0139 G00262:UG01:005:H09	Maximaler Sollwert	S0139_G00262	2	5	Register	1334
4	Test Mittelwert	S0139 G00262:UG01:005:H09	Minimaler Sollwert	S0139_G00262	2	5	Register	1335
5	Aussentemperatur	S0139 G00262:UG01:005:H09	Wert	S0139_G00262	2	5	Register	1001
6	Test Aussentemp	S0139 G00262:UG01:005:H09	Sollwert	S0139_G00262	2	5	Register	1336
7	Test Aussentemp	S0139 G00262:UG01:005:H09	Maximaler Sollwert	S0139_G00262	2	5	Register	1337
8	Test Aussentemp	S0139 G00262:UG01:005:H09	Minimaler Sollwert	S0139_G00262	2	5	Register	1338
9	Temp. VL Klassentrakt	S0139 G00262:UG01:005:H24	Eingangsadresse	S0139_G00262	2	5	Register	1002
10	Temp. VL Klassentrakt	S0139 G00262:UG01:005:H24	Sammelalarmgruppe	S0139_G00262	2	5	Register	1003
11	Temp. VL Klassentrakt	S0139 G00262:UG01:005:H24	Sammelalarmunterdrück	S0139_G00262	2	5	Register	1004
12	Temp. VL Klassentrakt	S0139 G00262:UG01:005:H24	Ersatzwert	S0139_G00262	2	5	Register	1005
13	Temp. VL Klassentrakt	S0139 G00262:UG01:005:H24	Zeitkonstante des Filter	S0139_G00262	2	5	Register	1006
14	Temp. VL Klassentrakt	S0139 G00262:UG01:005:H24	Grenzwert oben	S0139_G00262	2	5	Register	1007
15	Temp. VL Klassentrakt	S0139 G00262:UG01:005:H24	Verzögerung Störung G	S0139_G00262	2	5	Register	1008

Detailed descriptions for the following 7 columns can already be found under View: Digital signals.

Designations
DMS name
Comment
Channel
Station
T no.
Address

Type
The data type of the **analogue signal** can be modified here. Just click on the desired line in the **Type** column and in the subsequent dropdown list select from the data types **Timer, Counter, Register** and **RegFloat**.

PLC Lo, PLC Hi, Unit Lo and Unit Hi
These four columns are used for scaling. The best way to explain their use is with two examples:

Example 1:
The measurement of a 12-bit analogue card is recorded by the PLC in a register. The register content therefore corresponds to a range of values: 0 - 4095. The measured value is read by Visi.Plus from the PLC and converted, e.g. to a voltage range of 0 - 10VDC. Visi.Plus uses this converted value for further processing.

As a reminder, the most commonly used resolutions for Saia PCD® analogue cards are:

Measurement range	**8-bit resolution**	**10-bit resolution**	**12-bit resolution**
0-10V	0..255	0..1023	0..4095
0-20mA	0..255	0..1023	0..4095
4-20mA	51..255	205..1023	819..4095

According to this example, the columns will be defined for the corresponding data point as follows:

PLC Lo
The lower limiting value of the resolution of your analogue card is indicated here. This value depends on the resolution of the analogue card in use. According to Example 1: **0**

PLC Hi
The upper limiting value is indicated here, corresponding to the analogue card resolution. According to Example 1: **4095**

Unit Lo
In the Unit Lo column, enter the lower value to which the analogue signal should be converted. According to Example 1: **0**

Unit Hi
In this column enter the upper value to which the analogue signal should be converted. Example:

PLC Lo	PLC Hi	Unit Lo	Unit Hi
819	4095	0	50

A 4-20mA signal in 12bit resolution is imported and shown in Visi.Plus in the range of values from, e.g. 0 to 50. According to Example 1: **10**

Example 2:
The value of an analogue card PCD2.W210 (0-20mA (sensor 4-20mA), 10-bit resolution) is imported into Visi.Plus via a register and converted in the range of values from 0 to 100 (corresponding to 0 - 100%).

Columns in the data point line therefore appear as:

PLC Lo	PLC Hi	Unit Lo	Unit Hi
205*	1023	0	100

*corresponds to 4mA

Analogue signals can also be converted directly in the PLC. In doing so, it is no longer necessary to indicate conversion factors in the conversion columns (only 0:1 -> 0:1).

If the template object library of "MST Systemtechnik AG, Belp, CH" is used, the conversions take place directly in the object. Only the analogue card type must be specified.

Diff
Values in this column indicate the amount by which the signal must change before Visi.Plus updates the value.
Do not select a value that is too small, as any change to an analogue value (± 1 digit) is reported to the DMS (unnecessary load on the system). The Diff value is based on the DMS value.

6.10.5 View: Data Blocks

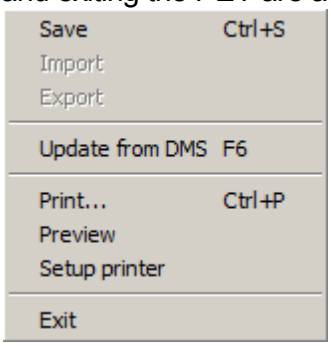
The following functions are planned:

- The data blocks have a special function with the SDriver.
- The configuration of objects can be saved in data blocks.
- For normal communication, data blocks should not be used
  (better only for configurations, for example, which are communicated with priority 0 or 99).

6.11 The PET menus

6.11.1 File menu

In the "**File**" menu, functions for saving, importing and exporting data, printing, print preview and exiting the PET are available.



6.11.1.1 Save

Save: **<CTRL+S>** or 

The project data that has been entered is saved in the current project. Alternatively, this can also take place through the menu command "File > Save".

All telegrams are automatically checked while saving. In doing so, assignment of the same telegram to multiple stations or data types is avoided.

If this check is not desired, this must be indicated in the file <proj>/cfg/pg5.cfg. Here the value for the multiple should be set to 0 in the section [Telegrams].

6.11.1.2 Import Data Points

Visi.Plus allows users to import data points via a text file with the ending ***.csv**.

Example:
"**System1.csv**"

csv (**c**omma **s**eparated **v**alue) means that the texts are saved in a simple format that can be processed in most programs.

The simplest way to do this is by entering the system data in rows and columns in a spreadsheet program, and saving the list as a "**csv**" **file**.

 As of the 1.6.103.30 setup, there are two ways of importing the data points.
Option 1: The extensions are defined and imported for all objects as done previously.
Option 2: The extensions are defined and imported for each object.

Both options are described in the following two chapters.

Visi.Plus offers the possibility of importing data points through a text file with the ending ***.csv**.

Example:
"**System1.csv**"

csv means that the text is saved in a simple format that can be processed by most programs (**C**omma **S**eparated **V**alue).

The easiest way to do this is by entering the system data in rows and columns in a spreadsheet program and saving the list as a "**csv**" **file**.

The type of data must be defined by a designation in the first line of the table. See the image with the information in the first line
Name, DMS Name, OBJECT, ...

The presence of these three pieces of information is mandatory!

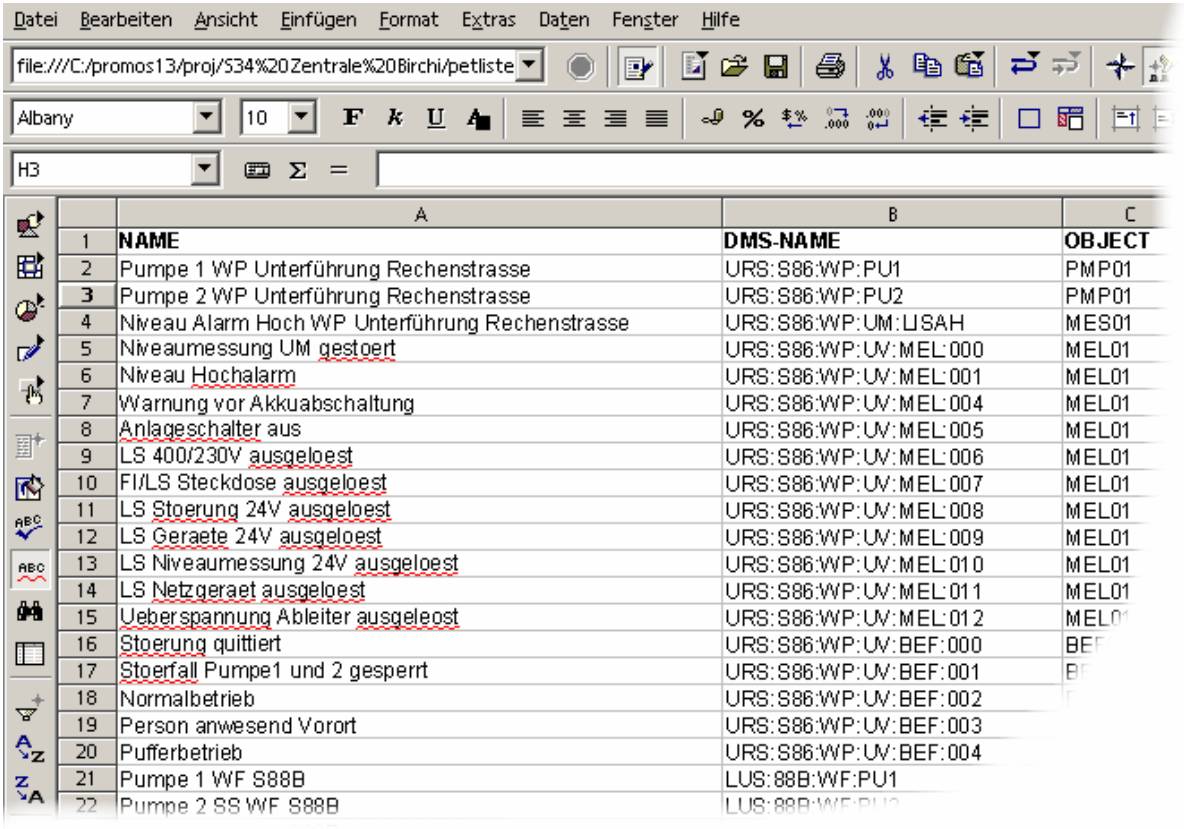
Additional columns can, of course, be added.

Example:
Requirement:
The "**csv**" file must contain at least the following information:

NAME
stands for object name

DMS Name
stands for DMS name

OBJECT
stands for object type



	A	B	C
1	**NAME**	**DMS-NAME**	**OBJECT**
2	Pumpe 1 WP Unterführung Rechenstrasse	URS:S86:WP:PU1	PMP01
3	Pumpe 2 WP Unterführung Rechenstrasse	URS:S86:WP:PU2	PMP01
4	Niveau Alarm Hoch WP Unterführung Rechenstrasse	URS:S86:WP:UM:LISAH	MES01
5	Niveaumessung UM gestoert	URS:S86:WP:UV:MEL:000	MEL01
6	Niveau Hochalarm	URS:S86:WP:UV:MEL:001	MEL01
7	Warnung vor Akkuabschaltung	URS:S86:WP:UV:MEL:004	MEL01
8	Anlageschalter aus	URS:S86:WP:UV:MEL:005	MEL01
9	LS 400/230V ausgeloest	URS:S86:WP:UV:MEL:006	MEL01
10	FI/LS Steckdose ausgeloest	URS:S86:WP:UV:MEL:007	MEL01
11	LS Stoerung 24V ausgeloest	URS:S86:WP:UV:MEL:008	MEL01
12	LS Geraete 24V ausgeloest	URS:S86:WP:UV:MEL:009	MEL01
13	LS Niveaumessung 24V ausgeloest	URS:S86:WP:UV:MEL:010	MEL01
14	LS Netzgeraet ausgeloest	URS:S86:WP:UV:MEL:011	MEL01
15	Ueberspannung Ableiter ausgeleost	URS:S86:WP:UV:MEL:012	MEL01
16	Stoerung quittiert	URS:S86:WP:UV:BEF:000	BEF
17	Stoerfall Pumpe1 und 2 gesperrt	URS:S86:WP:UV:BEF:001	BEF
18	Normalbetrieb	URS:S86:WP:UV:BEF:002	
19	Person anwesend Vorort	URS:S86:WP:UV:BEF:003	
20	Pufferbetrieb	URS:S86:WP:UV:BEF:004	
21	Pumpe 1 WF S88B	LUS:88B:WF:PU1	
22	Pumpe 2 SS WF S88B	LUS:88B:WF:PU2	

All objects are registered in the PET according to the list above.

Any additional parameters (columns) can be defined in this manner. For example, input or output addresses. You only have to make sure that the exact same name is written in the first line of the column as was defined as a template object attribute.

With the **Import** command, the file is selected and then all data defined in the PET or DMS is inserted into the "**csv**" file.

6.11.1.2.1 Import option 1

Determine the data type with a designation in the first row of the table. See the image with the designations in the first row.
Name, DMS name, OBJECT, etc.

These three designations must always be present!

You can of course add further columns.

Example:
Condition:
The "**csv**" file must contain at least the following information:

NAME
Stands for the object designation

DMS NAME
Stands for the DMS name

OBJECT
Stands for object type

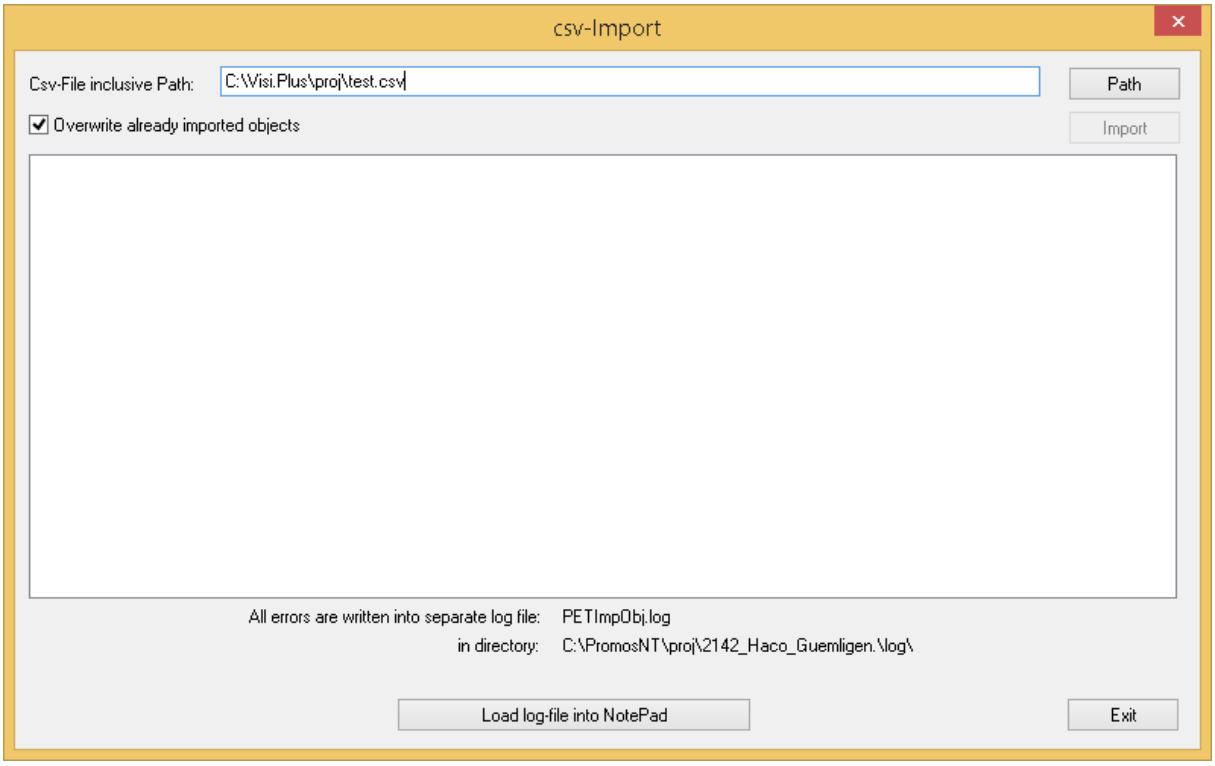
Any number of further parameters (columns) can be defined in this way, such as input and output addresses. All that is required is to ensure that the name is written in the first row of the column precisely as it is defined as a template object extension.

See this "**csv**" file as an example:

	A	B	C	D
1	NAME	DMS-NAME	OBJECT	ESchema
2	Temperatur AUL-Kanal	BHS60:MT:500	MES01	3310
3	ABL-Temperatur	BHS60:MT:502	MES01	3311
4	ZUL-Temperatur	BHS60:MT:503	MES01	3312
5	Druckdifferenz WRG	BHS60:MP:501	MES01	3313
6	ZUL-Rel. Feuchte	BHS60:MM:504	MES01	3314
7	ZUL-Abs. Feuchte	BHS60:MM:505	MES01	3315
8	Ventilator Zuluft	BHS60:AV:506	MOT01	3316
9	LE-Pumpe	BHS60:PW:507	MOT01	3317
10	LK-Pumpe	BHS60:PW:508	MOT01	3318

In this "**csv**" file, the data point ESchema has been included as an additional column.

Start the import via the menu **File > Import**. The following window opens:



Csv file including path
Enter the csv file here that you wish to import.

Path
Select the csv file here.

Import
Starts the import.

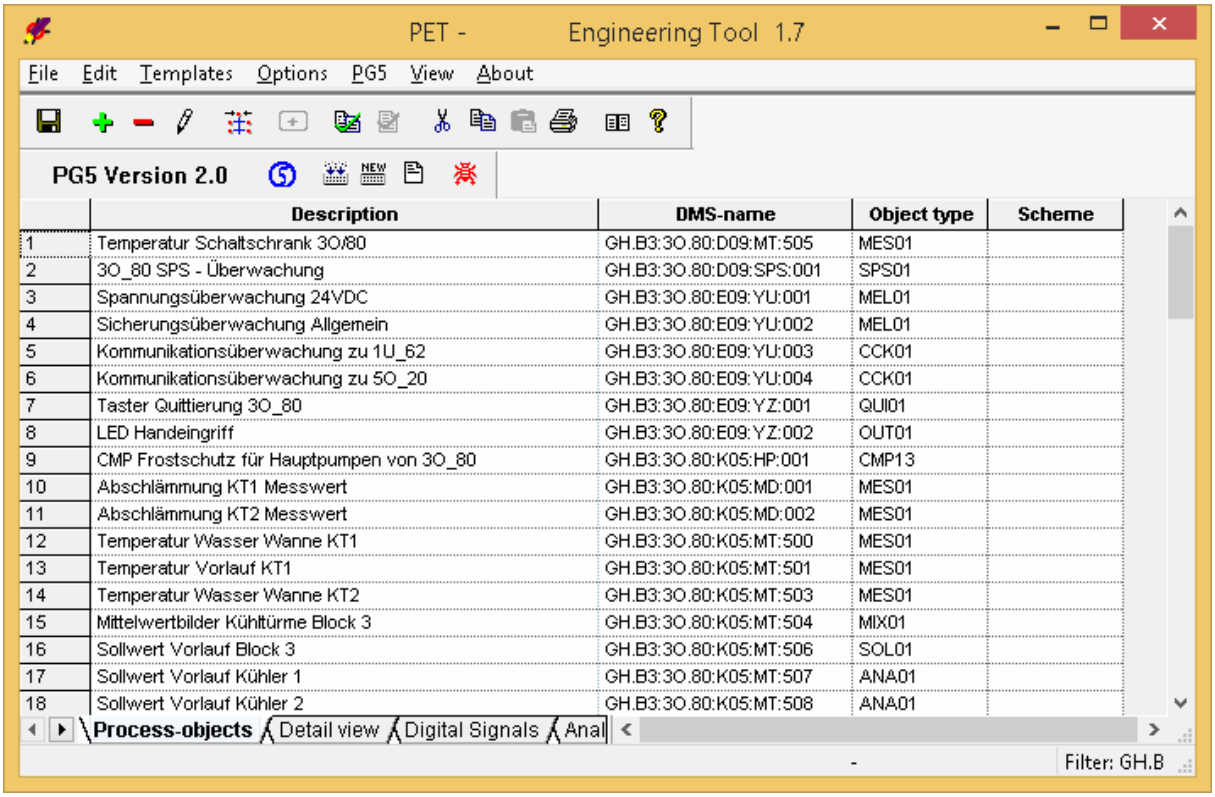
Overwrite previously imported objects
If active, the objects that have already been imported will be overwritten. This means that if the object type was changed, for example, the data point will be derived from the changed object type. All data points are also overwritten.

Debug window
This window displays all errors that occurred during the import.

Load log file in NotePad
Errors are recorded in the PETImpObj.log file. Use this button to open the file in NotePad.

End
Closes the window.

Once you have selected the "**csv**" file and clicked on **Import**, all of the data defined in the "**csv**" file is entered in the PET or DMS.



	Description	DMS-name	Object type	Scheme
1	Temperatur Schaltschrank 3O/80	GH.B3:3O.80:D09:MT:505	MES01	
2	3O_80 SPS - Überwachung	GH.B3:3O.80:D09:SPS:001	SPS01	
3	Spannungsüberwachung 24VDC	GH.B3:3O.80:E09:YU:001	MEL01	
4	Sicherungsüberwachung Allgemein	GH.B3:3O.80:E09:YU:002	MEL01	
5	Kommunikationsüberwachung zu 1U_62	GH.B3:3O.80:E09:YU:003	CCK01	
6	Kommunikationsüberwachung zu 5O_20	GH.B3:3O.80:E09:YU:004	CCK01	
7	Taster Quittierung 3O_80	GH.B3:3O.80:E09:YZ:001	QUI01	
8	LED Handeingriff	GH.B3:3O.80:E09:YZ:002	OUT01	
9	CMP Frostschutz für Hauptpumpen von 3O_80	GH.B3:3O.80:K05:HP:001	CMP13	
10	Abschlämmung KT1 Messwert	GH.B3:3O.80:K05:MD:001	MES01	
11	Abschlämmung KT2 Messwert	GH.B3:3O.80:K05:MD:002	MES01	
12	Temperatur Wasser Wanne KT1	GH.B3:3O.80:K05:MT:500	MES01	
13	Temperatur Vorlauf KT1	GH.B3:3O.80:K05:MT:501	MES01	
14	Temperatur Wasser Wanne KT2	GH.B3:3O.80:K05:MT:503	MES01	
15	Mittelwertbilder Kühltürme Block 3	GH.B3:3O.80:K05:MT:504	MIX01	
16	Sollwert Vorlauf Block 3	GH.B3:3O.80:K05:MT:506	SOL01	
17	Sollwert Vorlauf Kühler 1	GH.B3:3O.80:K05:MT:507	ANA01	
18	Sollwert Vorlauf Kühler 2	GH.B3:3O.80:K05:MT:508	ANA01	

6.11.1.2.2 Import option 2

It is possible to define the extensions to be imported for each object. This allows you to load entire configurations via the PET!
An import makes it possible to activate or deactivate alarms, for example.

The "**csv**" file is structured as follows:

	A	B	C	D	E	F
1	NAME	DMS-NAME	OBJECT	Datapoint	Value	Links

NAME
Stands for the object designation

DMS NAME
Stands for the DMS name

OBJECT
Stands for object type

Data point
Stands for the extension. The extension may contain a subtree (see the following table sub-datapoint RMP_Err:ALM:Alarm1:Active).

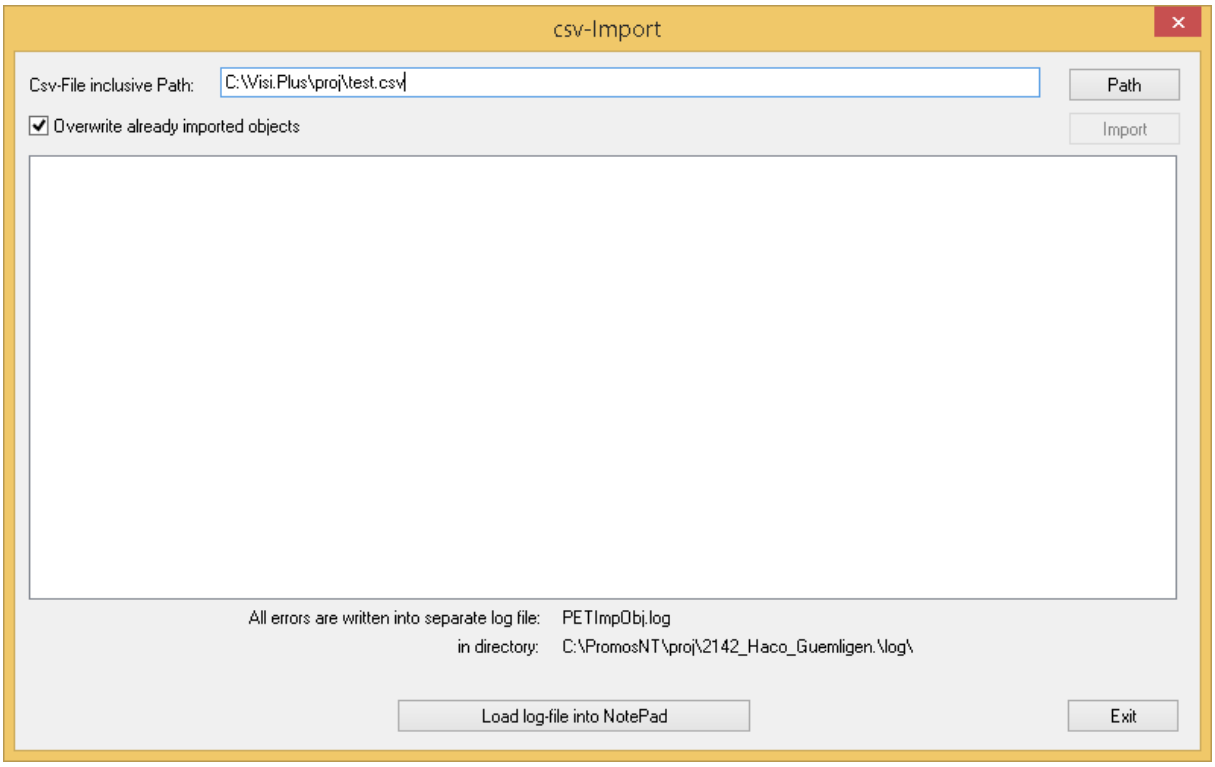
Value
Stands for the value (data point contents)

Links
Stands for ParIn

See the following "**csv**" file as an example:

	A	B	C	D	E	F
1	NAME	DMS-NAME	OBJECT	Datapoint	Value	Links
2	Temperatur AUL-Kanal	BHS60:MT:500	MES01	ESchema	3310	
3	Temperatur AUL-Kanal	BHS60:MT:500	MES01	Istwert	4711	
4	Ventilator Zuluft	BHS60:AV:506	MOT01	ESchema	3316	
5	Ventilator Zuluft	BHS60:AV:506	MOT01	ABS_Aktiv	1	
6	Ventilator Zuluft	BHS60:AV:506	MOT01	RMP_Err:ALM:Alarm1:Activ	1	
7	Ventilator Zuluft	BHS60:AV:506	MOT01	Freigabe	0	BH060:L01:LG:001:SWS01_Ein

Start the import via the menu **File > Import**. The following window opens:



csv file including path
Enter the csv file here that you wish to import.

Path
Select the csv file here.

Import
Starts the import.

Overwrite previously imported objects
If active, the objects that are already present during the import will be overwritten. This means that if the object type was changed, for example, the data point will be derived from the changed object type. All data points are also overwritten.

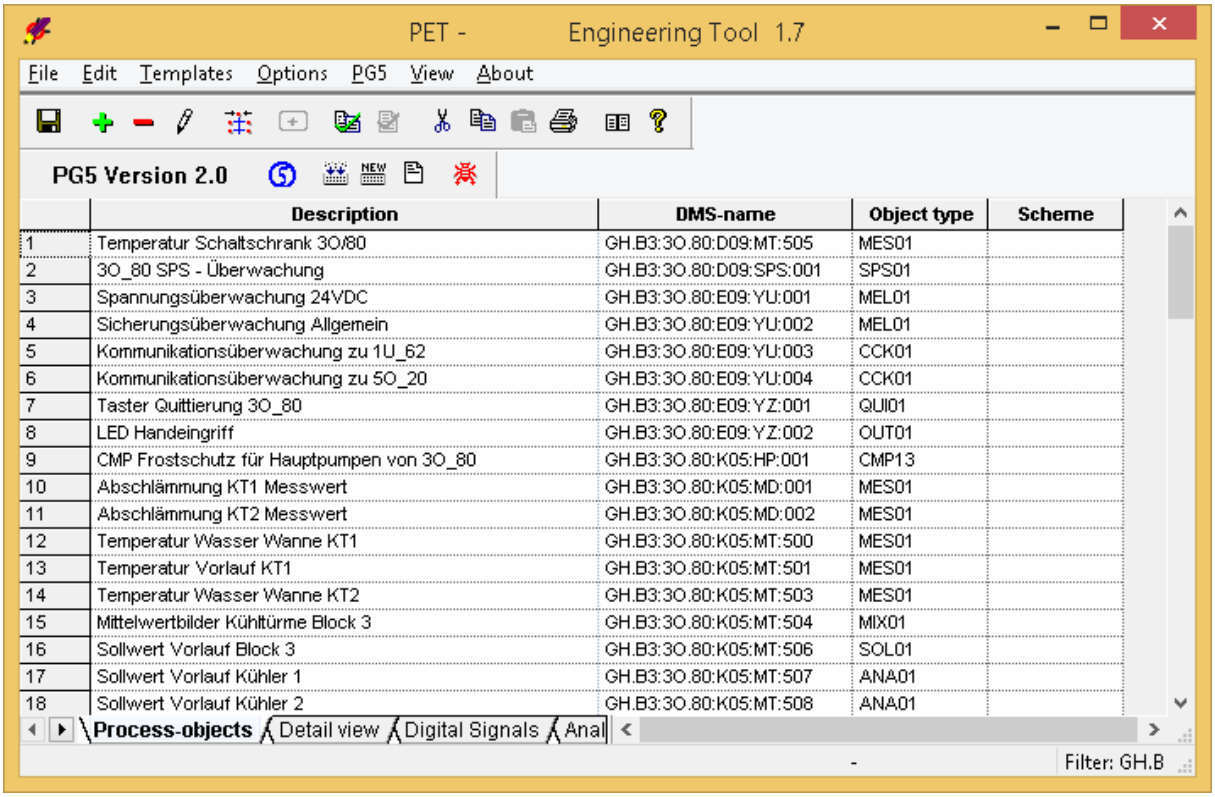
Debug window
This window displays all errors that occurred during the import.

Load log file in NotePad

Errors are recorded in the PETImpObj.log file. Use this button to open the file in a text editor.

End
Closes the window.

Once you have selected the "**csv**" file and clicked on **Import**, all of the data defined in the "**csv**" file is entered in the PET or DMS.



	Description	DMS-name	Object type	Scheme
1	Temperatur Schaltschrank 3O/80	GH.B3:3O.80:D09:MT:505	MES01	
2	3O_80 SPS - Überwachung	GH.B3:3O.80:D09:SPS:001	SPS01	
3	Spannungsüberwachung 24VDC	GH.B3:3O.80:E09:YU:001	MEL01	
4	Sicherungsüberwachung Allgemein	GH.B3:3O.80:E09:YU:002	MEL01	
5	Kommunikationsüberwachung zu 1U_62	GH.B3:3O.80:E09:YU:003	CCK01	
6	Kommunikationsüberwachung zu 5O_20	GH.B3:3O.80:E09:YU:004	CCK01	
7	Taster Quittierung 3O_80	GH.B3:3O.80:E09:YZ:001	QUI01	
8	LED Handeingriff	GH.B3:3O.80:E09:YZ:002	OUT01	
9	CMP Frostschutz für Hauptpumpen von 3O_80	GH.B3:3O.80:K05:HP:001	CMP13	
10	Abschlämmung KT1 Messwert	GH.B3:3O.80:K05:MD:001	MES01	
11	Abschlämmung KT2 Messwert	GH.B3:3O.80:K05:MD:002	MES01	
12	Temperatur Wasser Wanne KT1	GH.B3:3O.80:K05:MT:500	MES01	
13	Temperatur Vorlauf KT1	GH.B3:3O.80:K05:MT:501	MES01	
14	Temperatur Wasser Wanne KT2	GH.B3:3O.80:K05:MT:503	MES01	
15	Mittelwertbilder Kühltürme Block 3	GH.B3:3O.80:K05:MT:504	MIX01	
16	Sollwert Vorlauf Block 3	GH.B3:3O.80:K05:MT:506	SOL01	
17	Sollwert Vorlauf Kühler 1	GH.B3:3O.80:K05:MT:507	ANA01	
18	Sollwert Vorlauf Kühler 2	GH.B3:3O.80:K05:MT:508	ANA01	

You can check that the import has completed correctly in the detail view.

In this example, the data point **BHS60:MT:500:Actual** should have the value **4711**:

Istwert	Istwert		FLT	4711.000	SBUS N	

and the data point **BHS60:AV:506:Approval** should have the following values:

Freigabe	Freigabe Motor	BH060:L01:LG:001:SWS01_Ein	BIT	OFF	SBUS N

6.11.1.3 Export Data Points

Exports the data to a "**csv**" file. This file format can then be imported to practically any program, where the data can be processed further.

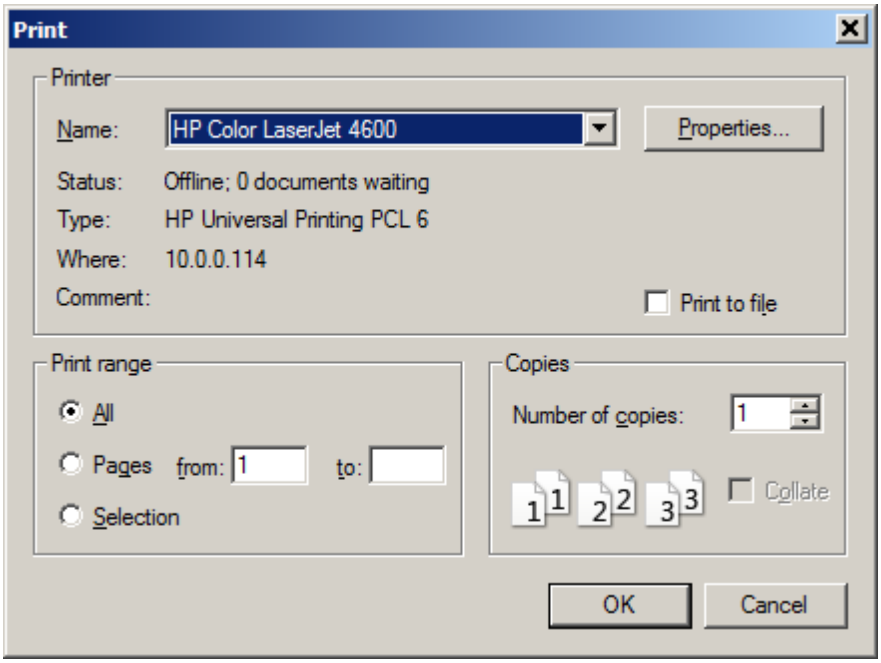
6.11.1.4 Update from DMS <F6>

Updates PET data to match the DMS data.

6.11.1.5 Print <CTRL+P>

Print: <CTRL+P> or 

Prints the current PET view.
The print settings can be changed in the Print dialog.



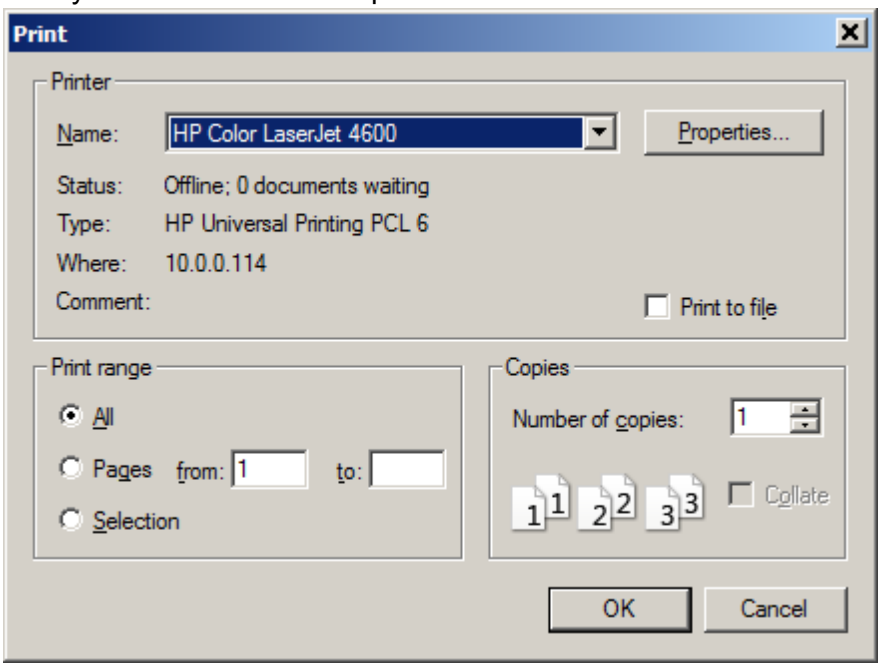
6.11.1.6 Print Preview

Print Preview: or 

Before printing a data point list, a preview of the printout can be displayed in order to adjust column widths, etc.

6.11.1.7 Printer Settings

Here you can set the default printer for the "**Print**" menu item.

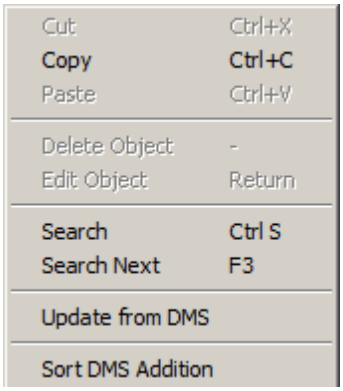


6.11.1.8 Exit

The PET is exited.

6.11.2 Edit menu

In the "**Edit**" menu there are different options for editing data points (Delete, Select, etc.).



*The commands "**Cut**", "**Copy**" and "**Paste**" are appropriate for editing text that is usually entered in the Detail View in the **Comments** column.*
No system objects can be edited with these commands!

6.11.2.1 Cut <CTRL+X>

 or **<CTRL+X>** or in the menu **Edit > Cut**

In order to cut text, select the desired text with the mouse and then select the function.
The cut segment is now located on the clipboard and can be reinserted in a new position.

6.11.2.2 Copy <CTRL+C>

 or **<CTRL+C>** or in the menu **Edit -> Copy**

In order to copy text, select the desired text with the mouse and then select the function.
The copied segment is now located on the clipboard and can be reinserted in a new position.

6.11.2.3 Paste <CTRL+V>

 or **<CTRL+V>** or in the menu **Edit -> Paste**

The paste command is used to insert the contents of the clipboard at the current cursor position.

6.11.2.4 Delete Object <->

 or **<DEL>** or in the **menu Edit > Delete Object**

Select the object for deletion in the System Object tab and then select the function. Multiple objects can be deleted simultaneously. To do so, select the system objects to be deleted with the left mouse button while holding down the Ctrl key.

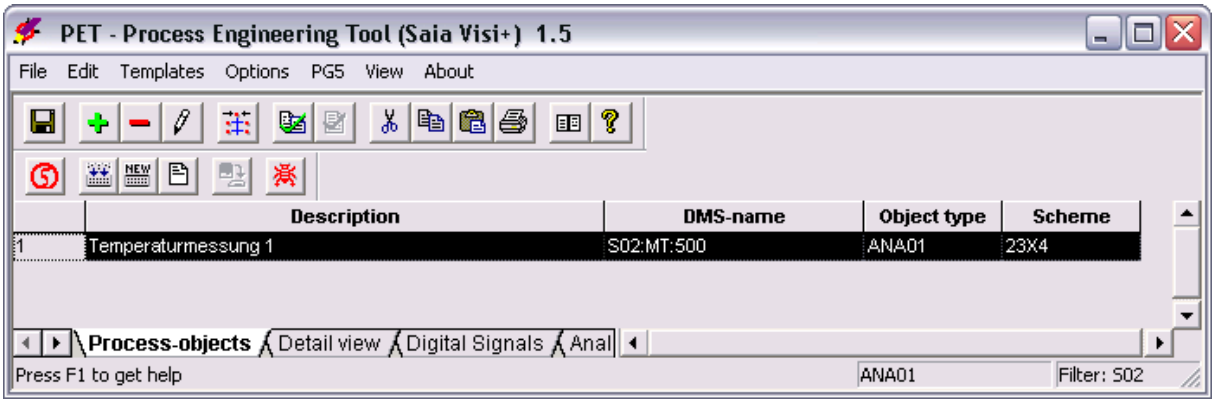
6.11.2.5 Edit Object <Return>

 or **<Return>** or in the menu **Edit > Edit Object**

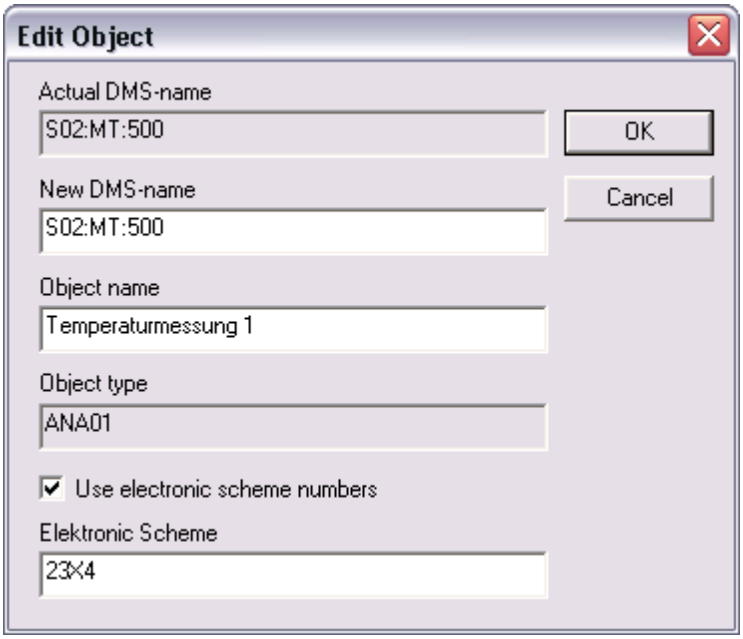
The attributes of a system object can be changed with this function. To do so, select the desired system object in the **System Objects** tab and then select the function.

Renaming can be better explained using an example:

For this purpose, create an object of type ANA01 (see the chapter Creating a system object):



Select only the first line and then press the **<Return>** key. The following input dialog opens:



The dialog adopts the attributes of the selected system objects. These can now be modified all the way up to the object type.

Current DMS Name
Displays the DMS name of the current object.

New DMS Name
The new DMS name can be entered here.

Object designation
A new object name can be defined here.

Object type
Displays the object type. This cannot be edited.

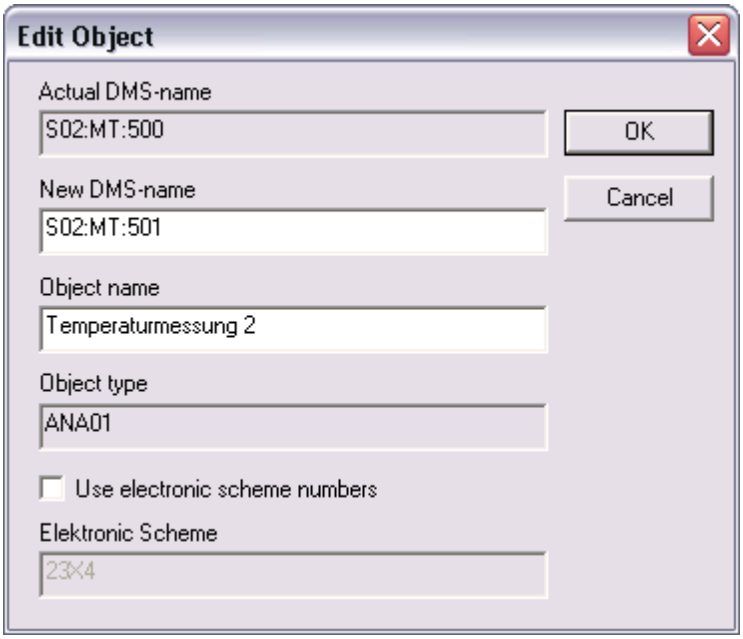
Insert Electrical Diagram Name
Defines whether an electrical diagram name should be inserted.

Electrical Diagram Name
Here a new name can be entered for the electrical diagram.

<OK>
Renames the data point in the DMS (if the DMS name was modified), writes the new object attributes to the DMS and modifies the files accordingly (see below).

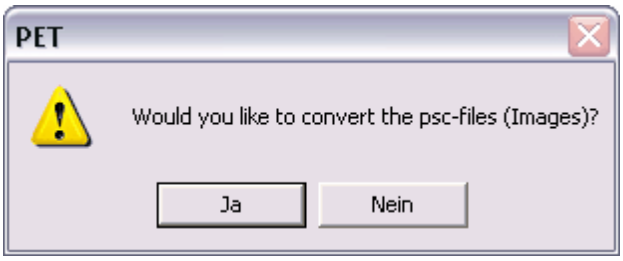
<Cancel>
Closes the dialog without adopting the attributes.

The object attributes should now be modified as follows:



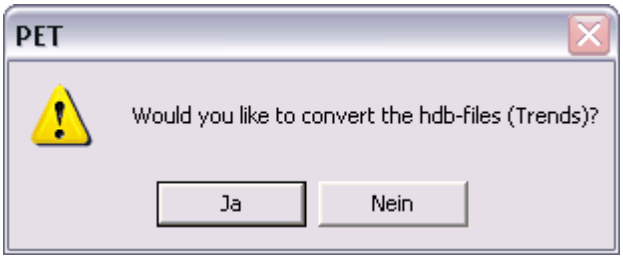
Then click on **<OK>**. By doing so, the DMS data point S01:MT:501 is renamed as S01:MT:502 and the object attributes are adopted to the DMS. In addition, the PSC, HDB and PDB files are modified, if desired. This is achieved by displaying the following message windows one after another:

- **PSC files (image files of the GE)**



Click on **<Yes>** if you want the files to be modified; if not, click on **<No>**. This also applies to the following dialogs. The filename and the DMS name are updated in the header in the HDB files.

- **HDB files (trend graphs)**

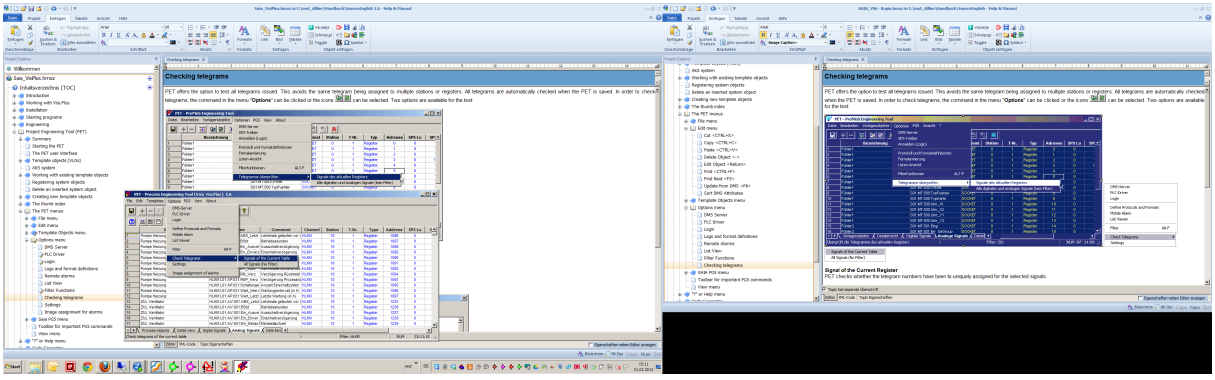


The initialised DMS names are updated in the PSC files. If images are open in the GE, they are then reloaded.

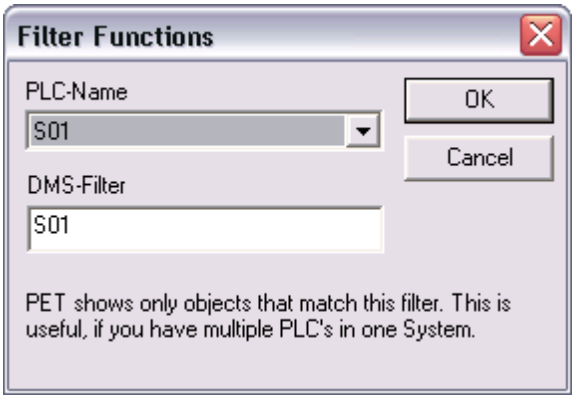
- **PCD files (*.scr, *.fup, *.rxp, *.sfc)**

The DMS names are updated in the PCD files.

A valid filter is necessary for modifying the PCD files. If the current DMS filter is set to default, the following message window is shown:

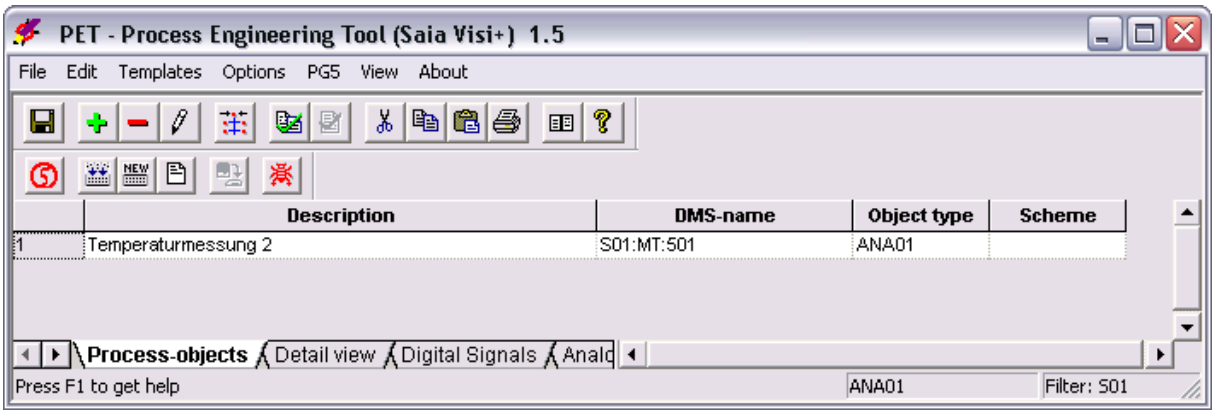


Click on **<Yes>** to open the filter dialog. Use the dropdown menu to select the PLC, in this case S01:



Then click on **<OK>**. The files are now converted.

Then the table of system objects is updated:



With larger projects, renaming can take several minutes! The process is completed when the hourglass turns into an arrow again.

6.11.2.6 Find <CTRL+F>

<CTRL+F> is a very helpful function for quickly finding specific data points again.
Alternatively, this can also take place through the menu "**Edit > Find**".
Before the Find command is selected, select the column to be searched through. After the command is select, the "**Find**" dialog window appears.

The desired search term is entered in the input field. The selected column is searched after clicking on **<OK>**.

6.11.2.7 Find Next <F3>

If a specific word, which appears several times in the data point list, is searched for, the menu command "**Edit > Find Next**" or the **<F3>** key should be used. The next find location is displayed, if available.

6.11.2.8 Update from DMS <F6>

This command re-reads all data (analogue values, digital values, etc.) from the DMS (Visi.Plus Data Management System) and updates the PET data.

6.11.2.9 Sort DMS Attributes

This function is only active in the **Digital Signals**, **Analogue Signals** or **Data Blocks** tabs.

The data is arranged hierarchically according to DMS name. For example, first all signals of temperature measurement S01:MES:000 are arranged alphabetically, then all data of measurement S02:MES:000.

The column is sorted by double-clicking on any column header.

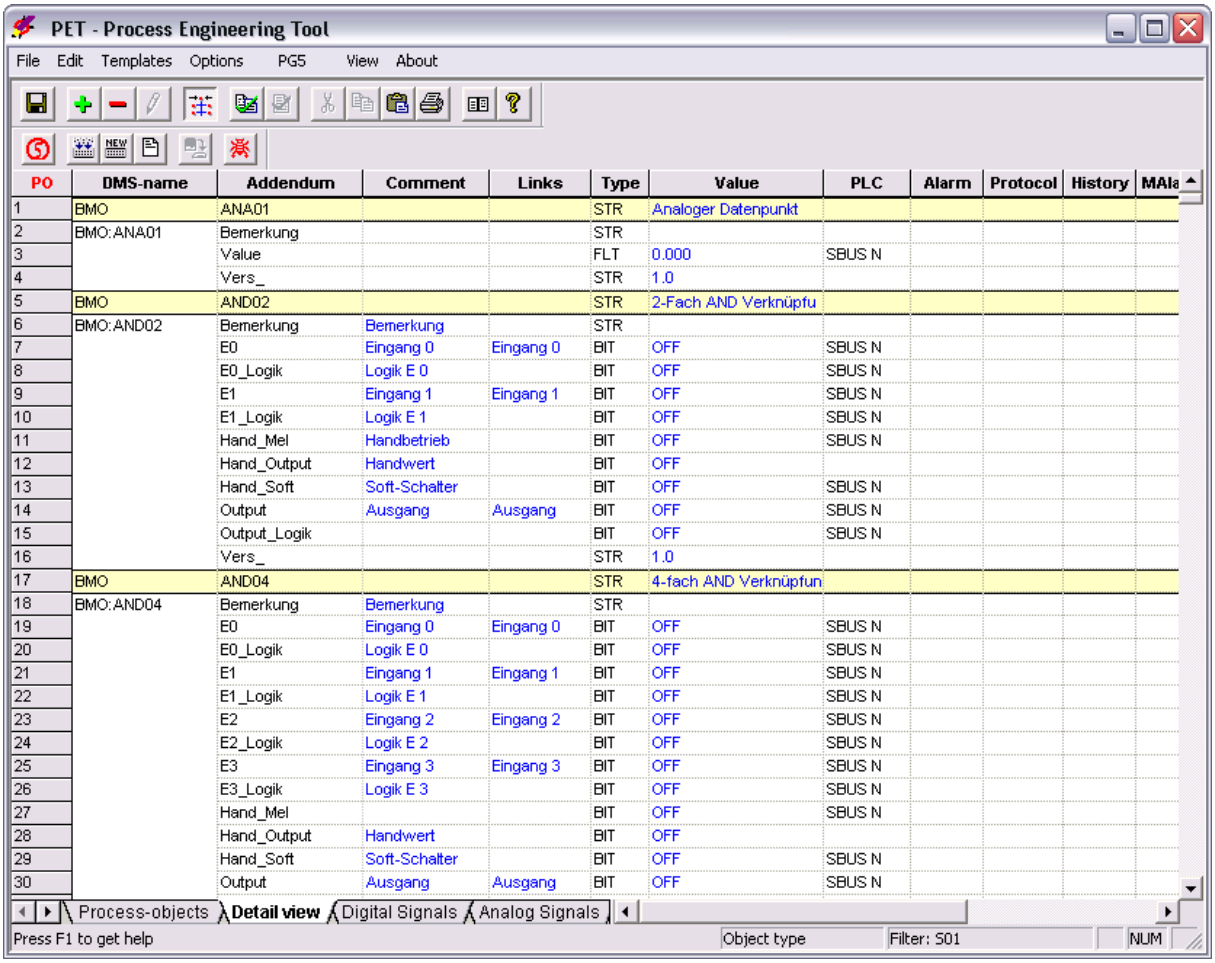
6.11.3 Template Objects menu

The "**Template Objects**" menu enables the creation of **new template objects** (**VLOs**) and their **attributes**. **VLOs** that have already been created or provided by third parties can be **integrated** here.

The chapter Template objects (VLOs) describes what VLOs are and how they are defined.

6.11.3.1 Template View <F5>

By pressing the  button in the toolbar, the **<F5>** key or with the menu command "**Template View**", all existing VLOs in the current project are listed.



PO	DMS-name	Addendum	Comment	Links	Type	Value	PLC	Alarm	Protocol	History	MAla
1	BMO	ANA01			STR	Analoger Datenpunkt					
2	BMO:ANA01	Bemerkung			STR						
3		Value			FLT	0.000	SBUS N				
4		Vers_			STR	1.0					
5	BMO	AND02			STR	2-Fach AND Verknüpfu					
6	BMO:AND02	Bemerkung	Bemerkung		STR						
7		E0	Eingang 0	Eingang 0	BIT	OFF	SBUS N				
8		E0_Logik	Logik E 0		BIT	OFF	SBUS N				
9		E1	Eingang 1	Eingang 1	BIT	OFF	SBUS N				
10		E1_Logik	Logik E 1		BIT	OFF	SBUS N				
11		Hand_Mel	Handbetrieb		BIT	OFF	SBUS N				
12		Hand_Output	Handwert		BIT	OFF					
13		Hand_Soft	Soft-Schalter		BIT	OFF	SBUS N				
14		Output	Ausgang	Ausgang	BIT	OFF	SBUS N				
15		Output_Logik			BIT	OFF	SBUS N				
16		Vers_			STR	1.0					
17	BMO	AND04			STR	4-fach AND Verknüpfun					
18	BMO:AND04	Bemerkung	Bemerkung		STR						
19		E0	Eingang 0	Eingang 0	BIT	OFF	SBUS N				
20		E0_Logik	Logik E 0		BIT	OFF	SBUS N				
21		E1	Eingang 1	Eingang 1	BIT	OFF	SBUS N				
22		E1_Logik	Logik E 1		BIT	OFF	SBUS N				
23		E2	Eingang 2	Eingang 2	BIT	OFF	SBUS N				
24		E2_Logik	Logik E 2		BIT	OFF	SBUS N				
25		E3	Eingang 3	Eingang 3	BIT	OFF	SBUS N				
26		E3_Logik	Logik E 3		BIT	OFF	SBUS N				
27		Hand_Mel			BIT	OFF	SBUS N				
28		Hand_Output	Handwert		BIT	OFF					
29		Hand_Soft	Soft-Schalter		BIT	OFF	SBUS N				
30		Output	Ausgang	Ausgang	BIT	OFF	SBUS N				

6.11.3.2 New Template Object (VLO)

Requirement:

Template View activated! (pressed!)

Creates a new template object. The description for this is found in the chapter under "Creating new template objects".

6.11.3.3 New Template Object Attribute <+>

Requirement:

Template View activated! (pressed!)

Creates a new attribute under the name of the template object (PLC element, etc.). The integration of template object attributes is described in this chapter under **New Template Object Attribute <+>**.

Deleting a template object or attribute

In order to delete a template object attribute, mark the respective attribute and then press the

<DEL> key or click on the **Delete Data Point** icon  .

6.11.3.4 Templates from Library

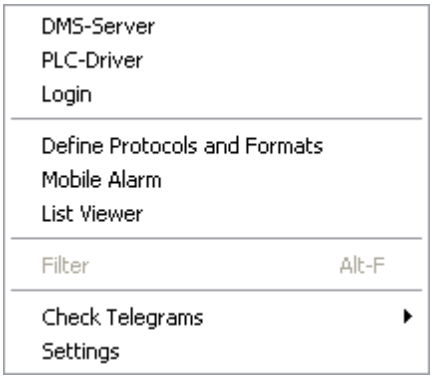
Requirement:

Template View activated! ( pressed!)

Template objects can be installed here. The exact procedure is described in the chapter under Installing existing template objects.

6.11.4 Options menu

In the "**Options**" menu bar, basic settings for the project can be made. In addition to driver selection and indication of the DMS server's location (the computer with the database), the definitions of the output and/or message texts for alarms and logs are also found here.



6.11.4.1 DMS Server

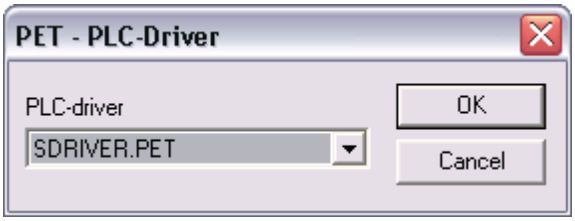
The **DMS Server** setting indicates the name of the computer that the DMS is running on. By default, i.e. after the installation of Visi.Plus, the name is a dot (.). meaning that the DMS is located on the computer where the visualisation project is located.

If an additional computer, e.g. with only the visualisation component installed, is set up, the network name of the computer where the DMS is located must be entered.

 *A default installation of Visi.Plus must also be carried out on this second computer, although only the GE is necessary for visualisation. In order for this computer to find the DMS, enter the network name of the computer with the DMS under "**Options > DMS Server**" in the menu.*

6.11.4.2 PLC Driver

A driver is required for exchanging data between Visi.Plus and the PLC. The driver **sdriver.pet** is available by default.



The desired driver can be selected by clicking on the dropdown menu. We recommend always using the latest driver.
For more information about drivers, see the **chapter** Communications drivers.

6.11.4.3 Login

Normally, different user levels are created in a project. Users with administrator rights can make all modifications that are possible within a project.

The Login **command** opens the Login dialog window.

User
Name of the user
Users must be registered in the **User Management** module before they can log in.

Password
The password of the user entered in the **User Management** module

The **chapter** User management describes how users are created.

In the PET a user can log in at any time by pressing the key combination ***<CTRL+L>*** *or selecting the command "**Login**" in the "**Options**" menu.*

6.11.4.4 Logs and format definitions

Before an alarm can be logged in Visi.Plus, an alarm log and its desired format must be defined. Selecting this function opens the Visi.Plus **Log Format Program**. The chapter **PRTFormat describes how a log** and its **format** are defined.

6.11.4.5 Remote alarms

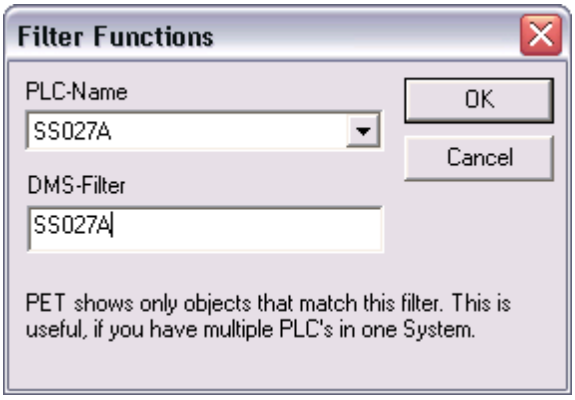
With this command the program **AlarmConfig** is started. How to define remote alarms is described in the **chapter** **MALM Configurator**.

6.11.4.6 List View

System data can be viewed and edited with the command **List view**. The command starts the program **pList**. The **chapter** **pList** describes how pList is used.

6.11.4.7 Filter Functions

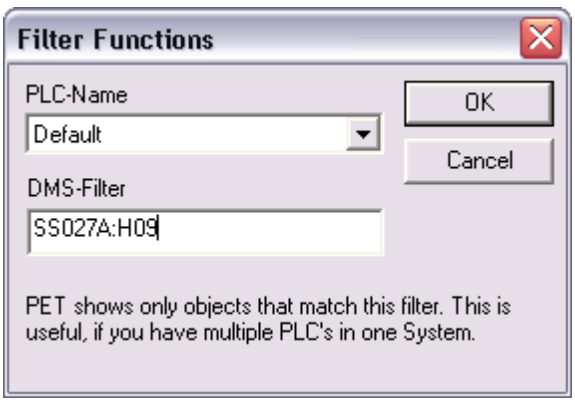
The Filter dialog can be called up through the menu or using the key combination **<Alt+F>**.



A **DMS filter** can be specified for each PLC created in PG5 that is to be **filtered** by. Only objects that match the filter criteria are still shown. This option is very useful if multiple PLC systems are present in a visualisation system.

If a new filter should be created for a PLC, the PLC must be selected via the dropdown menu and the filter must be entered in the field DMS Filter. The settings are adopted if a new PLC is selected or the dialog box is exited with **OK**.

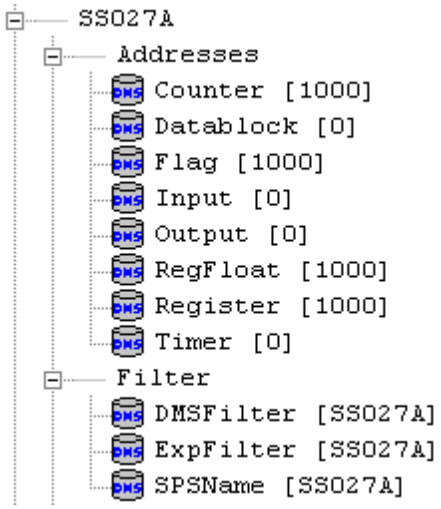
As its first entry**, the dropdown menu always contains** the value **Default**.



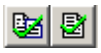
Any filter can be entered here.If **all objects should be displayed**, the input field must remain

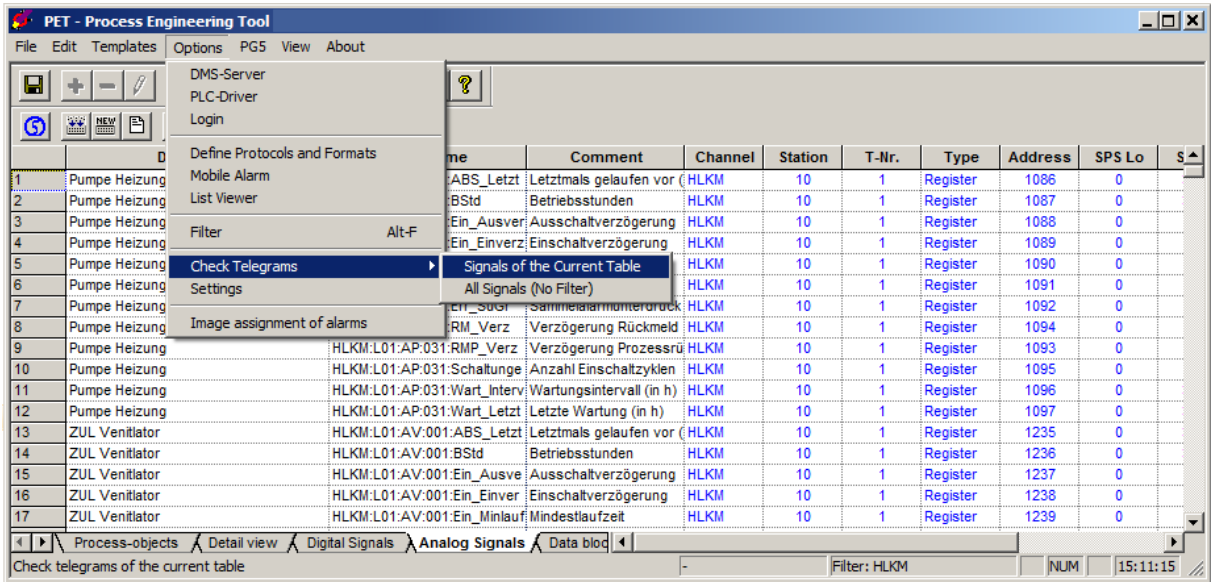
empty. The Default entry is necessary because it should be possible to filter even if no PLCs were created in PG5.

The filters are saved in the DMS in the data point system:Prg:PCD:<PLCName>:Filter.



6.11.4.8 Checking telegrams

PET offers the option to test all telegrams issued. This avoids the same telegram being assigned to multiple stations or registers. All telegrams are automatically checked when the PET is saved. In order to check telegrams, the command in the menu "**Options**" can be clicked or the icons  can be selected. Two options are available for the test:

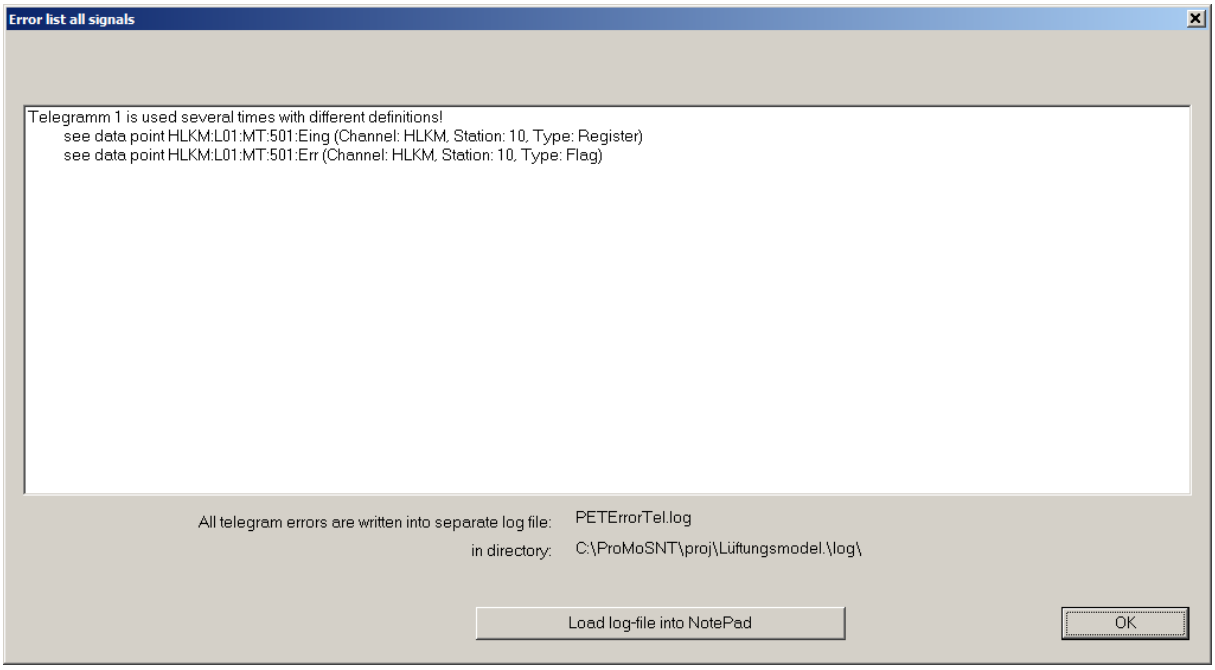


Signal of the Current Register
PET checks whether the telegram numbers have been bi-uniquely assigned for the selected signals.

All Digital and Analogue Signals (no filter)

If this command selected, the PET tests all digital and analogue signals for doubly-assigned telegrams.

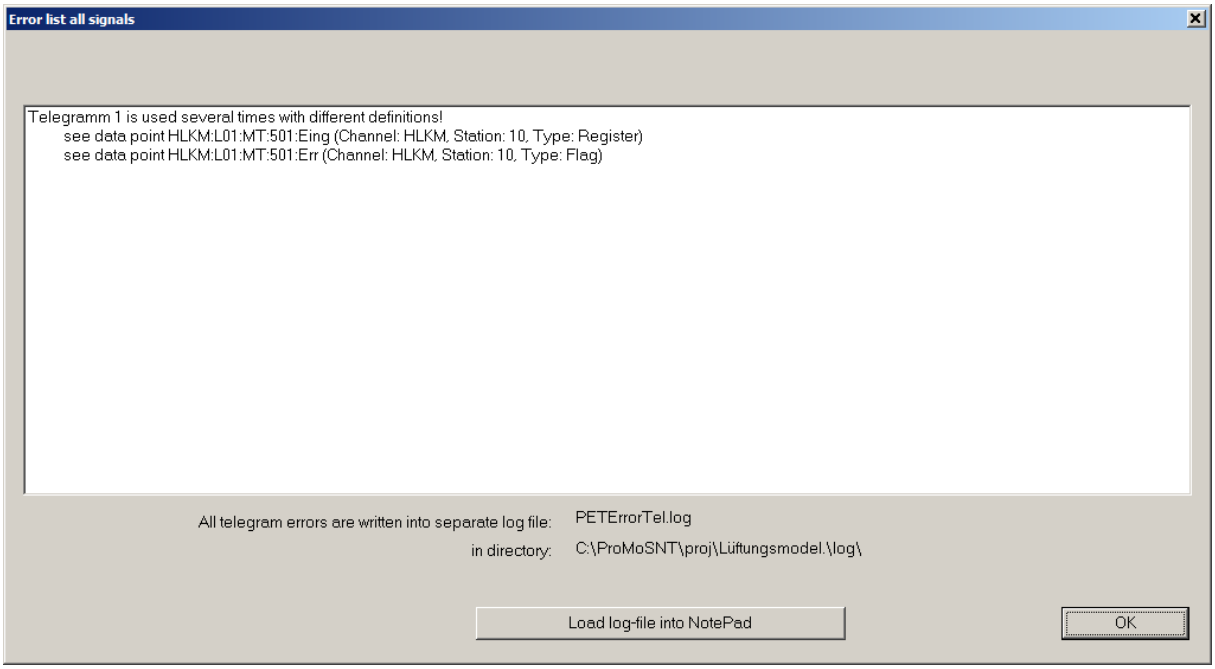
If some telegram numbers have been assigned twice in a project, the PET outputs them in an error list:



In the preceding image the telegram number 1 has been assigned for the digital and for the analogue signals.

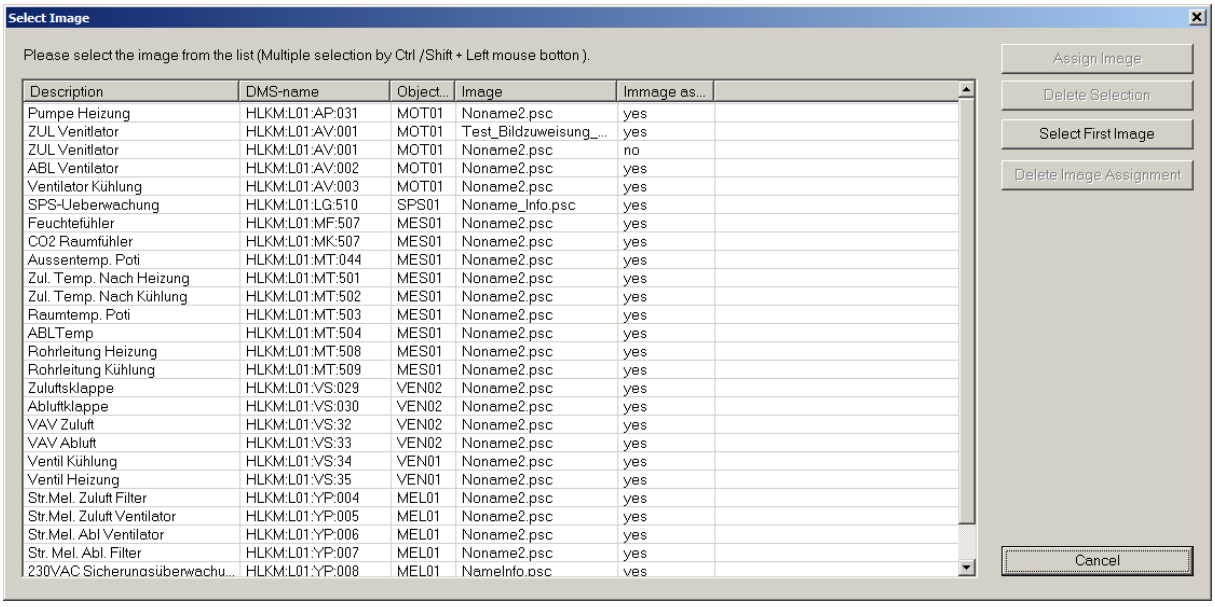
6.11.4.9 Settings

If this checkbox is activated, the telegrams from the digital signals, the analogue signals and the data blocks are compared to one another. If there are overlaps, a message is output when saving.



6.11.4.10 Image assignment for alarms

If an alarm is visualised in several images in the GE, the option **"Image assignment for alarms"** can be used to choose between the images. After the progress bar, the following screen opens:

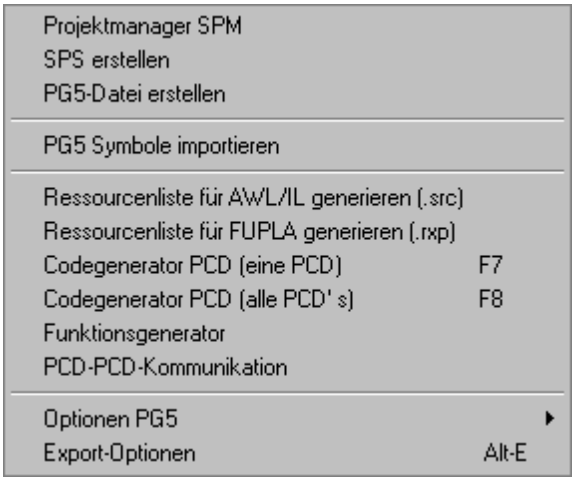


Description	DMS-name	Object...	Image	Immage as...
Pumpe Heizung	HLKM:L01:AP:031	MOT01	Noname2.psc	yes
ZUL Venitlator	HLKM:L01:AV:001	MOT01	Test_Bildzuweisung_...	yes
ZUL Venitlator	HLKM:L01:AV:001	MOT01	Noname2.psc	no
ABL Ventilator	HLKM:L01:AV:002	MOT01	Noname2.psc	yes
Ventilator Kühlung	HLKM:L01:AV:003	MOT01	Noname2.psc	yes
SPS-Ueberwachung	HLKM:L01:LG:510	SPS01	Noname_Info.psc	yes
Feuchtefühler	HLKM:L01:MF:507	MES01	Noname2.psc	yes
CO2 Raumfühler	HLKM:L01:MK:507	MES01	Noname2.psc	yes
Aussentemp. Poti	HLKM:L01:MT:044	MES01	Noname2.psc	yes
Zul. Temp. Nach Heizung	HLKM:L01:MT:501	MES01	Noname2.psc	yes
Zul. Temp. Nach Kühlung	HLKM:L01:MT:502	MES01	Noname2.psc	yes
Raumtemp. Poti	HLKM:L01:MT:503	MES01	Noname2.psc	yes
ABLTemp	HLKM:L01:MT:504	MES01	Noname2.psc	yes
Rohrleitung Heizung	HLKM:L01:MT:508	MES01	Noname2.psc	yes
Rohrleitung Kühlung	HLKM:L01:MT:509	MES01	Noname2.psc	yes
Zuluftsklappe	HLKM:L01:VS:029	VEN02	Noname2.psc	yes
Abluftklappe	HLKM:L01:VS:030	VEN02	Noname2.psc	yes
VAV Zuluft	HLKM:L01:VS:32	VEN02	Noname2.psc	yes
VAV Abluft	HLKM:L01:VS:33	VEN02	Noname2.psc	yes
Ventil Kühlung	HLKM:L01:VS:34	VEN01	Noname2.psc	yes
Ventil Heizung	HLKM:L01:VS:35	VEN01	Noname2.psc	yes
Str.Mel. Zuluft Filter	HLKM:L01:YP:004	MEL01	Noname2.psc	yes
Str.Mel. Zuluft Ventilator	HLKM:L01:YP:005	MEL01	Noname2.psc	yes
Str.Mel. Abl Ventilator	HLKM:L01:YP:006	MEL01	Noname2.psc	yes
Str. Mel. Abl. Filter	HLKM:L01:YP:007	MEL01	Noname2.psc	yes
230VAC Sicherungsüberwachu...	HLKM:L01:YP:008	MEL01	NameInfo.psc	yes

If several images are selected for one alarm, e.g. with <CTRL> + left mouse button, then the last one is selected automatically.

6.11.5 PG5 menu

This menu is used for working conjointly with **Visi.Plus** and **PG5**.



6.11.5.1 SPM Project Manager

The PG5 programming software can be started directly from the **PET** with the menu command "**SPM Project Manager**".

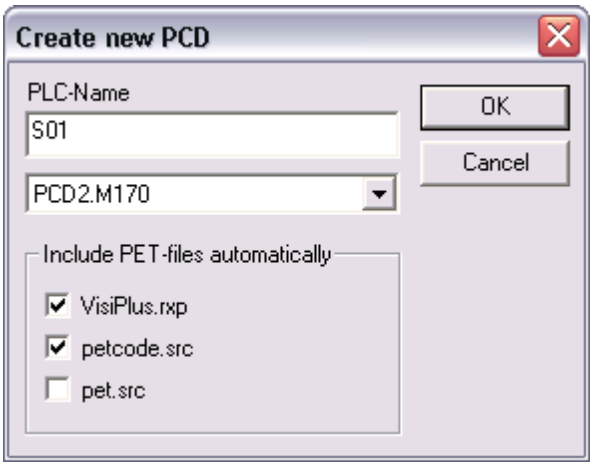
Alternatively, the PG5 icon can also be clicked on.

6.11.5.2 Create PLC

Requirement:

SPM Project Manager has already been started (see the menu command "**PG5 -> SPM Project Manager**")!

In the current PG5 project this menu item places a new PLC (CPU) in the Project Manager. The following window is shown so it can be given a name and CPU type can be selected:



Name of the PLC
Enter the name of the PLC here. We recommend also using the name of the PLC as the first part of the DMS name, e.g. "**S01**:MT:500". "S01:" stands for Station1.

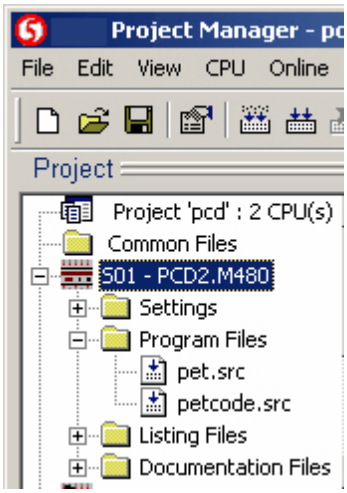
Dropdown menu

Enables the selection of the desired PLC type. In the example PCD2.M170.

Automatically Insert PET Files
When checked, it creates the PLC code files **VisiPlus.rxp, petcode.src** and **pet.src** in the new PLC.

<OK>
Creates the PLC in the PG5 "**Project Manager**"

The results could look like the following image.



*The two files contain data for the **Code Generation (AWL)** module of Visi.Plus (optional).*
If desired, the PET generates the following data in the files:

pet.src
All variable declarations and assignments

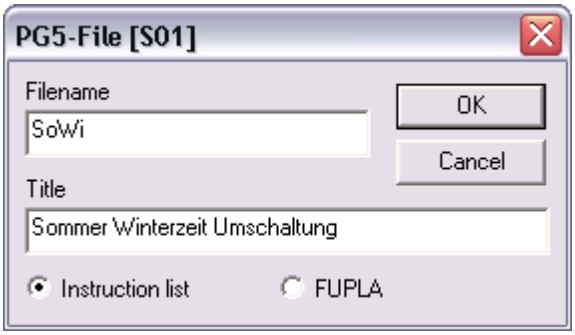
petcode.src
All FB calls and their parameters

visiplus.rxp
All variable declarations and assignments as a resource file

6.11.5.3 Create PG5 File

In order to not have to switch to PG5 for each new program file, a **new file** can be created under the active PLC with this menu command in the Saia PG5® Project Manager.

By selecting the menu item "**Create PG5 File**", the following window appears:



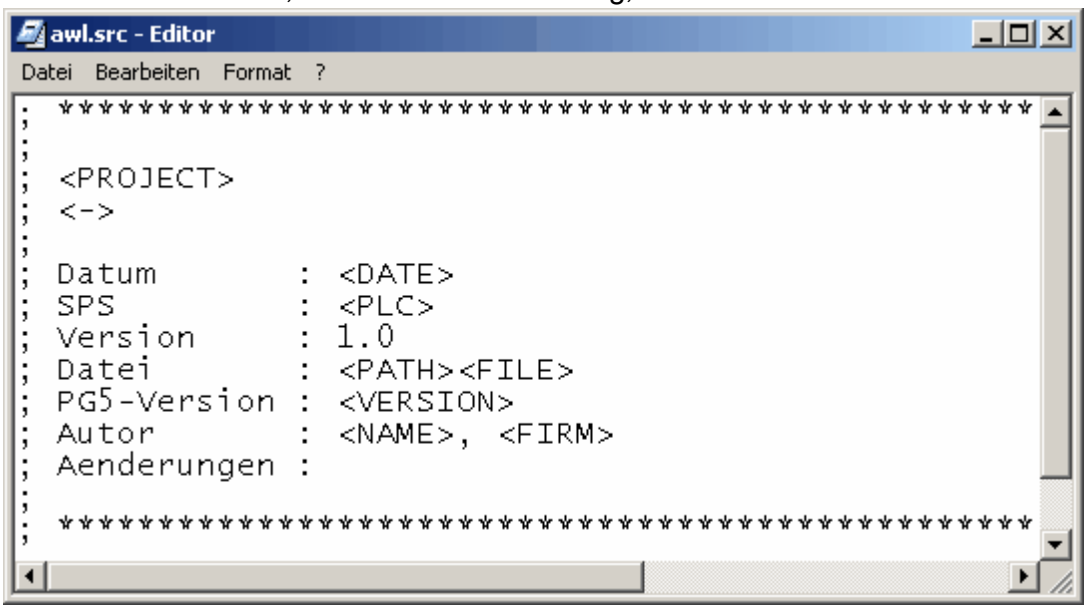
Window Title Bar
The name of the active PLC in which a new file will be created is shown in square brackets.

Filename
Name of the file to be created

Title
Additional title (is shown in the Saia PG5® Project Manager)

Instruction List (IL)
Continuous Function Chart (Fupla)
Selection of file type (AWL or Fupla file)

The AWL file contains, as shown in the following, a "**text header**".

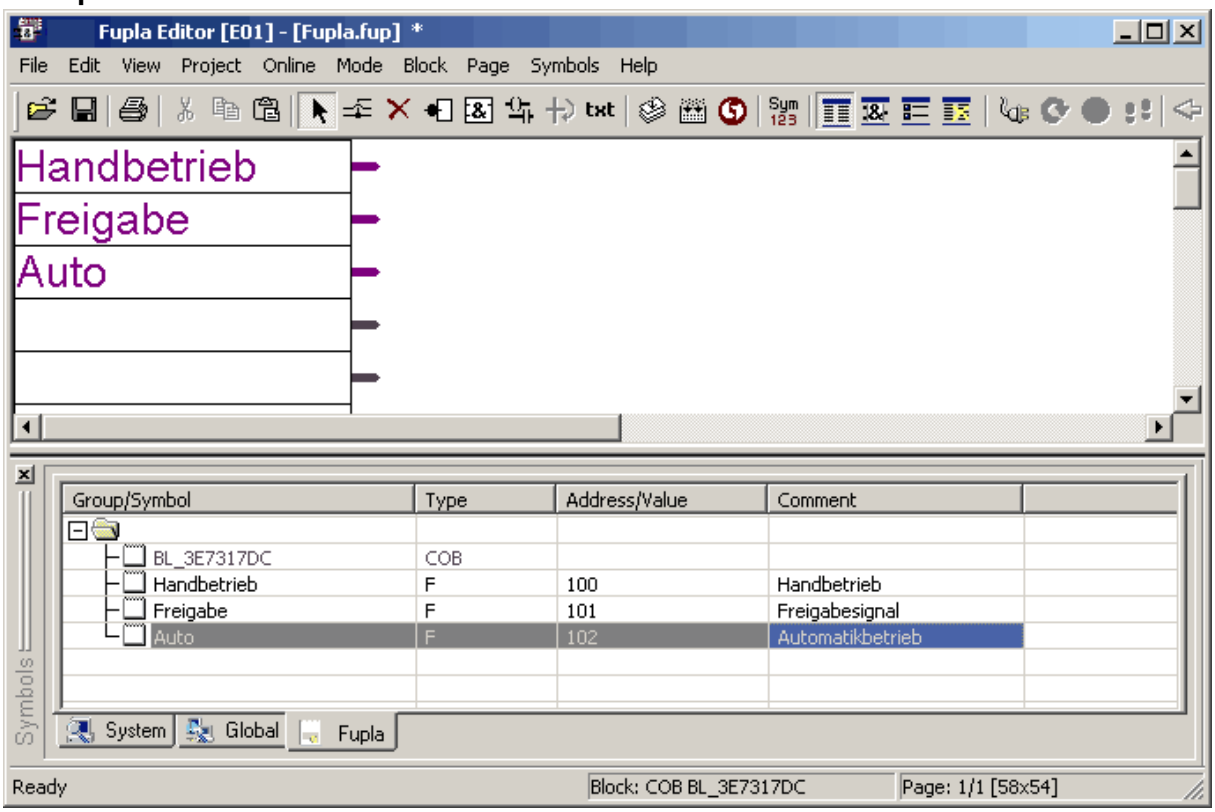


*When using the function "**Create PG5 File**" for the first time, the control window "**AWL Options**" opens up on creation of an AWL file.*
More information about this text header file can be found under AWL Settings.

6.11.5.4 Import PG5 Symbols

With this command the Fupla symbols defined in the Saia PCD® project are imported into the PET.

Example:



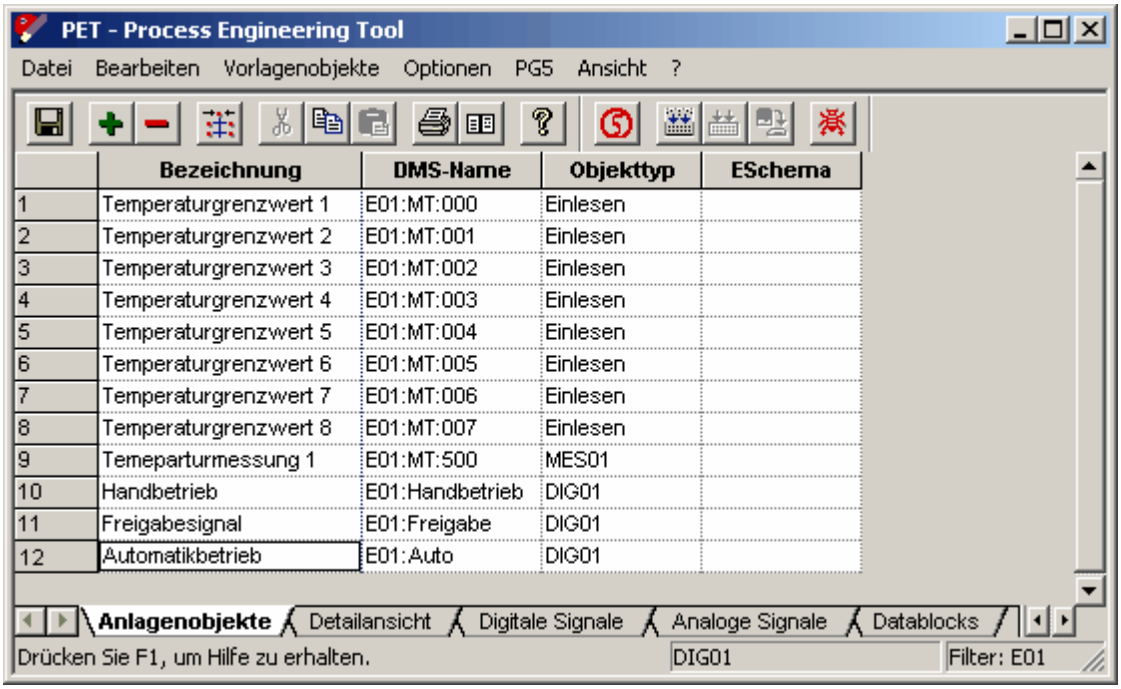
*In order to import symbols from PG5, the two template objects **ANA01** and **DIG01** must be installed from the template object library (see Installing existing template objects).*

*A valid PLC filter must also be set in the PET under the menu command "**Options / Filter Functions**", otherwise all symbols are imported from all CPUs.*

In the preceding image three digital signals of the type 'flag' with the addresses 100-102 were created.

As soon as changes to the Saia PG5® Fupla symbol table or program changes have been made, they must be saved in the current PLC before importing them into the PET.

After executing the command "**Import PG5 Symbols**" in the PET, the following view may appear.



	Bezeichnung	DMS-Name	Objekttyp	ESchema
1	Temperaturgrenzwert 1	E01:MT:000	Einlesen	
2	Temperaturgrenzwert 2	E01:MT:001	Einlesen	
3	Temperaturgrenzwert 3	E01:MT:002	Einlesen	
4	Temperaturgrenzwert 4	E01:MT:003	Einlesen	
5	Temperaturgrenzwert 5	E01:MT:004	Einlesen	
6	Temperaturgrenzwert 6	E01:MT:005	Einlesen	
7	Temperaturgrenzwert 7	E01:MT:006	Einlesen	
8	Temperaturgrenzwert 8	E01:MT:007	Einlesen	
9	Temeparturmessung 1	E01:MT:500	MES01	
10	Handbetrieb	E01:Handbetrieb	DIG01	
11	Freigabesignal	E01:Freigabe	DIG01	
12	Automatikbetrieb	E01:Auto	DIG01	

The generated digital Fupla symbols were adopted into the **PET** as object type "**DIG01**" (the last three lines).

*The **template objects DIG01** and **ANA01** are only used for importing digital or analogue signals from PG5 into the PET.*

*As shown in the example above, the digital and analogue signals are always derived from template objects **DIG01** and **ANA01** on import. This approach for the insertion of objects is not optimal, because a separate object is created for each register and flag. This function should only be used for smaller projects or existing projects!*

The following files are necessary for the import:

- **FDB file**
  Is generated by PG5 when compiling the project and contains the name of the FBox. The DMS name for the new system object can be generated from this.

- **Resources file**
  Is generated by PG5 when compiling the project and contains the resources for the FBox. When creating FBoxes of the DDC suite family, this is generated in the global symbol table and saved in the file **...pcd\<PLCname>_Global.sy5**. The resources of FBoxes from the HVAC family are saved in the file **...pcd\<PLCname>\sym_System.sy5** and displayed in the System tab in the Fupla Editor in PG5.

- **HW5 file**
  Is generated by PG5 and contains the channel and the station of a CPU.

- **Porting list**
  The import is based on a so-called porting list. This is a text file that can be added to or

modified by the SI. The porting list is provided with the setup in Version 1.5 or higher.

An entry in the list is structured as follows:

_AnyMacroName; Line; BMO

Example:
_SHOWTIME2;3;vloTimeWhere (FBoxes of the DDC suite family)

The meaning of the entries is as follows:

1. __AnyMacroName:
Unique macro name (=FBox)

2. Line:
Provides information about the source of the resources. A number is always entered for FBoxes of the DDC suite family (in this case 3). The number indicates indirectly that the resources are to be taken from the file _Global.sy5. The entry always begins with SYSTEM for FBoxes of the HVAC family. This indicates that the resources are to be taken from the file _System.sy5.

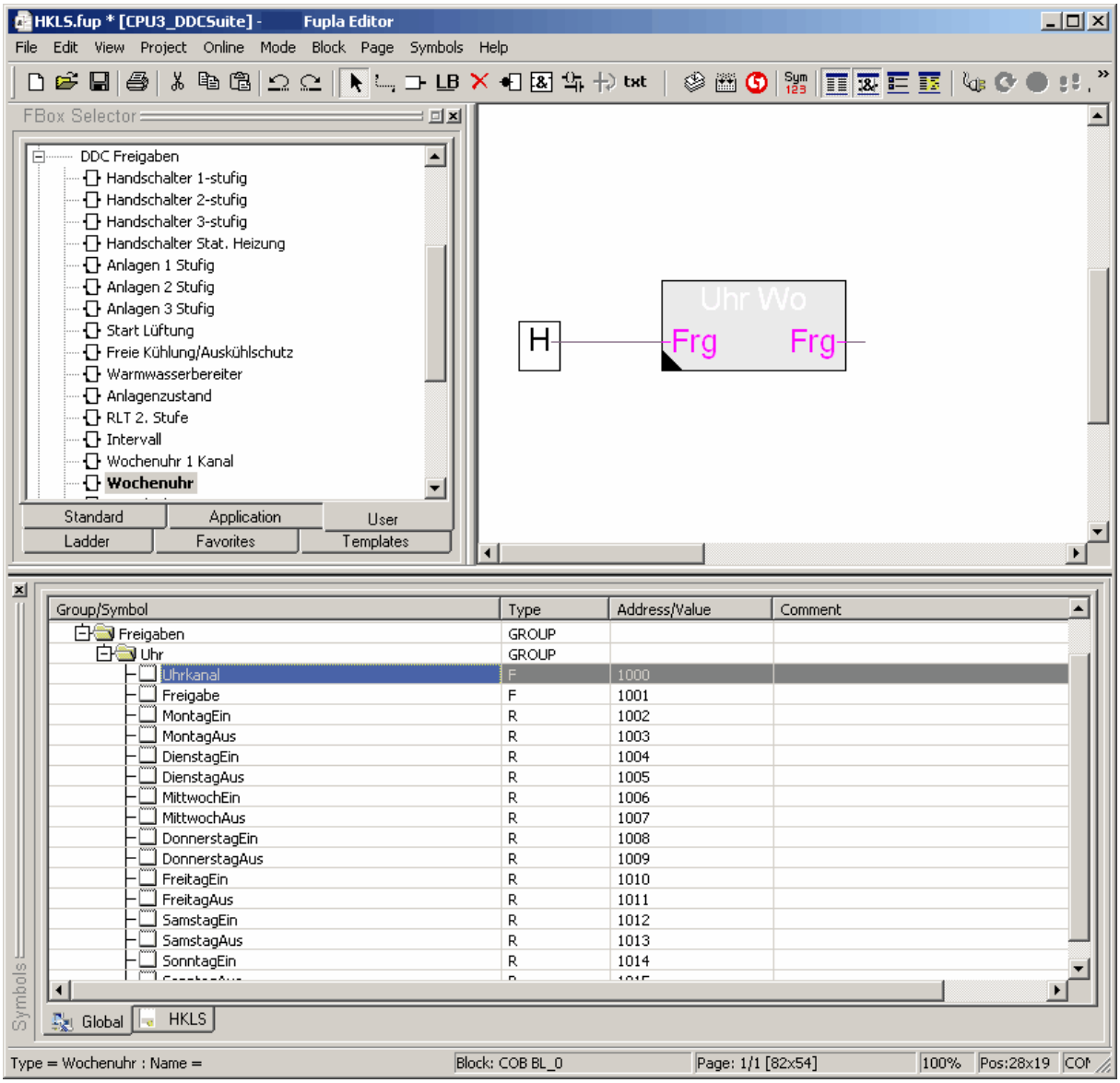
3. BMO:
Indicates the BMO object from which the system object for the FBox should be derived.

The porting list should be copied to the directory <proj>\cfg in the project. The path for the file must be indicated in the file <proj>/cfg/pg5.cfg in the [Import] section, e.g.:
[Import]
File = C:\Visi.Plus\proj\Import\cfg\BoxVLO.txt

In order to be able to import resources of the Fboxes from PG5, the indicated BMO objects must be installed in the porting list. If this is not the case, ***DIG01*** *and* ***ANA01*** *are taken as a template object.*

Example 1: Importing an FBox of the "DDC" derivation

The **FBox "week clock"** from the **"Releases" DDC suite family** should be used as an example here. After inserting the FBox, the internal resources are created automatically with predefined group and symbol names. These are available individually in the Symbol Editor in the global table.



The groups can be renamed and moved. The symbol names may not be changed, because they are used in Visi.Plus (SWeb templates, BMO objects).

The resources normally do not have fixed addressing, meaning that the addresses can be available, but do not have to.

The entry for this FBox in the porting list is:

_SHOWTIME2;3;vIoTimeWhere

The following now happens when importing:

1. The resource name is searched for in the **FDB file** in the 4th line of **Macros _SHOWCLOCK2**:

   Excerpt from the FDB file:

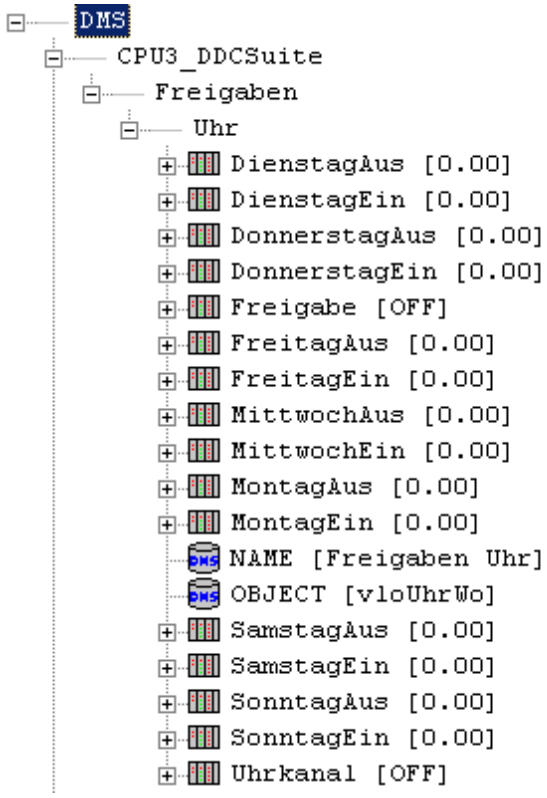
```
_SHOWCLOCK2(     10304,                    ;;Version Indicator.
             ___FlgDyn_C0001+8,            ;;Input #1 (Frg)
             ___FlgDyn_C0001+9,            ;;Output #1 (Frg)
```

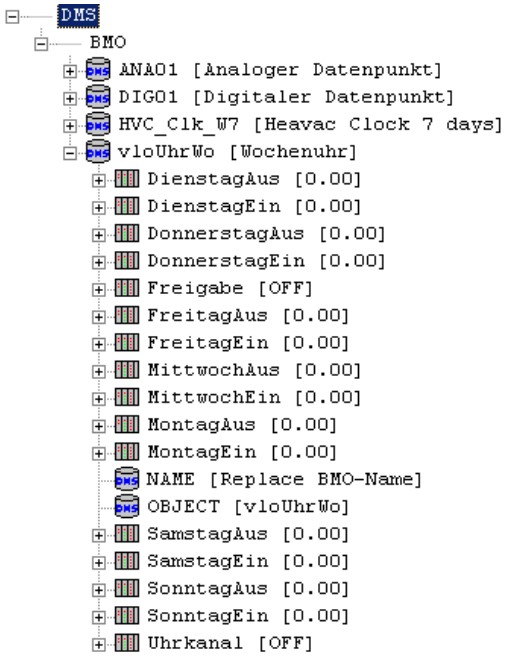
```
Releases.Clock.Clockchannel,
Releases.Clock.Release,
Releases.Clock.MondayOn,
Releases.Clock.MondayOff,
Releases.Clock.TuesdayOn,
Releases.Clock.TuesdayOff,
Releases.Clock.WednesdayOn,
Releases.Clock.WednesdayOff,
Releases.Clock.ThursdayOn,
Releases.Clock.ThursdayOff,
Releases.Clock.FridayOn,
Releases.Clock.FridayOff,
Releases.Clock.SaturdayOn,
Releases.Clock.SaturdayOff,
Releases.Clock.SundayOn,
Releases.Clock.SundayOff,
800,                    ;;Init variable (Monday On).
1900,                   ;;Init variable (   ... Off).
800,                    ;;Init variable (Tuesday On).
1900,                   ;;Init variable (   ... Off).
800,                    ;;Init variable (Wednesday On).
1900,                   ;;Init variable (   ... Off).
800,                    ;;Init variable (Thursday On).
1900,                   ;;Init variable (   ... Off).
800,                    ;;Init variable (Friday On).
1900,                   ;;Init variable (   ... Off).
0,                    ;;Init variable (Saturday On).
0,                    ;;Init variable (   ... Off).
0,                    ;;Init variable (Sunday On).
0)                      ;;Init variable (   ... Off).
```

In this case it is called Release.Clock. The DMS name for the new system object is generated from this, whereby the period is replaced by a colon and the CPU name is placed in front of it. Therefore the DMS name is: CPU3_DDCSuite:Releases:Clock.

2. A new system object is created with the DMS name **CPU3_DDCSuite:Releases:Clock**. This object is derived from the BMO object **vIoClockWhere**.

DMS View **BMO View**





3. The addresses are read from the file **...pcd\<PLCname>_Global.sy5** and inserted into the DMS.

Excerpt from the file* _Global.sy5*:

```
$GROUP    Releases                    ;
$GROUP    Clock                ;
ClockChannel      EQU   F      1000
PUBL      ClockChannel
Release   EQU   F      1001
PUBL      Release
MondayOn  EQU   R      1002
PUBL      MondayOn
MondayOff EQU   R      1003
PUBL      MondayOff
TuesdayOn EQU   R      1004
PUBL      TuesdayOn
TuesdayOff EQU  R      1005
PUBL      TuesdayOff
WednesdayOff      EQU   R      1006
PUBL      WednesdayOff
WednesdayOff      EQU   R      1007
PUBL      WednesdayOff
ThursdayOn        EQU   R      1008
PUBL      ThursdayOn
ThursdayOff       EQU   R      1009
PUBL      ThursdayOff
FridayOn  EQU   R      1010
PUBL      FridayOn
FridayOff EQU   R      1011
```

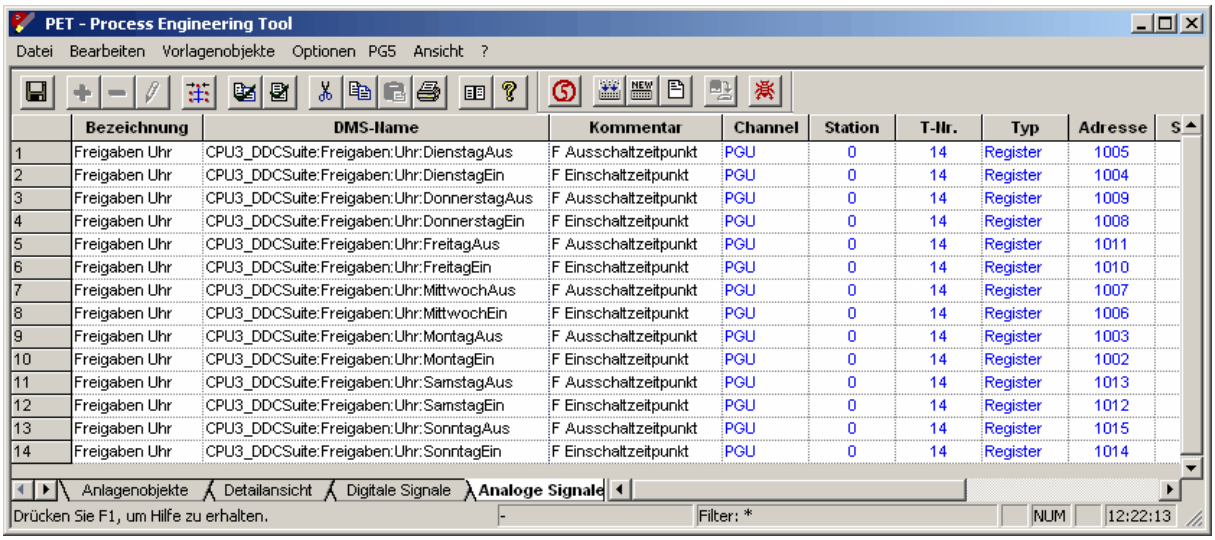
```
PUBL        FridayOff
SaturdayOn  EQU   R       1012
PUBL        SaturdayOn
SaturdayOff EQU   R       1013
PUBL        SaturdayOff
SundayOn    EQU   R       1014
PUBL        SundayOn
SundayOff   EQU   R       1015
PUBL        SundayOff
$ENDGROUP
$ENDGROUP
```

4. The channel and the station are taken from the file **...pcd\<PLCname>\<PLCname>.hw5** and inserted into the DMS.

   Excerpt from the HW5 file:

```
[GoOnline]
Version=1
Channel=PGU
Cpu=0
Station=0
```

After executing the command "**Import PG5 Symbols**" in the PET, the following view appears (Analogue Signals tab):



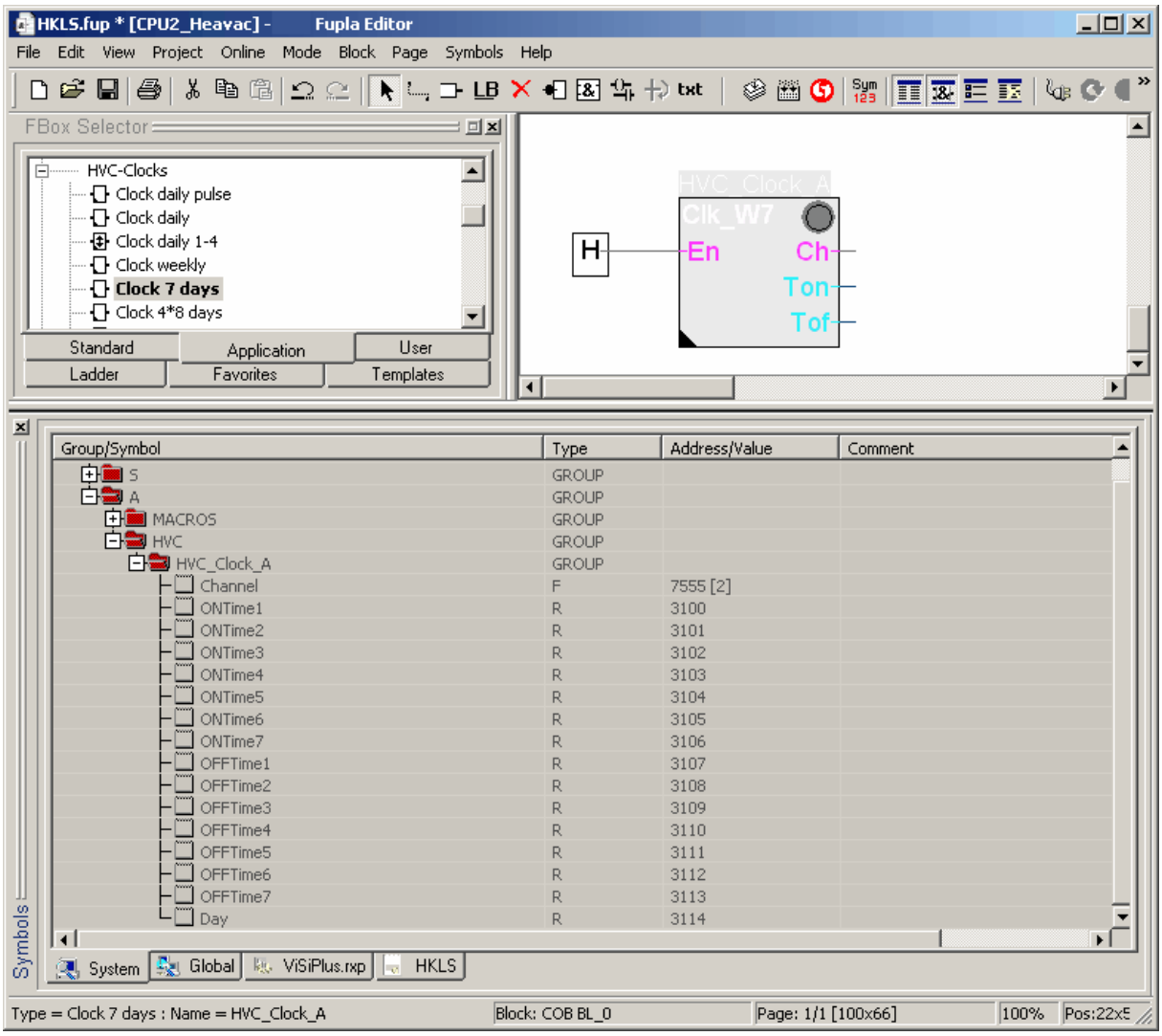
	Bezeichnung	DMS-Name	Kommentar	Channel	Station	T-Nr.	Typ	Adresse
1	Freigaben Uhr	CPU3_DDCSuite:Freigaben:Uhr:DienstagAus	F Ausschaltzeitpunkt	PGU	0	14	Register	1005
2	Freigaben Uhr	CPU3_DDCSuite:Freigaben:Uhr:DienstagEin	F Einschaltzeitpunkt	PGU	0	14	Register	1004
3	Freigaben Uhr	CPU3_DDCSuite:Freigaben:Uhr:DonnerstagAus	F Ausschaltzeitpunkt	PGU	0	14	Register	1009
4	Freigaben Uhr	CPU3_DDCSuite:Freigaben:Uhr:DonnerstagEin	F Einschaltzeitpunkt	PGU	0	14	Register	1008
5	Freigaben Uhr	CPU3_DDCSuite:Freigaben:Uhr:FreitagAus	F Ausschaltzeitpunkt	PGU	0	14	Register	1011
6	Freigaben Uhr	CPU3_DDCSuite:Freigaben:Uhr:FreitagEin	F Einschaltzeitpunkt	PGU	0	14	Register	1010
7	Freigaben Uhr	CPU3_DDCSuite:Freigaben:Uhr:MittwochAus	F Ausschaltzeitpunkt	PGU	0	14	Register	1007
8	Freigaben Uhr	CPU3_DDCSuite:Freigaben:Uhr:MittwochEin	F Einschaltzeitpunkt	PGU	0	14	Register	1006
9	Freigaben Uhr	CPU3_DDCSuite:Freigaben:Uhr:MontagAus	F Ausschaltzeitpunkt	PGU	0	14	Register	1003
10	Freigaben Uhr	CPU3_DDCSuite:Freigaben:Uhr:MontagEin	F Einschaltzeitpunkt	PGU	0	14	Register	1002
11	Freigaben Uhr	CPU3_DDCSuite:Freigaben:Uhr:SamstagAus	F Ausschaltzeitpunkt	PGU	0	14	Register	1013
12	Freigaben Uhr	CPU3_DDCSuite:Freigaben:Uhr:SamstagEin	F Einschaltzeitpunkt	PGU	0	14	Register	1012
13	Freigaben Uhr	CPU3_DDCSuite:Freigaben:Uhr:SonntagAus	F Ausschaltzeitpunkt	PGU	0	14	Register	1015
14	Freigaben Uhr	CPU3_DDCSuite:Freigaben:Uhr:SonntagEin	F Einschaltzeitpunkt	PGU	0	14	Register	1014

The addresses can be edited and exported. Resources from FBoxes of the DDC suite family in the DMS are identified by the data point PLC:PG5Export = On.

Example 2: Importing an FBox of the "HVAC" derivation

The **FBox "Clock 7 days"** from the **HVAC family HVC clocks** is used as an example here. After inserting the FBox, the internal resources are created automatically with predefined group and symbol names. These are available individually in the Symbol Editor in the global

table.



The groups can be renamed and moved. The symbol names may not be changed, because they are used in Visi.Plus (SWeb templates, BMO objects).

The resources normally have fixed addressing. This can be defined from the SI. However, these are frequently addresses assigned by the compiler which can be changed after compilation.

The entry for this FBox in the porting list is:

_HEACLKW7B;SYSTEM.A.HVC;HVC_Clk_W7

The following now happens when importing:

5. The DMS name is searched for in the last line of the **macro _HEACLKW7B** in the **FDB file**:

Excerpt from the FDB file:

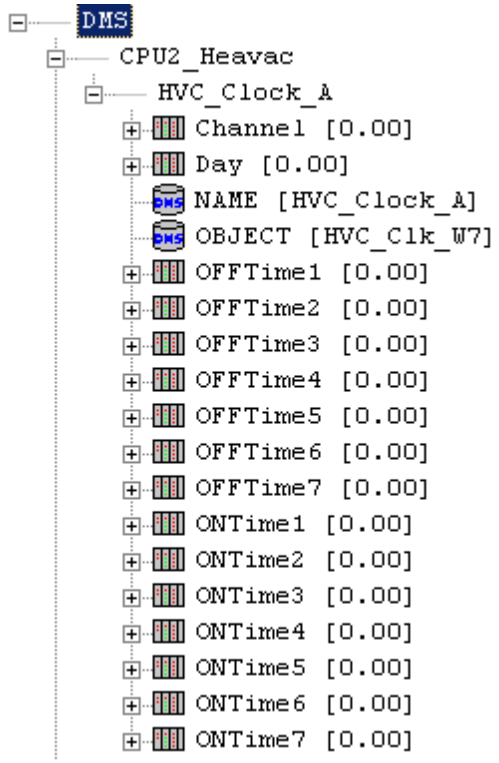
```
_HEACLKW7B(    5,                    ;;Version Indicator.
          ___FlgDyn_C0001+3,         ;;Input #1 (En)
```

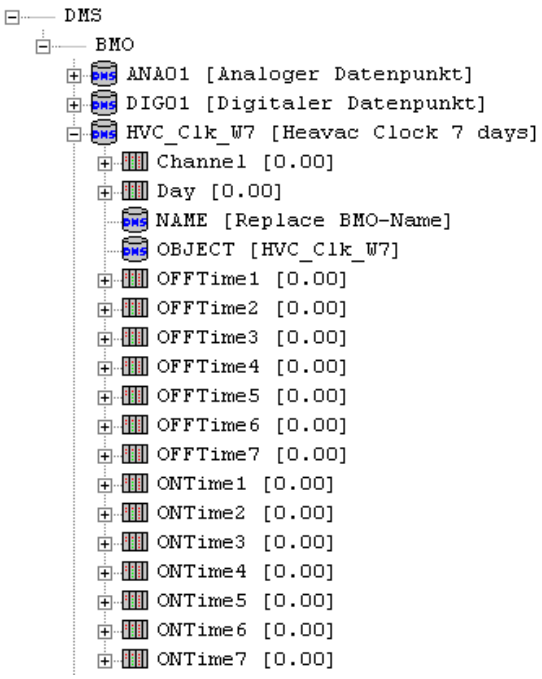
```
___FlgDyn_C0001+4,              ;;Output #1 (Ch)
___RegDyn_C0001+0,               ;;Output #2 (Ton)
___RegDyn_C0001+1,               ;;Output #3 (Tof)
___RegDyn_C0001+2,               ;;Dynamic variable Rd[4].
__stc_id_0000C_var_00,           ;;Static variable Fs[2].
HVC_Clock_UserDef_mit,
__stc_id_0000C_var_02,           ;;Static variable LED[1].
0,                          ;;Init variable (Objects for HMI Editor ).
1200,                         ;;Init variable (Switch On Monday).
1200,                         ;;Init variable (Switch Off Monday).
1200,                         ;;Init variable (Switch On Tuesday).
1200,                         ;;Init variable (Switch Off Tuesday).
1200,                         ;;Init variable (Switch On Wednesday).
1200,                         ;;Init variable (Switch Off Wednesday).
1200,                         ;;Init variable (Switch On Thursday).
1200,                         ;;Init variable (Switch Off Thursday).
1200,                         ;;Init variable (Switch On Friday).
1200,                         ;;Init variable (Switch Off Friday).
1200,                         ;;Init variable (Switch On Saturday).
1200,                         ;;Init variable (Switch Off Saturday).
1200,                         ;;Init variable (Switch On Sunday).
1200,                         ;;Init variable (Switch Off Sunday).
HVC_Clock_A)                      ;;FBox user name.
```

In this case it is called HVC_Clock_A. The DMS name for the new system object is generated from this, whereby the CPU name is placed in front of it. Therefore the DMS name is: CPU2_Heavac:HVC_Clock_A.

6. A new system object is created with the DMS name **CPU2_Heavac:HVC_Clock_A**. This object is derived from the BMO object **HVC_Clk_W7**.

DMS View **BMO View**





7. The addresses are read from the file **...pcd\<PLCname>_Global.sy5** and inserted into the DMS.

   Excerpt from the file _Global.sy5:

```
A.MACROS.HVC_Clock_A EQU  12 ;_HEACLKW7B
A.HVC.HVC_Clock_A.Channel EQU F 7555[2] ;
A.HVC.HVC_Clock_A.ONTime1 EQU R 3100 ;
A.HVC.HVC_Clock_A.ONTime2 EQU R 3101 ;
A.HVC.HVC_Clock_A.ONTime3 EQU R 3102 ;
A.HVC.HVC_Clock_A.ONTime4 EQU R 3103 ;
A.HVC.HVC_Clock_A.ONTime5 EQU R 3104 ;
A.HVC.HVC_Clock_A.ONTime6 EQU R 3105 ;
A.HVC.HVC_Clock_A.ONTime7 EQU R 3106 ;
A.HVC.HVC_Clock_A.OFFTime1 EQU R 3107 ;
A.HVC.HVC_Clock_A.OFFTime2 EQU R 3108 ;
A.HVC.HVC_Clock_A.OFFTime3 EQU R 3109 ;
A.HVC.HVC_Clock_A.OFFTime4 EQU R 3110 ;
A.HVC.HVC_Clock_A.OFFTime5 EQU R 3111 ;
A.HVC.HVC_Clock_A.OFFTime6 EQU R 3112 ;
A.HVC.HVC_Clock_A.OFFTime7 EQU R 3113 ;
A.HVC.HVC_Clock_A.Day EQU R 3114 ;
```

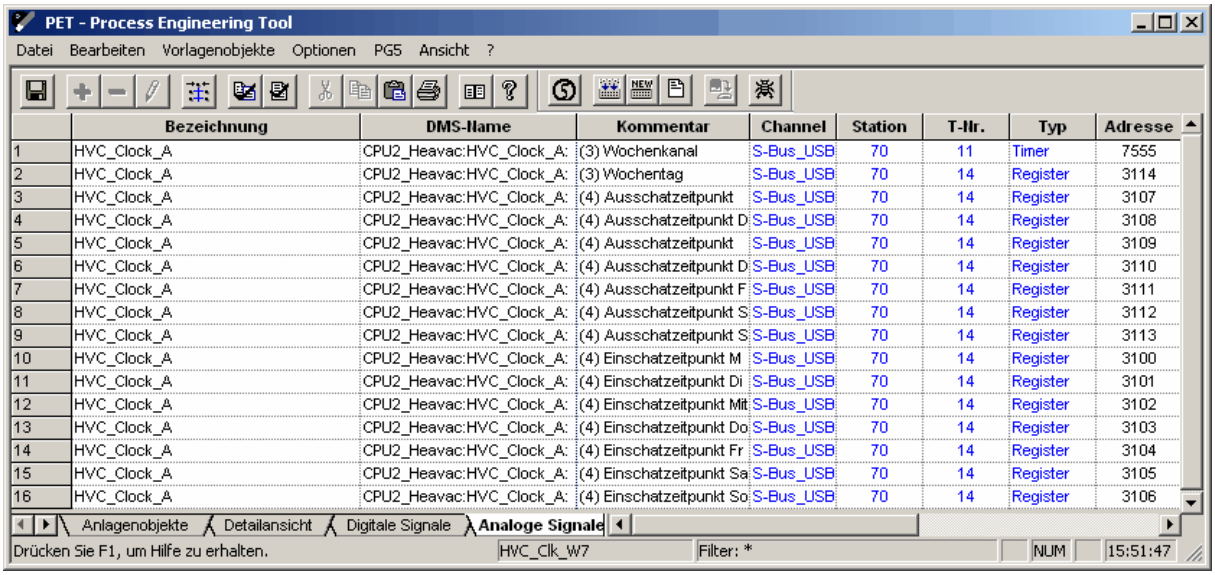
8. The channel and the station are taken from the file **...pcd\<PLCname>\<PLCname>.hw5** and inserted into the DMS.

   Excerpt from the HW5 file:

```
[GoOnline]
Version=1
Channel=S-Bus_USB
```

Cpu=0
Station=70

After executing the command "**Import PG5 Symbols**" in the PET, the following view appears (Analogue Signals tab):

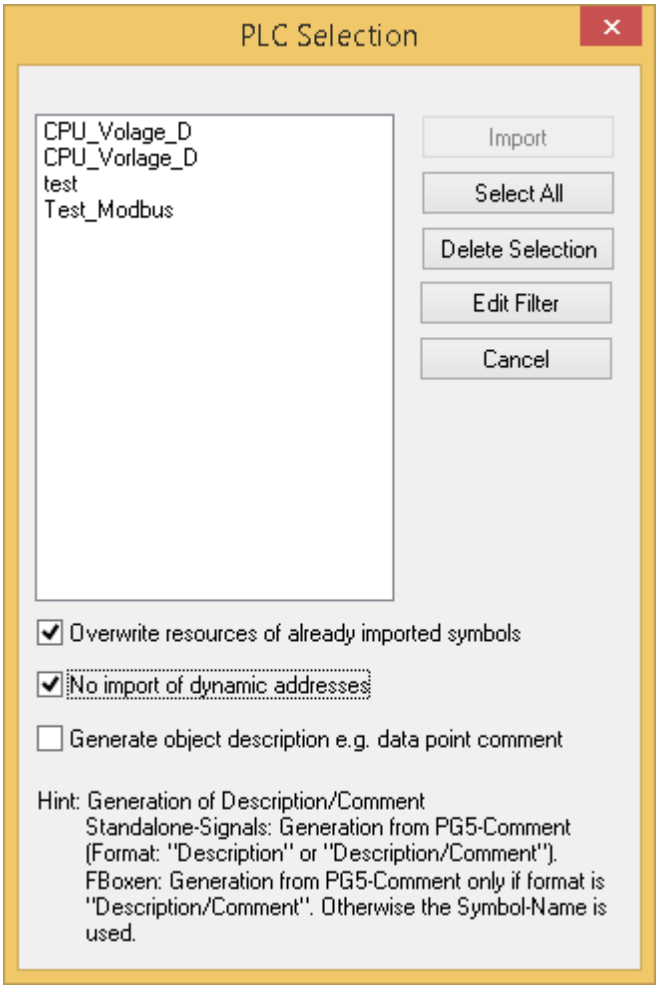


	Bezeichnung	DMS-Name	Kommentar	Channel	Station	T-Nr.	Typ	Adresse
1	HVC_Clock_A	CPU2_Heavac:HVC_Clock_A:	(3) Wochenkanal	S-Bus_USB	70	11	Timer	7555
2	HVC_Clock_A	CPU2_Heavac:HVC_Clock_A:	(3) Wochentag	S-Bus_USB	70	14	Register	3114
3	HVC_Clock_A	CPU2_Heavac:HVC_Clock_A:	(4) Ausschatzeitpunkt	S-Bus_USB	70	14	Register	3107
4	HVC_Clock_A	CPU2_Heavac:HVC_Clock_A:	(4) Ausschatzeitpunkt D	S-Bus_USB	70	14	Register	3108
5	HVC_Clock_A	CPU2_Heavac:HVC_Clock_A:	(4) Ausschatzeitpunkt	S-Bus_USB	70	14	Register	3109
6	HVC_Clock_A	CPU2_Heavac:HVC_Clock_A:	(4) Ausschatzeitpunkt D	S-Bus_USB	70	14	Register	3110
7	HVC_Clock_A	CPU2_Heavac:HVC_Clock_A:	(4) Ausschatzeitpunkt F	S-Bus_USB	70	14	Register	3111
8	HVC_Clock_A	CPU2_Heavac:HVC_Clock_A:	(4) Ausschatzeitpunkt S	S-Bus_USB	70	14	Register	3112
9	HVC_Clock_A	CPU2_Heavac:HVC_Clock_A:	(4) Ausschatzeitpunkt S	S-Bus_USB	70	14	Register	3113
10	HVC_Clock_A	CPU2_Heavac:HVC_Clock_A:	(4) Einschatzeitpunkt M	S-Bus_USB	70	14	Register	3100
11	HVC_Clock_A	CPU2_Heavac:HVC_Clock_A:	(4) Einschatzeitpunkt Di	S-Bus_USB	70	14	Register	3101
12	HVC_Clock_A	CPU2_Heavac:HVC_Clock_A:	(4) Einschatzeitpunkt Mit	S-Bus_USB	70	14	Register	3102
13	HVC_Clock_A	CPU2_Heavac:HVC_Clock_A:	(4) Einschatzeitpunkt Do	S-Bus_USB	70	14	Register	3103
14	HVC_Clock_A	CPU2_Heavac:HVC_Clock_A:	(4) Einschatzeitpunkt Fr	S-Bus_USB	70	14	Register	3104
15	HVC_Clock_A	CPU2_Heavac:HVC_Clock_A:	(4) Einschatzeitpunkt Sa	S-Bus_USB	70	14	Register	3105
16	HVC_Clock_A	CPU2_Heavac:HVC_Clock_A:	(4) Einschatzeitpunkt So	S-Bus_USB	70	14	Register	3106

The addresses cannot be edited (therefore they are marked black). The resources from FBoxes of the HVAC families are normally found in arrays. If readdressing is permitted, this can lead to address overlapping. These resources cannot be exported either, because this is automatically managed by PG5. Resources from FBoxes of the HVAC family in the DMS are identified by the data point PLC:PG5Export = Off.

If resources not listed in the porting list are found in the files _Global.sy5 and _System.sy5, they are derived from the BMO objects ***DIG01*** *and* ***ANA01.***

Selection List
If multiple PLCs are present in the project, a selection can be made prior to import:



The list contains the PLCs generated in PG5. No PLC is selected by default. A PLC can be selected by clicking on the corresponding PLC entry in the list (it is then highlighted in blue). Clicking on it again removes the PLC from to the selection.

<Import>
Starts PG5 symbol import.

<Select All>
Selects all PLCs in the list (highlighted in blue).

<Deselect All>
Deselects the selected PLCs. The highlighting is removed from all entries in the list.

<Cancel>
The dialog is closed.

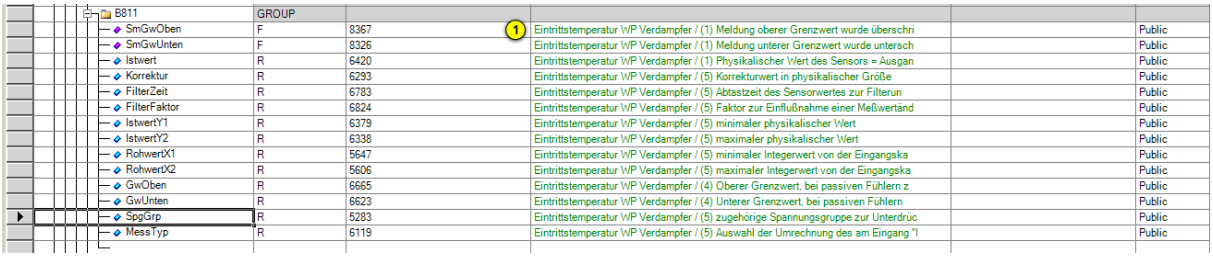
<Edit Filter>
Open the Menue for filter settings for single symbol and FBox.

<Overwrite resources of already imported symbols>
If this checkbox is activated all Telegram number, Station number, Channel name and Address will be overwritten.

<No import of dynamic addresses>

If this setting is activated no Media (ANA01, DIGI01) with dynamic addresses in PG5 are imported.

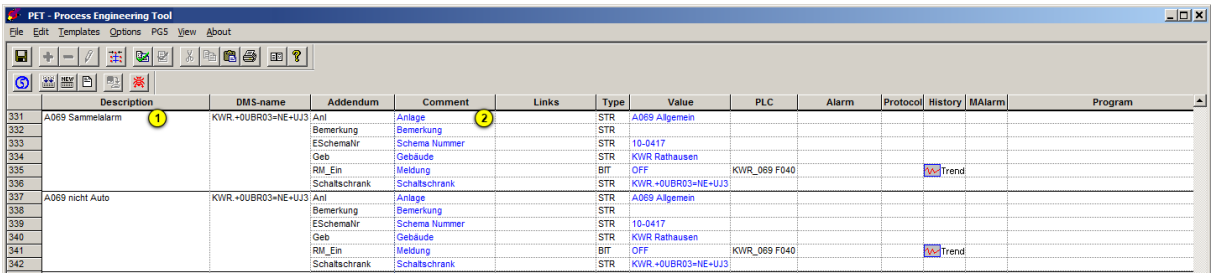
<Generate object description from comment>
With this checkbox, the name is generated from the comment. In PG5, comments must have the following format to enable each comment to be transferred correctly:



B811	GROUP				
SmGwOben	F	8367	1 Eintrittstemperatur WP Verdampfer / (1) Meldung oberer Grenzwert wurde überschri		Public
SmGwUnten	F	8326	Eintrittstemperatur WP Verdampfer / (1) Meldung unterer Grenzwert wurde untersch		Public
Istwert	R	6420	Eintrittstemperatur WP Verdampfer / (1) Physikalischer Wert des Sensors = Ausgan		Public
Korrektur	R	6293	Eintrittstemperatur WP Verdampfer / (5) Korrekturwert in physikalischer Größe		Public
FilterZeit	R	6783	Eintrittstemperatur WP Verdampfer / (5) Abtastzeit des Sensorwertes zur Filterun		Public
FilterFaktor	R	6824	Eintrittstemperatur WP Verdampfer / (5) Faktor zur Einflußnahme einer Meßwertänd		Public
IstwertY1	R	6379	Eintrittstemperatur WP Verdampfer / (5) minimaler physikalischer Wert		Public
IstwertY2	R	6338	Eintrittstemperatur WP Verdampfer / (5) maximaler physikalischer Wert		Public
RohwertX1	R	5647	Eintrittstemperatur WP Verdampfer / (5) minimaler Integerwert von der Eingangska		Public
RohwertX2	R	5606	Eintrittstemperatur WP Verdampfer / (5) maximaler Integerwert von der Eingangska		Public
GwOben	R	6665	Eintrittstemperatur WP Verdampfer / (4) Oberer Grenzwert, bei passiven Fühlern z		Public
GwUnten	R	6623	Eintrittstemperatur WP Verdampfer / (4) Unterer Grenzwert, bei passiven Fühlern		Public
SpgGrp	R	5283	Eintrittstemperatur WP Verdampfer / (5) zugehörige Spannungsgruppe zur Unterdrüc		Public
MessTyp	R	6119	Eintrittstemperatur WP Verdampfer / (5) Auswahl der Umrechnung des am Eingang "I		Public

A forward slash is used to divide the comment. The first part will then be inserted as the object name. The second part will remain as a comment with the add-on.

In PET, the import looks like this:



PET - Process Engineering Tool

File Edit Templates Options PG5 View About

	Description	DMS-name	Addendum	Comment	Links	Type	Value	PLC	Alarm	Protocol	History	MAlarm	Program
331	A069 Sammelalarm 1	KWR.+0UBR03=NE+UJ3	Anl	Anlage 2		STR	A069 Allgemein						
332			Bemerkung	Bemerkung		STR							
333			ESchemaNr	Schema Nummer		STR	10-0417						
334			Geb	Gebäude		STR	KWR Rathausen						
335			RM_Ein	Meldung		BIT	OFF	KWR_069 F040			Trend		
336			Schaltschrank	Schaltschrank		STR	KWR.+0UBR03=NE+UJ3						
337	A069 nicht Auto	KWR.+0UBR03=NE+UJ3	Anl	Anlage		STR	A069 Allgemein						
338			Bemerkung	Bemerkung		STR							
339			ESchemaNr	Schema Nummer		STR	10-0417						
340			Geb	Gebäude		STR	KWR Rathausen						
341			RM_Ein	Meldung		BIT	OFF	KWR_069 F040			Trend		
342			Schaltschrank	Schaltschrank		STR	KWR.+0UBR03=NE+UJ3						

1 Object name

2 Comment of object add-on

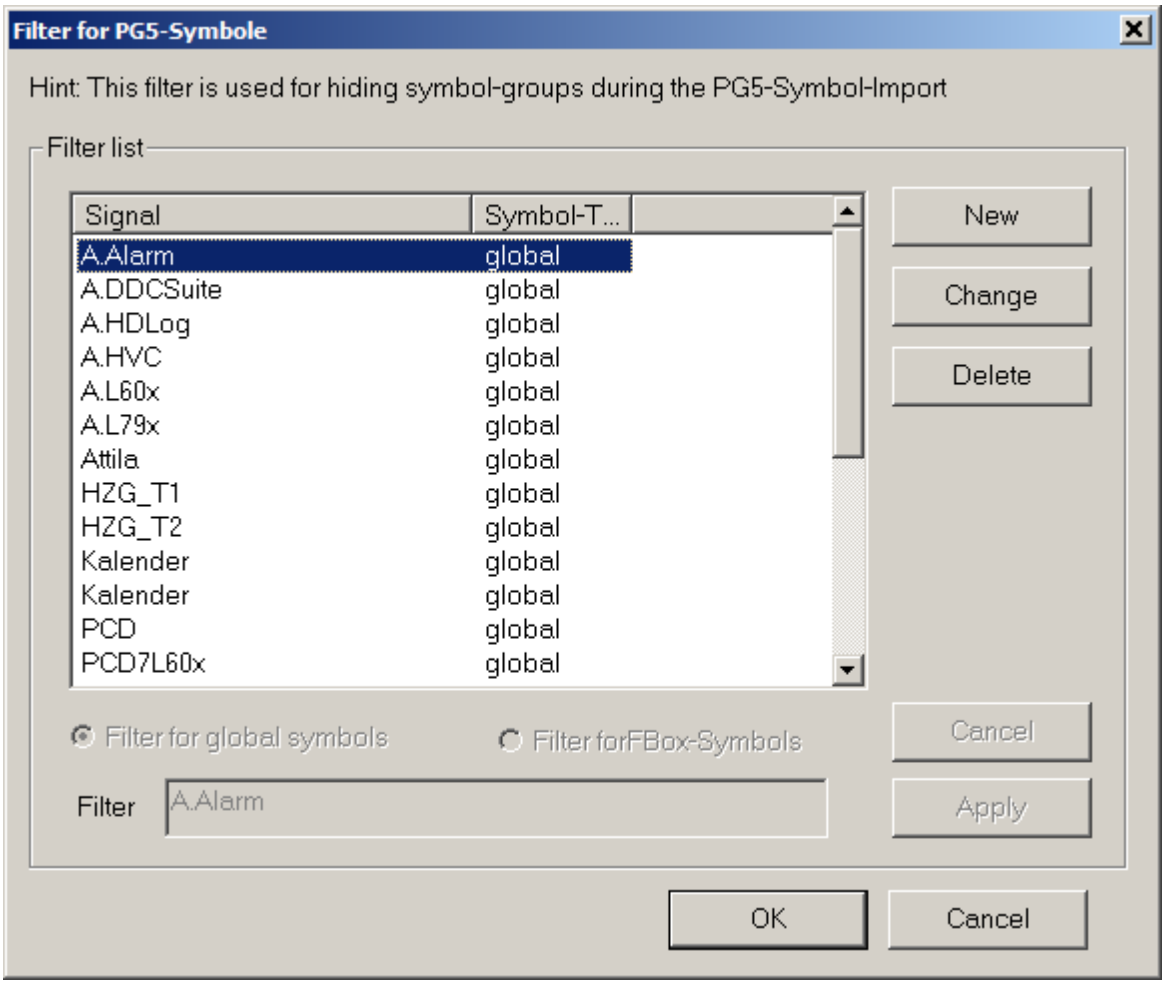
Names can also be exported in this format as an rxp file. However, to do this the PET file SDriver.pet must be modified as follows:

```
[Export2]
Comment = ;
DMSChar = .
Space = .
Dot = .
Slash = .
Plus = .
Minus = .
SChar =
FileExt=.rxp
Select=Saia PG5® Fupla-Resource file (*.rxp)|*.rxp|All Files (*.*)|*.*||
Title=Symbol;Media;Address;Comment := Value ;Scope
1=@DMSName
2=;
3=@Type[0]
4=;
5=@Address
6=:=
7=@PlcValue
```

```
8=;
9=@Name
10=" / "
11=@Comment
12=;
13=1
14=\n
```

Detailed information on the PET file is provided in the chapter: PET file.

Edit Filter



In this Window the Import filters for single media (ANA01, DIGI01) and FBox symbol groups can be defined.

If the checkbox ***Filter for single symbols*** is active you can define symbol-group names which will not be imported as ANA01 or/and DIG01.
If the checkbox ***Filter for FBox-symbols*** is active you can define symbol-group names which will not be imported as FBox-Objects.

<New>
Create a new filter.

<Change>

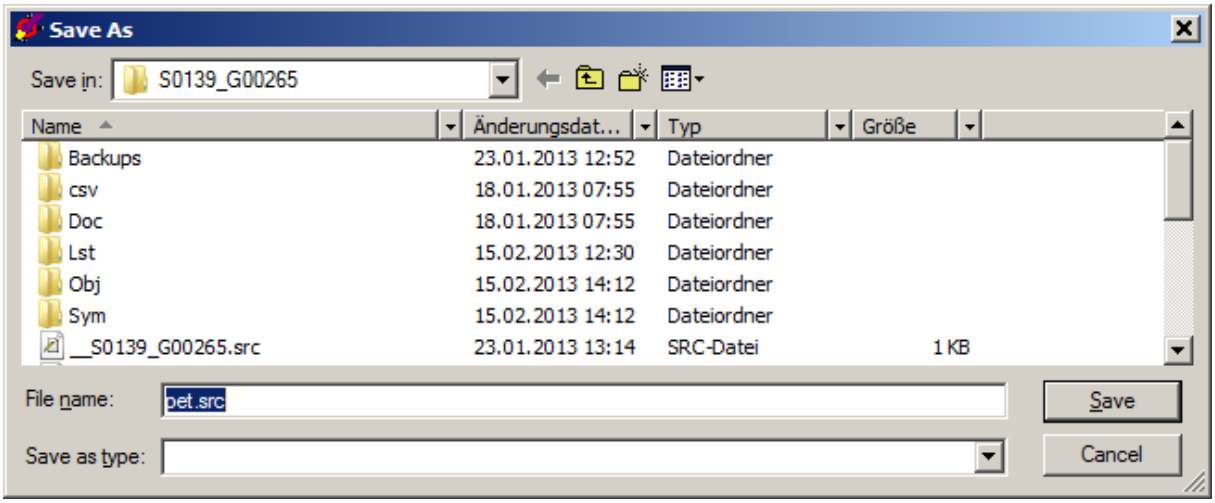
change a selected filter.

<Delete>
Delete a selected filter.

Automatic update
The PET has automatic updating of PG5 resources, meaning that the PET is automatically updated if the resources in PG5 change (for this purpose, compile Fupla file in PG5). This function is activated when the value 'Auto' in the [Update] section in the file <proj>/cfg/pg5.cfg is set to '1'.

6.11.5.5 Generate Resource List for AWL/IL (.src)

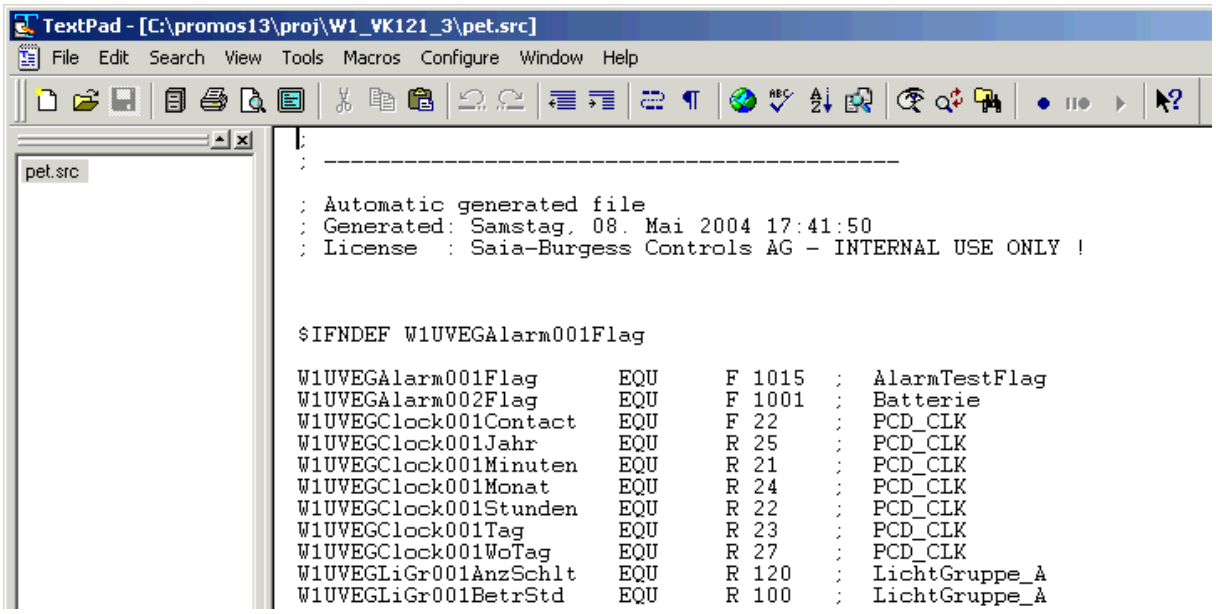
Generate Resource List for AWL/IL creates an "**src**" file (Saia PG5® AWL source file) with all **symbols**, **media codes**, **element addresses** and **comments/descriptions**.



It is best to save the file in the root directory (in this case in the directory "c:\Visi.Plus\proj \promos") of the current project.

The file can then be linked with the rest of the project, in other words with other program files ("src" files).

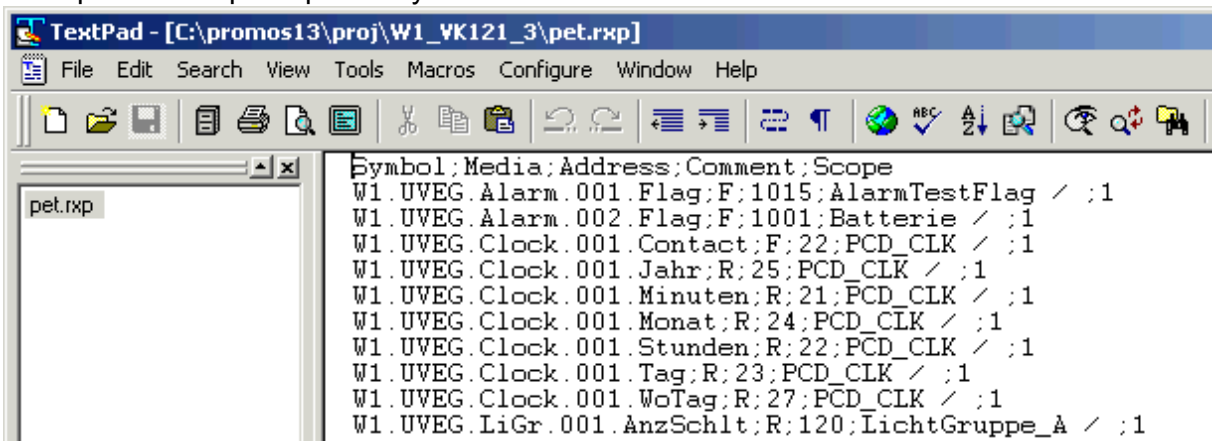
Example of an AWL exported symbol table:



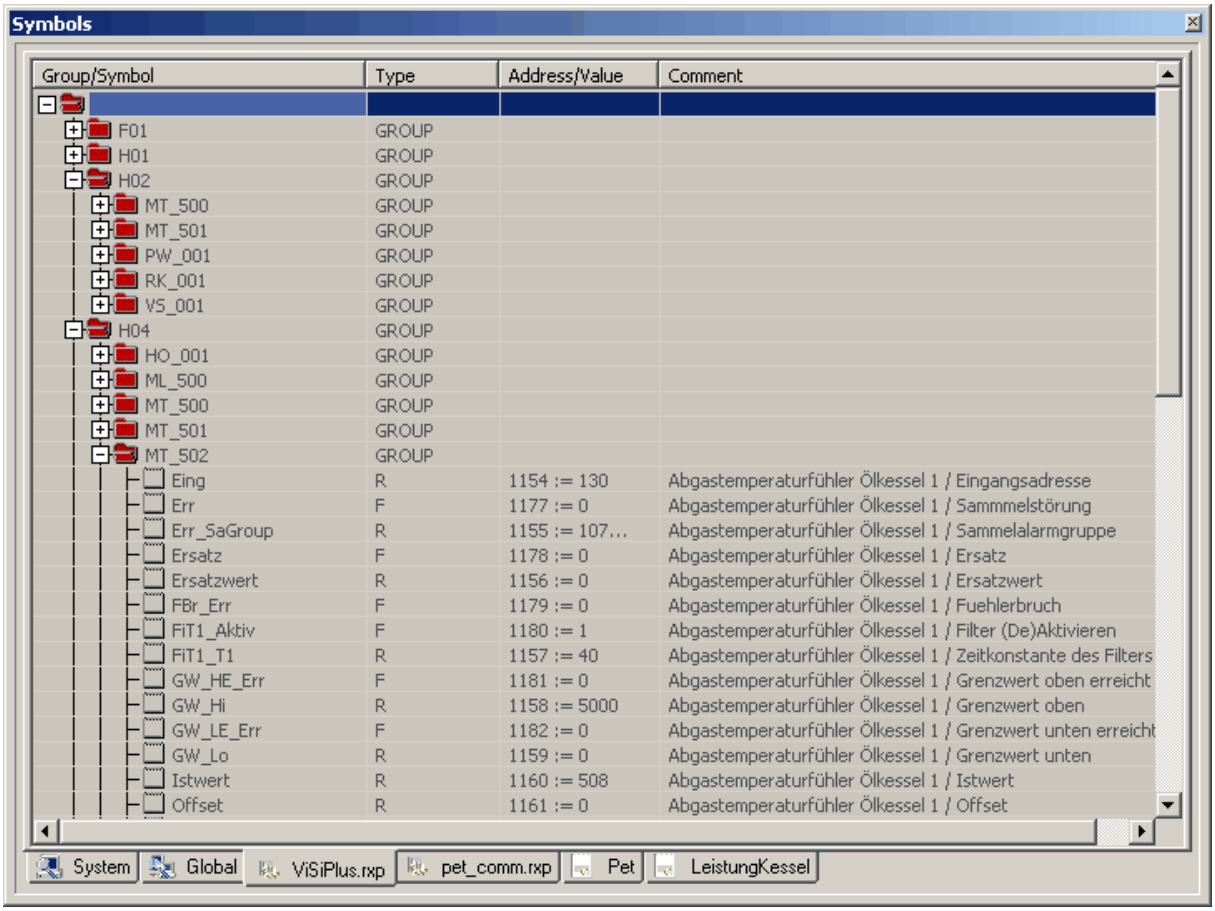
6.11.5.6 Generate Resource List for FUPLA (.rxp)

With "**Generate Resource List for Fupla**" a symbol list for a Saia PG5® Fupla program is created in a file with the extension "**rxp**" (export resources). This file can then be imported with PG5 into the desired Fupla program.

Example of an Fupla exported symbol table:



These resources can be used directly in PG5.

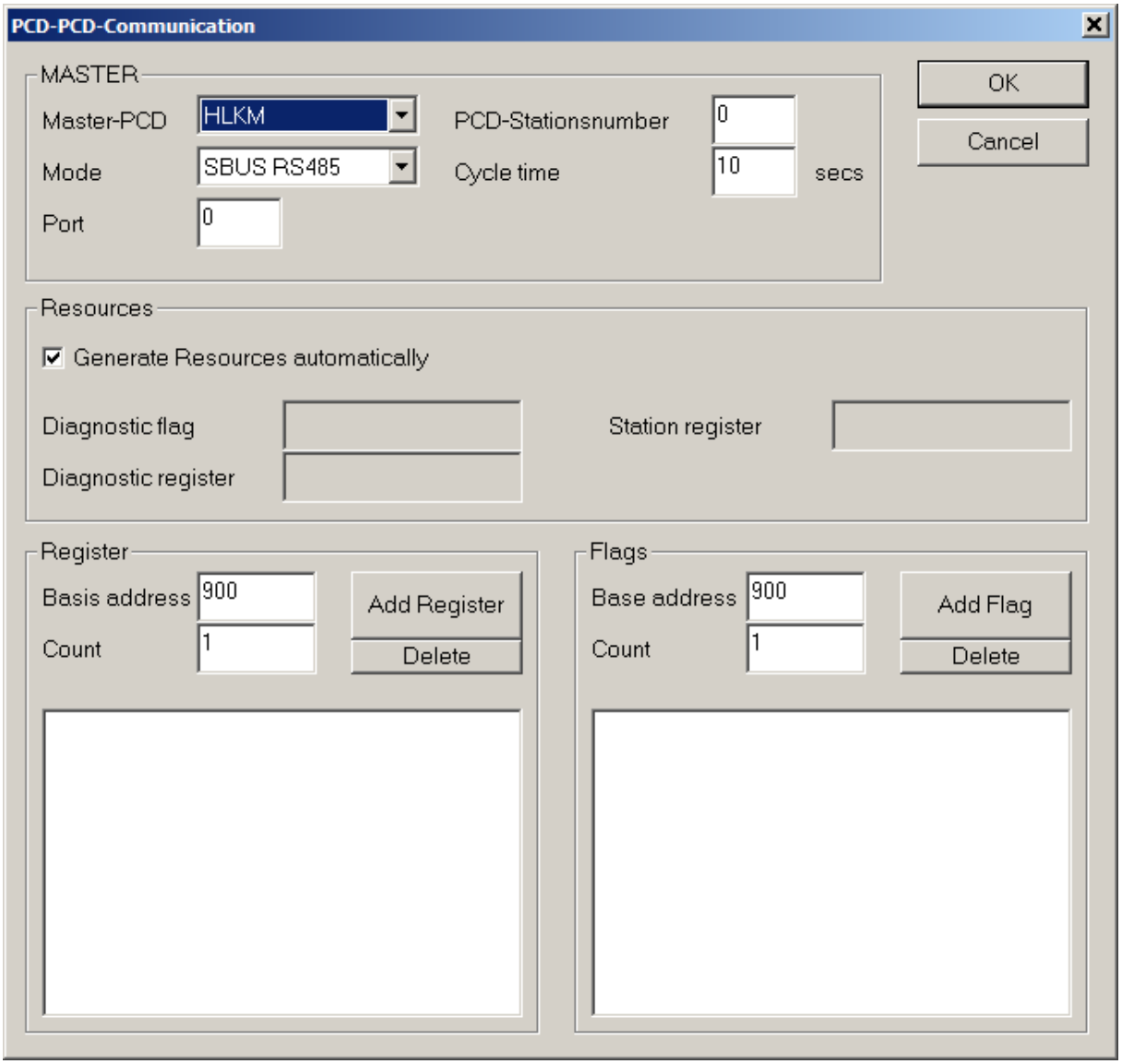


Resources from Fboxes with the "HVAC" derivation are not exported, because they are automatically managed by PG5.

6.11.5.7 PCD/PCD Communication

In larger systems, data from one PLC is used on additional PLCs (for example, the outside temperature is read on one PLC and the control cycles are stored on various other PLCs).

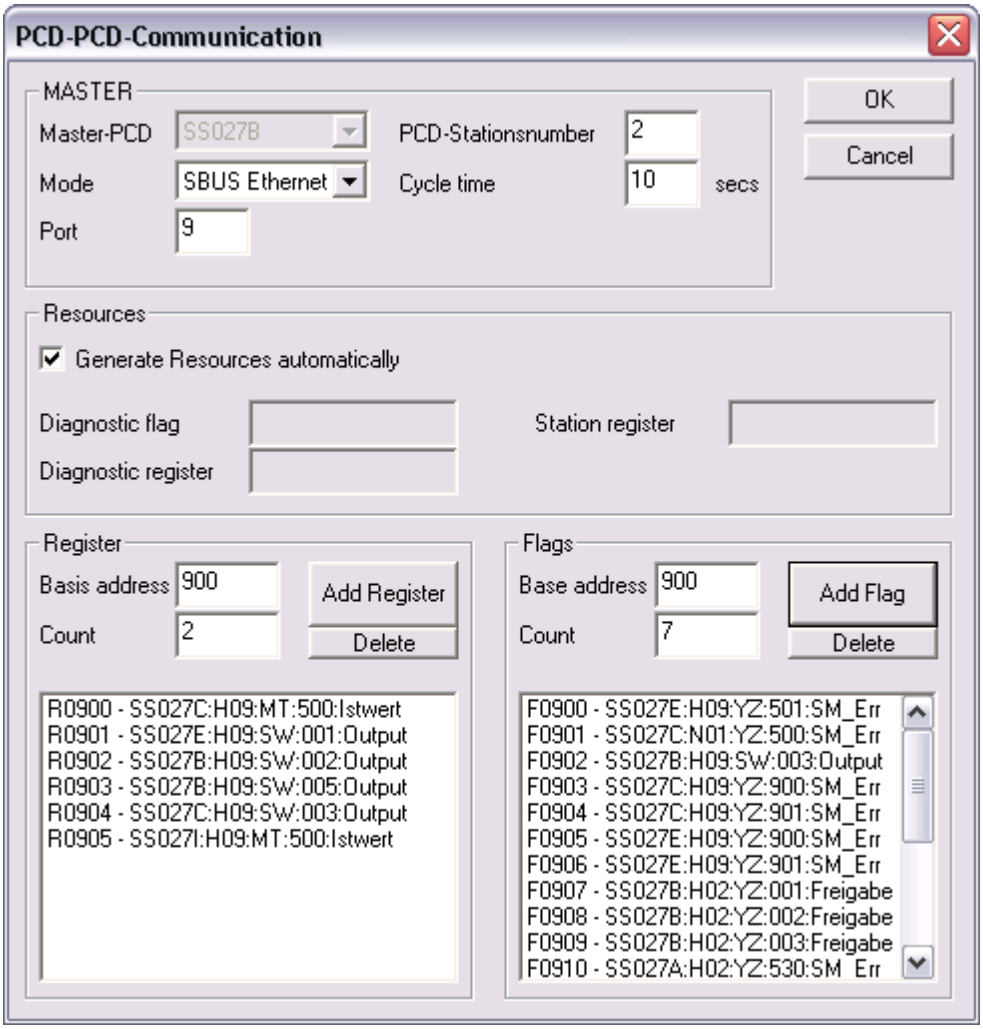
Different data points from various PLC controllers can then be defined through the menu item PCD/PCD Communication, which should be available universally. The idea behind PCD/PCD communication is as follows:
A master PLC, where all communication is processed, is defined. The master obtains the required data on all PLC controllers and then writes it to a defined area of all PLC controllers. Thus the general data is made directly available to all PLCs.



Now the various parameters can be defined in the dialog:

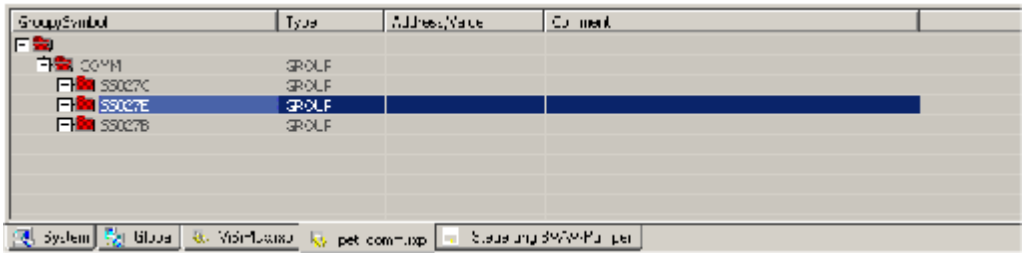
MASTER

- Master PCD — Defines the PLC master
- Mode — Defines RS485 or ethernet
- Port — Defines the communications port
- PCD station number — SBUS station of the master
- Cycle time — Read/write interval (in seconds)
- Register and flags — -

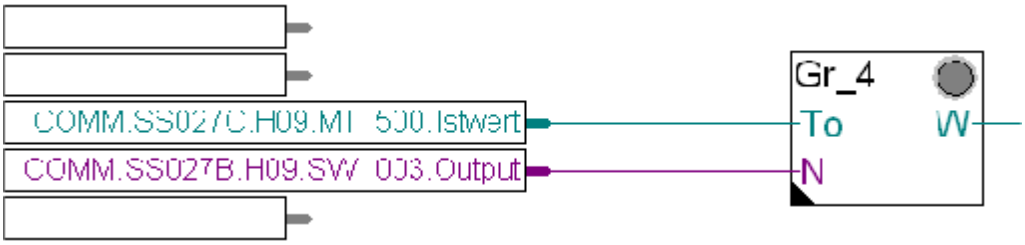


By adding registers and flags, resources from various PLC controllers which should be copied to all PLC controllers can be defined. Selection takes place through the DMS Selection dialog. The Code Generator automatically generates the PLC code required to communicate the data. A resources file is also generated for each PLC (pet_comm.rxp).
This can be attached to the project on any PLC. The file is inserted into the common files and added to all PLCs

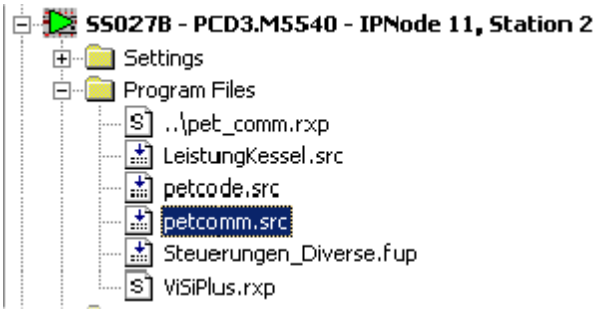
The resources begin with the designation 'COMM'. The name of the PLC is inserted as an additional group name.



These resources can be accessed like any other resources:
The resources can now be used in the PLC programs.



The source code 'petcomm.src' must be added to the master PLC. This code is automatically generated (in the directory of the master PLC).



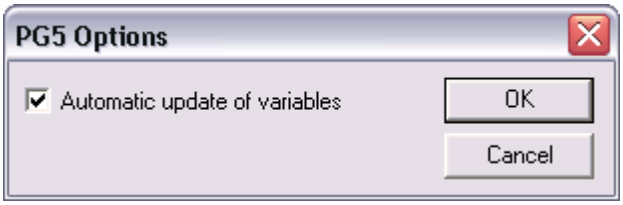
The communication interval should be made as slow as possible. It does not make sense to communicate the outside temperature every second.
Past experience has shown that the communication interval should be set to 10-30 seconds. For rapid control (e.g. control output of a valve), communication must be programmed separately. An interval of a few seconds is usually sufficient in the HVAC area.

The definition of PCD/PCD communication is stored in the file '*pcdcomm.def*' in the directory of the master PLC.

6.11.5.8 PG5 Options

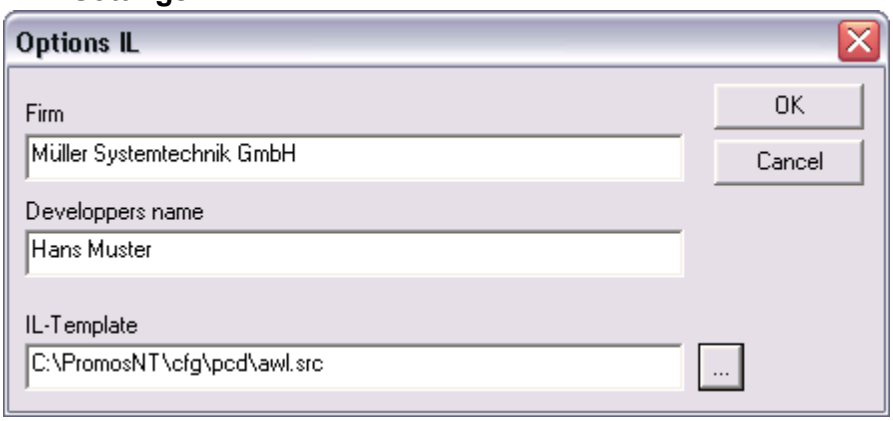
This is where the settings are configured for importing and generating the AWL code.

Import Settings



Marking the **Automatically Update Variables** checkbox prompts the PET to automatically reimport the toolbar for Saia PG5® Fupla in case of any changes.

AWL Settings



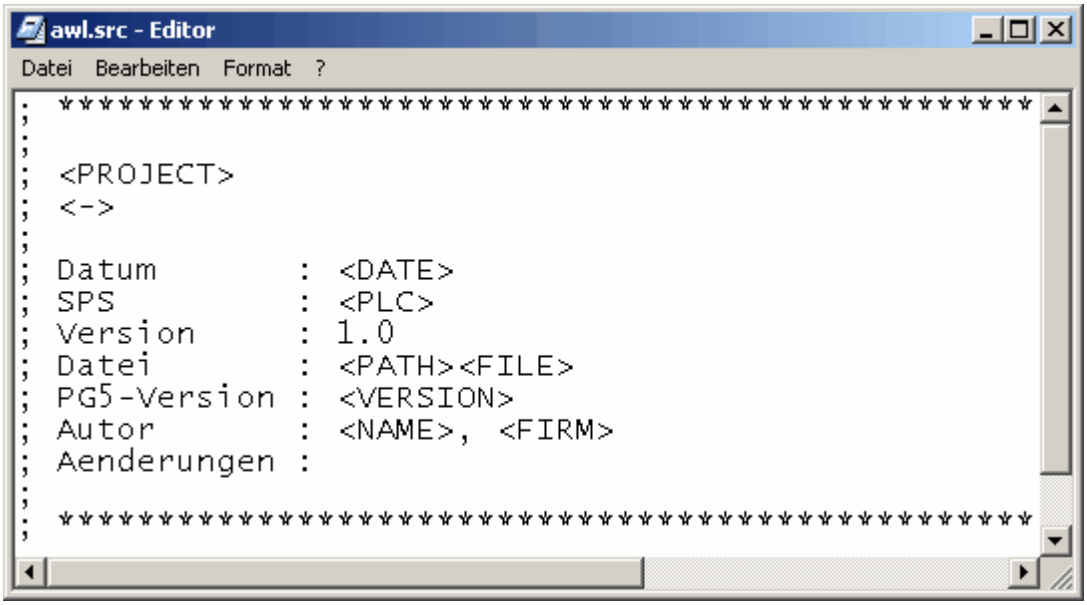
The data for the text header that is inserted on **Generate PLC Code** can be entered in this dialog window.

Company
This input field is used for entering the name of the development company that is entered in the text header in the <Company> space.

Name of the Developer
This is also entered in the text header in the <Name> space.

Template for AWL Programs
A file with a template text frame is indicated here.

The template text frame may appear as follows:



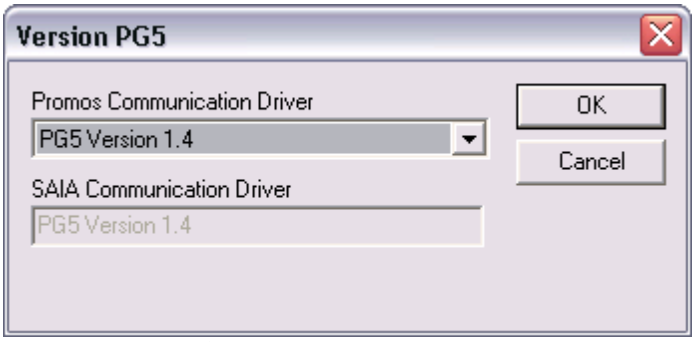
The words in angle brackets are so-called placeholders. They are replaced by predefined data on generation of the AWL file by Visi.Plus.

We recommend not changing the placeholders manually. Their positions, however, can be changed.

The **template file "awl.src"** is located by default in the directory "c:\Visi.Plus\cfg\pcd\". Of course this can be changed.

In doing so, it must be noted that the modified file should be saved during any update of Visi.Plus.

PG5 Version



The installed PG5 version is selected here. This setting is then entered in the file **pcd.5pj** in the \pcd project folder:

```
[PG5 Project File - DO NOT EDIT]
PG5Version=V2.1
```

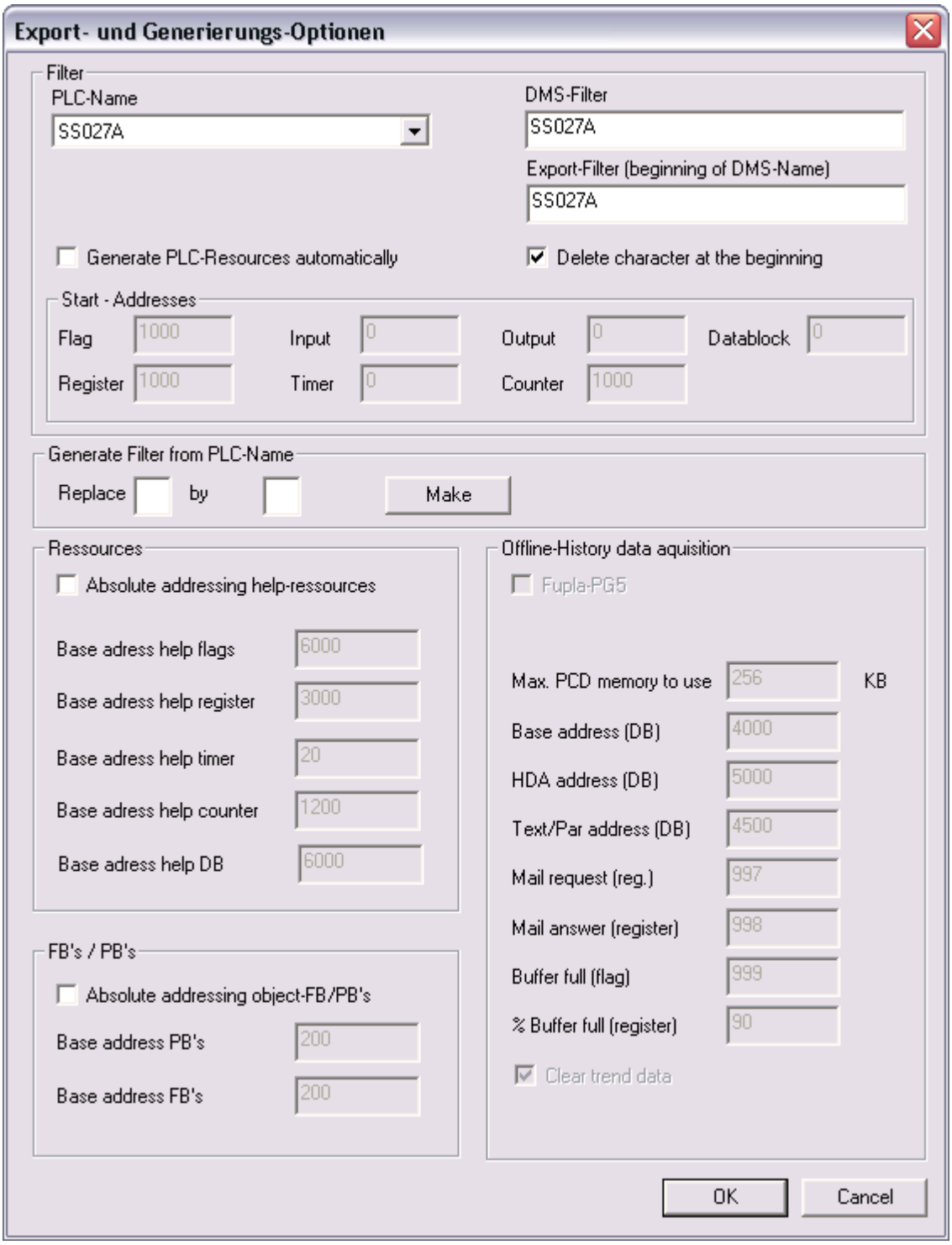
The actual version number is extracted from this string (-> "14") in entered in the registry under:

```
"HKEY_CURRENT_USER\Software\Visi.Plus\PET\Settings" "PG5" = "14"
```

This registry entry is consequently referenced by other Visi.Plus modules, e.g. SDriver.

6.11.5.9 Export Options

Here, settings can be made that have an influence on code generation.



Filter / PLC Names
Only data for the selected PLC is generated. If no PLC is selected, but multiple PLCs are present in the project, no usable code can be generated.

DMS Filter / Export Filter
The filters in **Export and Generation Options** only filter for the data points that match the characters in the input field.

Remove beginning characters
Makes it possible to use the characters entered not only for filtering, but also to omit them when generating the PLC symbols. Example: WL027:H02:MT:500:ActualValue becomes H02.MT_500.ActualValue. It hardly makes sense to have all resources defined on PLC WN027 begin with WL027 (unnecessary extra writing).

Automatically Assign PLC Resources
If this option is activated, the PLC resources are automatically assigned when objects are inserted. The new object PLC addresses are added to the end in each case.

Start Addresses
In this section the respective start addresses of the PLC resources are defined. These addresses may not conflict with the automatically assigned resources in PG5.

Generate Filter from PLC Names
If the PLC does not have the same name as the DMS name (beginning), then it is possible to replace individual parts with others.


If multiple PLC stations are defined in the project, a filter can be set which only generates the PLC code for the respective defined station.

We recommend truncating each of the station names. The PLC symbols are shortened and there is less typing work for later programming. All symbols begin on the same PLC with the same DMS name.

Auxiliary Resources
If all the resources must be permanently defined in a project, the start addresses of the resources can be entered in this section.

Absolute Addressing of Auxiliary Resources
If the checkbox is activated, the basic address of the auxiliary resources to be used can be entered manually.

If the checkbox is unmarked, the auxiliary resources are assigned dynamically (see PG5 dynamic element assignment).

Absolute Addressing of FB/PB
If the checkbox is activated, the basic address of the FB and PB to be used can be entered manually.

If the checkbox is not marked, the FBs and PBs are assigned dynamically (see PG5 dynamic element assignment).


If the file generated by the PET (which then only contains PLC codes) is opened, the number of auxiliary flags, auxiliary registers, etc used by the PET is entered in the file header.

It must be ensured that Visi.Plus has sufficient auxiliary flags available and these flags are not being used in the PLC program.
Recommendation:*Assign the auxiliary resources dynamically.*

Offline Historical Data Acquisition
*It is possible to register the historical data directly on the PLC and later (e.g. once daily) import it into the Visi.Plus database.*This allows data acquisition on stations which are not always online (e.g. via modem).

6.11.6 Toolbar for important PG5 commands

The toolbar provides quick access to some important commands in PG5.

Starts the Saia PG5® Project manager. This command can also be accessed via the menu command "PG5 -> Project Manager".

Build recompiles the respective active PLC file.

Rebuild ALL recompiles the entire project, in other words all files of the active CPU.

Open a new PG5 file in the project.

Download Program loads the PLC code into the PLC controller (not active yet).

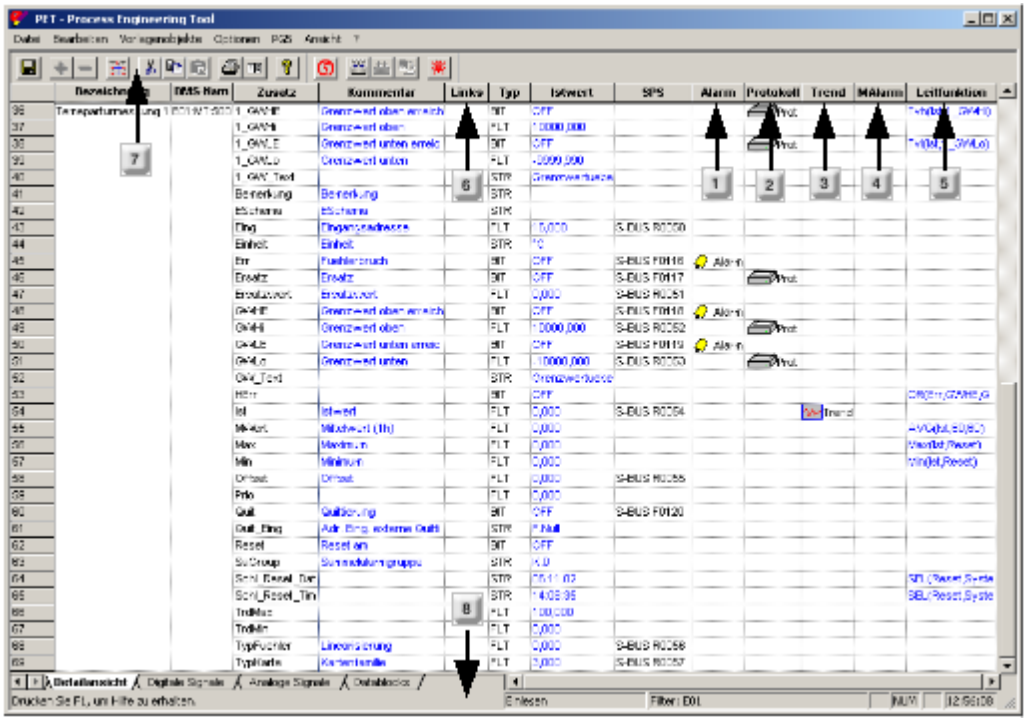
Starts the PG5 Online Debugger.

6.11.7 View menu

In the "**View**" menu, additional columns and window bars can be switched on or off in the PET.

The columns listed in the following image are used to hold actions, depending on their name.

In the following image, the Detail View of a system object is shown with all columns and window bars switched on.



1 _ 6 **Alarm, Log, Trend, MAlarm, Control Function and Link columns**

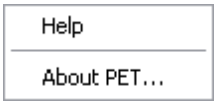
Toolbar

7 **Toolbar**

8

6.12 "?" or Help menu

The Visi.Plus online help and current information about the PET version are found in the "**?**" menu.



6.12.1 Help Topics

The Visi.Plus online help can be accessed through the menu "Help topics", via the corresponding **Help** button or by pressing the **<F1>** key.

6.12.2 About PET

The version number of the PET can be viewed here, among other things.

Important
The version number must always be indicated in case of support queries.

We recommend our internet forum for support queries: This offers two advantages: First, help is available more quickly; and secondly, other users can benefit from the entries.

6.13 PET-File

The PET file (extension PET) describes the structure of the two PET lists for analogue and digital values. The PET has been developed to be as independent as possible of any products. The preferred drive can be set via the menu item 'Options' -> 'PLC driver'.

All PET files must be in the BIN directory of the Visi.Plus installation.

Any file name can be chosen. The extension must be '.pet'. PET displays all found file names in the list of PLC drivers.
Definitions of views in the PET must be provided at program startup. Export definitions are read again with each export.

The file is structured like a standard Windows INI file.

6.13.1 Section for digital and analogue values

Example:

```
[Digital]
Show = :PLC:Logic
Text = SBUS N
SChar= #
1=Bezeichnung
2=DMS-Name
3=Kommentar
4=Telegramm
5=Station
6=Datentyp1
7=Adresse1
8=Logik

PLC = PLC
PLC1 = SBUS N
PLC2 = 5
PLC2Format = %03d
PLC2Space = 1
PLC3 = 6
PLC3Format = F;I;O
PLC4 = 7
PLC4Format = %04d

Signal1=Address
Type1=DWS
```

```
Signal2=Logic
Type2=BIT
Signal3=Station
Type3=DWS
Signal4=Telegram
Type4=DWS
Signal5=Type
Type5=STR
Value5=Flag
```

The numbers in the upper area define the columns in the PET. The definitions with PLC are responsible for the format in the Detail view. Signal, Type and Value define the appearance of an analogue/digital data point in the DMS.

Show=	Defines which string should be included in the DMS name for the signal to be either digital or analogue. A signal should be a defined that only fits in the appropriate category.
Text=	Short description of driver. This is used internally as a driver value for identification. It must be defined in both areas [Digital] and [Analogue]. It should be identical in both areas, since this makes driver data easier to locate. The text is also displayed as the default value with Paste. Example: PLC
Schar=	Character that determines when the data point is to be displayed in red.
x=	Marker that is included in the title bar. This text also serves as a marker for supplying more detailed information on the column display (x = column number). Numbering must be continuous. The numbers determine the number of columns.
PLC=	Defines the string for driver definition. In the example, PLC=PLC means that the DMS includes a data point named 'PLC', which contains the driver definitions. The string describing the data point in the PLC is stored in the DMS under the name defined here. This name is reserved by the system and must not appear in any normal DMS name.
PLCx=	Text to be displayed at the short description in the Detail view. X should be replaced by a sequential number (e.g. PLC1, PLC2, etc.). If a number is defined instead of a text, the content of the column will be transferred. Using the definitions PLC1 .. PLCx, it is possible to define how the string in the Detail view should be structured.
PLCxFormat=	Defines the output format (e.g. %03d = 3 digits including leading zeros, %5.2f = 5 digits with two decimal places). C syntax. Alternatively, a selection can be specified (e.g. F;I;O)
PLCxSpace=	Number of spaces to be placed between two outputs.
Signalx=	Name of the data point to be integrated in the DMS under the driver.
Typex=	Data type of data point to be integrated.
Valuex=	Initial value of data point.

X must be a consecutive number.

Example for output to the Detail view:

```
PLC1=SBC
PLC1Space=1
PLC2=5
PLC2Format=%03d
PLC3=PCD
PLC3Space=1
```

e.g. outputs the following text to the Detail view: `SBC 312 PCD`

In the DMS, data points are defined via Signal, Type and Value. The Signals, Types and Values must be numbered consecutively.

Signalx can have any choice of DMS name (extension in driver description).

Typex can be of the type: BIT, BYS, BYU, WOS, WOU, DWS, DWU, FLT or STR.

Valuex can be any choice of value, which will be transferred as the default value. It is important to ensure that the value matches the data type defined under Type.

6.13.2 Section for data types

In this section the data type selection is defined. The choice of data type can therefore be limited per driver.
Example:

```
[DataType]
0=NONE
1=BIT
2=FLT
3=STR
```

Caution: Numbering starts at zero!

In this example, when adding a new data point the user can only select NONE, BIT, FLT or STR.

6.13.3 Section with freely deinable column defintions

In the section for digital and analogue values, columns are defined by numbers.

```
1=Bezeichnung
2=DMS-Name
3=Kommentar
```

For each definition, a Detail definition must be created.

Example:

```
[Bezeichnung]
Value = ..:NAME
Size = 200
ReadOnly = 1
```

The text in square brackets [] must exactly match the text in the section for digital/analogue values (caution with upper/lower case).

Value=	Value to be displayed in the column. The current data point is always assumed, whose value is stored in the PLC. Special characters: = DMS name
..:xyz=	Content of the data point located one level higher (e.g. ..:NAME)
xyz=	Content of the data point, where xyz also works on several levels.
Title=	Column title, if this differs from the text in square brackets.
Size=	Column width in pixels.
ReadOnly=1	Read only
Min=	Minimum input (numerical value)
Max=	Maximum input (numerical value)
Error=	Error message if input range is exceeded
Control=	Selection field. Possible selections must be separated by semicolons (;). Example: Control=Flag;Input;Output
Increment=	Individual values will be incremented by this value if an area of the column is selected and the right-hand mouse button is pressed.
Komma=	Number of decimal places to be displayed.
Position=	0 = left justified, 1 = centred

6.13.4 Section for code generation

This section comprises the general definitions for code generation.

```
[Generate]
Menu1=Concept-Datei erstellen (Modicon CCN-File)
Menu2=
PCD=0
Title=NO
Select=Modicon OPC-Files (*.ccn)|*.ccn|All Files (*.*)|*.*||
```

Menu1=	Defines the first export menu item (text will be displayed in PET menu). If no Menu1 is defined, the menu item will be hidden in PET.
Menu2=	Defines the second export menu item (text will be displayed in PET menu). If no Menu2 is defined, the menu item will be hidden in PET.
PCD=0	No Saia PCD®, i.e. no $ifdef/$endif will be generated. In addition, the Saia PG5®-specific menus (Fupla, code generation) will be hidden.
Title=NO	No title lines will be generated. By default, two lines are created with the generation date.
Select=	Definition of selection for file selection dialog box. This string can be used to define precisely what types of files can actually be generated. Example: `Concept-Files (*.ccn)|*.ccn|All Files (*.*)|*.*||` allows files with the extension .cnn to be selected, or all files.

6.13.5 Section for data export

For data export, there are two different areas:

Menu item: Create definition file	[Export]
Menu item: Export resource file	[Export2]

Example:

```
[Export]
Comment = ;
DMSChar =
Space = _
Dot = _
Slash = _
Plus = _
Minus = _
SChar =
FileExt=.src
Title=
1=@DMSName
2=\t
3=EQU
4=\t
5=@Type[0]
6=\s
7=@Address
8=\t
9=;
10=\s
11=@Comment
12=\s
13=@Name
14=\n

[Export2]
Comment = ;
DMSChar =
Space = _
Dot = _
Slash = _
Plus = _
Minus = _
SChar =
FileExt=.rxp
Title=Name;Media;Address;Count;Scope;Comment;DB/Text_Size;Initial_Value
1=@DMSName
2=;
3=@Type[0]
4=;
5=@Address
6=;1;Public
7=;
8=@Comment
9=\s
10=@Name
11=;;
12=\n
```

Comment= Character with which a comment can be defined in the source code of the

target system.

DMSChar= Character with which the DMS character (:) is to be replaced. This is to ensure that the special character : can be replaced or removed from the DMS string (Remove: DMSChar=).

Space= Character with which a space in the DMS name is to be replaced (if no character, a space will be eliminated).

Dot= Character with which a dot in the DMS name is to be replaced (if no character, a dot will be eliminated).

Slash= Character with which a forward slash (/) in the DMS name is to be replaced.

Plus= Character with which a plus (+) in the DMS name is to be replaced.

Minus= Character with which a minus (-) in the DMS name is to be replaced.

SChar= No function yet

FileExt= Extension of the file to be created.

Title= String will be inserted as the first line of the generated file.

x= Consecutive number (1=, 2=, etc.). Defines a line to be output.
The following special characters can be used:
\t = Tab
\n = New line
\s = Space
\\ = Backslash (\)

@DMSName is replaced by object DMS name, including extension
=

@Name= is replaced by name of object

@Comment= is replaced by the comment (per data point)

@xyz= Content of DMS data point xyz. The number of characters to be transferred can be defined in square brackets (e.g. @Type[1] only transfers the first character of the DMS value).
Only values from the driver definition in the DMS can be transferred.
A format can also be attached, e.g. (%04d)
including round brackets: 7=@Address(%04d).
It is also possible to generate a string depending on the DMS value.
Syntax: @DMSName=OK?On:Off
If the DMS contains value 'OK', an 'On' will be output. If not, an 'Off' will be output.
Example: 6=@Type=INT?04:00
Otherwise, any choice of text can be defined.

Example:

```
1=N;
2=@DMSName
3=;
4=@Type
5=;
6=@Type=INT?04:00
7=@Address(%04d)
8=;;
9=@Name
10=\n
```

e.g. outputs a line with the following content:

```
N;H02MT500Strg;BOOL;040124;;Störung
```

6.13.6 SDriver.pet

```
// Version:  1.6.1.13
// Date:     13.06.13 SBC:AK/GM

[Generate]
Menu1_German=Ressourcenliste für AWL/IL generieren (.src)
Menu2_German=Ressourcenliste für Fupla generieren (.rxp)
Menu1_English=Generate resources for AWL/IL (.src)
Menu2_English=Generate resources for Fupla (.rxp)
PCD=1
Title=NO
Select=Saia PG5® AWL/IL-Resourendatei (*.src)|*.src|All Files (*.*)|*.*||


[DataType]
0=NONE
1=BIT
2=DWU
3=FLT
4=STR

[Digital]
Show = :PLC:Logic
Text = SBUS N
SChar= #
1=Bezeichnung
2=DMS-Name
3=Kommentar
4=Channel
5=Station
6=Telegramm
7=Datentyp1
8=Adresse1
9=Logik
Ressourcen=4;5;6;7;8

PLC = PLC
PLC1 = 4
PLC1Space = 1
PLC2 = 7
PLC2Format = F;I;O
PLC3 = 8
PLC3Format = %04d

Signal1=Address
Type1=DWS
Signal2=Logic
Type2=BIT
Signal3=Station
Type3=DWS
Signal4=Telegram
```

```
Type4=DWS
Signal5=Type
Type5=STR
Value5=Flag
Signal6=Channel
Type6=STR
Value6=SBUS


[Analog]
Show = :PLC:Diff
Text = SBUS N
SChar = #
1=Bezeichnung
2=DMS-Name
3=Kommentar
4=Channel
5=Station
6=Telegramm
7=Datentyp2
8=Adresse1
9=SPS_Lo
10=SPS_Hi
11=Unit_Lo
12=Unit_Hi
13=Diff
Ressourcen=4;5;6;7;8

PLC = PLC
PLC1 = 4
PLC1Space = 1
PLC2= 7
PLC2Format = T;C;R;D
PLC3= 8
PLC3Format = %04d

Signal1=Address
Type1=DWS
Signal2=Diff
Type2=FLT
Signal3=Station
Type3=DWS
Signal4=PLC_Hi
Type4=FLT
Value4=1.0
Signal5=PLC_Lo
Type5=FLT
Value5=0.0
Signal6=Telegram
Type6=DWS
Signal7=Type
Type7=STR
Value7=Register
Signal8=Unit_Hi
```

```
Type8=FLT
Value8=1.0
Signal9=Unit_Lo
Type9=FLT
Value9=0.0
Signal10=Channel
Type10=STR
Value10=SBUS


[Datablock]
Show = :PLC:DBIndex
Text = SBUS N
SChar = #
1=Bezeichnung
2=DMS-Name
3=Kommentar
4=Channel
5=Station
6=TelegrammDB
7=Datentyp3
8=AdresseDB
9=DBIndex
10=SPS_Lo
11=SPS_Hi
12=Unit_Lo
13=Unit_Hi
Ressourcen=4;5;6;7;8

PLC = PLC
PLC1 = 4
PLC1Space = 1
PLC2= 7
PLC2Format = D
PLC3= 8
PLC3Format = %04d
PLC4= .
PLC5= 9
PLC5Format = %d

Signal1=Address
Type1=DWS
Signal2=DBIndex
Type2=DWS
Signal3=Station
Type3=DWS
Signal4=PLC_Hi
Type4=FLT
Signal5=PLC_Lo
Type5=FLT
Signal6=Telegram
Type6=DWS
Signal7=Type
Type7=STR
```

```
Value7=Datablock
Signal8=Unit_Hi
Type8=FLT
Signal9=Unit_Lo
Type9=FLT
Signal10=Channel
Type10=STR
Value10=SBUS



[Export]
Comment = ;
DMSChar =
Space = _
Dot =
Slash = _
Plus = _
Minus = _
SChar =
FileExt=.src
Select=Saia PG5® AWL/IL-Resourendatei (*.src)|*.src|All Files (*.*)|*.*||
Title=
1=@DMSName
2=\t
3=EQU
4=\t
5=@Type[0]
6=\s
7=@Address
8=\s
9=@PlcEQU
10=@PlcValue
11=\t
12=;
13=\s
14=@Comment
15=\s
16=@Name
17=\n

[Export2]
Comment = ;
DMSChar = .
Space = .
Dot = .
Slash = .
Plus = .
Minus = .
SChar =
FileExt=.rxp
Select=Saia PG5® Fupla-Resourendatei (*.rxp)|*.rxp|All Files (*.*)|*.*||
Title=Symbol;Media;Address;Comment;Scope
1=@DMSName
```

```
2=;
3=@Type[0]
4=;
5=@Address
6=;
7=@Comment
8=;
9=1
10=\n


[Bezeichnung]
Title_German = Bezeichnung
Title_English = Description
Value = ..:NAME
Size = 250
ReadOnly = 1

[DMS-Name]
Title_German = DMS-Name
Title_English = DMS-Name
Value = .
Size = 350
ReadOnly = 1

[Kommentar]
Title_German = Kommentar
Title_English = Comment
Value= Comment
Size=350
ReadOnly = 1

[Channel]
Title_German = Channel
Title_English = Channel
Value = PLC:Channel
Size=60

[Telegramm]
Title_German = T-Nr.
Title_English = T-Nr.
Value = PLC:Telegram
Size = 40
Min = 0
Max = 999
Error=Es sind max. 999 Telegramme erlaubt
Position= 1

[TelegrammDB]
Title_German = T-Nr.
Title_English = T-Nr.
Value = PLC:Telegram
Size = 40
```

```
Min = 0
Max = 999
Error=Es sind max. 999 Telegramme erlaubt
Position= 1
Increment=1
DBSort = 1

[Station]
Title_German = Station
Title_English = Station
Value = PLC:Station
Size = 60
Min = 0
Max = 253
Error=Es sind nur Stationsnummern im Bereich 0 .. 253 erlaubt
Position= 1

[Datentyp1]
Title_German  = Typ
Title_English = Type
Value = PLC:Type
Size = 60
Control = Flag;Input;Output

[Adresse1]
Title_German=Adresse
Title_English=Address
Value = PLC:Address
Size = 55
Min = 0
Max = 16384
Error=Es sind nur Adressen im Bereich 0 .. 16383 erlaubt
Increment = 1
Position= 1

[AdresseDB]
Title_German=Adresse
Title_English=Address
Value = PLC:Address
Size = 55
Min = 0
Max = 16000
Error=Es sind nur Adressen im Bereich 0 .. 16000 erlaubt
Increment = 1
Position= 1
DBSort=1

[DBIndex]
Title_German=DB-Index
Title_English=DB-Index
Value = PLC:DBIndex
Size = 60
Min = 0
Max = 383
```

```
Error=Es sind nur Adressen im Bereich 0 .. 383 erlaubt
Increment = 1
Position= 1
DBSortIndex=1

[Logik]
Title_German=Logik
Title_English=Logic
Value = PLC:Logic
Size = 60
Control = Normal;Invers

[SPS_Lo]
Title_German=SPS Lo
Title_English=SPS Lo
Value = PLC:PLC_Lo
Size = 45
Increment = 0
Position= 1

[SPS_Hi]
Title_German=SPS Hi
Title_English=SPS Hi
Value = PLC:PLC_Hi
Size = 45
Increment = 0
Position= 1

[Unit_Lo]
Title_German=Unit Lo
Title_English=Unit Lo
Value = PLC:Unit_Lo
Size = 45
Increment = 0
Position= 1
Komma=2

[Unit_Hi]
Title_German=Unit Hi
Title_English=Unit Hi
Value = PLC:Unit_Hi
Size = 45
Increment = 0
Position= 1
Komma=2

[Diff]
Title_German=Diff
Title_English=Diff
Value = PLC:Diff
Size = 45
Increment = 0
Position= 1
Komma=2
```

```
[Datentyp2]
Title_German=Typ
Title_English=Type
Value = PLC:Type
Size = 70
Control = Timer;Counter;Register;RegFloat;RegFloatIEEE

[Datentyp3]
Title_German=Typ
Title_English=Type
Value = PLC:Type
Size = 75
Control = Datablock;DBFloatIEEE
.
```

Saia Visi.Plus

Database programs and systems

Chapter 7

7 Database programs and systems

Visi.Plus contains two proprietary databases:

- **DMS** – **D**ata **M**anagement **S**ystem

Fast database (RAM-based) to hold all current values. In addition, all configuration values are also saved in this database.

- **PDBS** – **P**rocess **D**ata**B**ase **S**ystem

Disk-oriented database system. Historical data is saved in this database. All alarms and logs are also managed by the PDBS.

7.1 Summary

This chapter explains the functioning of and worthwhile information about the DMS and PDBS, the two database systems of Visi.Plus.
The following topics are discussed:

- What do the abbreviations DMS and PDBS mean?
- How are database variables and control functions generated?
- How is the DMS data automatically saved?
- Which data is logged by which events in the PDBS?
- How is a backup of the PDBS generated?

7.2 The Data Management System DMS (dms.exe)

The DMS (Data Management System) module is the central element of Visi.Plus and runs in the memory (RAM) of the PC designated as the host computer. The DMS has the important task of managing all emerging data. Regardless of the size of a Visi.Plus project, there is only one DMS (aside from Multi DMS installations or in a redundant system)!

Since many system objects (pumps, sensors, valves, etc.) have similar attribute and data point structures (target value, actual value, status, thermo contact, diagram number, designation, manufacturer, output, etc.), it is logical to combine these similar properties and define them as so-called template objects (BMO). For instance, a template object called "MES01" which contains all possible attributes for sensors could be created. This would include, for example, measurement value, default value, unit, maximum value, minimum value, type, etc. System objects are then created with the help of such template objects. This approach also gives the entire project a clear data structure. Therefore it is imperative to define as few template objects as possible. If these templates have been designed in detail, no data will be forgotten when creating the system objects.

The template objects and subsequently created system objects are defined by the project engineer in a convenient tabular form in the PET module (chapter PET).

The DMS can also be used to define template objects. However, this requires detailed knowledge of Visi.Plus. The advantage of the DMS is that the entire data structure of a directory on the hard disk is displayed (with subdirectories).

After starting the DMS module, all data is managed exclusively in the computer's memory (RAM) for speed-related reasons.

All Visi.Plus modules (component programs) access DMS data via a predefined interface (via pipe or socket connection). Drivers have the task of retrieving data from the controllers and entering it in the DMS or vice-versa. Information about which data this is, in turn, contained in the DMS data, which was defined by the PET used by the programmer.

Access to the hard disk only takes place when the program starts up, when the program is exited, cyclically (e.g. once daily) or when the project engineer explicitly saves the DMS.

If you want data and signals to be visible on the screen or changed via input fields and screen buttons, the GE (Graphical Editor) is used. All graphical elements can be linked to the data in the DMS in order to bring about changes in colour, size or other modifications.

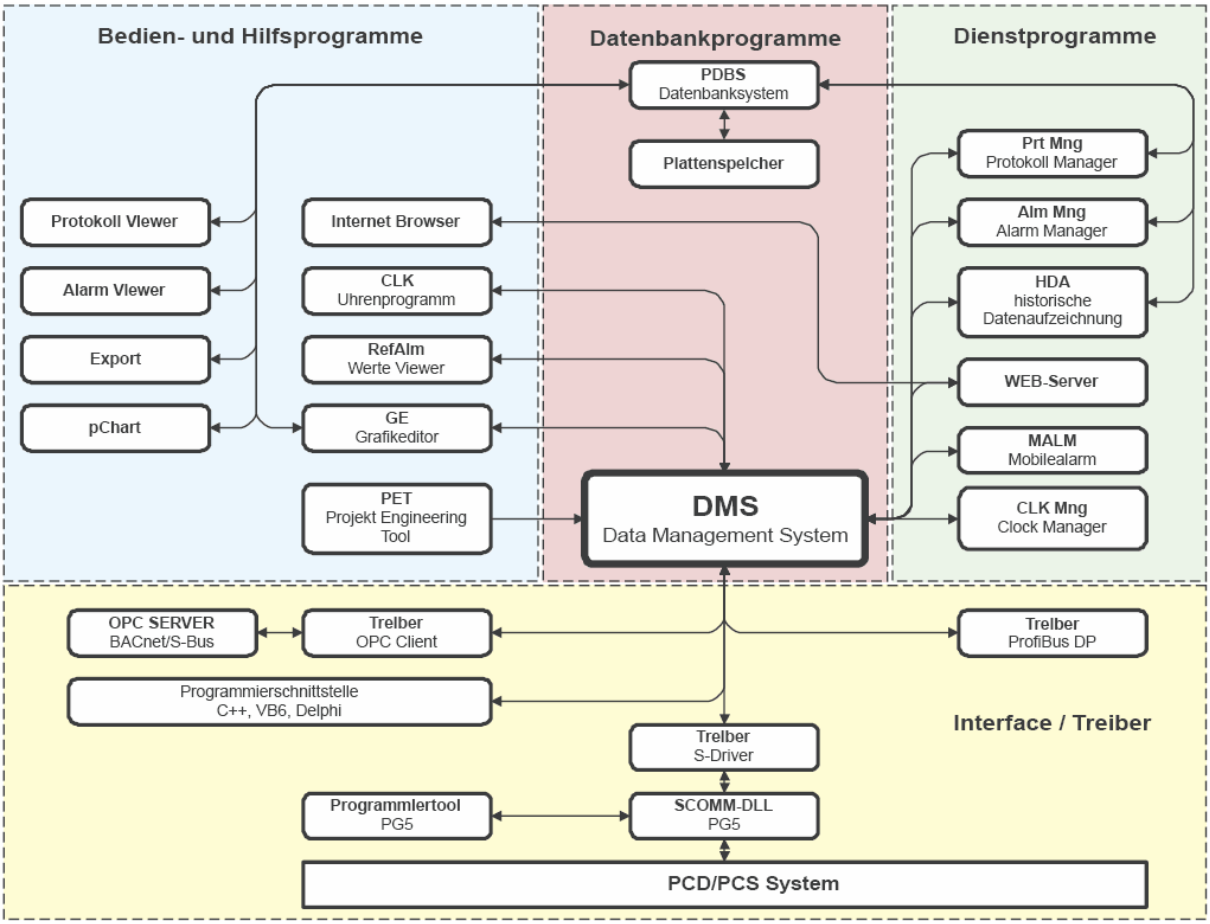
The DMS is therefore the hub for the exchange of data between the various program modules.

The communication of the individual program modules takes place in an event-oriented manner. Each Visi.Plus module registered with the desired data points and is automatically informed in case of a change in value.

The DMS can manage up to 10 million data points (depending on the memory size and performance of the PC). More about this topic is found in the chapter Performance characteristics and technical data.

It is possible to start up the DMS using a system with one or more PLC drivers on the same PC and to view the visual output on other PCs. It is also possible to install drivers and modules on external PCs. They can then exchange data over the network with the DMS in the host computer. In the process, up to 100 control stations can access one DMS simultaneously.

For more information relevant to this subject, also see the **chapter PET.**



7.2.1 Starting the DMS

The DMS is automatically started as the first module at every Visi.Plus system startup.

It is possible to start the DMS by itself. This might be useful if you do not want every module to start. This enables you, for example, to start just the PET, separately.

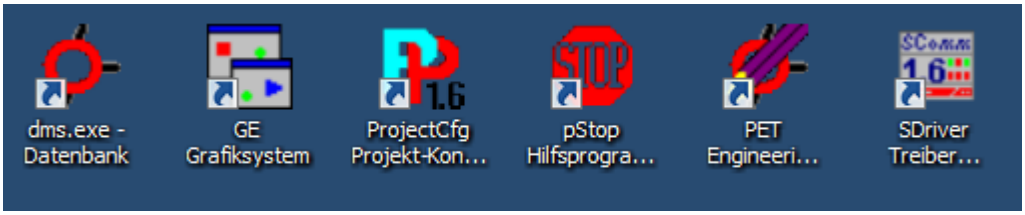
The module normally runs as a background process (on the status line, not as a service).

For verification purposes, this is displayed on the Windows taskbar by a red ring (bottom right at the edge of the screen in the system tray).

The DMS control window can be opened in the following manner:

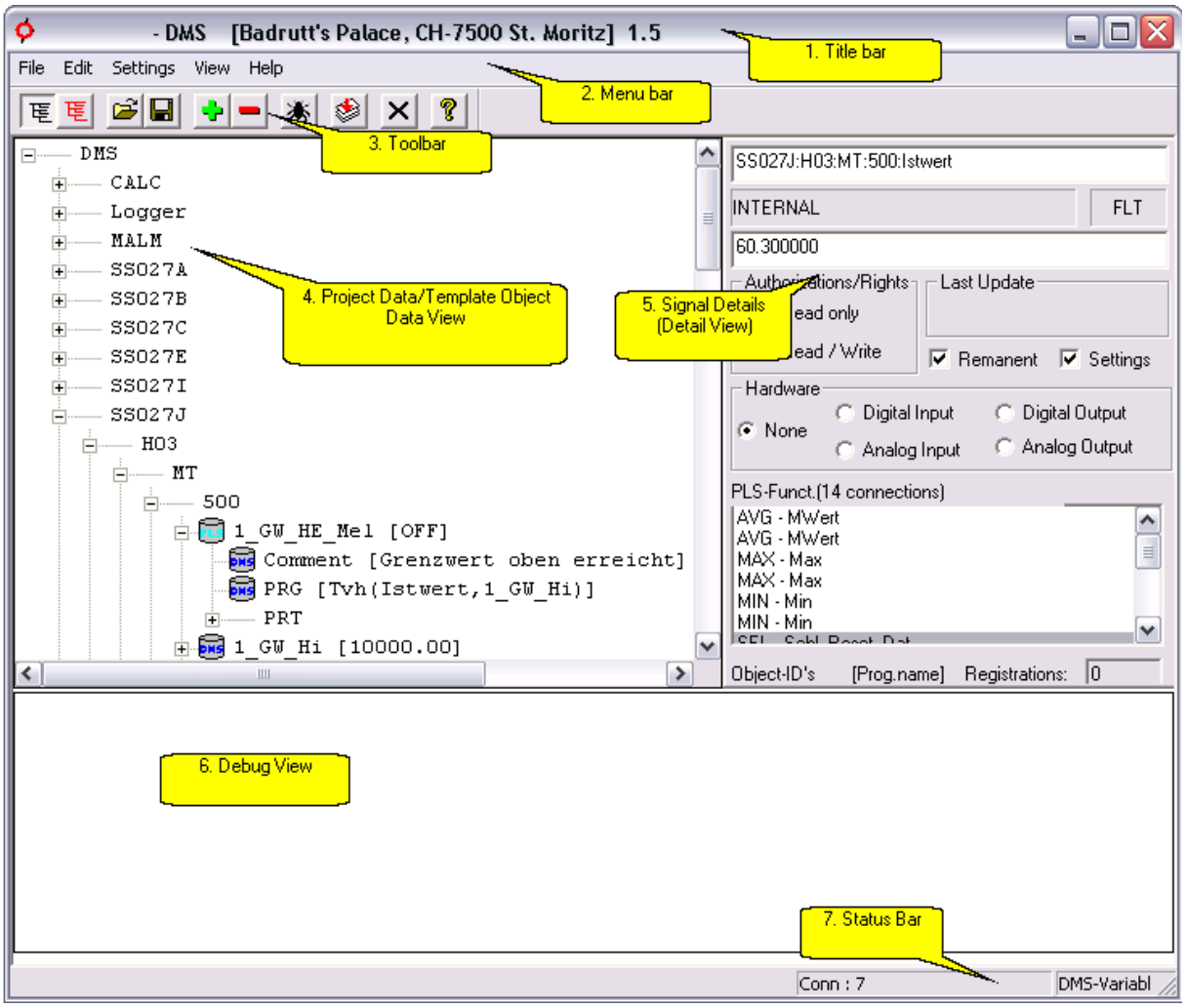
By double-clicking on the **icon** or by pressing the right mouse button and selecting **Display**.

For programming and commissioning, it is best to place shortcuts to the most important modules on the Desktop, e.g. as per the following recommendation:



*Please note that the **DMS** must always be **started first**!*

7.2.2 DMS user interface



The user interface in the Data Management System (DMS)

1 Title bar

Along with the **DMS** module name, the licencee and the licence location are shown in square brackets.

Menu bar

The menus are sorted according to basic functions. Under the **Edit** menu are the functions required for processing the current data (e.g. New Data Point, Delete Data Point). Under the **View** menu, you can adjust how the data (e.g. Object View or Operating Resource View) and toolbars are displayed. Commands used for processing and observing a data point can only be selected if the corresponding data point is active. Active means: the data point is selected on the screen (blue background).

Toolbar

This bar uses icons to offer quick access to the most important commands and functions.

Project Data/Template Object Data View

Signals (Project Data) and System Data View.

Signal Details (Detail View)

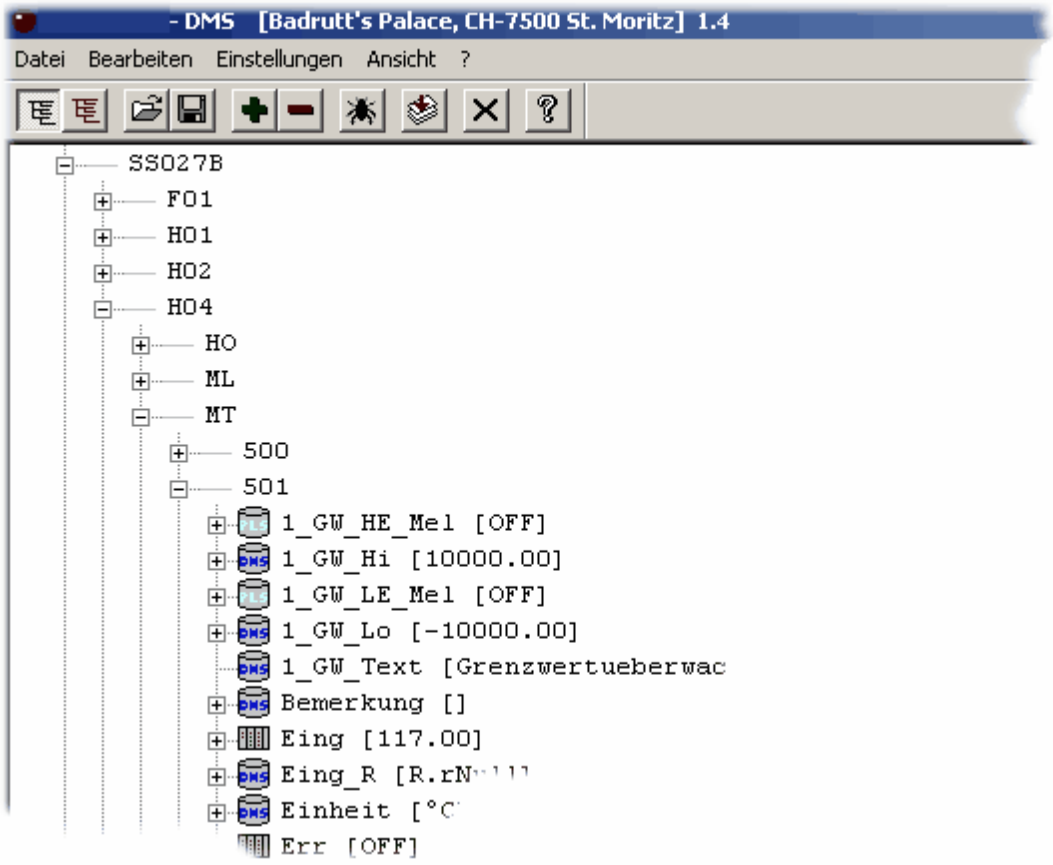
Here the data points are shown in detail. The data point content can be edited in online mode.

Debug View

The current telegrams (value changes) can be shown here.

Status Bar

The number of connections between the DMS and other Visi.Plus modules is shown in the status bar. After that, the number of DMS variables used can be seen. The text "changes" shows the number of changes of all variables used in the project.



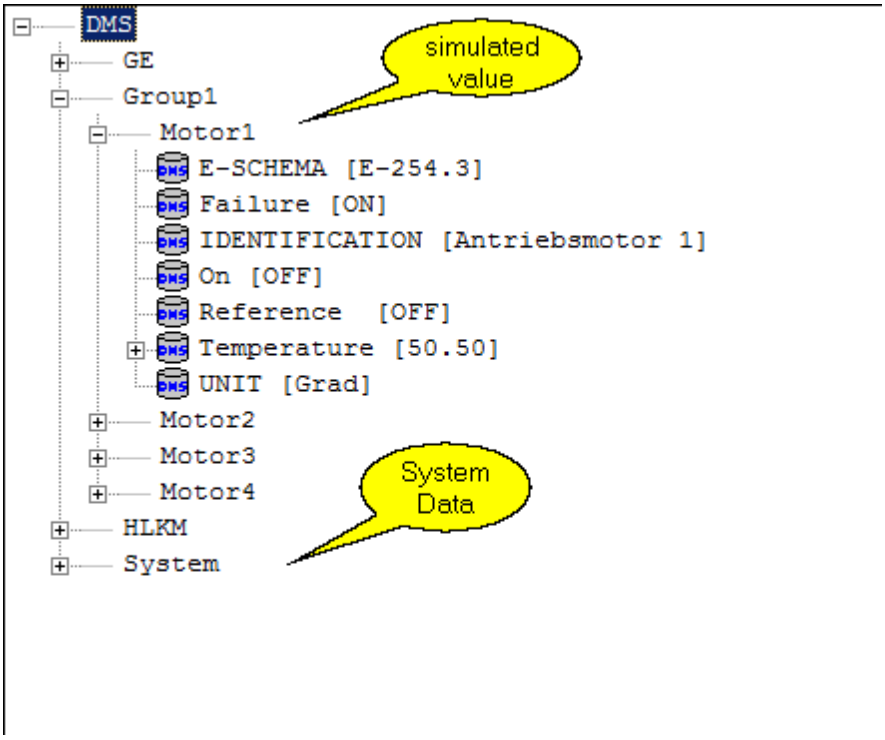
All integrated signals are shown in the left part, beginning with the DMS in the tree structure. By clicking on the ⊟ and ⊞ icons or by double-clicking on the respective AKS designation, the sub-paths open or close.

The view of the tree structure can be switched by clicking the following buttons:

If "Project Data (Object View)" is active, the data which actually being used is displayed.

If "Template Object (Tools View)" is active, the parent objects are displayed. These are used to create the system objects in the PET.

Simulation of values:



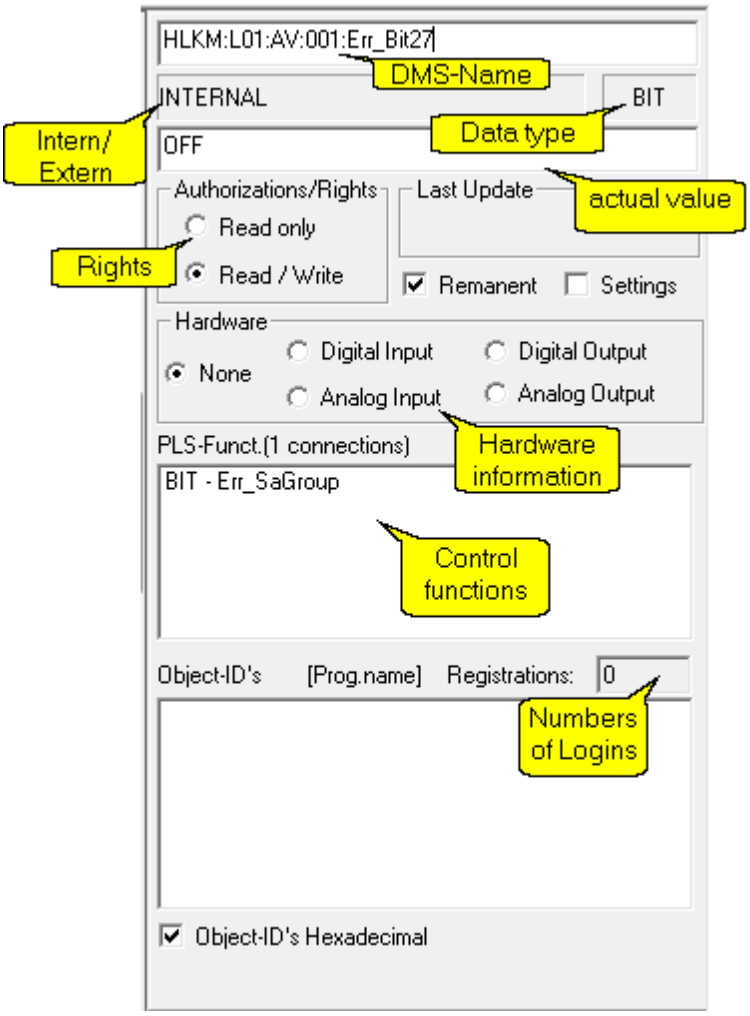
The signals of **Group1** are based on the values simulated by Visi.Plus through the PC. This allows various initialisations to be tested in the graphical system without being connected to a PLC.

System data points:

The signals of the **System Group** contain some system-internal data of Visi.Plus (date, time, blinkers, operating system versions, licence data, etc.).

*Click on **Group1** then on **Motor1** to see all template object attributes and their changes in value. By clicking on the **Value Simulation** command in the **Settings** menu, the simulated signals for the whole of **Group1** can be switched off.*

Data point details



The signal details can be viewed and the value of a selected signal can be changed in the upper right part of the user interface.

1: DMS name of the selected signal.

2: All DMS data created by Visi.Plus is **INTERNAL** data. If a program generates new data into the DMS while the DMS is running, it is declared as **EXTERNAL** until the next backup of the DMS data.

3: Data type of the signal (BIT, BYS, BYU, WOS, WOU, DWU, DWS, FLT, STR).

4: Display of the **current value**. Input field for changing the value or the text of the signal. For a binary signal, either a 0 [OFF] or a 1 [ON] must be entered. In order to adopt the changes, press the <ENTER> key.

5: Last Update shows the date and time of the last value change. If the value has not been changed since the beginning, the field will be blank.

6: If a constant that may not be changed is required, "Read Only" **rights** should be given to the data point. Then the data point cannot be overwritten by the visualisation (Graphical Editor). If a data point has "Read/Write" rights, it can be read and written to.

7:I f the checkbox **Remanent** is marked, the value of the signal while saving the DMS to the

hard disk is stored. If the checkbox is not marked, the value is not saved.

8: With the **Configuration** checkbox you can define whether the data point or its content should be written to the corresponding PLC register.

Example:
A data point "S01:MT:500:TypeCard" is created. The data point should save the analogue card type as a number (e.g. Value 2). When generating the PLC code, the following code line is generated (and is written to the PLC when the program is downloaded) if the Configuration checkbox is marked:

```
S01:MT:500:TypeCard     EQU     R100 := 2
```

9: The **Hardware** details are for informational purposes only.

10: PLS Functions (control functions) are listed. The number of connected control functions is indicated in the text.

11: Number of Logins at this data point.

12: All programs can log in to data points. When they do, each login is given an **Object ID** number. In addition to the the Object ID, the program and the name of the PC on which the program is running are displayed.

13: The Object IDs can be displayed in **hexadecimal** or decimal.

7.2.3 Data types

The following list of data types is available, depending on the PLC driver being used.

NONE
No data type. NONE is used if no data should be used.
Example: Data point: *Group1:**Motor1:***

BIT
Binary signal (for example on/off). Used to process digital signals (switch on/off, errors, etc.)
Example: Data point: *Group1:Motor1:**On***

BYS
Byte (8 bits) with sign (-128 to 127) inclusion of integer affected by a sign in the specified value range

BYU
Byte (8 bits) without sign (0 to 255) integer in the specified value range

WOS
Word (16 bits) without sign (0 to 65,535) integer in the specified value range

WOU

Word (16 bits) with sign (-32,768 to 32,767) inclusion of integer affected by a sign in the specified value range

DWS
Double word (32 bits) with sign (-2,147,483,648 to 2,147,483,647) inclusion of integer affected by a sign in the specified value range

DWU
Double word (32 bits) without sign (0 to 4,294,967,295) integer in the specified value range

FLT
Floating point number with sign (+5.42101E-20 to +9.22337E+18 (to an accuracy of 5 places) otherwise (-2.71056E-20 to -9.22337E+18). This type is used for analogue signals such as temperatures and pressures.
Example: Data point: *Group1:Motor1:**Temperature***

STR
String max. 80 characters long. Contains text, names, diagram numbers, etc.
Example: Data point: *Group1:Motor1:**Comment***

7.2.4 Creation of the DMS structure

The DMS structure, that is to say the database structure with all its data points and attributes, can be defined as follows in the DMS module or typically in the PET (see the chapter Project Engineering Tool).
Since the following operations on the DMS structure are used for template objects as well as addendums, attributes and data points, the word "element" was chosen.

What is a/n:

***Data point**:*
Data communicated with the PLC (flags, registers ..) or static data (e.g. electrical diagram number)

***Addendum**:*
Alarm, Trend, etc.

***Attribute**:*
Value, colour, size, position, diagram no., unit, etc.

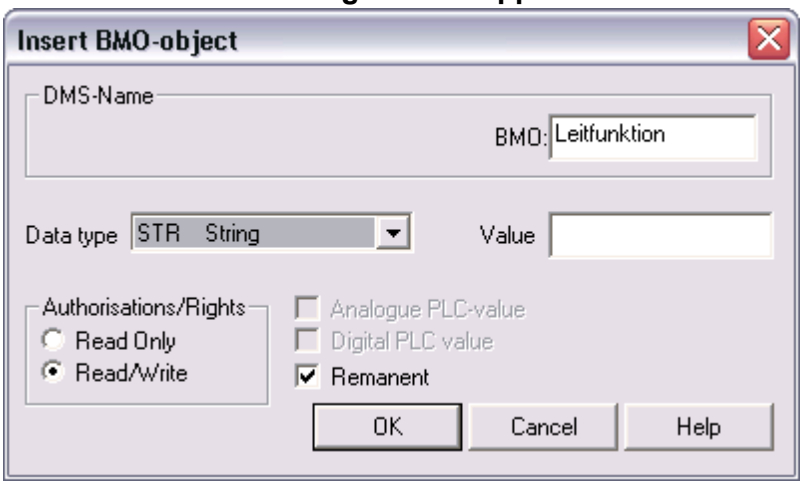
***Template object**:*
Templates for system objects (also called BMOs) consisting of data points, addendums and attributes

7.2.4.1 Insertion of DMS elements

Before a new element is inserted, it should be ensured that the correct element, superior to the one being inserted, is marked.

New element or in the menu, Edit > New Element or **<+>** key or **<Insert>** key.

The **Insert Element dialog window appears:**



DMS name
Shows the existing part of the selected DMS name. In the input field to the right, the DMS name must be completed with the name of the new element.

Data Type
The data type can be determined by clicking on the dropdown menu.

Value
A numerical value can be entered depending on the data type.

Rights
Enables **read**-only or **write and read rights** to be assigned.

Remanent
The contents of the element is only saved when the DMS is saved to the hard disk if the check-box is marked.

<OK> to adopt the settings.

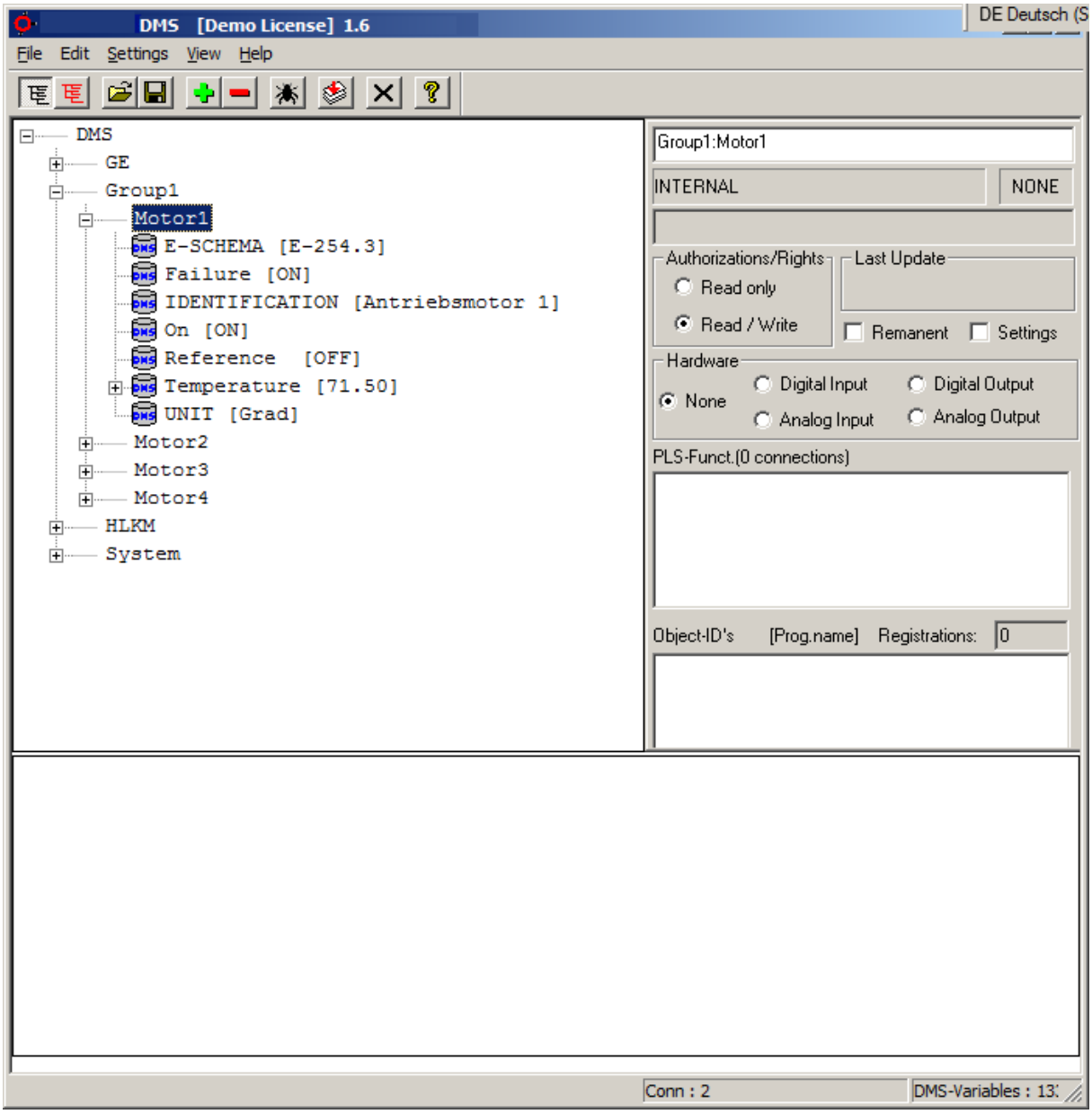
We recommended creating all template and system objects in the PET. This is also much more convenient.

If elements have been inserted in a system object that was derived from a template object, the inserted elements are lost as soon as something in the template object is altered. Any change to a template object has the consequence of automatically changing all system objects of the same type to match the template object!

7.2.4.2 Editing DMS elements

Left part of the window contents (tree structure)

Click on an element for the following explanations.



Right part of the window contents (with various fields)
The following changes can be made to the elements in the fields depending on the data type:

In the white input field that currently has the value "OFF", the value can be changed and accepted with **<ENTER>** at any time. The value from the DMS is applied and the display in the tree structure is updated (if the automatic update function is active). Digital values can be overwritten with 0 or 1 (ON and OFF respectively).

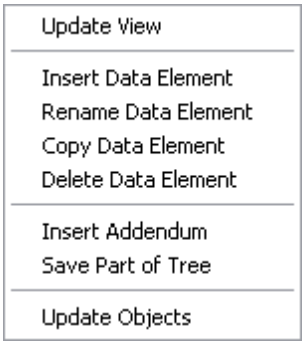
In the **Rights** field you can set whether the element may only be read or can also be written.

If the checkbox **Remanent** is marked, the value of the signal while saving the DMS to the hard disk is stored.

By switching to the template view, elements in the template objects can also be edited as described above. These changes affect all system objects created with this current

template object. In other words, they adjust themselves (via the context menu item "Update Objects").

Right mouse button
By pressing the right mouse button on a selected element, various commands can be accessed.



Description of the individual functions:

Update Display
Shows the current DMS values in the tree structure (update of values).

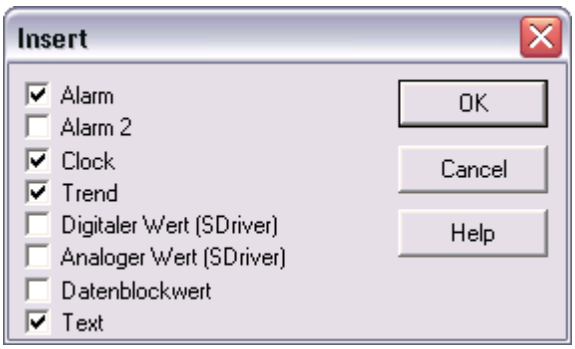
Insert Data Element
A new element is inserted. Elements are inserted at the marked position (in the subordinate "branch" of the tree).

Rename Data Element
The data point name can be changed. Under certain circumstances, this also has an influence on the process diagrams, the PLC program, etc. When an element is renamed, the diagrams and the PLC programs are automatically adjusted.

Copy Data Element
The complete tree under the active data point is copied (including all values).

Delete Data Element
The data element is deleted.

Add Attribute
Adds an attribute to the element, such as an alarm.



Various attributes can also be integrated into the element by selecting the respective checkbox.

The configuration for this dialog is saved in the `\InstallationPath\cfg\dms.add` file.

An attribute should never be added to a system object!
This should always be done in the template object. With any modification of the template, the system object ***loses the inserted attribute****!*

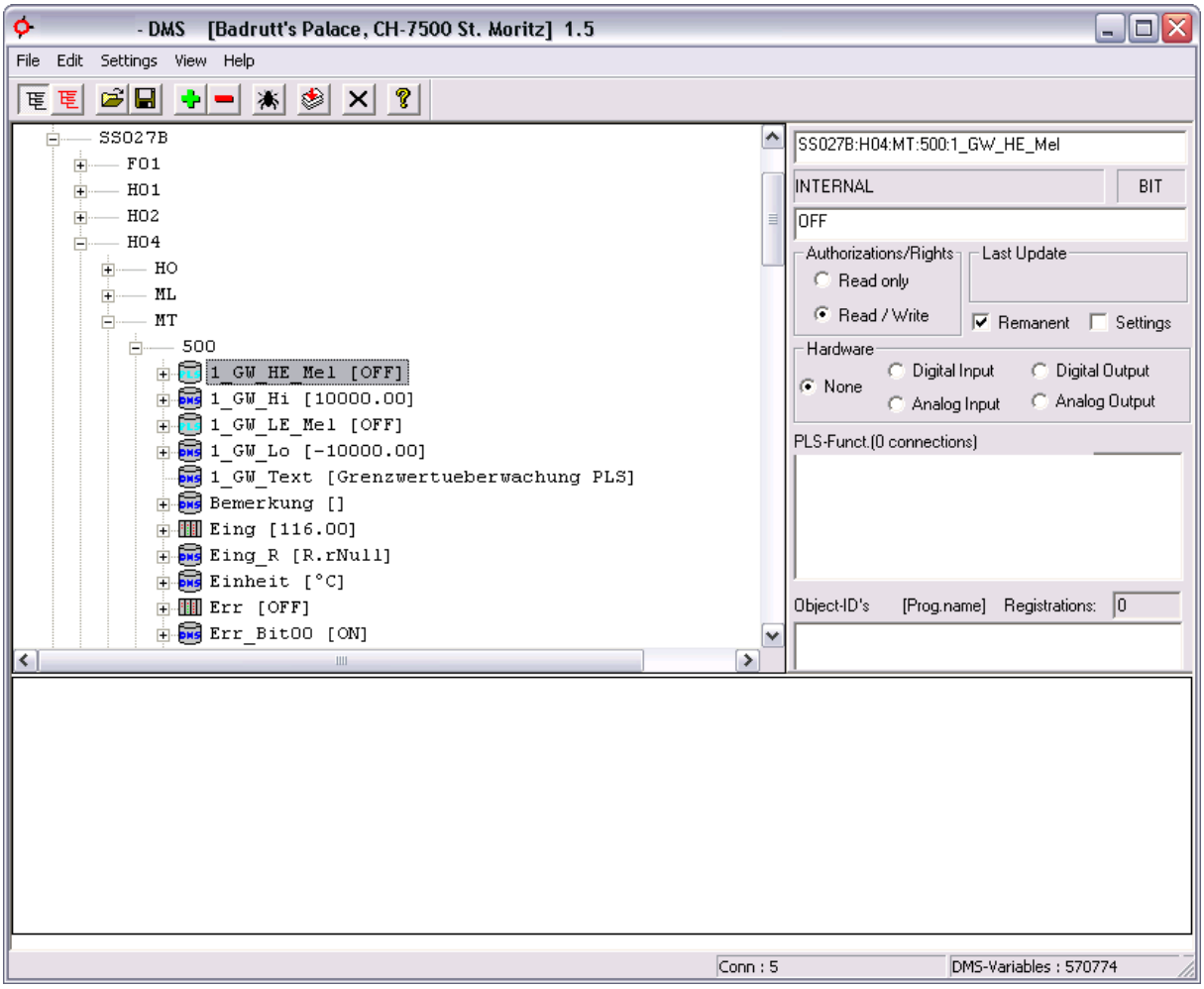
We recommend editing all template and system objects in the PET. This is much more convenient.

Any change to a template object has the consequence of automatically changing all system objects of the same type to match the template object.
This is recognisable when a change originating from a template object is made in a system object. The system object always adopts the status of the template object after a change to the template object.

In the Template Object view the ***"Update Objects"*** *function is available in addition to the functions described above. When this command is selected, all system objects are recompared with the VLO object and its attributes (insertion of new alarms in the VLO –> inheritance by all existing system objects).*

7.2.4.2.1 Delete DMS Elements

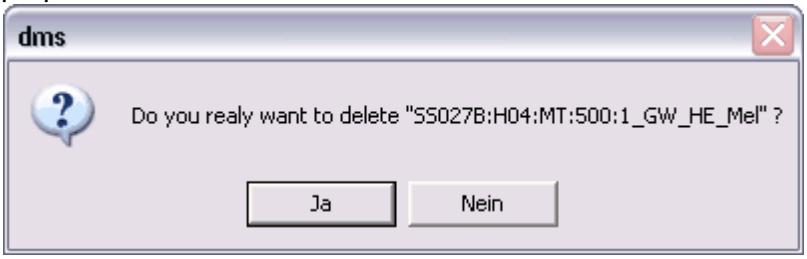
Select the element to be deleted with the **left mouse button**.



There are four available methods for deleting elements:

- Select the menu item **Edit -> Delete Data Element**
- **Right-click** and select **Delete Data Element**.
- **Press**  **Delete Data Element** in the toolbar.
- **Press the <DEL> key** on the keyboard.

A confirmation prompt showing the element to be deleted again is displayed for verification purposes:



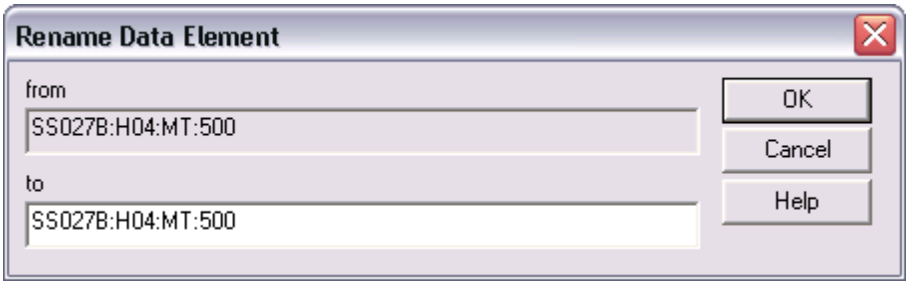
  *By switching to the template-object tree structure, it is also possible to delete*

elements in the template objects, as described above.
We recommend editing all template and system objects in the PET. This is much more convenient.

Any change to a template object has the consequence of automatically changing all system objects of the same type to match the template object. This is recognisable when a change originating from a template object is made in a system object. The system object always adopts the status of the template object after a change to the template object.

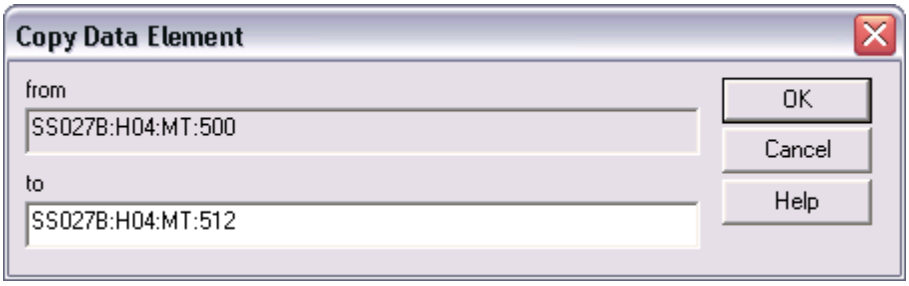
7.2.4.2.2 Rename Data Element

A data element on any level can be renamed. Renaming can have the consequence of changing not only the names in the DMS, but also the historical data on hard disk, the diagrams in the graphics system as well as the PLC programs that use this data point. You are asked by the program whether these dependencies should be changed automatically.



7.2.4.2.3 Copy Data Element

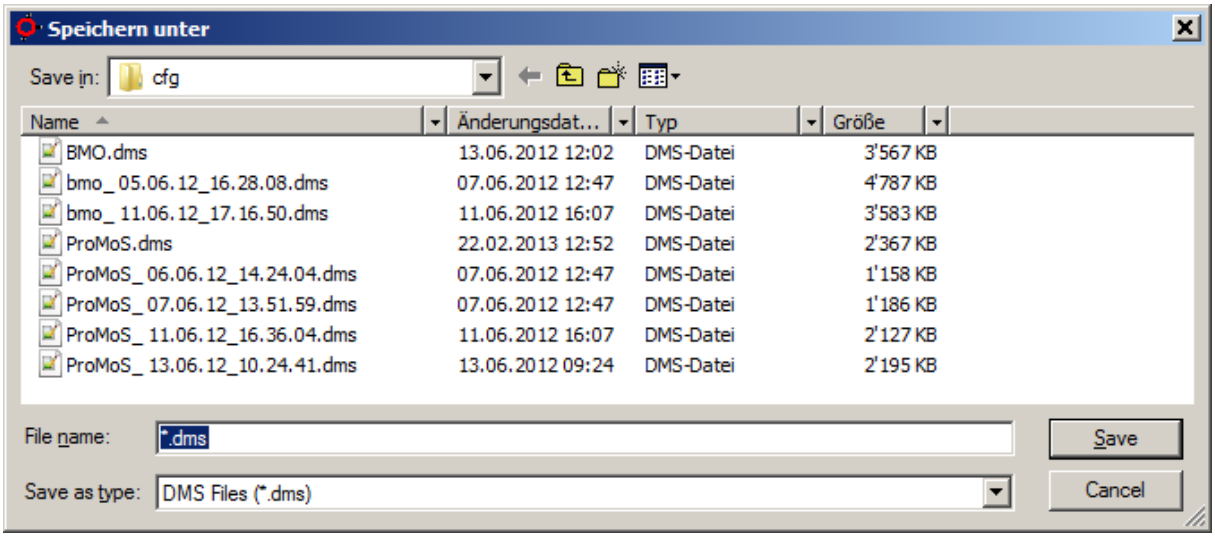
Complete "data trees" in the DMS can be copied. If, for example, you would like to create an entire system component twice, copy the respective data tree.



7.2.4.2.4 Export/Import Sub-Tree

Export
If you would like to use part of your project in another project, a sub-tree can be completely exported and imported into another DMS.

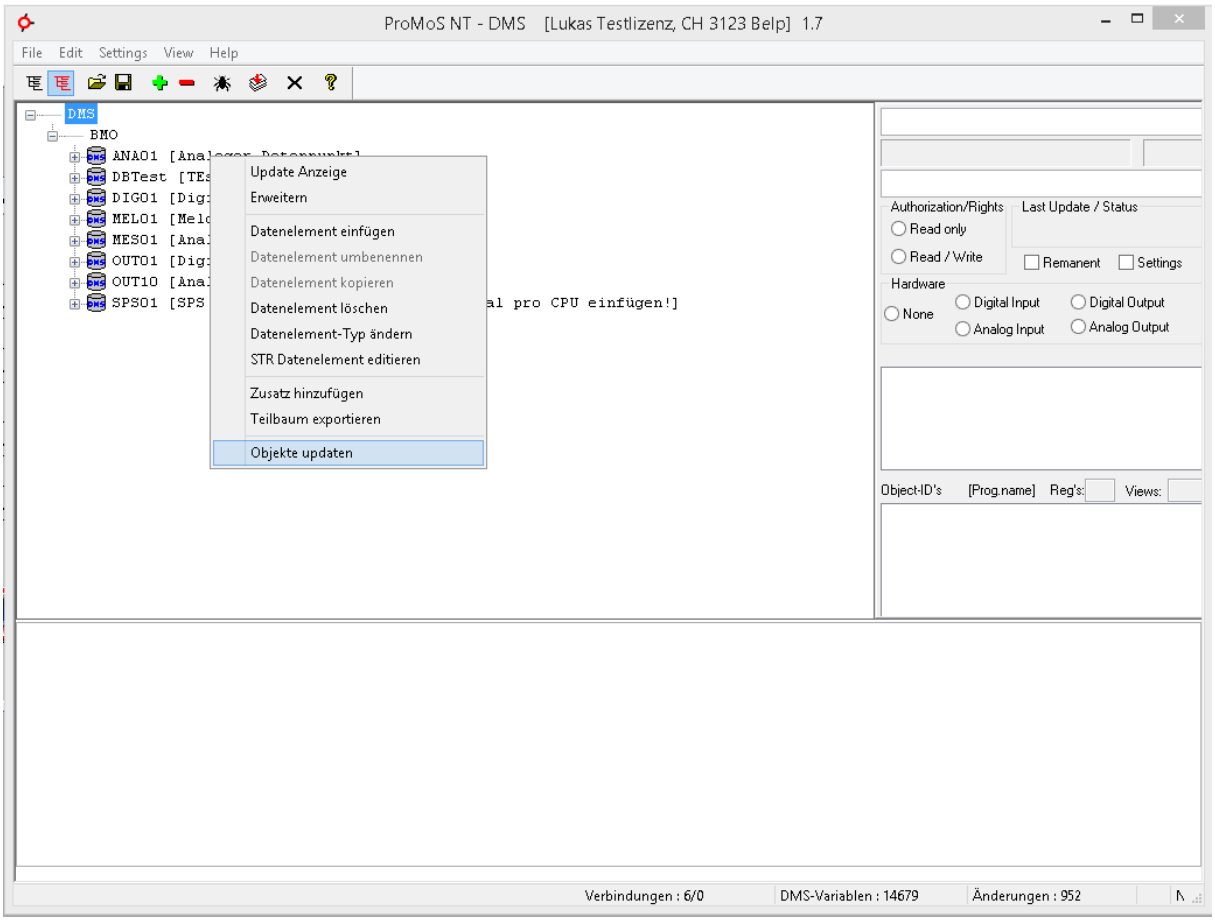
Select the section to be exported by choosing the top-most element in the sub-tree.



Import
The sub-tree is imported into another DMS by opening the exported file (menu: Open). In the process, any existing data points are overwritten.

7.2.4.2.5 Update Objects

If adaptations are made in the templates, all objects in the DMS can be automatically adjusted to the template via this menu item. To do this, right-click and open the context menu in the BMO tree and update the object.



The following rules must be observed when updating the template:

- New data points are added. This means that data points already deleted in the DMS for an object are also added.
- Deleted data points in the BMO tree are also deleted in the DMS.
- The content of the existing data types is not overwritten.
- The data type is not overwritten.
- If a data type is renamed in the template, this results in the deletion of the data point in the DMS when updating and the new one is added. The existing content is lost in the process and is adopted by the template. If the variable is used in a Fupla file, it is not automatically renamed there.

7.2.5 System data points

The DMS automatically manages a series of data points (they do not have to be defined). These data points are integrated into the "System" sub-tree.

These data points are subdivided into the following areas:

- Path settings (project, programs, backup, etc.)
- Alarm data (number of pending alarms, alarm history, etc.)
- Blinkers (data points whose value changes at intervals)
- Time Switch Program data
- Date and time in various formats
- Error information
- Licence data

- Information for connected graphical systems
- PC-related data (information about memory, disk, graphics, etc.)
- PCD system (information about PET settings, PG5 version, etc.)
- Error messages for control functions
- Status of the individual Visi.Plus modules
- User data (login name, rights, etc.)
- Versions (Visi.Plus modules)

These system data points are explained individually in the following.

System

Data point	Description
System:BackupPath	Data path through backup
System:BinPath	Data path via binary files and DLLs
System:HelpPath	Data path through help files
System:IniPath	Data path for the "promos.ini" configuration file
System:Pipes	Number of connections to DMS
System:Points	Number of data points in the DMS
System:Project	Current project path
System:Show DMSName	Auxiliary data points for showing and hiding DMS names
System:Text	Text storage for language versions

System:ALM

Information about alarms is found in this sub-tree:

Data point	Description
System:ALM:Count	Number of pending alarms
System:ALM:Enable	Activates/deactivates alarm logging
System:ALM:GRP:xxx	Number of alarms in group
System:ALM:GRP:xxx:GrpActiv	Activates/deactivates individual alarm groups
System:ALM:GRP:xxx:GrpQuit	Group alarm acknowledgement for all alarms in this group
System:ALM:Last	Last alarm (DMS name)
System:ALM:LastEntry	Last alarm (incoming plain text alarm)
System:ALM:LastPri	Last alarm (priority)
System:ALM:LastText	Last alarm (plain text alarm, in/out/ack.)
System:ALM:New Alarm	Number of unacknowledged alarms
System:ALM:Priority:PriX:APCount	Number of pending alarms with priority X
System:ALM:Priority:PriX:APNew Alarm	Number of new (not acknowledged) alarms with priority X
System:ALM:SOCKET_COMM	Socket alarm settings (see Alarm Manager)
System:ALM:SOCKET_COMM_NTP	NTP settings (see Alarm Manager)
System:ALM:View	AlarmViewer status

System:MALM

This sub-tree contains information about remote alarm processes:

Data point	Description
System:MALM:General:xxx	Information about defined types of notification (e-mail, SMS, ...)
System:MALM:Enable	Activates/deactivates remote alarms
System:MALM:Groups:xxx	Information about defined groups and their members
System:MALM:Prio:xxx	Configuration of individual priorities
System:MALM:Rcp	Configuration of individual users
System:MALM:ACK	User name (acknowledger) of the last acknowledged alarm
System:MALM:Status	Status reports of the last active alarm output type (e-mail, SMS, etc.)
System:MALM:TAPI	Set by the TAPIDriver with acknowledgment by telephone

System:Blinker

Auxiliary variables for automatically triggering events. These variables enable, for example, the intialisation of colour changes in the graphical system, which then change the status in the clock.
We recommend only using this blinker for test purposes, because the system load increases considerably if, for example, an alarm button is shown flashing. Since the data can also be displayed over a network, the network traffic is also multiplied, which can even cause the entire system to hang in extreme cases!

Data point	Description
System:Blink0.25	Value change every 0.25 sec.
System:Blink0.5	Value change every 0.5 sec.
System:Blink1.0	Value change every second
System:Blink10.0	Value change every 10 sec.
System:Blink60.0	Value change every 60 sec.
System:Blink600.0	Value change every 10 minutes

System:CLK

Data storage of the Time Switch Program. Holidays and special days are stored in this component.

System:Date

Data point	Description
System:Date	Current date (DD.MM.YY)
System:Date:DateLong	Current date (DD.MM.YYYY)
System:Date:Day	Current day of the month (1..31)
System:Date:Day of Week	Day of the week (Monday = 1)
System:Date:Month	Current month (1..12)
System:Date:Year	Current year (- 2000)

System:Driver

All PLC drivers store the data in this sub-tree. The data points of the SDriver are described in the following:

Data point	Description

System:Driver:SDriver:xxx	Channel name with current status of the driver (ON/OFF). Only correct if the driver is active.
System:Driver:SDriver:xxx:Activated	Starting and stopping the driver. The Activated data point should not be used in new systems!
System:Driver:SDriver:xxx:TpS	Number of telegrams per channel
System:Driver:SDriver:xxx:Update	The driver is stopped and restarted (reimporting the DMS). Effectively, the channel is just restarted.
System:Driver:SDriver:xxx:LastErr	Error code of the last (error) message
System:Driver:SDriver:xxx:LastMsg	Last (error) message
System:Driver:SDriver:xxx:StationXXX	Data of the PLC
System:Driver:...:StationXXX:CodeFreeSystem:Date:Year	Free memory of the PLC for code
System:Driver:...:StationXXX:CodeUsed	Used memory in the PLC
System:Driver:...:StationXXX:Firmware	Firmware version of the PLC
System:Driver:...:StationXXX:Modified	ON if the code was altered by a debugger.
System:Driver:...:StationXXX:ProgName	Name of the PLC program
System:Driver:...:StationXXX:TextFree	Free text memory on PLC

For more SDriver data points, see Status of channels and stations in the DMS in the chapter Communications drivers, The SDriver.

System:Error

The DMS stores the last error message in this tree.

Data point	Description
System:Error:LastError	Last DMS error message

System:HDLog

The SDriver (Saia) creates the operating commands for the HDLog readouts in this tree.

Data point	Description
System:HDLog:CHANNEL:UpdateHDA_CSV	Activates the readout of historical data from the PCD

System:Licence

Storage of the licence information.

Data point	Description
System:Licence:Copyright	Copyright of the product
System:Licence:Customer	Customer name (from the SW key)
System:License:CustomerAddress	Customer address (from the SW key)
System:Licence:CustomerCity	Location of the customer
System:License:DongleID	Unique ID of the hardware dongle
System:Licence:DriverPoints	Number of data points for the licence.
System:Licence:ID	Unique licence number
System:Licence:Location	Location of the end customer
System:Licence:ProductName	Product name
System:Licence:Type	Type of licence (demo, development or runtime)
System:Licence:User	End customer name
System:Licence:Prog:XXX	Enablement of the individual program modules

System:Node

The Graphical Editor saves the data in this sub-tree. An individual sub-tree is created for each PC (client).

Data point	Description
System:Node:XXX:Close	The current screen can be closed by setting the data point.
System:Node:XXX:GE_UP	Indicates whether a GE has been started.
System:Node:XXX:Image	Most recently accessed screen in the GE. Any screen can be opened by setting the data point. The filename of the screen must be written in the data point.
System:Node:XXX:ImgReInit	Current screen reinitialisation
System:Node:XXX:Layer:All	Activates/deactivates all defined layers in the GE
System:Node:XXX:Layer:LINK	Activates/deactivates the LINK layer in the GE
System:Node:XXX:Layer:LINKBOXEN	Activates/deactivates the LinkBox layer in the GE
System:Node:XXX:Layer:YYY	Activates/deactivates the layer YYY in the GE
System:Node:XXX:Preview	The print preview function is activated by setting the data point.
System:Node:XXX:Print	The Windows print dialog is displayed. The user can select a printer
System:Node:XXX:PrintAll	Prints out all screens (only works via the GE menu)
System:Node:XXX:PrintDirect	The current screen is printed out on the default printer.
System:Node:XXX:Runtime	If the value is 1, the editor is in runtime mode.
System:Node:XXX:Save	The current screen can be saved by setting the data point.
System:Node:XXX:Type	Indicates the type of GE

System:NT

All computer-specific data is stored in this tree. By doing so, it is possible to shut down the system, for example. In addition, system data such as memory size, disk space, video data, etc. is stored.

Data point	Description
System:NT:Computername	Name of the PC
System:NT:Disk	Data to the hard drive (without solid state disk)
System:NT:Disk:Drive C:Free	Percentage of free disk space. An alarm threshold is automatically defined for this data point (disk > 90%)
System:NT:Disk:Drive C:FreeSpace	Free disk space in KBytes
System:NT:Disk:Drive C:Quit	Acknowledgment of the disk space alarm
System:NT:Disk:Drive C:Serial	Serial number of the disk
System:NT:Disk:Drive C:Volume	Name of the disk

These data points are repeated for all local and network drives

Data point	Description
System:NT:Display:Colour	Number of colours that can be displayed
System:NT:Display:MaxResolution	Resolution of the graphics card
System:NT:Display:Resolution	Selected resolution
System:NT:DMS_Starttime	Starting date and time of the DMS
System:NT:DMS_InfMinuteCnt	Number of INFORM transactions per minute in the DMS

System:NT:DMS_RdMinuteCnt	Number of READ transactions per minute in the DMS
System:NT:DMS_WrtMinuteCnt	Number of WRITE transactions per minute in the DMS
System:NT:DMSlife	Internal DMS watchdog
System:NT:Hostname	Name of the PC
System:NT:Language	Selected language
System:NT:Memory:Free	Percentage of free RAM
System:NT:Memory:FreePageFile	Free storage space in swap file
System:NT:Memory:FreePhysicalMemory	Free memory space in KBytes
System:NT:Memory:FreeVirtual	Total free virtual memory (KB)
System:NT:Memory:TotalPageFile	Size of the swap file
System:NT:Memory:TotalPhysicalMemory	Size of the RAM
System:NT:Memory:TotalVirtual	Total available memory
System:NT:NoSave	Help data point for the program pStop.exe
System:NT:Perf:ALMMNG	System load of the AlarmMng
System:NT:Perf:DMS	System load of the DMS
System:NT:Perf:GE	System load of the GE
System:NT:Perf:HDAMNG	System load of the HDAMng
System:NT:Perf:PDBS	System load of the PDBS
System:NT:Perf:PRTMNG	System load of the PRTMng
System:NT:Perf:SDRIVER	System load of the of the SDriver
System:NT:Perf:SYSTEM	Overall system load

All performance values are saved as historical data (only if the corresponding option "Performance Trend Acquisition" is set in the Gen. Settings dialog box):

Data point	Description
System:NT:PrimaryIP	IP address of the PC
System:NT:ProcessorSpeed	Speed of the CPU
System:NT:Quit	The program can be exited by setting this data point.
System:NT:QuitWarn	The program can be exited by setting this data point (incl. alarm generation).
System:NT:Restart	The PC is restarted (reboot) by setting this data point.
System:NT:SaveDMS	The DMS is saved by setting this data point.
System:NT:ShutDown	The PC is shut down by setting this data point.
System:NT:Update	The internal DMS tree is rebuilt (tree structure).
System:NT:Username	Name of the PC user (Windows login)
System:NT:WatchDog	Auxiliary variable for monitoring all manager programs
System:NT:WindowsBuild	Internal Windows version number (build)
System:NT:WindowsVersion	Official Windows version number

System:PCD

The following signals are only present when working with the PG5 Development Tool. Details about the data points can be obtained from the PG5 documentation (Saia).

Data point	Description
System:PCD:PG5Version	PG5 version in use
System:PCD:PLCName:Error	Last error message
System:PCD:PLCName:ErrorNumber	Error number
System:PCD:PLCName:ErrorText	Error text
System:PCD:PLCName:GoOnline	Channel setting data
System:PCD:PLCName:IPModule	Ethernet card data (PLC)
System:PCD:PLCName:MediaAssignment	PLC data configuration
System:PCD:PLCName:Pcd	PLC internal data
System:PCD:PLCName:ProgramNames	Programs on the PLC
System:PCD:PLCName:SbusGatew ay	PLC gatew ay settings
System:PCD:PLCName:SbusPgu	PGU interface settings

System:PDBS

The PDBS stores the last error message in this tree.

Data point	Description
System:Error:Status:ErrorCode	Last error code from the PDBS
System:Error:Status:LastError	Last PDBS error message / status message

System:Prg

Data point	Description
System:Prg:LoadData System:Prg:SaveDef	Any number of DMS data points can be imported/exported (see the pmosFilePicker tool) by setting this data point (filename of the data to be imported). Filename of the data to be exported
System:Prg:PCD:CPUName:Addresses	Addresses for the PET (automatic resource assignment)
System:Prg:PCD:CPUName:Filter	Possible filter settings (PET)
System:Prg:PCD:CPUName:PetCode	Details for code generation (PET)
System:Prg:Count	Number of compilable control functions (PLS)
System:Prg:Error	If an error occurs w hile compiling control functions, it is stored here. In the case of multiple errors, only the most recent is mentioned in this system data point.

System:Prog

Indicates the current status of the individual program modules

Data point	Description
System:Prog:XXX	XXX = program module (ON = is active, OFF = not active)
System:Prog:GE:YYY:UP	YYY = PC name (ON = is active)
System:Prog:GE:YYY:LastError	Last error message of the GE
System:Prog:HDA:QueXX	Data of the internal queues (buffers)
System:Prog:HDA:QueSize	Size of the internal queues (buffers)
System:Prog:WatchDog:DMSPeriod	Watchdog signal period of the DMS in seconds

System:Prog:WatchDog:xxxMNG	Status of the manager program (Alm, Hda, etc.)
System:Prog:WatchDog:LifePeriod	Lifetime period for all programs
System:Prog:WatchDog:Period	Watchdog period for all programs
System:Prog:WatchDog:Timeout	Watchdog timeout time

System:PRT

The PrtMng stores its data points in this tree.

Data point	Description
System:PRT:Count	Number of PRT data points (logged data points)
System:PRT:Enable	Activates/deactivates logging
System:PRT:Last	Last log entry (if no History is activated)
System:PRT:History:XXX:00	Last log entry (XXX= log name)
System:PRT:History:XXX:00:Name	DMS name of the last logged data point

System:Text

The GE uses language dependent texts from this area (must be defined in Engineering).

Data point	Description
System:Text:FRENCH	All texts are in French
System:Text:GERMAN	All texts are in German

System:Time

System data points for the current time indication:

Data point	Description
System:Time	Time in the format HH:MM:SS
System:Time:HHMMSS	Time in floating point format
System:Time:Hours	Hours
System:Time:Minutes	Minutes
System:Time:Seconds	Seconds

System:User

All user information (logins and rights) is stored in this sub-tree. Please note that the data points always depend on the PC where login took place. The PC name is always automatically exchanged for the current PC name for initialisations.

Data point	Description
System:User:Name	Is no longer used (still included for compatibility reasons)
System:User:PCName:Admin	Is set if the user has administrator rights (editing in the GE)
System:User:PCName:AutoLogin	A user can be automatically logged into the system. If this user is active, the data point is set.
System:User:PCName:AutoLogout	If the data point is set, the user is automatically logged out after a specified time.
System:User:PCName:Comment	Comments for users (see pUser program)
System:User:PCName:FTP	The current user has access to the FTP server
System:User:PCName:HTTP	The current user has access to the WebServer
System:User:PCName:Name	Name of the logged-in user

System:User:PCName:Rights1	Access levels Level01 .. Level16
System:User:PCName:Rights2	Access levels to systems Level01 .. Level08
System:User:PCName:UserAdmin	Access through user administration

System:ActUser
This sub-tree stores the data of the user currently logged in (login time and PC name/IP adr.). <no.> is a counter running from 1 up to the max. number of users which can be displayed. This information is defined in the settings, see Login monitoring. The data where <no.> = 1 concerns the user who has been logged in for the longest; the higher the <no.>, the more recent the login time.

Data point	Description
System:ActUser:User<no.>	Name of the user logged in.
System:ActUser:User<no.>:LoginTime	Login time, e.g.: "15.10.09 16:42:16"
System:ActUser:User<no.>:PC	The user's PC name, e.g.: "NODE3", or when logging into the WebServer: "10.0.0.63 - 011D811187Aa"

This data is listed separately in the DMS for each PC.

System:Version
All version information for the programs of Visi.Plus is found in this tree.

7.2.6 Control functions

7.2.6.1 General

So-called control functions can be integrated into the DMS. Control functions in Visi.Plus can compare values, calculate and influence elements in the controllers (flags, registers, etc.).

That also means that Visi.Plus can assume control functions. Caution: if the computer is switched off or fails, these control functions can no longer be executed. That means that, under some circumstances, the controllers will no longer receive important data from Visi.Plus.

There are two different types of control functions. These are control functions which require compiling and those which require interpreting. Compiling functions are directly executable by Visi.Plus after one-time compiling. Interpreting control functions are always re-interpreted. Interpreting control functions need more processing power from the PC and are therefore significantly slower.

Note: Control functions cannot process endless recursions (standard: 40 levels).

System data points
- The total number of control functions is written to "**System:Prg:Count**".
- If an error occurs during the compilation of control functions, it is recorded in "**System:Prg:Error**". In the case of multiple errors, only the most recent is mentioned in this system data point.

Computational accuracy

The accuracy of calculation of float data type (FLT) is limited to 6 decimal places.

Example:
1. **MUL(Parm1,Parm2)**, where Parm1=0.001 and Parm2=4444.01. The result of type FLT equals 4.44402 and this value change is detected by DMS.
2. If Parm1=0.00001 and Parm2=4444.01, then the result=0.00444402; but DMS detects only 0.004444.

7.2.6.2 Settings and troubleshooting

Troubleshooting

All control function compilation and execution errors are held in a separate log file "**DMSErrorPLS.log**" in the project log directory.

 Control functions which cause a compilation error are normally reported (beep and/or error display) and not inserted into the DMS structure, so that they cannot be executed.

See also the chapter "Control functions, settings and troubleshooting" for more details.

Debugging

When the debug option is enabled, a trace log of all control functions called is written to the "**DMSErrorPLS.log**" log file.
As an additional option, the result as well as the parameter values before and after can also be logged.

Also see the chapter "Debugging control functions" for more details.

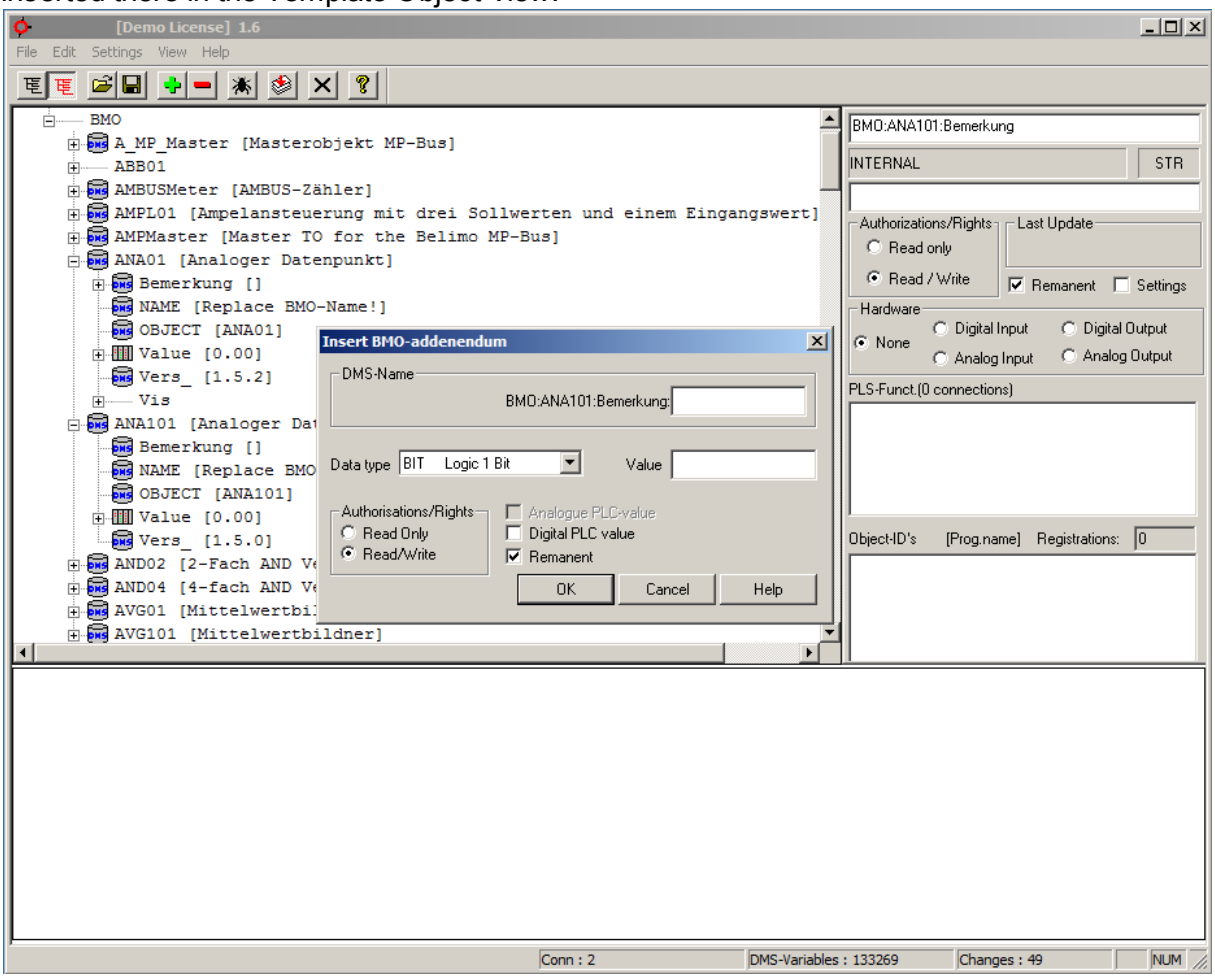
7.2.6.3 Compiled control functions

In the following, the **control function** is shown by Visi.Plus using an addition.
The goal is to add the **Actual value**, **Target value** and **Offset value** data originating from a PLC and to place it in the DMS element **Result** (not in the PLC).

The keyword for control functions is:**PRG**
All addenda with this name are interpreted as control functions.

 *Then the result could only be shown in the GE (Graphical Editor). This method can be used for data which is only necessary for display purposes. For example, this might be hours of operation of various pumps, etc. If Visi.Plus does not run (hard disk change, etc.), this is no longer a problem, because the PLC should process the most important current data and not a PC. The PLC should not be dependent on a visualisation program in order for a system or machine to function.*

In order to install a compiled control function in the DMS, a new template object must be inserted there in the Template Object View.

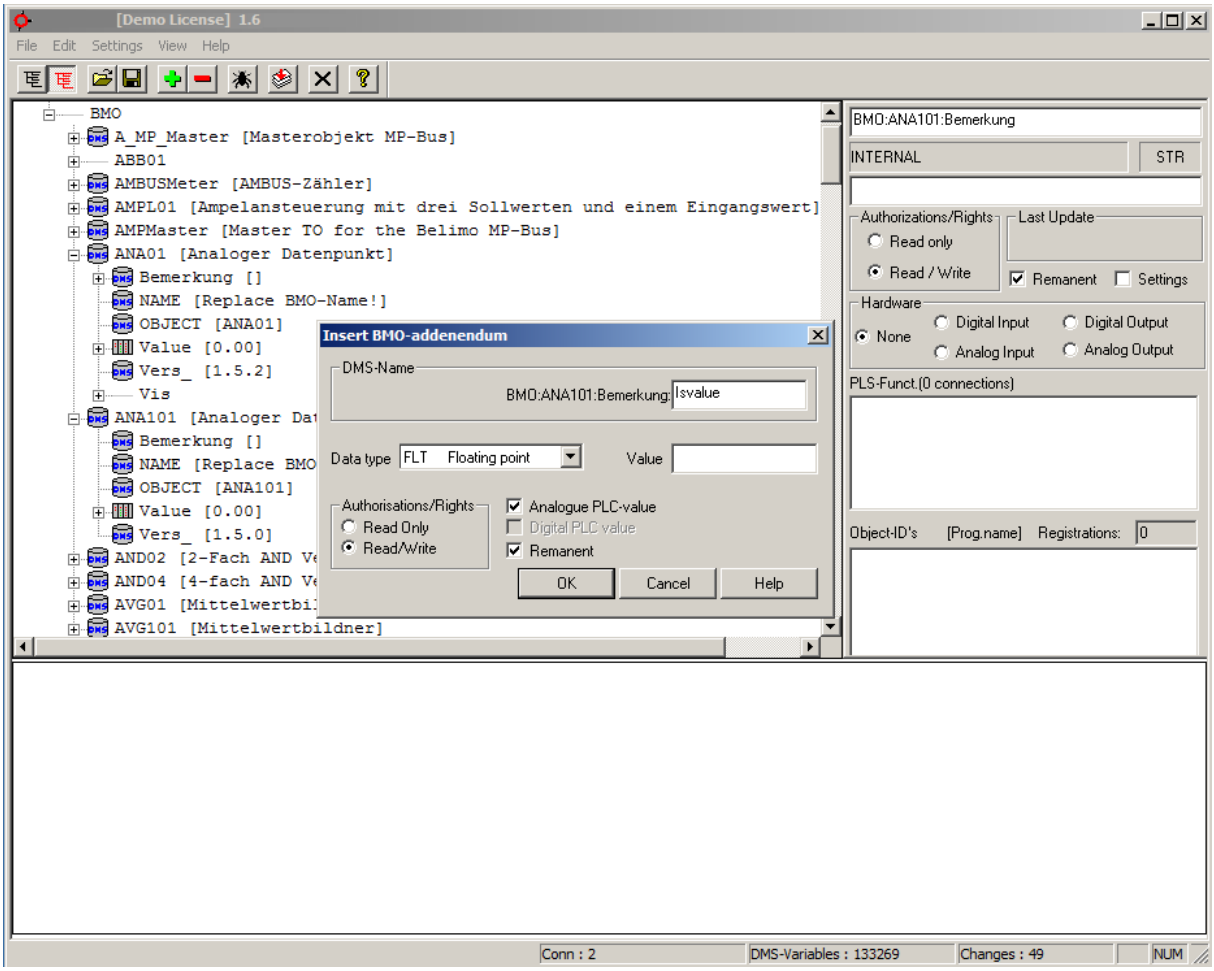


In the preceding image, a new template object with the name **Control function** (may also be another name) of the data type **STR string** was just inserted.
Through the creation of this new template object, the elements **NAME** and **OBJECT** were automatically inserted by the DMS. The element **OBJECT** adopts the assigned template object name **Control function**. In the element **NAME** it says **Replace BMO name!**

As soon as the creation of a template object is completed, it can be used in the PET for defining system objects like pumps, valves, sensors, etc.
*If, for example, a "SensorNorth" is created for the system in the **PET**, it must be assigned a template object, e.g. **MES01**. This is an example of "inheriting". That means the system object "SensorNorth" now receives all the attributes of the template object **MES01**.*
*Through inheriting, the entry of the data point **NAME** (content thus far is **Replace BMO name**) is replaced by the DMS name of the system object. The data point **NAME** is immediately given the DMS name **"SensorNorth"** from the system object.*

Next, the template object attributes **Actual value**, **Target value**, and **Offset** should be added to the new template object. Before this can take place, it is essential that the **Control function** template object be marked.

*If the cursor constantly jumps back to the original position after inserting an attribute (data point) (in this example **Control function**), make the following settings:*
*Select **Settings > General > Return to Output in the Tree upon Insertion > Activate** from the menu.*



The template object with the name **Control function** is marked (grey display column BMO:Control function)!

After selecting the command "**Insert Data Element**" via the right mouse button, the window **Insert Operating Resource Attribute** appears. The field **DMS Name** contains part of the DMS name **BMO: Control function** to the left of the input field. This must now be completed by entering the name **Actual value** in the input field.

*The word **BMO** (Operating Resource Object) is a so-called placeholder, which will be replaced later by the name of the **system object** (PumpXY, ValveXY, etc.).*

After this the attributes of the new element are defined.

Attribute	Description
Data Type	FLT floating point number
Value	blank (an initialisation value could be entered here, corresponding to the data type)

Attribute	Description
Rights	read/w rite
Analogue PLC value	yes
Digital PLC value	no (can no longer be selected because it is already predetermined by the data type)
Remanent	yes (the content of the element is saved w hile saving the DMS to the hard disk)

In the **template object attributes actual value, target value** and **offset**, the DMS has automatically inserted the data points **Comment** and **PLC**. A comment regarding the element can be entered in the data point **Comment**.
The data point **PLC** was created because the checkbox **Analogue PLC value** was marked and contains all necessary parameters for communication with the PLC.

The additional template object elements **Target value** and **Offset** are, as just described, also completed with the same attributes.

*Make sure that the **Control function** template object is marked before insertion!*

Then insert the element **Result** with the following data under control function:

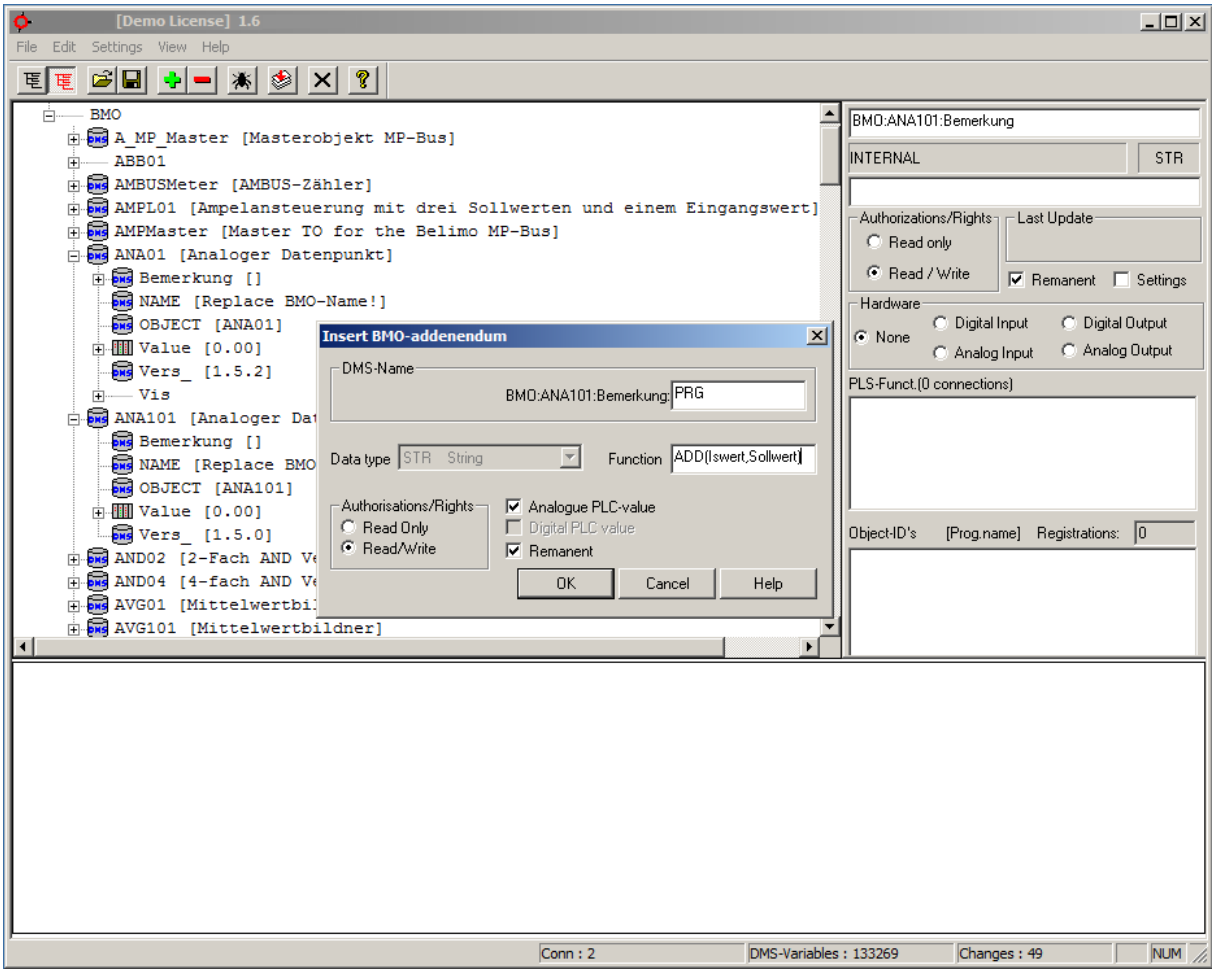
Assigned attribute	Description
DMS name	BMO:Control function:**Result**
Data Type	FLT floating point number
Value	blank (an initialisation value could be entered here, corresponding to the data type)
Rights	Read/w rite
Analogue PLC value	no
Digital PLC value	no
Remanent	yes

The DMS should now appear as shown below.
For the next step, mark the element **Result**!
The control function **ADD** should inserted in this element. To do so, click the right mouse button and select **Insert Element**. The dialog window that appears, **Insert Element**, is then to be filled out as pictured below:

In the window **Insert Element** you can see that the DMS name has been completed with the attribute **Result**. In the input field to the right of this the element **PRG** is now attached.

At this point in time, the DMS recognises that a control function is installed. The **data type** can no longer be selected.

*In order to install a control function, always enter **PRG** as the name in the DMS Name input field.*

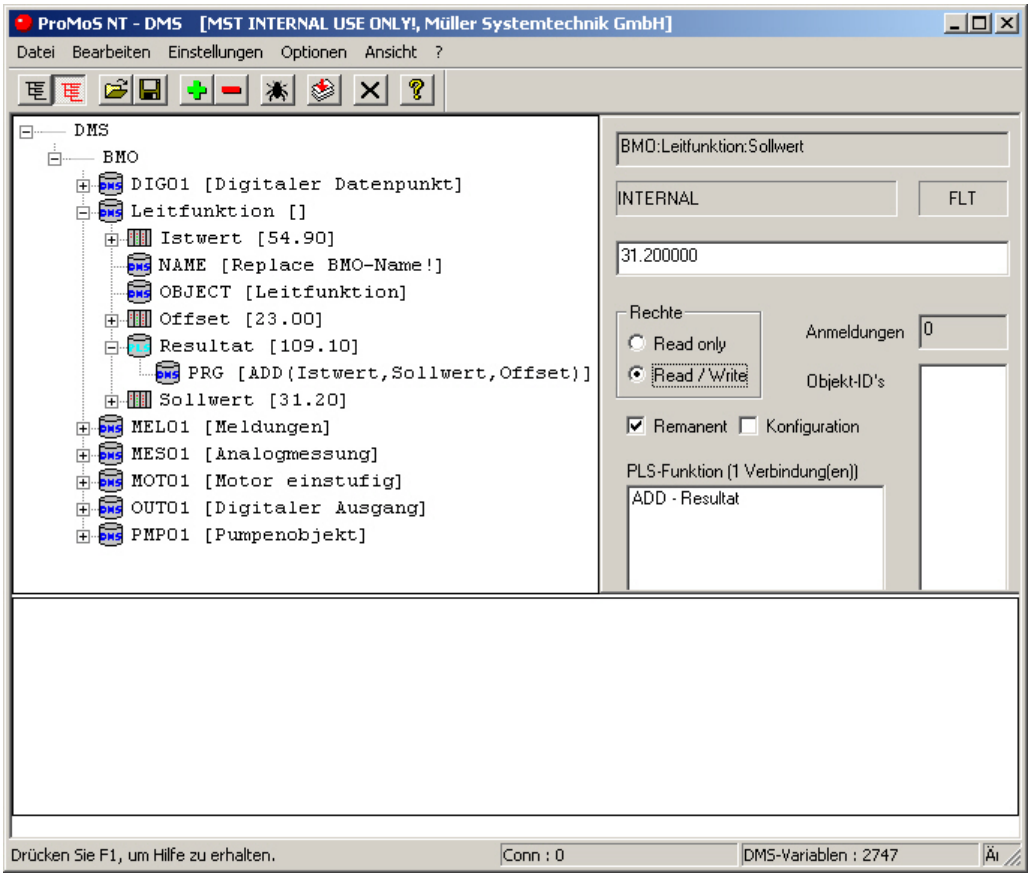


In the **Function** field, add the following note:

ADD (ActualValue,TargetValue,Offset)

By clicking on the **<OK>** button, the control function is installed in the template object element **Result**.

 Finally, the added control function must be compiled. This is carried out by clicking on the **Compile** icon in the toolbar, pressing the **<F2>** key or selecting **Settings > Compile Function** in the menu and then **Settings > Execute Functions**.



Testing a newly created control function:
By clicking on the respective element **Actual value**, **Target value** and **Offset** under the template object (BMO) **Control function**, a value can be entered in the input field shown in the preceding image (where it says "31.200000").
The element **Result** calculates the total of the elements **Actual value**, **Target value** and **Offset** (the result is shown after a delay, according to the value entered under **Update interval(s)** in the menu **Settings > General**).

Several control functions on the same data point

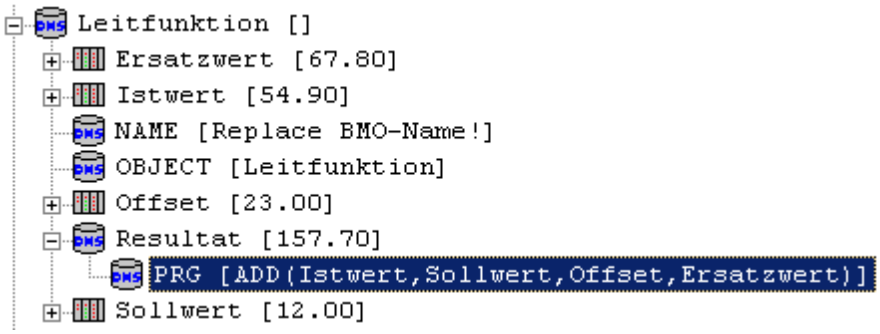
There can be several control functions (PRG) attached to a single data point. In this case, the keyword PRG must be numbered consecutively:

```
PRG
PRG1
PRG2
```
etc.

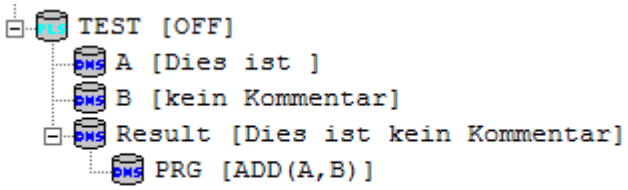
7.2.6.3.1 Addition (ADD)

Attribute	Description
Function	Adds up the indicated values
Supported data types	All data types except BIT and STR
Maximum number of parameters	4
Syntax	ADD

Attribute	Description
Example	ADD (ActualValue,TargetValue,Offset,DefaultValue)

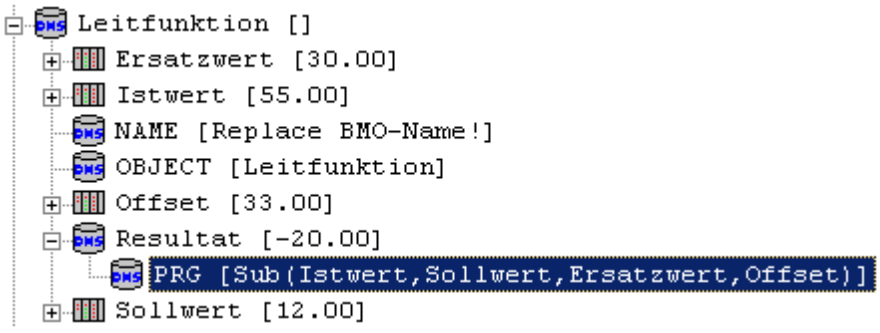


Example for strings:



7.2.6.3.2 Subtraction (SUB)

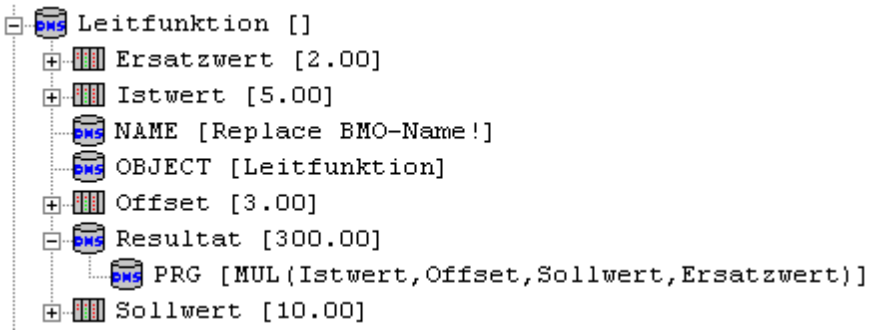
Attribute	Description
Function	Subtracts the indicated values from one another
Supported data types	All data types except BIT and STR
Maximum number of parameters	4
Syntax	SUB
Example	SUB(ActualValue,TargetValue,DefaultValue,Offset)



7.2.6.3.3 Multiplication (MUL)

Attribute	Description
Function	Multiplies the indicated values together
Supported data types	All data types except BIT and STR
Maximum number of parameters	4
Syntax	MUL

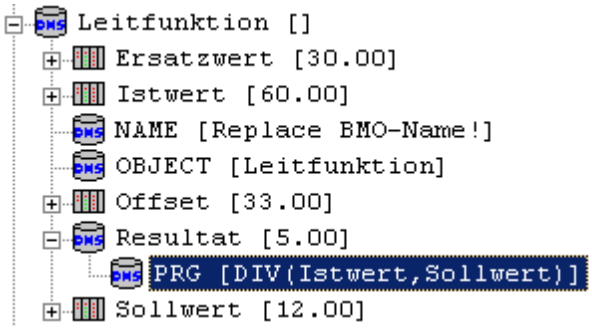
Attribute	Description
Example	MUL(ActualValue,Offset,TargetValue,DefaultValue)



The parameters TargetValue and DefaultValue are only taken into account for the function data type **FLT**.

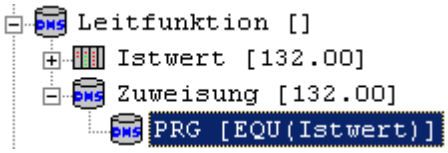
7.2.6.3.4 Division (DIV)

Attribute	Description
Function	Divides 2 values
Supported data types	All data types except BIT and STR
Maximum number of parameters	2
Syntax	DIV
Example	DIV(ActualValue,TargetValue,Offset,DefaultValue)



7.2.6.3.5 Equate (EQU)

Attribute	Description
Function	Assignment of one DMS value to another
Supported data types	All data types
Maximum number of parameters	1
Syntax	EQU
Example	EQU(ActualValue)

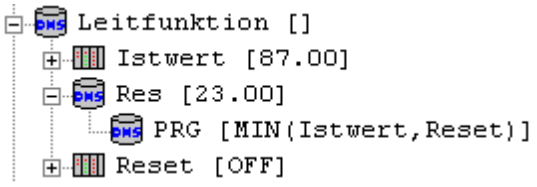


If the value of data point **ActualValue** changes, it will also be applied in the data point **Assignment**.

If the data types of the parameter ActualValue *and the result do not match, the parameter data type is converted automatically.*

7.2.6.3.6 Minimum (MIN)

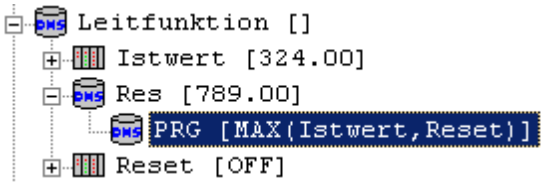
Attribute	Description
Function	Minimum value processing w ith reset function (trailing pointer)
Supported data types	All data types except for STR
Maximum number of parameters	2
Syntax	MIN
Example	MIN(ActualValue,Reset)



In the data point **Res** the smallest occurring value of the data point **ActualValue** is adopted, as long as the data point **Reset** is not set.

7.2.6.3.7 Maximum (MAX)

Attribute	Description
Function	Maximum value processing w ith reset function (trailing pointer)
Supported data types	All data types except for STR
Maximum number of parameters	2
Syntax	MAX
Example	MAX(ActualValue,Reset)

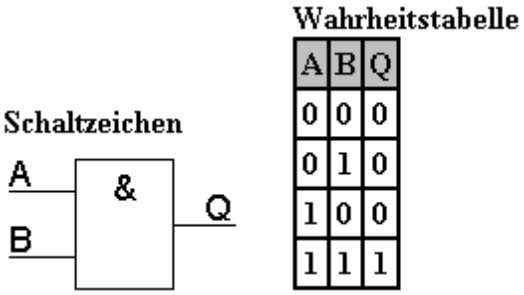


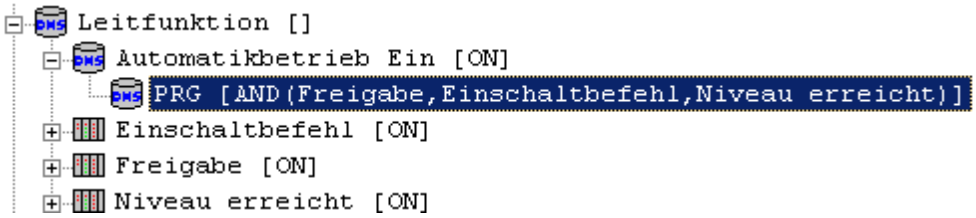
In the data point **Res** the largest occurring value of the data point **ActualValue** is adopted, as long as the data point **Reset** is not set.

7.2.6.3.8 And (AND)

Attribute	Description
Function	Logical AND function

Attribute	Description
Supported data types	Only BIT
Maximum number of parameters	4
Syntax	AND
Example	AND (Release,Switch on command,Level reached)

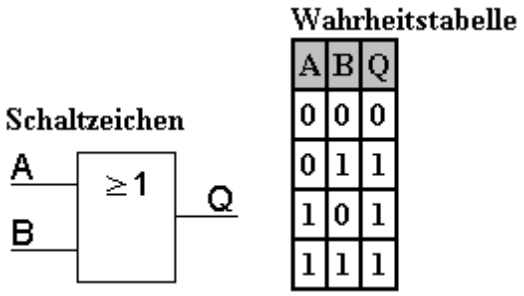


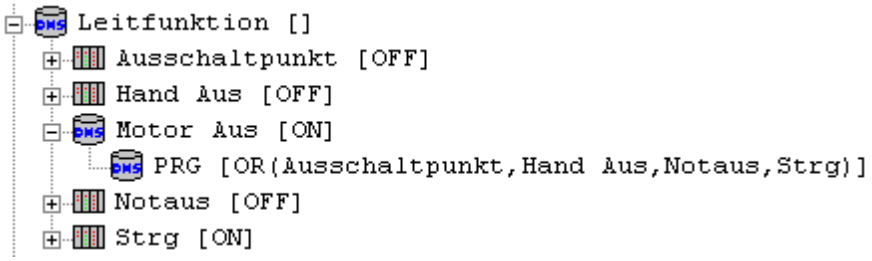


The data point **Automatic operation ON** is only set if the data points **Release, Switch on command** and **Level reached** are all also set.

7.2.6.3.9 Or (OR)

Attribute	Description
Function	Logical OR function
Supported data types	Only BIT
Maximum number of parameters	4
Syntax	OR
Example	OR(TrippingPoint, Manual OFF, Emergency OFF, CTRL)

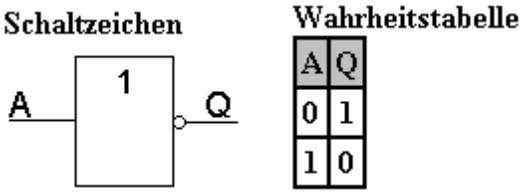


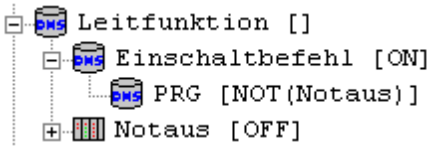


The data point **Motor OFF** is set as soon as one of the data points **TrippingPoint**, **ManualOFF**, **EmergencyOFF** or**CTRL** is set.

7.2.6.3.10 Not (NOT)

Attribute	Description
Function	Logical NOT function
Supported data types	All data types
Maximum number of parameters	1
Syntax	NOT
Example	NOT(Emergency OFF)





The data point **Switch-on command** remains set until the data point **EmergencyOFF** is set.

For parameter types BYS, BYU, WOS, WOU, DWS, DWU and FLT, please note:

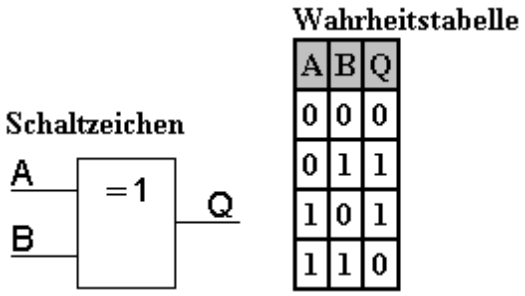
- If the parameter is zero, the result is TRUE (and vice-versa).

For parameter type STR, please note:

- If the string is empty, the result is TRUE (and vice-versa).

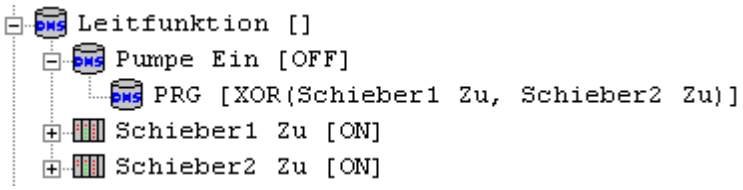
7.2.6.3.11 Exclusive Or (XOR)

Attribute	Description
Function	Logical XOR function
Supported data types	Only BIT
Maximum number of parameters	2
Syntax	XOR
Example	XOR (Gate1 Closed,Gate2 Closed)



Wahrheitstabelle

A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0



The data point Pump On is set as long as either the data point Gate1 Closed or the data point Gate2 Closed is set. If both data points (Gate1 Closed, Gate 2 Closed) are logic 0, the data point Pump On is reset.

7.2.6.3.12 Program Start (PRG)

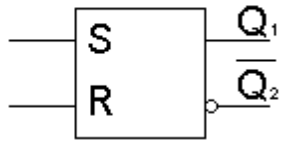
Attribute	Description
Function	Starting of programs (only positive edge)
Supported data types	Only BIT
Maximum number of parameters	1(program name including complete path statement)
Syntax	PRG
Example	PRG(c:\winnt\system32\calc.exe)

As soon as the **Computer** bit is set, the Windows computer starts up.

7.2.6.3.13 RS FlipFlop (RSF)

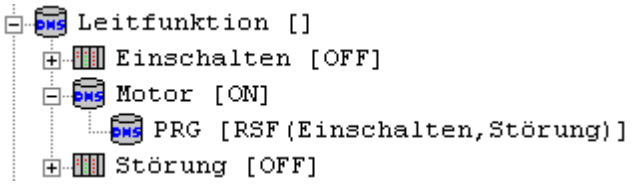
Attribute	Description
Function	RS FlipFlop (S = Set, R = Reset)
Supported data types	Only BIT
Maximum number of parameters	2
Syntax	RSF
Example	RSF(Switch on,Error)

Schaltzeichen



Wahrheitstabelle

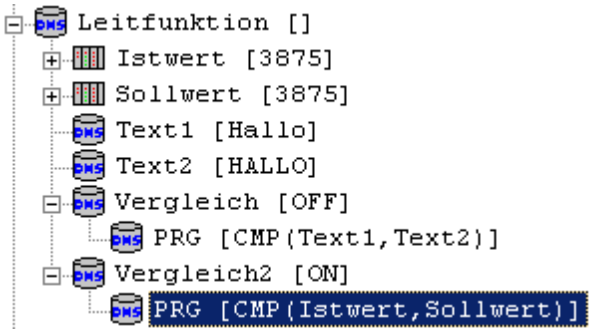
S	R	Q_1	Q_2	Zustand
1	0	1	0	Setzen
0	0	x	x	Speichern
0	1	0	1	Rücksetzen
1	1	x	x	Unbestimmt



With the occurrence of an impulse at the data point **Switch on**, the data point **Motor** is set. With the occurrence of an impulse at the data point **Error**, the data point **Motor** is reset.

7.2.6.3.14 Compare (CMP)

Attribute	Description
Function	Compares integers or strings for parity
Supported data types	All data types
Maximum number of parameters	2
Syntax	CMP
Example	CMP(Text1,Text2)



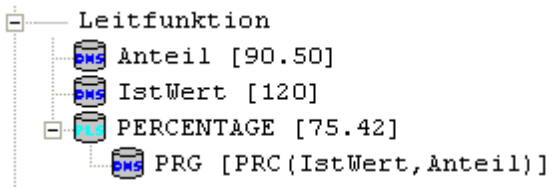
As shown in the preceding image, the control function is case-sensitive, meaning that upper/ lower-case characters are respected. If the data point **Text2** had the content **Hello,** the data point **Compare** would be set (to OFF here, because Text1 and Text2 are not identical). The data point **Compare2** is set, because the data points **ActualValue** and **TargetValue** exhibit identical values.

Caution with FLT values. A comparison is not recommended (better > or <).

If the data types of the parameters ActualValue *and* TargetValue *do not match, the parameter data types are converted automatically.*

7.2.6.3.15 Percent calculation (PRC)

Attribute	Description
Function	Calculates the percentage of the share (Param2) relative to the base value (Param1)
Supported data types	All data types except BIT and STR
Maximum number of parameters	2
Syntax	PRC
Example	PRC(BaseValue,share)

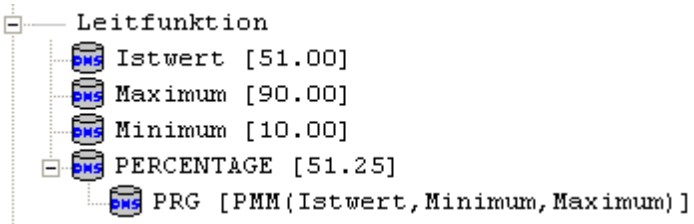


The data point **PERCENTAGE** shows a percentage of 75.42% for the data point **Share** relative to **ActualValue**.

The input parameters do not have to be of the same type (automatic conversion in FLT).

7.2.6.3.16 Percent calculations betw een Minimum and Maximum (PMM)

Attribute	Description
Function	Calculates the position of the actual value (Param 1) betw een Minimum (Param 2) and Maximum (Param 3). The result is a percentage.
Supported data types	All data types except BIT and STR
Number of parameters	3
Syntax	PMM
Example	PMM(ActualValue,Minimum,Maximum)

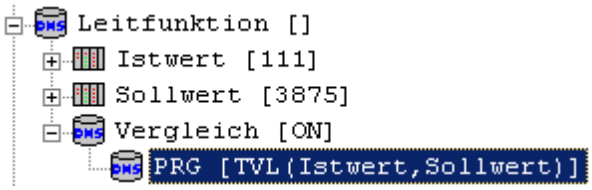


The data point **PERCENTAGE** shows a position of 51.25% for the data point **ActualValue** between **Minimum** and **Maximum**.

The input parameters do not have to be of the same type (automatic conversion in FLT).

7.2.6.3.17 Threshold Value Low (TVL)

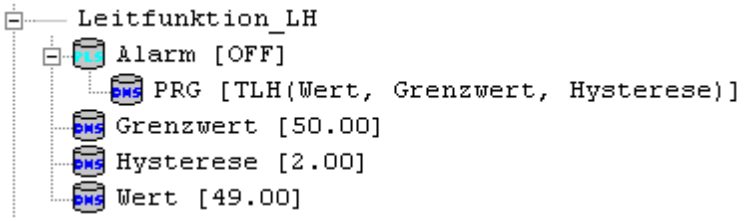
Attribute	Description
Function	Compares w hether the first parameter is smaller than the second.
Supported data types	All except BIT, STR
Maximum number of parameters	2
Syntax	TVL
Example	TVL(ActualValue,TargetValue)



The data point **Compare** is set as soon as the data point **ActualValue** exhibits a smaller value than the data point **TargetValue**.

7.2.6.3.18 Low hysteresis threshold (TLH)

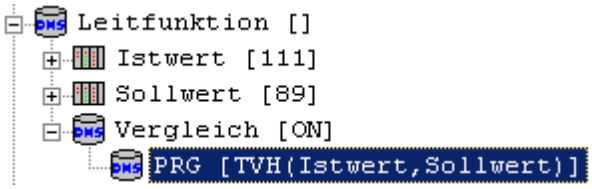
Attribute	Description
Function	Compares w hether the first parameter is smaller than the second. The output size is only reset if the 1st parameter > the sum (2nd param. + 3. param = Hysteresis).
Supported data types	All except BIT, STR
Maximum number of parameters	3
Syntax	TLH
Example	TLH(Value, Threshold, Hysteresis)



The data point **Alarm** is set as soon as the data point **Value** exhibits a smaller value than the data point **Threshold**. The data point **Alarm** is only reset when the value is greater than the **Threshold** + **Hysteresis**.

7.2.6.3.19 Threshold Value High (TVH)

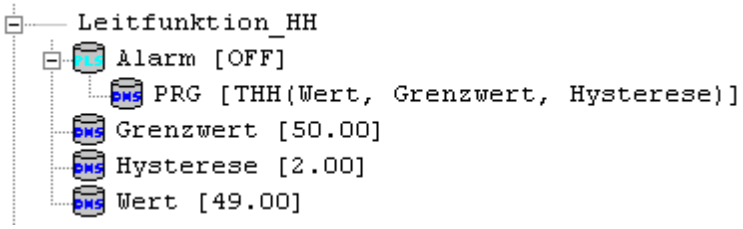
Attribute	Description
Function	Compares w hether the first parameter is larger than the second.
Supported data types	All except BIT, STR
Maximum number of parameters	2
Syntax	TVH
Example	TVH(ActualValue,TargetValue)



The data point **Compare** is set as soon as the data point **ActualValue** exhibits a greater value than the data point **TargetValue**.

7.2.6.3.20 High hysteresis threshold (THH)

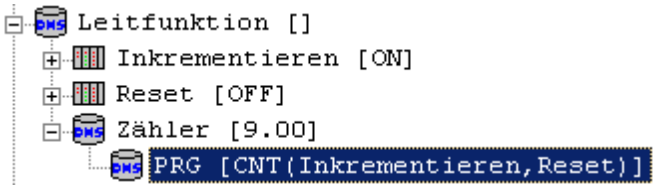
Attribute	Description
Function	Compares whether the first parameter is larger than the second. The output size is only reset if the 1st parameter < the difference (2nd param. - 3. param = Hysteresis).
Supported data types	All except BIT, STR
Maximum number of parameters	3
Syntax	THH
Example	THH(Value, Threshold, Hysteresis)



The data point **Alarm** is set as soon as the data point **Value** exhibits a greater value than the data point **Threshold**. The data point **Alarm** is reset as soon as the data point **Value** exhibits a smaller value than **Threshold** - **Hysteresis**.

7.2.6.3.21 Counter (CNT)

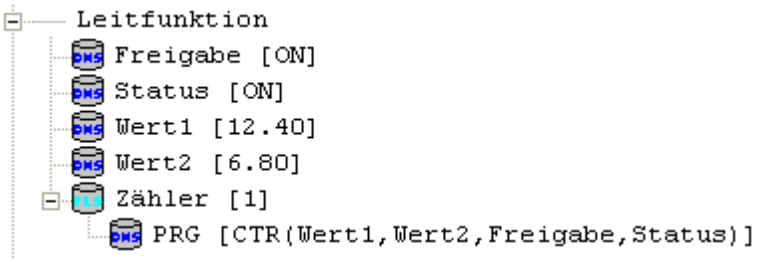
Attribute	Description
Function	Generates a set/reset counter
Supported data types	All data types except for STR
Maximum number of parameters	2
Syntax	CNT
Example	CNT(Increment, Reset)



The data point **Counter** counts each pulse occurring at the data point **Increment**. If the data point **Reset** is set, the **Counter** will be reset.

7.2.6.3.22 Counter (CTR)

Attribute	Description
Function	The value transitions are counted error free.
Supported data types	Counter: All data types except for STR Value1, Value2 = FLT; release, status = BIT
Maximum number of parameters	4
Syntax	CTR
Example	CTR(value1, value2, release, status)



The **counter** data point counts upwards for every value transition which occurs at data points **Value1** and **Value2** if **Value1 > Value2**.

If the **Release** data point is not set, the **counter *will not be*** incremented.
The counter can be reset by writing a 0 directly to the **counter**.

For every value change, a check is first made whether the **Status** is already 1. If yes, the **counter is not** incremented.
If Value1 once again becomes smaller than Value2, then the **Status** is reset to 0.

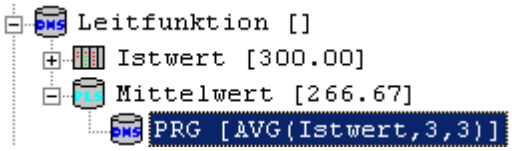
The setting or resetting of the data point **Release** has no effect on the data point **Status** .

If the **counter is to be restarted and the previous value transitions are to be ignored** when the **Release** data point is set, then **Status** should be reset by directly writing a 0.

*Because the function also changes the **Status**, the registered clients are also informed of every change.*

7.2.6.3.23 Average Value (AVG)

Attribute	Description
Function	Establishes the arithmetic average of a value signal
Supported data types	DWU, FLT
Maximum number of parameters	3
Syntax	AVG
Example	AVG(ActualValue,3,3)



The second parameter indicates the number of measurements to be taken. The third parameter indicates the interval (in seconds) in which the average is recalculated. In the preceding image 3 measurements are taken and the average is recalculated every 3 seconds.

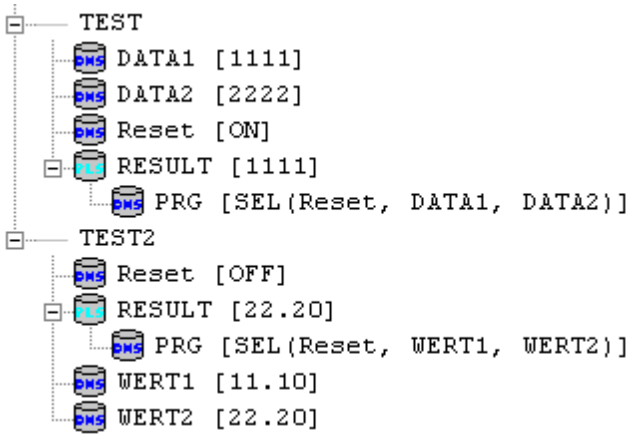
It suffices to recalculate the average value once per minute with one measurement being taken per second.

Recommendation:
AVG(ActualValue,60,60).

*In the preceding image the symbol for the data point **average value** is coloured light blue. This colouring is a visual indicator that a control function is embedded in this data point. After compiling, the DMS view must be reimported to see the colouration.*

7.2.6.3.24 Select (SEL)

Attribute	Description
Function	Selects one of two parameters according to the status of a value signal.
Supported data types	All data types - the value signal must be of the BIT data type
Maximum number of parameters	3
Syntax	SEL
Example	SEL(Signal, Value1, Value2)



The data point **RESULT** is set according to the value signal:
Value signal = 1 -> **RESULT** = Parameter Value1
Value signal = 0 -> **RESULT** = Parameter Value2

Example 1 (String):
If the value signal is **Reset** = 0 (OFF), then the data point **RESULT** = Data2 = "2222" is set,

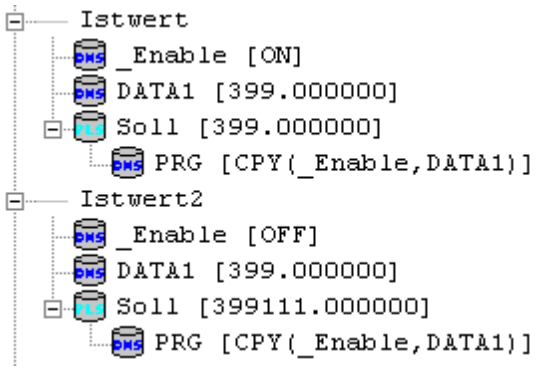
If the value signal is **Reset** = 1 (ON), then the data point **RESULT** = Data1 = "1111" is set.

Example 2 (Floating):
If the value signal is **Reset** = 0 (OFF), then the data point **RESULT** = Value2 = "22.20" is set,
If the value signal is **Reset** = 1 (ON), then the data point **RESULT** = Value1 = "11.10" is set.

The result data type must, of course, be the same data type as both parameters.

7.2.6.3.25 Copy (CPY)

Attribute	Description
Function	Copies the value of the parameters according to the status of a value signal.
Supported data types	All data types - the value signal must be of the BIT data type
Maximum number of parameters	2
Syntax	CPY
Example	CPY(_Enable, DATA1)



The data point **Soll** is set according to the value signal **_Enable**:
value signal = 1 (ON) -> **Soll** = Parameter DATA1
value signal = 0 (OFF) -> **Soll** stays unchanged (the function CPY is not executed at all)

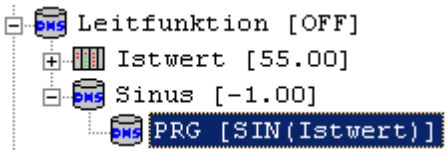
Example (Float):
If the value signal **_Enable** = 0 (OFF), then the data point **Soll** stays unchanged = 399111.000000.
If the value signal **_Enable** = 1 (ON), then the data point **Soll** = DATA1 = 399.000000 is set.

The result data type must, of course, be the same data type as both parameters.

7.2.6.3.26 Sine (SIN)

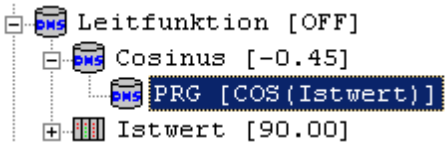
Attribute	Description
Function	Returns the sine value in the radian of a value
Supported data types	DWU, FLT
Maximum number of parameters	1
Syntax	SIN

Attribute	Description
Example	SIN(ActualValue)



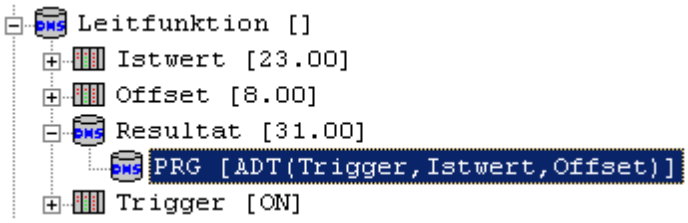
7.2.6.3.27 Cosine (COS)

Attribute	Description
Function	Returns the cosine value in the radian of a value
Supported data types	DWU, FLT
Maximum number of parameters	1
Syntax	COS
Example	COS(ActualValue)



7.2.6.3.28 Triggered Addition (ADT)

Attribute	Description
Function	Adds tw o values w ith the occurrence of a positive edge at the trigger signal.
Supported data types	DWU, FLT
Maximum number of parameters	3
Syntax	ADT
Example	ADT(Trigger,ActualValue,Offset)

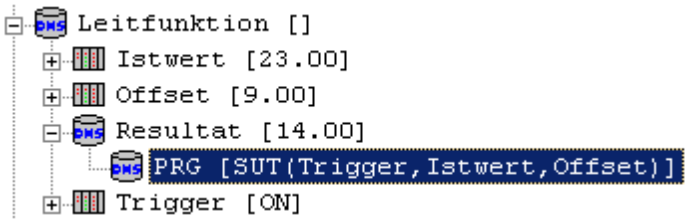


The result is calculated from the data points **ActualValue** and **Offset** (parameters 2 and 3), if a positive edge is set at the data point **Trigger**.

7.2.6.3.29 Triggered subtraction (SUT)

Attribute	Description
Function	Subtracts tw o values w ith the occurrence of a positive edge at the trigger signal.
Supported data types	DWU, FLT
Maximum number of parameters	3
Syntax	SUT

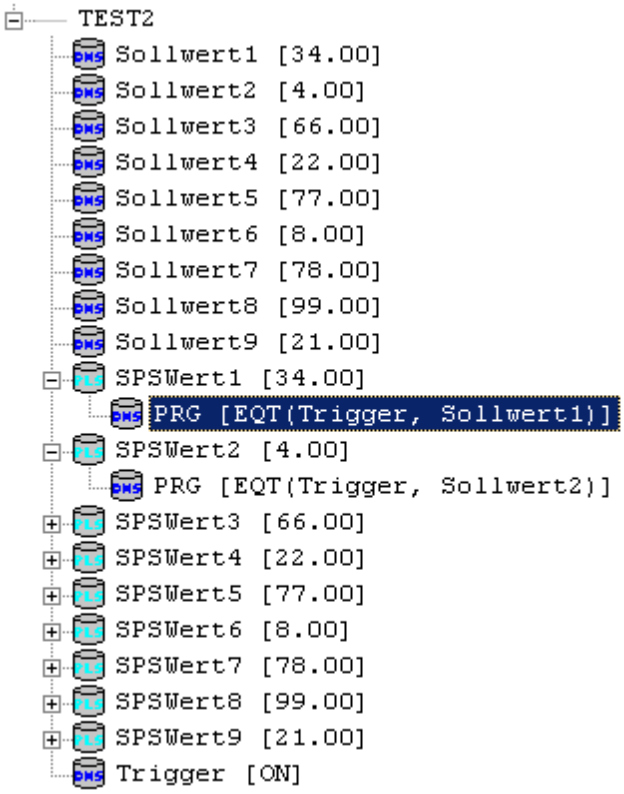
Attribute	Description
Example	SUT(Trigger,ActualValue,Offset)



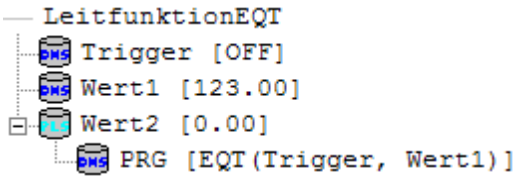
The result is calculated from the data points **ActualValue** and **Offset** (parameters 2 and 3), if a positive edge is set at the data point **Trigger**.

7.2.6.3.30 Equate triggered (EQT)

Attribute	Description
Function	Assigns a value to the trigger signal w ith the occurrence of a positive edge
Supported data types	all data types
Maximum number of parameters	2
Syntax	EQT
Example	EQT(Trigger, TargetValue1)



This function is used, for example, if multiple target values should be adjusted and (almost) simultaneously written to the PLC (input of multiple target values which are written to the PLC via an "Apply" button).



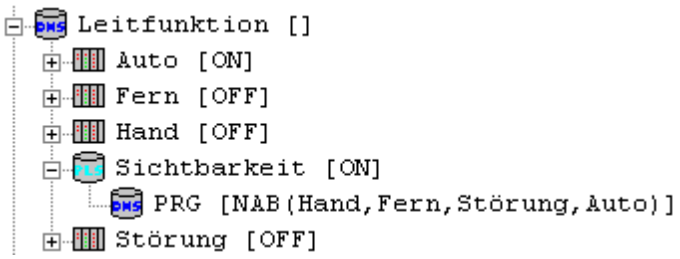
The first parameter indicates the data point that triggers the copying of a value. The second parameter indicates the source value. If the trigger is set, Value1 is copied to Value2.

If the data types of the parameters Trigger *and* TargetValue *do not match, the parameter data types are converted automatically.*

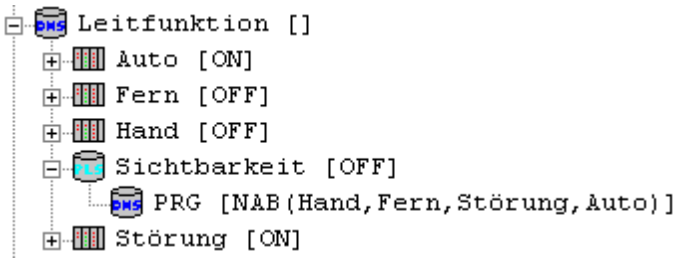
7.2.6.3.31 Not A and B (NAB)

Attribute	Description
Function	NAB sets the data point if the last parameter is set and the previous parameters are all not set.
Supported data types	Only BIT
Maximum number of parameters	4
Syntax	NAB
Example	NAB(Manual,Remote,Error,Auto)

Function:
The NAB control function can be used elegantly to show the visibility of objects depending on signals.



The **Visibility** data point is set if the data point **Auto** is also set. If one of the data points **Manual**, **Remote** or **Error** is now set, the data point **Visibility** is reset.

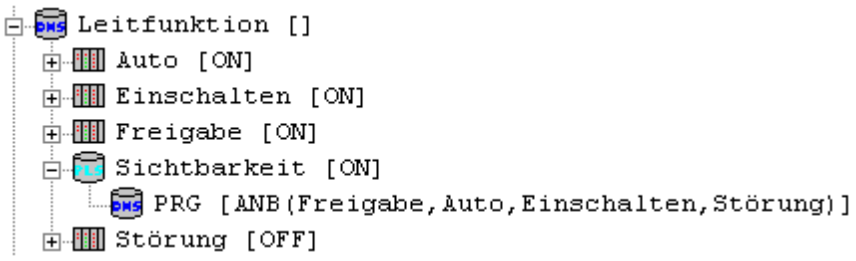


In this example the **Error** data point is set. In doing so, the NAB control function is no longer fulfilled and the data point **Visibility** is reset.

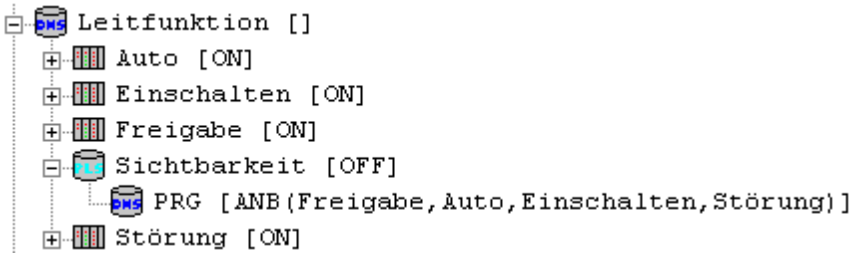
7.2.6.3.32 A and not B (ANB)

Attribute	Description
Function	ANB sets the data point if the last parameter is reset and all preceding parameters are set.
Supported data types	Only BIT
Maximum number of parameters	4
Syntax	ANB
Example	ANB(Release,Auto,SwitchOn,Error)

Function:
ANB can be set in order to show the visibilities of objects depending on signals.



The data point **visibility** is set if the data points **Release, Auto** and **SwitchOn** are all also set.



However, if the data point **Error** is set, the data point **Visibility** is reset.

7.2.6.3.33 EQU from file (EQF)

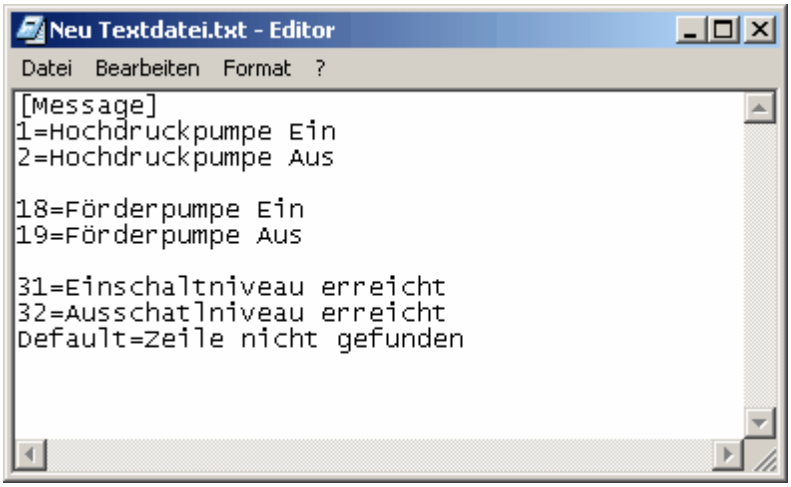
Attribute	Description
Function	The command transmits a string from a text file on the basis of a specified line value.
Supported data types	STR
Maximum number of parameters	2, line number and data point name of the file
Syntax	EQF
Example	EQF(Line,File)

Function:

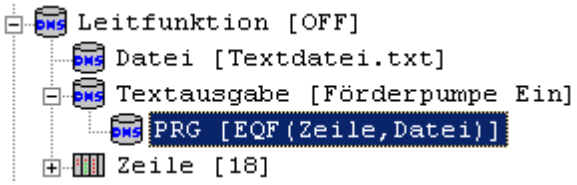
- Text file path
  The text file to be output can also include the full path, e.g. "**D:\Visi.Plus\cfg\BacDriver\LoopUnits.txt**".
  If no path is included, the text file to be output in the current project must be located in the Directory \cfg (standard: C:**Visi.Plus**\"Projektname"\cfg).
  The text file name can also include the following string macros:

“{INSTDIR}”	e.g. “C:\Visi.Plus”
“{PROJDIR}”	e.g. “C:\Visi.Plus\proj”
“{CFGDIR}”	e.g. “C:\Visi.Plus\cfg”
“{BINDIR}”	e.g. “C:\Visi.Plus\bin”
“{ACTPROJDIR}”	e.g. “C:\Visi.Plus\proj\<ProjectName>“

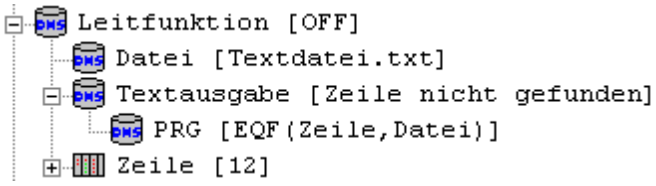
- Text file
  The text file must be structured as follows:



Message must be written in square brackets so that the control function **EQF** recognises the text file. Afterwards, the text line is entered. The line numbers can be freely assigned. A default line can be created. This will then be output whenever the indicated line number is not found in the text file.



The generated text file must be given a data point of type STR (in the preceding image the generated file has been given the data point **File**). In the data point **Line** any numerical value can now be entered. If a line number is defined in the text file for the numerical value, this is output in the data point of type STR **Text output**.

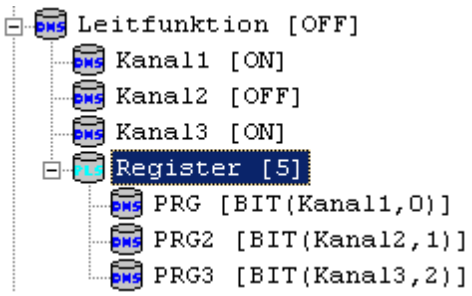


If no text line is defined for the value entered in the data point **Line**, the default line is output in the data point **Text output**.

This function is ideal, for example, if error messages must be output in plain text on the basis of error numbers.

7.2.6.3.34 Copy BIT to DW (BIT)

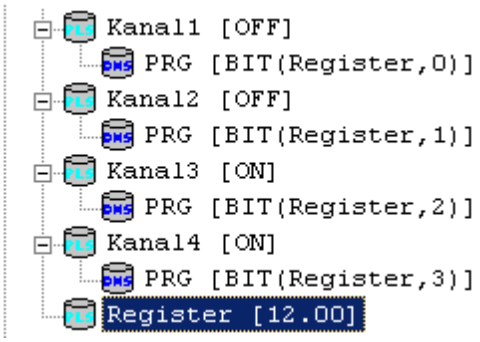
Attribute	Description
Function	BIT copies various data points of the type BIT to a register or vice versa. The BIT function, therefore, is bidirectional.
Supported data types	DWU, DWS, FLT
Maximum number of parameters	2, data point name and bit position
Syntax	BIT
Example	BIT(Channel1,0)



The first parameter of the control function is a digital signal. The second parameter is the position of the bit, which is set in the register on occurrence of the digital signal.
By setting the respective channels, the register value is also adjusted.

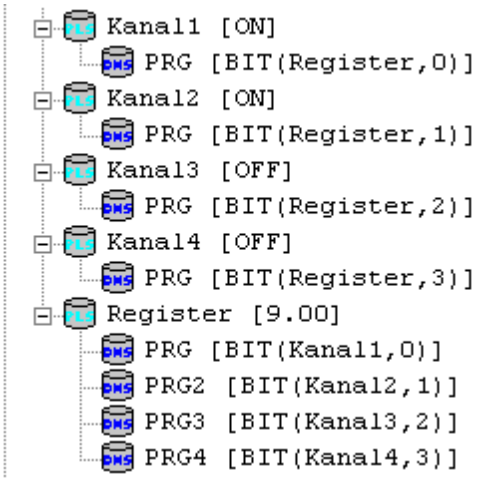
In order to install control functions in a data point, they must have different names. The name must also begin with PRG. In the preceding image, the additional control functions were numbered.

The **BIT** function is also possible in reverse, meaning the respective bits are set on the basis of a register value.



By setting the register value the individual channels are set accordingly.

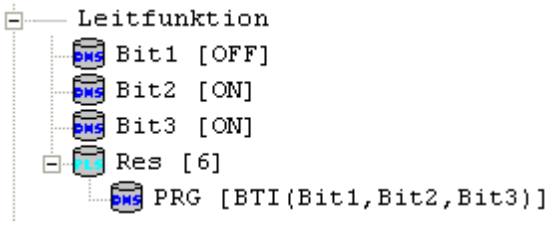
The **BIT** function is bidirectional, meaning the function can write a register value on the basis of modified bit values or set the corresponding bit values on the basis of a register value. A bidirectional example is shown in the following image:



If register values are changed in the DMS, the corresponding bits are set or reset in the BIT control function.
If the channel values are modified in the DMS, the BIT control function adjusts the register value accordingly.

7.2.6.3.35 Converting bit data to DW (BTI)

Attribute	Description
Function	BTI converts various data points of the type BIT to a register.
Supported data types	Result: All data types except for STR and BIT Bit1, Bit2, Bit3, Bit4 = BIT
Maximum number of parameters	4
Syntax	BIT
Example	BTI(Bit1,Bit2,Bit3)



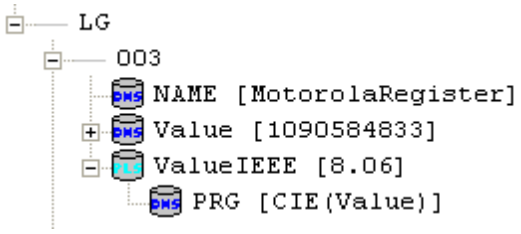
The BIT data points Bit1, Bit2, Bit3, Bit4 are weighted accordingly and are copied to the 0 nibble of the Res data point:

- Bit1 0/1 ➔ 0/1
- Bit2 0/1 ➔ 0/2
- Bit3 0/1 ➔ 0/4
- Bit4 0/1 ➔ 0/8

Example: Bit1 = 1, Bit2 = Bit3 = 0, Bit4 = 1 becomes Res = 9.

7.2.6.3.36 Converting DW data to float IEEE (CIE)

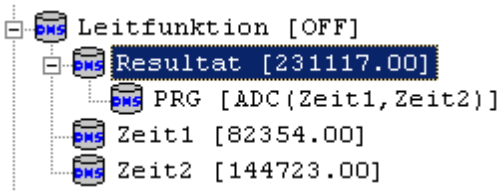
Attribute	Description
Function	CIE interprets DW data as FFP (Motorola Fast Floating Point Format), converts them to IEEE Floating Point Format (FTL) and copies them to the corresponding DMS FTL data point.
Supported data types	All data types except BIT and STR. Result: only FLT.
Maximum number of parameters	1
Syntax	CIE
Example	CIE(Value)



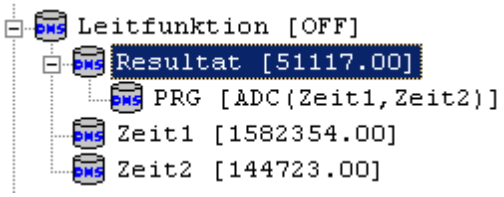
In this example, the data in the DP "Value" of the type DWU = 1090584833 are interpreted as FFP Motorola Fast Floating Point and are converted to DP "ValueIEEE" of the type FLT. Result = 8.06.

7.2.6.3.37 Add / Subtract Times (HHMMSS) ADC

Attribute	Description
Function	ADC allow s the addition/subtraction of times in the format HHMMSS
Supported data types	FLT
Maximum number of parameters	2
Syntax	ADC
Example	ADC(DMSName1,DMSName2)

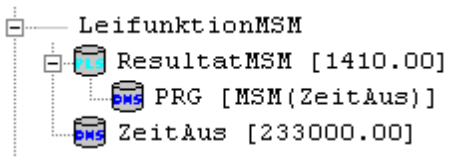


If the entered Time2 is negative, a subtraction of both time values takes place. If the two values added together come to more than 23:59, the display starts over at 0:00.



7.2.6.3.38 Minutes Since Midnight (MSM)

Attribute	Description
Function	Calculates the number of minutes since midnight (12:00am)
Supported data types	FLT
Maximum number of parameters	1
Syntax	MSM
Example	MSM(TimeOff)



The sole parameter (type FLT) indicates the time in the format HHMMSS.

The result (type FLT) is in the format MMMM (minutes, max. 1440).

Example of a possible use: in order to graphically represent a period of time, a switch-on and switch-off time must be converted into Minutes Since Midnight, so that it can be represented in a rational manner.

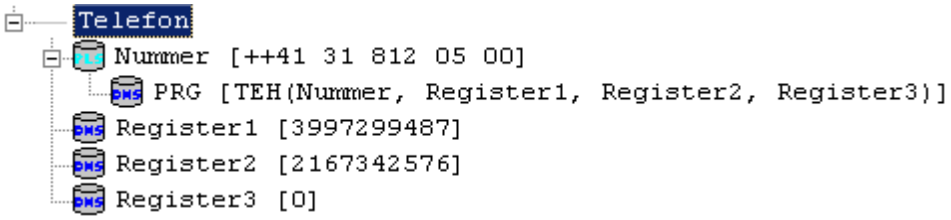
7.2.6.3.39 Convert telephone number to hex (TEH)

Attribute	Description
Function	TEH converts a telephone number to a hex number
Supported data types	FLT
Maximum number of parameters	4, telephone number and storage registers
Syntax	TEH
Example	TEH(Number, Reg1, Reg2, Reg3)

The telephone number is of the type STR.
The three registers for the PLC are of the type DWU.

The following parameters must be given in the control function:

Par1:
Telephone number
Par2:
Register for first value
Par3:
Register for second value
Par4:
Register for third value



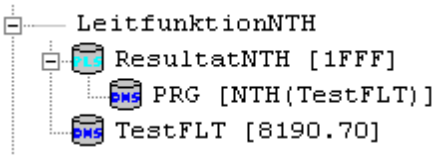
The control function can only be triggered directly from the DMS. The telephone number must be entered via the GE.

The register cannot be displayed perfectly in the GE (since it is type DWS).

For checking purposes, the number must be typed to a computer and then converted to hex (or be shown in the PCD as hex).

7.2.6.3.40 Convert number to hex (NTH)

Attribute	Description
Function	NTH converts a number into a hex number w hich is displayed in a string
Supported data types	All data types
Maximum number of parameters	1
Syntax	NTH
Example	NTH(number)



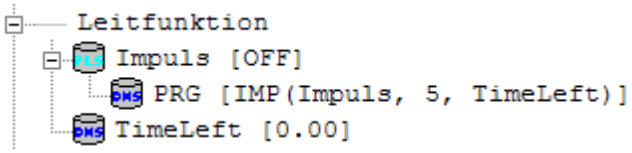
The only parameter (all data types) is the input number.

If the parameter is of the type FLT, the number of rounded to the nearest whole number, digits after the decimal point are disregarded.

The result (type STR) is the input number converted into hex.

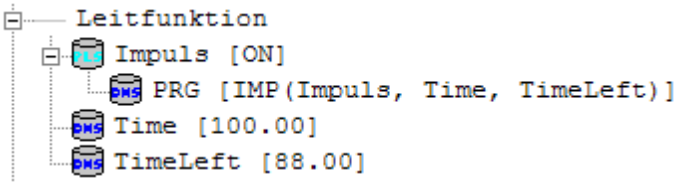
7.2.6.3.41 Impulse (IMP)

Attribute	Description
Function	Resets a set bit after a prescribed length of time
Supported data types	BIT
Maximum number of parameters	3
Syntax	IMP
Example	IMP(Signal,3,RemainingTime)



The first parameter shows the data point that triggers the impulse (and sets the data point to 1). This can be the same data point which is then reset.
The second parameter shows the number of seconds that are waited until the data point is reset.
The third parameter points to an FLT data point which shows the remaining time (optional).

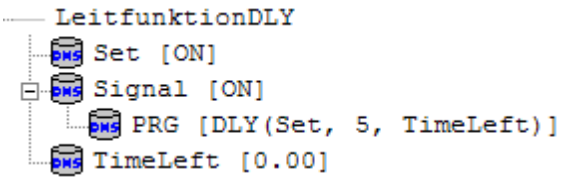
The second parameter can also be defined as a data point (and thus be dynamically adjusted):



This function is very CPU-intensive. We recommended only limited use of this function (do not trigger hundreds of impulse functions simultaneously). It should also be noted that the indication of time is not very precise (Windows Timer). This function may not be used for time-critical applications.

7.2.6.3.42 Delay (DLY)

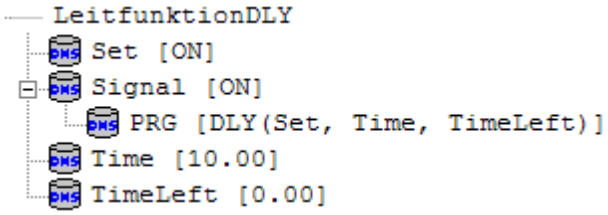
Attribute	Description
Function	After a prescribed length of time, sets a bit which has not been set to 1
Supported data types	BIT
Maximum number of parameters	3
Syntax	DLY
Example	DLY(Signal,3,RemainingTime)



The first parameter shows the data point that triggers the impulse (and sets the data point to 1 after a delay).
The second parameter shows the number of seconds that are waited until the data point is set.
The third parameter points to an FLT data point, which indicates how much time is left until the data point is set. This parameter is optional and does not have to be specified (example: DLY(Set, Time))

As soon as the data point being set returns to zero, the set-delayed data point is also set to zero.

The second parameter can also be defined as a data point (and thus be dynamically adjusted):



This function is very CPU-intensive. We recommended only limited use of this function (do not trigger hundreds of delay functions simultaneously). It should also be noted that the indication of time is not very precise (Windows Timer). This function may not be used for time-critical applications.

7.2.6.4 Compiling control functions

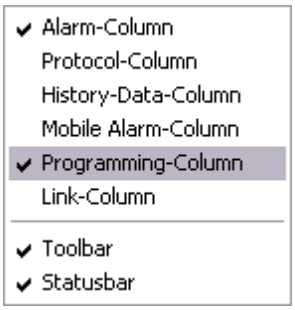
For a **compiling control function** to be inserted into the DMS, it must first be compiled:

Compile

or press the **<F2> key** on the keyboard

or select**Menu > Settings > Compile Function**

Control functions are only executed if the DMS value changes.
*We recommend entering control functions in the PET in the column **Control Function**. If the Control Function column is not visible, it can be activated later in the PET menu **View > Control Function Column**:*

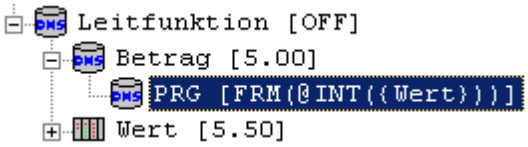


Advantage:
It is not necessary to manually perform compilations of the control functions in the PET. The PET automatically triggers the compilation in the DMS.

Program
Tvh(Istwert,1_GW_Hi)
Tvl(Istwert,1_GW_Lo)

7.2.6.5 Interpreted control functions (FRM)

All interpreted control functions begin with the keyword **FRM** (for formula). The control function itself is defined like compiled functions with the keyword **PRG** (or **PRGx**).



Structure of the command line:
FRM(formula)

Syntax definition of a formula:

- max. 80 characters
- Max. four DMS values per formula
- DMS values in curly brackets (e.g. {actual value})
- The formula can be any expression

Examples:
FRM({Value1}+{Value2})
FRM(@sum({Value3},{Value4}))
FRM({Value1} / {Value2} * 4.16)
FRM(@COS({Value}))

In the current version a DMS value can only be assigned once to a control function!

Interpreted control functions are inserted into the DMS like compiled control functions. The interpreted control functions must also be compiled.

7.2.6.5.1 Mathematical functions

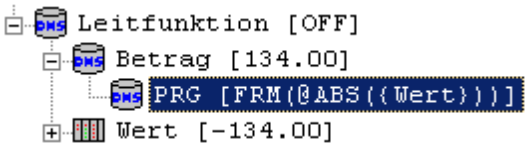
An almost unlimited number of functions can be defined with basic arithmetic operations and any number of parentheses levels. Constants are also allowed directly in the formula.

Example: `FRM({Value1} / {Value2} * 4.16)`

Because the functions are interpreted, they can also be defined at runtime (e.g. in an input field in the GE).

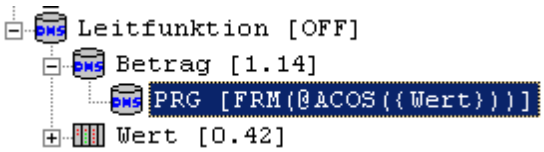
7.2.6.5.1.1 Absolute value (@ABS(x))

Attribute	Description
Function	Absolute value returns the unsigned numerical value of given variables
Supported data types	DWU, DWS, FLT
Maximum number of parameters	1
Syntax	@ABS(x)
Example	FRM(@ABS({Value}))



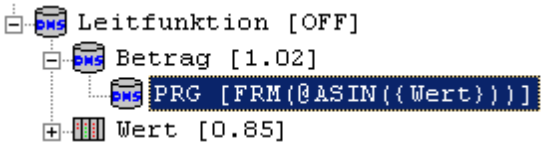
7.2.6.5.1.2 Arc cosine (@ACOS(x))

Attribute	Description
Function	Arc cosine returns the inverse cosine value in radians
Supported data types	DWU, DWS, FLT
Maximum number of parameters	1
Syntax	@ACOS(x)
Example	FRM(@ACOS({Value}))



7.2.6.5.1.3 Arc sine (@ASIN(x))

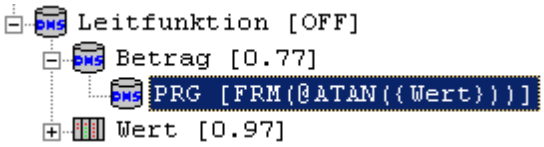
Attribute	Description
Function	Arc sine returns the inverse sine value in radians
Supported data types	DWU, DWS, FLT
Maximum number of parameters	1
Syntax	@ASIN(x)
Example	FRM(@ASIN({Value}))



7.2.6.5.1.4 Arc tangent (@ATAN(x))

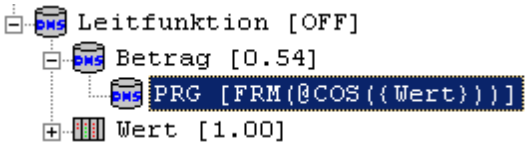
Attribute	Description
Function	Arc tangent returns the inverse tangent value in radians
Supported data types	DWU, DWS, FLT

Attribute	Description
Maximum number of parameters	1
Syntax	@ATAN(x)
Example	FRM(@ATAN({Value}))



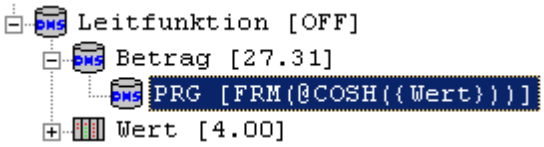
7.2.6.5.1.5 Cosine (@COS(x))

Attribute	Description
Function	Returns the cosine value of a value in radians
Supported data types	DWU, DWS, FLT
Maximum number of parameters	1
Syntax	@COS(x)
Example	FRM(@COS({Value}))



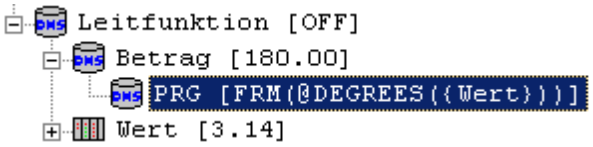
7.2.6.5.1.6 Cosine hyperbolic (@COSH(x))

Attribute	Description
Function	Cosine hyperbolic returns the hyperbolic cosine value in radians
Supported data types	DWU, DWS, FLT
Maximum number of parameters	1
Syntax	@COSH(x)
Example	FRM(@COSH({Value}))



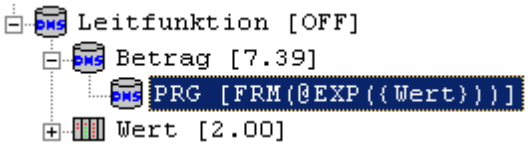
7.2.6.5.1.7 Radians to degrees (@DEGREES(x))

Attribute	Description
Function	Converts the angle from radians to degrees
Supported data types	DWU, DWS, FLT
Maximum number of parameters	1
Syntax	@DEGREES(x)
Example	FRM(@DEGREES({Value}))



7.2.6.5.1.8 Exponential (@EXP(x))

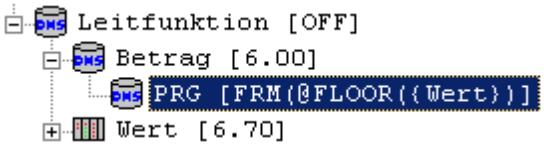
Attribute	Description
Function	Constructs the exponential function of variable x
Supported data types	DWU, DWS, FLT
Maximum number of parameters	1
Syntax	@EXP(x)
Example	FRM(@EXP({Value}))



The example shown in the preceding image corresponds to the function e^2.

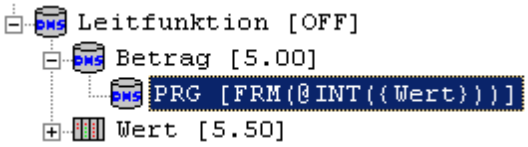
7.2.6.5.1.9 Round down to smaller integer (@FLOOR(x))

Attribute	Description
Function	Rounds a floating-point number down to the next smaller integer
Supported data types	FLT
Maximum number of parameters	1
Syntax	@FLOOR(x)
Example	FRM(@FLOOR({Value}))



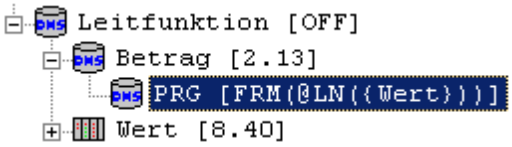
7.2.6.5.1.10 Integer part of a floating-point number (@INT(x))

Attribute	Description
Function	Returns the integer part of a floating-point number
Supported data types	FLT
Maximum number of parameters	1
Syntax	@INT(x)
Example	FRM(@INT({Value}))



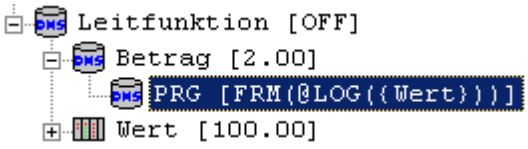
7.2.6.5.1.11 Natural logarithm (@LN(x))

Attribute	Description
Function	Constructs the natural logarithm of variable x
Supported data types	FLT
Maximum number of parameters	1
Syntax	@LN(x)
Example	FRM(@LN({Value}))



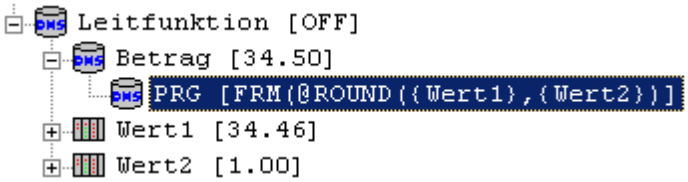
7.2.6.5.1.12 Base 10 logarithm (@LOG(x))

Attribute	Description
Function	Constructs the base 10 logarithm of variable x
Supported data types	DWU, DWS, FLT
Maximum number of parameters	1
Syntax	@LOG(x)
Example	FRM(@LOG({Value}))



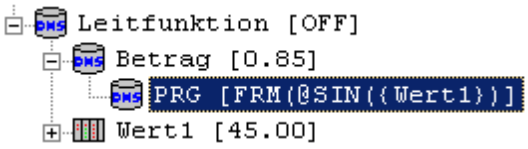
7.2.6.5.1.13 Round to n digits (@ROUND(x,n))

Attribute	Description
Function	Rounds variable x to the number of digits specified in variable n
Supported data types	FLT
Maximum number of parameters	1
Syntax	@ROUND(x)
Example	FRM(@ROUND({Value1},{Value2}))



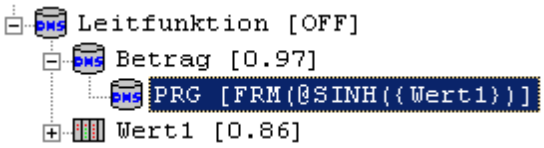
7.2.6.5.1.14 Sine (@SIN(x))

Attribute	Description
Function	Returns the sine value of a value in radians
Supported data types	DWU, DWS, FLT
Maximum number of parameters	1
Syntax	@SIN(x)
Example	FRM(@SIN({Value}))



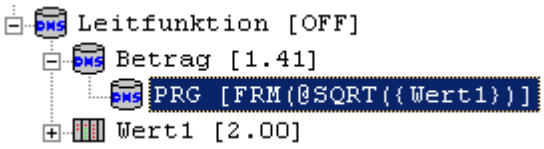
7.2.6.5.1.15 Sine hyperbolic (@SINH(x))

Attribute	Description
Function	Sine hyperbolic returns the hyperbolic sine value in radians
Supported data types	DWU, DWS, FLT
Maximum number of parameters	1
Syntax	@SINH(x)
Example	FRM(@SINH({Value1}))



7.2.6.5.1.16 Square root (@SQRT(x))

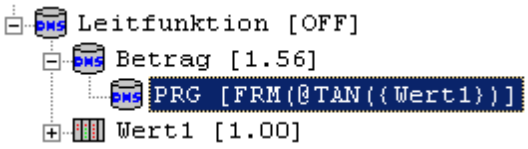
Attribute	Description
Function	Calculates the square root of variable x
Supported data types	FLT
Maximum number of parameters	1
Syntax	@SQRT(x)
Example	FRM(@SQRT({Value1}))



7.2.6.5.1.17 Tangent (@TAN(x))

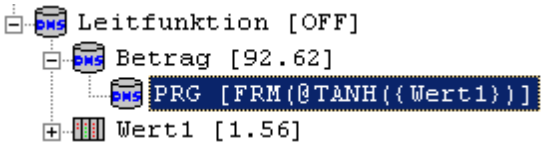
Attribute	Description
Function	Returns the tangent value of a value in radians

Attribute	Description
Supported data types	FLT
Maximum number of parameters	1
Syntax	@TAN(x)
Example	FRM(@TAN({Value1}))



7.2.6.5.1.18 Tangent hyperbolic (@TANH(x))

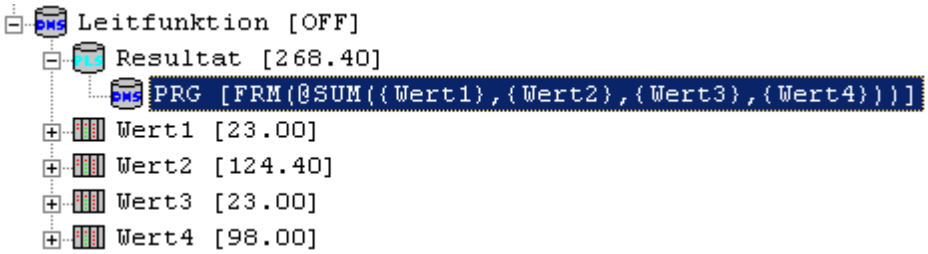
Attribute	Description
Function	Tangent hyperbolic returns the hyperbolic tangent value in radians
Supported data types	DWU, DWS, FLT
Maximum number of parameters	1
Syntax	@TANH(x)
Example	FRM(@TANH({Value1}))



7.2.6.5.2 Statistical functions

7.2.6.5.2.1 Sum (@SUM(..))

Attribute	Description
Function	Calculates the total of the indicated parameters
Supported data types	DWU, DWS, FLT
Maximum number of parameters	4
Syntax	@SUM(...)
Example	FRM(@SUM({Value1},{Value2},{Value3},{Value4}))



*All functions are only triggered if a value is changed or the command **Execute***

__Function__ is selected in the __Settings__ menu.
We recommend entering all control functions in the PET in the Control Function column, because the compilations of the control functions are automatically executed as a result.

7.2.7 XML data transfer from and to a DMS

Any DMS data points can be copied between two DMSes by means of an XML file exchange.

The DMS which requests data is the client DMS and the DMS which delivers data is the server DMS.
The PWEB web server must be running on both systems.

Server DMS

There must be a configuration file (customer definition file) in the <proj>\cfg folder, which contains the following specifications of the client DMS:

- Client identification (customer section)
- DMS definitions which are transmitted to the client
- Time specification of the last successful data transfer (date/time stamp)
- Time specification indicating how often the client can request the data (interval)
- Total number of successful data transfers

Example of a customer section (Customer ID = 3D78AE66):

```
.....
[3D78AE66]
Customer=MST Systemtechnik AG
Location=3123 belp
Interval=1440
LastUpdate=25.03.10 11:41:24
Count=123
Error=OK
Weather:3D78AE66:Forecast:DAY_1:Radiation=Weather:DAY_1:Radiation
Weather:3D78AE66:Forecast:DAY_2:Rain=Weather:DAY_2:Rain
Weather:3D78AE66:Forecast:DAY_3:Text=Weather:DAY_3:Text
.....
```

Client DMS

Starts the data transfer at the program start of pXMLdata.exe + the corresponding command line parameters.

 See the chapter "Auxiliary programs", Sending command line to web server with parameters (pXMLdata.exe).

7.2.7.1 Sample Data Transfer Procedure

1. The client requests data from the server:
via GE button, program start e.g.:
"pXMLData /ADR:10.0.0.39 /PORT:80 /PARM:DMSXMLGET.HTM?cust=3D78AE66?srvc=forecast?path=pxmldata.cfg?port=80"

where:
ADR: 10.0.0.39 = Server IP
PORT: 80 = Server Port
cust: 3D78AE68 = Customer ID

srvc: forecast = Type of query; if present, it is used as a sub-folder in the \dat directory.
path: pxmldata.cfg = Customer definition file (see below) in <proj>\cfg folder, contains DMS definitions of XML clients
port: 80 = XML Client Port


The data transfer can also be started manually using query commands directly from Internet Explorer (or other browsers). This can be especially useful during the debugging phase.
Example of a command line:
"HTTP://10.0.0.39/DMSXMLGET.HTM?CUST=3D78AE68?SRVC=FORECAST?PATH=pxmldata?PORT=80"

2. From this, the server determines the cfg file <path> and customer section <cust> contained there.
3. From this customer section, the server reads all key lines with the following specifications:
- Interval=1440 ; in minutes, only one access per day here
- Count ; number of correct client accesses = number of data deliveries
- Error ; last error message, e.g.:
  "XMLcust[3D78AE66] ERROR: data not sent,interval not passed min=[1440], still remaining=[1421]"
- <DMSServer>=<DMSclient> ; DMS definitions

4. The server checks whether the last access occurred longer ago than the minutes specified in the interval.
If not, then an XML file is created without DMS data but with the corresponding status:
<status>XMLcust[3D78AE66] ERROR: data not sent,interval not passed min=[1440], still remaining=[1439]</status>

5. The server reads the requested DMS data points,
writes them to an XML file, incl. DMS client names and DMS type and a time stamp.
In addition, a <deliverydate> field with the current time is created as well as <nextdelivery> = current time + interval.
The XML file is saved under the following name in <proj>\dat\<srvc>\<cust>.xml before being sent.
If the folder \dat\<srvc> does not exist, it is created.

6. The XML file is sent to the client
at the same IP address from which the query came, port = <PORT> (from the command line of the query).
If the transmission is OK, then the count is incremented in the customer section of the <path> (pxmldata.cfg) file and the date/time stamp LastUpdate is refreshed.

7. The client detects the command line
and extracts the necessary information:
cust: 3D78AE68 = Customer ID
srvc: FORECAST = Type of query; if present, it is used as a sub-folder in the \dat directory.
path: 3D78AE68.xml = The XML file to be received with the requested DMS data of the server.

8. Subsequently, the client receives the XML file from the server
and writes it to <proj>\dat\<srvc>\<cust>.xml.

9. The client parses the XML file
The following DMS data are determined: DMS name, type, value and date stamp.
This information is encoded in the XML file in the following manner:
Example:

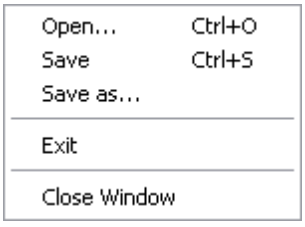
```
<?xml version="1.0" encoding="UTF-8"?>
<data>version="1.5.1" link="http://www.mst.ch"
<status>OK</status>
<deliverydate>26.03.2010 12:39:14</deliverydate>
<nextdelivery>27.03.2010 12:38:57</nextdelivery>
<set>
       <parameter name="MeteoForecast:DAY_1:Radiation">
              <value datetime="2010-03-12 00:00:00" type="FLT">4094.000000</value>
       </parameter>
       <parameter name="MeteoForecast:DAY_3:Text">
              <value datetime="2010-03-14 00:00:00" type="STR">SnowShowers</value>
       </parameter>
       <parameter name="MeteoForecast:DAY_2:Radiation">
              <value datetime="2010-03-13 00:00:00" type="FLT">3817.000000</value>
       </parameter>
</set>
</data>
```

All DMS data points are created new corresponding to the type and the value is saved.

 All data transfers are carried out by the web servers on the server and client sides. All XML events are logged by the the web server and are also displayed in the PWEB Debug window. There are also additional log options in the PWEB menu, log settings Dlg.: Log Send/ Receive XML data details.

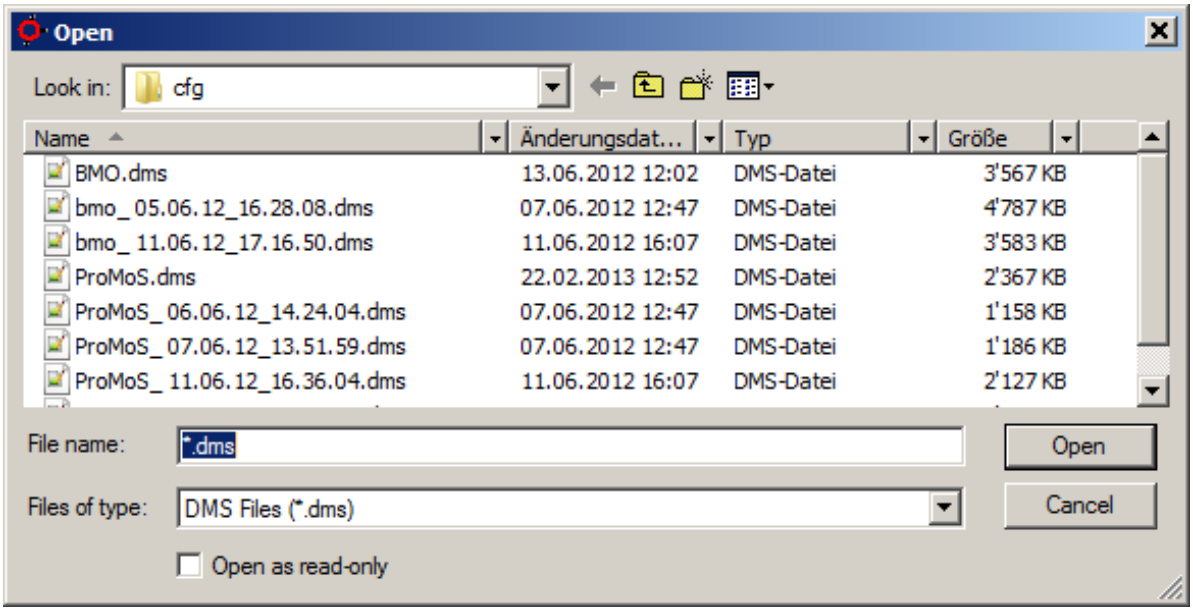
7.2.8 File menu

Different options for saving, opening and exiting the DMS area available under the "**File**" menu.



7.2.8.1 Open... <CTRL+O>

 Project data or existing template object structures can be imported with the "**Open**" command or with the key combination **<CTRL+O>** . Select the DMS file to be imported in the **Open** window.



If there is already DMS data with the same name, it will not be overwritten. Only the value will be reimported. Existing DMS data is retained.

7.2.8.2 Save <CTRL+S>

The current project data can be saved by pressing the **Save** icon in the toolbar, the keyboard combination **<CTRL+S>** or the menu entry **Save**.

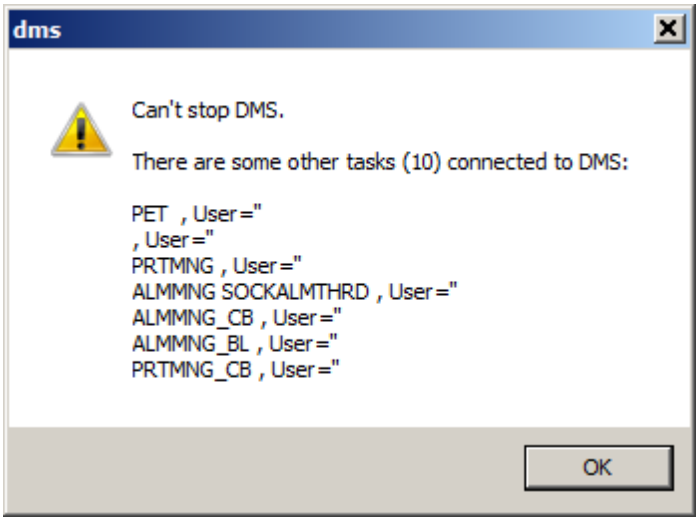
7.2.8.3 Save As...

"**Save As...**" can be used to specify a different filename. It may makes sense, for example, to save copies before starting a test.

7.2.8.4 Exit

In order to close the application, all modules (Graphical Editor, PDBS, Alarm Manager, etc.) registered with the DMS must first be exited.

As long as programs are still registered with the DMS, it cannot be exited! A corresponding message will notify you of this.



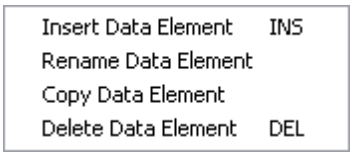
The entire system can be terminated with the supplemental program pStop (also from the GE).

7.2.8.5 Close Window

This menu item closes the DMS window. This does not close the DMS program itself; it is only minimised to the taskbar.

7.2.9 Edit menu

Under the "Edit" menu there are options are available for editing data points.

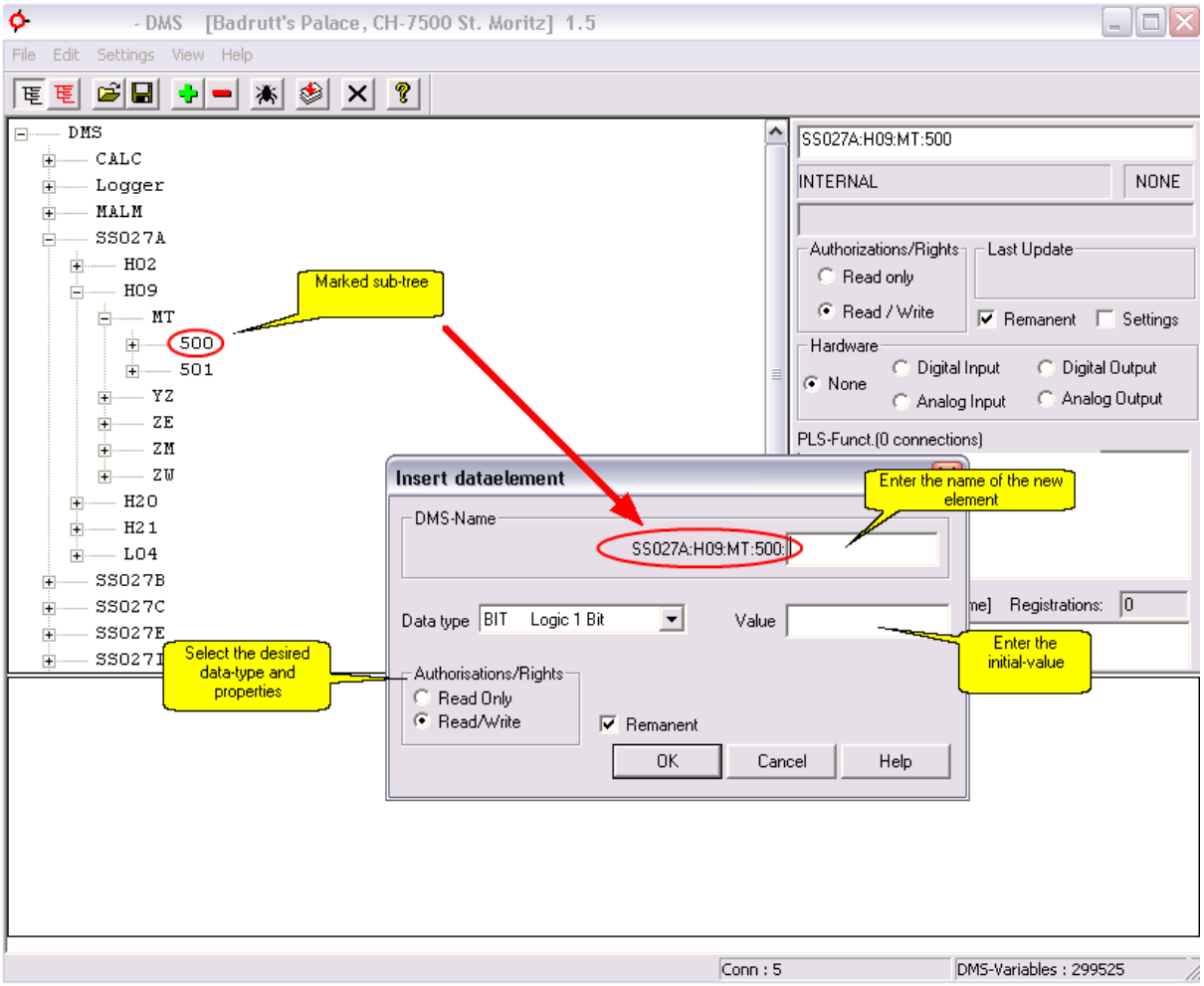


The two menu items in the centre are only displayed actively if a data point is selected.

7.2.9.1 Insert Data Element <INSERT>

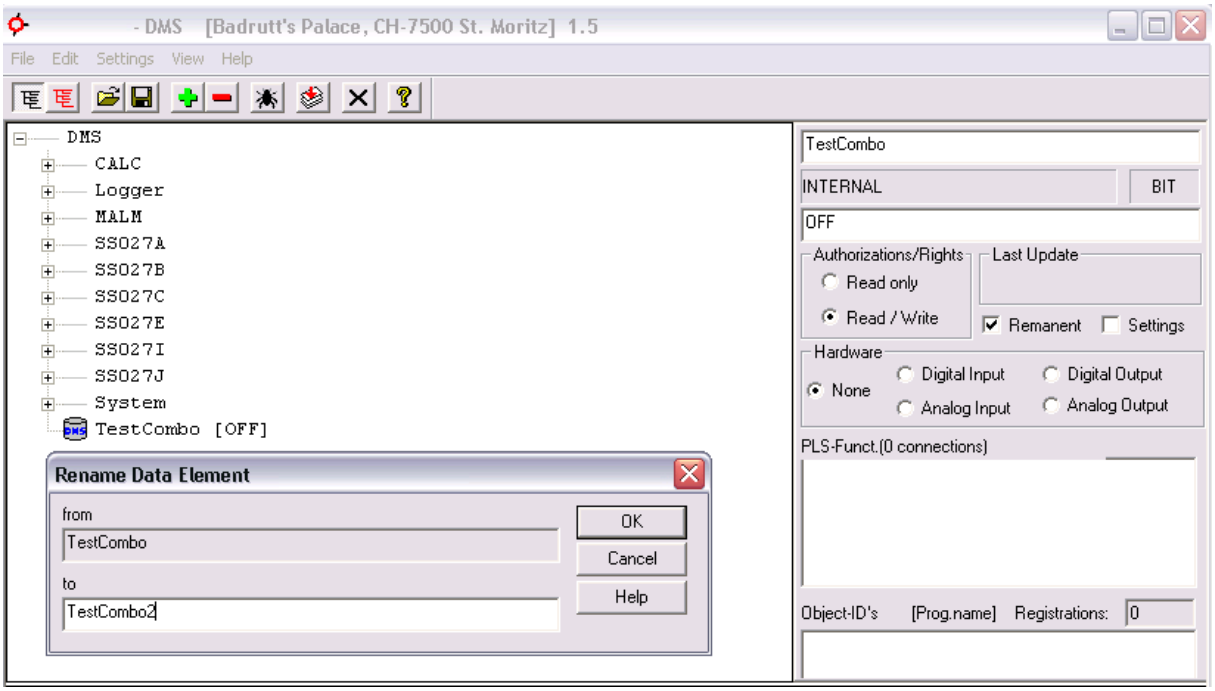
Insert a new element by clicking on the **Insert Data Element** icon in the toolbar, or pressing the **<+>** or **<Insert>** keys.

Make absolutely sure that the desired sub-tree is marked!



7.2.9.2 Rename Data Element

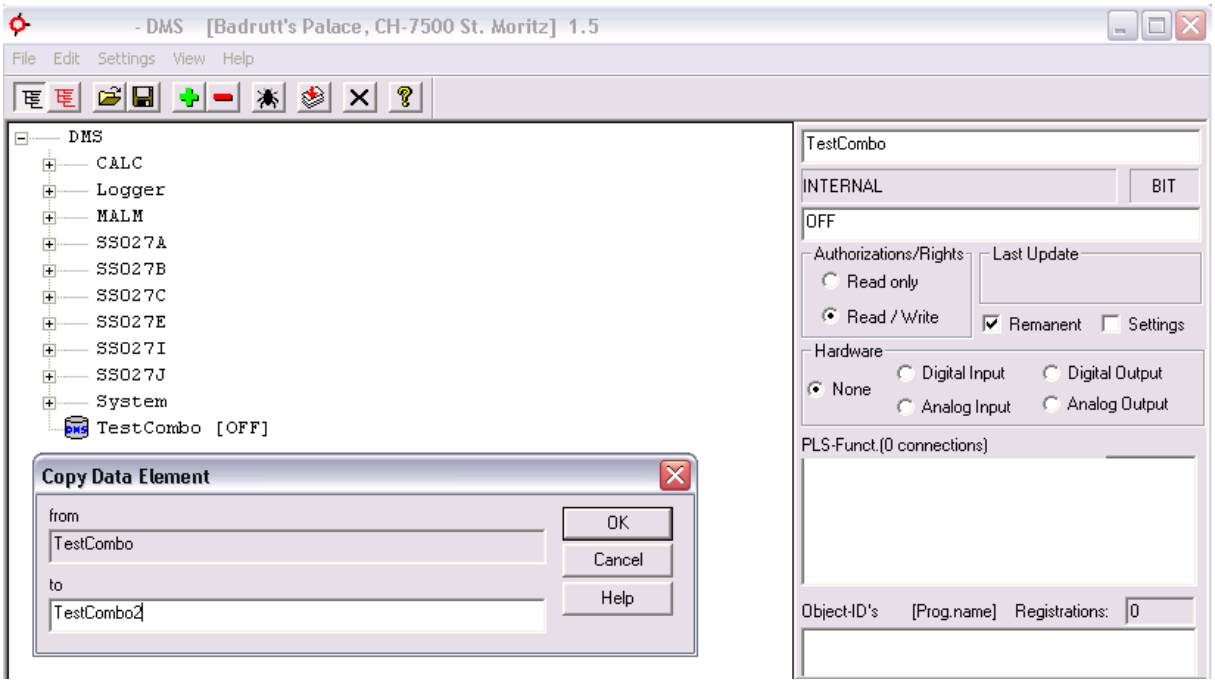
Rename a marked element by clicking on the menu item **Rename Data Element** in the Edit menu.
The **Rename Data Element** dialog appears:



Make absolutely sure that the desired sub-tree is marked!

7.2.9.3 Copy Data Element

Copy a marked element by clicking on the menu item **Copy Data Element** in the Edit menu. The **Copy Data Element** dialog appears:



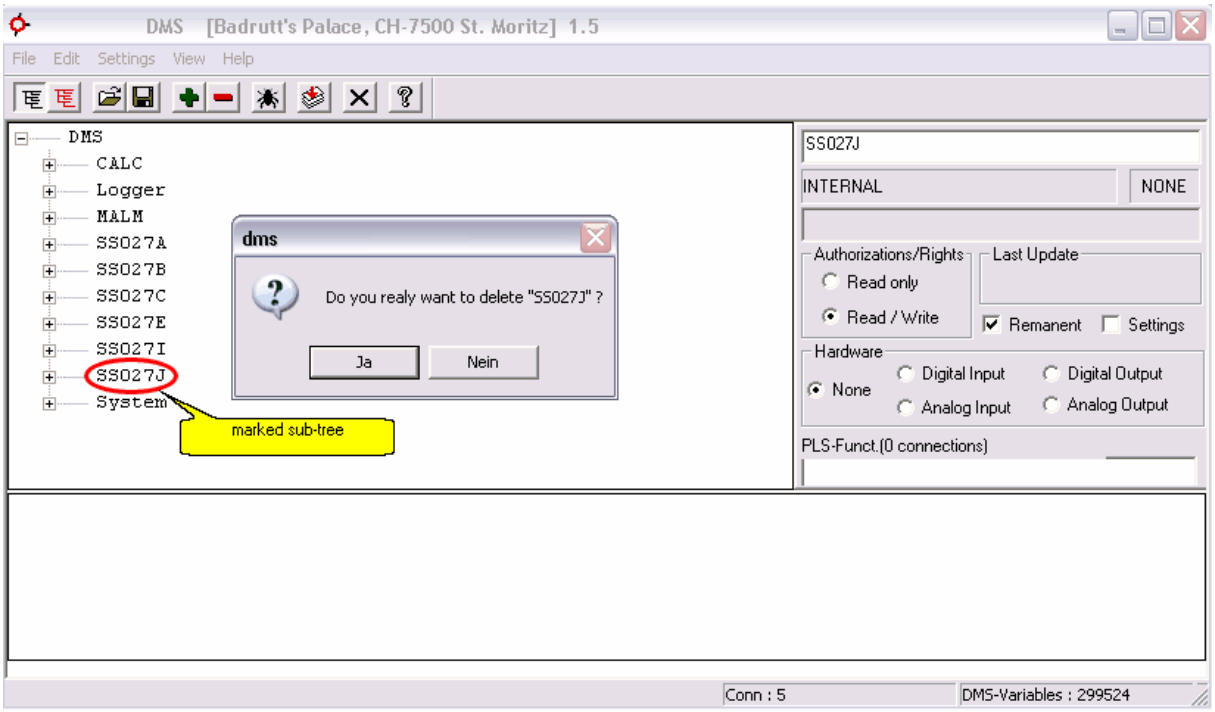
Make absolutely sure that the desired sub-tree is marked!

7.2.9.4 Delete Data Element <DEL>

Select the data point to be deleted (the data point is marked with a blue background). Carry out the deletion by clicking on the **Delete Data Element** icon in the toolbar or with the **<DEL>** key.

A confirmation prompt appears, asking whether the data point should really be deleted.

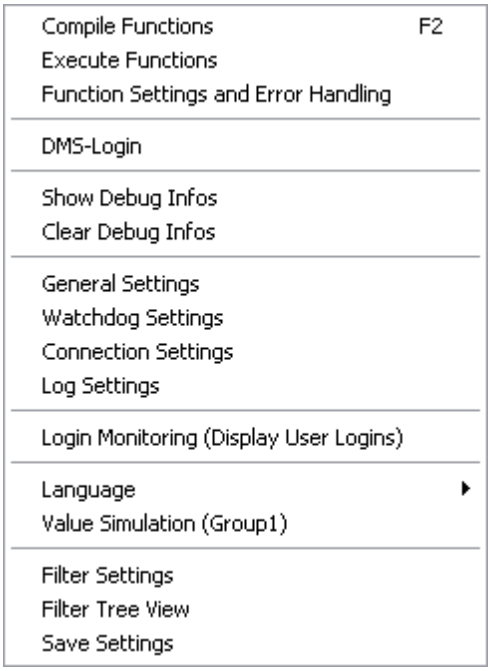
Make absolutely sure that the desired sub-tree is marked!



All data points under the deleted data point are also deleted, but they remain physically present in memory so that other programs which use these data points do not have problems. After the DMS is restarted, the data points are finally deleted.

7.2.10 Settings menu

The "**Settings**" menu offers settings for control functions, value displays, operating language and filter settings for the DMS.



7.2.10.1 Compile control functions <F2>

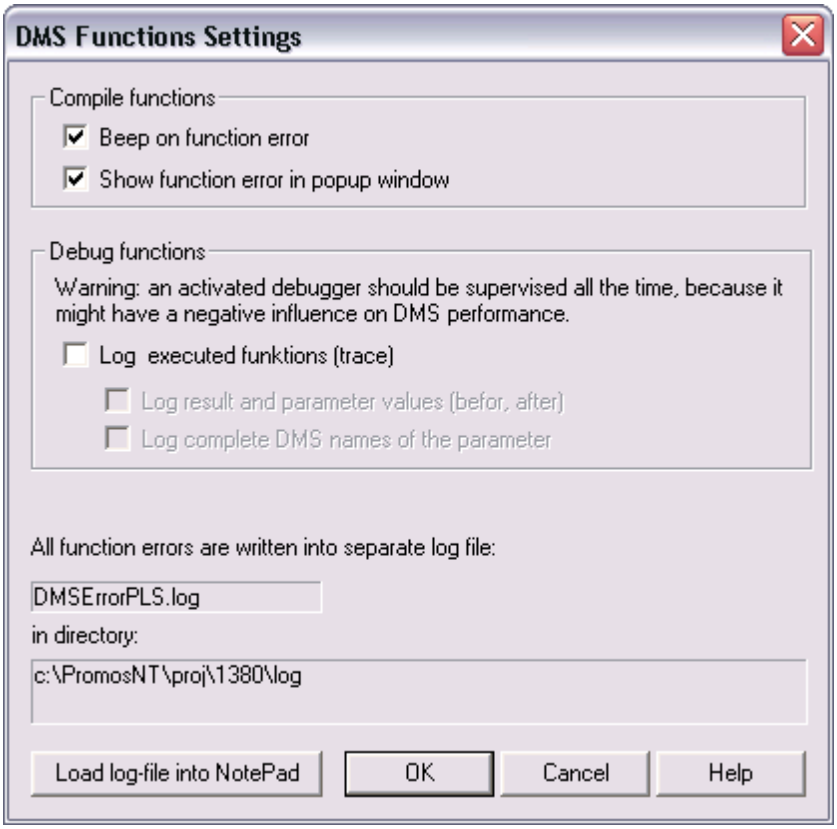
By selecting the "**Compile Functions**" command or by pressing the **<F2>** key, all control functions in the DMS are compiled.

7.2.10.2 Execute control functions

All control functions are executed with the command "**Execute functions**". This means that no manual value changes need to be made.

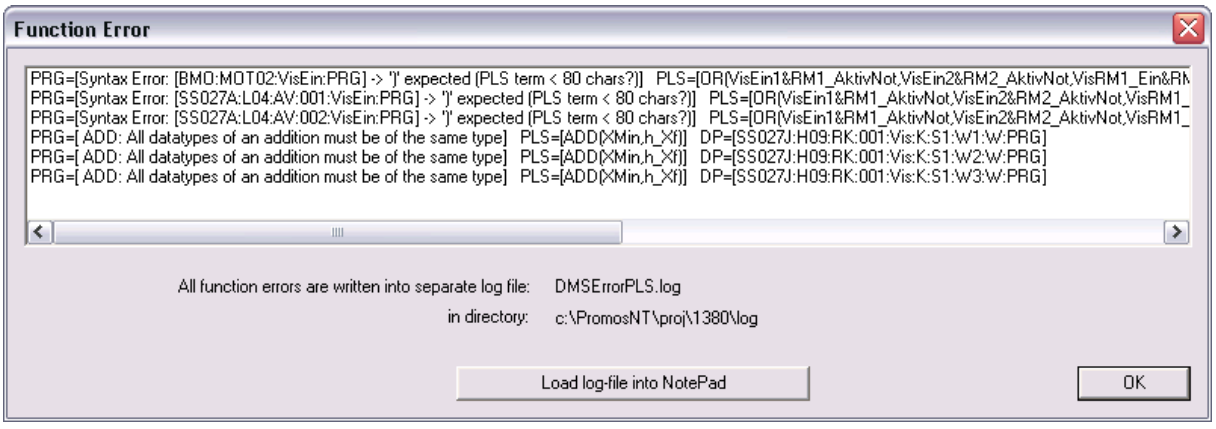
7.2.10.3 Control functions and troubleshooting

In the window "**DMS Control Function Settings**", various settings can can be changed which have an influence on control functions troubleshooting:



Beep for Control Function Error
If activated, a beep sounds during compilation if one or more errors occur.

Show Control Function Errors in Popup Window List
If activatedt, all errors are listed in a separate popup window. The window is non-modal, and so does not need to be immediately acknowledged; you can simply leave it alone.
Example:



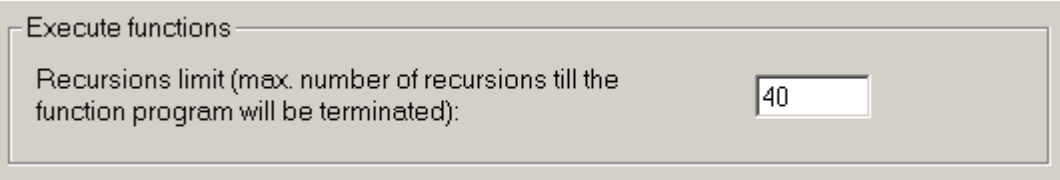
Load the log file in Notepad
The DMSErrorPLS.log file for the current project is displayed in Notepad.

Control function error log file
All control function compilation and execution errors are held in a separate log file

"**DMSErrorPLS.log**" in the project log directory.

Control functions which cause a compilation error are normally reported (beep and/or error display) and not inserted into the DMS structure, so that they cannot be executed.

7.2.10.3.1 Execute control functions (recursion)



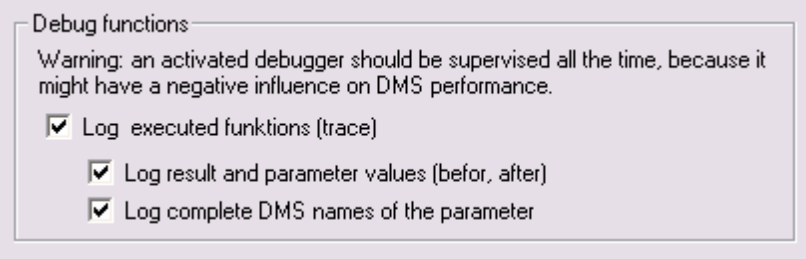
The number of levels on which a recursion is allowed can be defined in this dialog.

Default: 40

Attention: A very small value causes the menu displays in the DDC Suite to stop working!

7.2.10.3.2 Debugging Control Functions

The following two options are available:



- **Control functions cause logging (Trace)**
  When the debug option is enabled, a trace log of all control functions called is written to the "**DMSErrorPLS.log**" log file.

  Example: Test run of a sequence: the debug option was enabled before the value change of the relevant variable "**COP_Actual**" and was then switched off again afterward:

```
08.03.10 14:45:38 ======================= PLS tracing ON ============================
08.03.10 14:53:21  Tvh [OverRed] (COP_Actual, COP_Red, -, -)
08.03.10 14:53:21   And [CompareYellow] (UnderGreen, OverRed, -, -)
08.03.10 14:53:21    Anb [Yellow] (Operation, CompareYellow, Delay, -)
08.03.10 14:53:21     Cnt [CounterYellow] (Yellow, Reset, -, -)
08.03.10 14:53:21    Anb [Red] (Operation, CompareRed, Delay, -)
08.03.10 14:53:21    Anb [Green] (Operation, CompareGreen, Delay, -)
08.03.10 14:53:21  Add [COP_Red_Offset] (COP_Red, Value_Offset, -, -)
08.03.10 14:53:21  Tvl [CompareRed] (COP_Actual, COP_Red_Offset, -, -)
08.03.10 14:53:21   Anb [Red] (Operation, CompareRed, Delay, -)
08.03.10 14:53:21    Cnt [CounterRed] (Red, Reset, -, -)
08.03.10 14:53:21    Cnt [CounterGreen] (Green, Reset, -, -)
08.03.10 14:53:21    Cnt [CounterYellow] (Yellow, Reset, -, -)
08.03.10 14:53:21   Anb [Green] (Operation, CompareGreen, Delay, -)
08.03.10 14:53:21  Add [COP_Yellow_Offset] (COP_Yellow, Value_Offset, -, -)
```

```
08.03.10 14:53:21  Tvh [CompareGreen] (COP_Actual, COP_Yellow_Offset, -, -)
08.03.10 14:53:21  Tvl [UnderGreen] (COP_Actual, COP_Yellow_Offset, -, -)
08.03.10 14:53:45 ======================= PLS tracing OFF ==========================
```

- Interpretation:
  Every control function executed is listed according to the time sequence together with its name and the names of the parameters used.
- Recursions / nesting depth:
  When one control function is triggered by another one, e.g. by a value change of a data point, if the result is used as a parameter in another control function(s), this is shown using a variable indentation.

- **Log result and parameter values (before, after)**
  As an additional option, the result as well as the parameter values before and after can also be logged.

  Here is the example from above once again with this option:

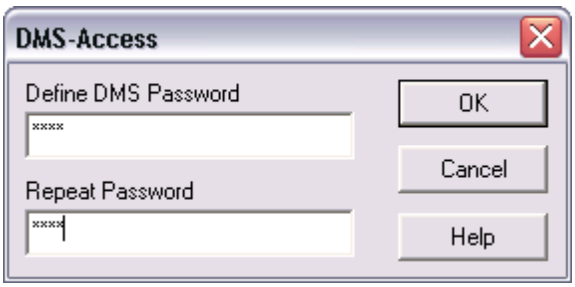
```
08.03.10 15:18:08 ======================= PLS tracing ON ===========================
08.03.10 15:18:20  --> [BIT:ON] (FLT:0.800000, FLT:0.800000, -, -)
08.03.10 15:18:20  Tvh [OverRed] (COP_Actual, COP_Red, -, -)
08.03.10 15:18:20
08.03.10 15:18:20   --> [BIT:OFF] (BIT:OFF, BIT:OFF, -, -)
08.03.10 15:18:20   And [CompareYellow] (UnderGreen, OverRed, -, -)
08.03.10 15:18:20   <-- (And) [BIT:OFF] (BIT:OFF, BIT:OFF, -, -)
08.03.10 15:18:20  <-- (Tvh) [BIT:OFF] (FLT:0.800000, FLT:0.800000, -, -)
08.03.10 15:18:20
08.03.10 15:18:20  --> [FLT:0.805000] (FLT:0.800000, FLT:0.005000, -, -)
08.03.10 15:18:20  Add [COP_Red_Offset] (COP_Red, Value_Offset, -, -)
08.03.10 15:18:20  <-- (Add) [FLT:0.805000] (FLT:0.800000, FLT:0.005000, -, -)
08.03.10 15:18:20
08.03.10 15:18:20  --> [BIT:OFF] (FLT:0.800000, FLT:0.805000, -, -)
08.03.10 15:18:20  Tvl [CompareRed] (COP_Actual, COP_Red_Offset, -, -)
08.03.10 15:18:20
08.03.10 15:18:20   --> [BIT:OFF] (BIT:ON, BIT:ON, BIT:OFF, -)
08.03.10 15:18:20   Anb [Red] (Operation, CompareRed, Delay, -)
08.03.10 15:18:20
08.03.10 15:18:20    --> [FLT:2.000000] (BIT:ON, BIT:OFF, -, -)
08.03.10 15:18:20    Cnt [CounterRed] (Red, Reset, -, -)
08.03.10 15:18:20    <-- (Cnt) [FLT:3.000000] (BIT:ON, BIT:OFF, -, -)
08.03.10 15:18:20
08.03.10 15:18:20    --> [FLT:2.000000] (BIT:ON, BIT:OFF, -, -)
08.03.10 15:18:20    Cnt [CounterGreen] (Green, Reset, -, -)
08.03.10 15:18:20    <-- (Cnt) [FLT:3.000000] (BIT:ON, BIT:OFF, -, -)
08.03.10 15:18:20
08.03.10 15:18:20    --> [FLT:5.000000] (BIT:OFF, BIT:OFF, -, -)
08.03.10 15:18:20    Cnt [CounterYellow] (Yellow, Reset, -, -)
08.03.10 15:18:20    <-- (Cnt) [FLT:5.000000] (BIT:OFF, BIT:OFF, -, -)
08.03.10 15:18:20   <-- (Anb) [BIT:ON] (BIT:ON, BIT:ON, BIT:OFF, -)
08.03.10 15:18:20
08.03.10 15:18:20   --> [BIT:ON] (BIT:ON, BIT:ON, BIT:OFF, -)
08.03.10 15:18:20   Anb [Green] (Operation, CompareGreen, Delay, -)
08.03.10 15:18:20   <-- (Anb) [BIT:ON] (BIT:ON, BIT:ON, BIT:OFF, -)
08.03.10 15:18:20  <-- (Tvl) [BIT:ON] (FLT:0.800000, FLT:0.805000, -, -)
08.03.10 15:18:20
08.03.10 15:18:20  --> [FLT:0.885000] (FLT:0.880000, FLT:0.005000, -, -)
08.03.10 15:18:20  Add [COP_Yellow_Offset] (COP_Yellow, Value_Offset, -, -)
08.03.10 15:18:20  <-- (Add) [FLT:0.885000] (FLT:0.880000, FLT:0.005000, -, -)
08.03.10 15:18:20
08.03.10 15:18:20  --> [BIT:ON] (FLT:0.800000, FLT:0.885000, -, -)
08.03.10 15:18:20  Tvh [CompareGreen] (COP_Actual, COP_Yellow_Offset, -, -)
08.03.10 15:18:20
```

```
08.03.10 15:18:20   --> [BIT:ON] (BIT:ON, BIT:OFF, BIT:OFF, -)
08.03.10 15:18:20   Anb [Green] (Operation, CompareGreen, Delay, -)
08.03.10 15:18:20
08.03.10 15:18:20    --> [FLT:3.000000] (BIT:OFF, BIT:OFF, -, -)
08.03.10 15:18:20    Cnt [CounterGreen] (Green, Reset, -, -)
08.03.10 15:18:20    <-- (Cnt) [FLT:3.000000] (BIT:OFF, BIT:OFF, -, -)
08.03.10 15:18:20   <-- (Anb) [BIT:OFF] (BIT:ON, BIT:OFF, BIT:OFF, -)
08.03.10 15:18:20  <-- (Tvh) [BIT:OFF] (FLT:0.800000, FLT:0.885000, -, -)
08.03.10 15:18:20
08.03.10 15:18:20  --> [BIT:OFF] (FLT:0.800000, FLT:0.885000, -, -)
08.03.10 15:18:20  Tvl [UnderGreen] (COP_Actual, COP_Yellow_Offset, -, -)
08.03.10 15:18:20
08.03.10 15:18:20   --> [BIT:OFF] (BIT:ON, BIT:OFF, -, -)
08.03.10 15:18:20   And [CompareYellow] (UnderGreen, OverRed, -, -)
08.03.10 15:18:20   <-- (And) [BIT:OFF] (BIT:ON, BIT:OFF, -, -)
08.03.10 15:18:20  <-- (Tvl) [BIT:ON] (FLT:0.800000, FLT:0.885000, -, -)
08.03.10 15:18:26 ======================= PLS tracing OFF ==========================
```

Every control function trace line is complemented by additional input/output lines:
"`-->`" Input parameter types and values and
"`<--`" output result (again, with control function name and the unchanged parameter types and values for a better overview for recursive calls).

7.2.10.4 DMS Password

By clicking on the menu line **DMS Password** in the Settings menu, the dialog **Define DMS Access** appears:



By pressing the **<OK>** button without entering a password, the DMS is enabled again.

By defining a DMS password, unauthorised users can be prevented from opening the DMS main window. If a password is defined, a password query appears when the DMS icon is double-clicked on in the Windows Start menu:

The encrypted password is saved in the registry under
HKEY_CURRENT_USER\Software\Saia Visi.Plus 1.6_Promos16

7.2.10.5 Debug Display (changes in value)

 If the **Debug** icon is pressed or the command "**Display Value Change**" is activated, all changes in value are logged in the Debug Display.

```
16:53:57.148   WRITE      : 520Ba:510F:L02:AK:03:ERR_Mec_Trav        -  Value [BIT]: ON (Type: RO)
16:53:57.148   READ TYPE  : 520Ba:510F:L02:AK:03:ERR_Mec_Trav        -  Value [BIT]: OFF (Type: RW Rem)
16:53:57.148   READ       : 520Ba:510F:L02:AK:03:ERR_Mec_Trav        -  Value [BIT]: OFF (Type: RW Rem)
16:53:57.098   WRITE      : 520Ba:510F:L02:AK:03:ERR_Mec_Overload    -  Value [BIT]: ON (Type: RO)
16:53:57.098   READ TYPE  : 520Ba:510F:L02:AK:03:ERR_Mec_Overload    -  Value [BIT]: OFF (Type: RW Rem)
16:53:57.098   READ       : 520Ba:510F:L02:AK:03:ERR_Mec_Overload    -  Value [BIT]: OFF (Type: RW Rem)
16:53:57.058   INFORM     : System:Time                              Value [STR]: 16:53:57 (Type: RO) (ID:2)
16:53:57.048   WRITE      : 520Ba:510F:L02:AK:03:ERR_Damper_Travel   -  Value [BIT]: ON (Type: RO)
16:53:57.048   READ TYPE  : 520Ba:510F:L02:AK:03:ERR_Damper_Travel   -  Value [BIT]: OFF (Type: RW Rem)
16:53:57.048   READ       : 520Ba:510F:L02:AK:03:ERR_Damper_Travel   -  Value [BIT]: OFF (Type: RW Rem)
16:53:56.998   WRITE      : 520Ba:510F:L02:AK:03:ERR_Damper_Temp     -  Value [BIT]: ON (Type: RO)
16:53:56.998   READ TYPE  : 520Ba:510F:L02:AK:03:ERR_Damper_Temp     -  Value [BIT]: OFF (Type: RW Rem)
16:53:56.998   READ       : 520Ba:510F:L02:AK:03:ERR_Damper_Temp     -  Value [BIT]: OFF (Type: RW Rem)
16:53:56.948   WRITE      : 520Ba:510F:L02:AK:03:ERR_Error           -  Value [BIT]: ON (Type: RO)
```

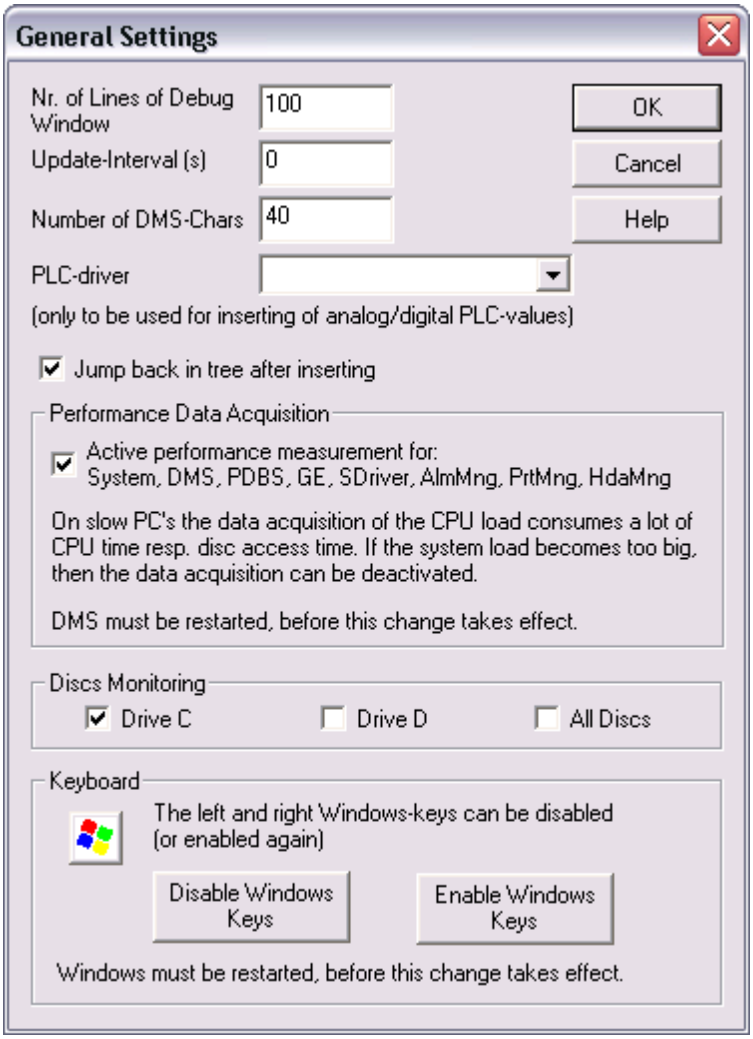
 *Value changes executed manually in the DMS are not logged in the Debug Display. If a change in value is executed through an initialised button in the Graphical Editor, it is shown in the Debug Display. More information can be found in the chapter* ***Graphical Editor****.*

7.2.10.6 Delete Debug Display

All events logged in the debug display are deleted with the "**Delete Debug Display**" command.

7.2.10.7 General Settings

In the "**General Settings**" window, various settings can can be changed which have an influence on the appearance of the DMS.



Number of Lines for the Debug Display:
The number of lines shown in the debug window can be entered in the input field, e.g. 100 lines.

Update Interval(s):
Here you can set the intervals at which value updates of the displayed DMS data should take place. It is every 5 seconds in the example.

The shorter the update interval, the more processing power that is necessary!

Number of DMS characters:
Indicates how many blank spaces should be output after the DMS names. The number must lie between 20 and 80 characters.

PLC Driver:

By clicking on the drop-down menu a list of possible PLC communications drivers is displayed for selection. This is not necessary if data points are integrated directly into the DMS. If the data points are entered in the PET, the **PLC Driver** setting is irrelevant (it is included for reasons of compatibility).

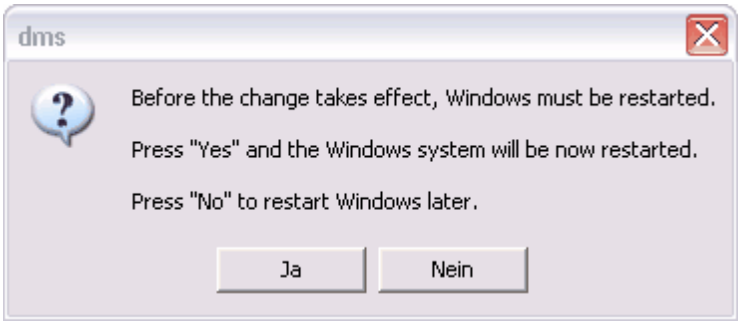
Return to Output in the Tree upon Insertion:
By marking this checkbox, you are automatically returned to the point of origin after insertion of an element.

Performance Trend Logging
When this option is activated, trend logging points are built into the DMS automatically ("System:NT:Perf:ALMMNG:TRD..."), which are used to log the processor time for the respective module as trend data: System, DMS, PDBS, GE, SDriver, AlmMng, PrtMng and HdaMng.
On slower PCs, trend logging consumes a great deal of the CPU capacity as processor time or disk access time. For this reason, trend logging can be deactivated for high system loading. The DMS must be restarted afterward for the change to take effect.

Hard Disk Monitoring
When this checkbox is marked, various hard disk parameters are logged in DMS Tree System:NT:Disk.
Normally, only C: (or the DMS hard drive) should be monitored. Otherwise, for example, if the option "All Disks" is activated and there is a network link and this network drive cannot be reached, the DMS can end up being blocked regularly for minutes at a time because of system timeouts.

Keyboard

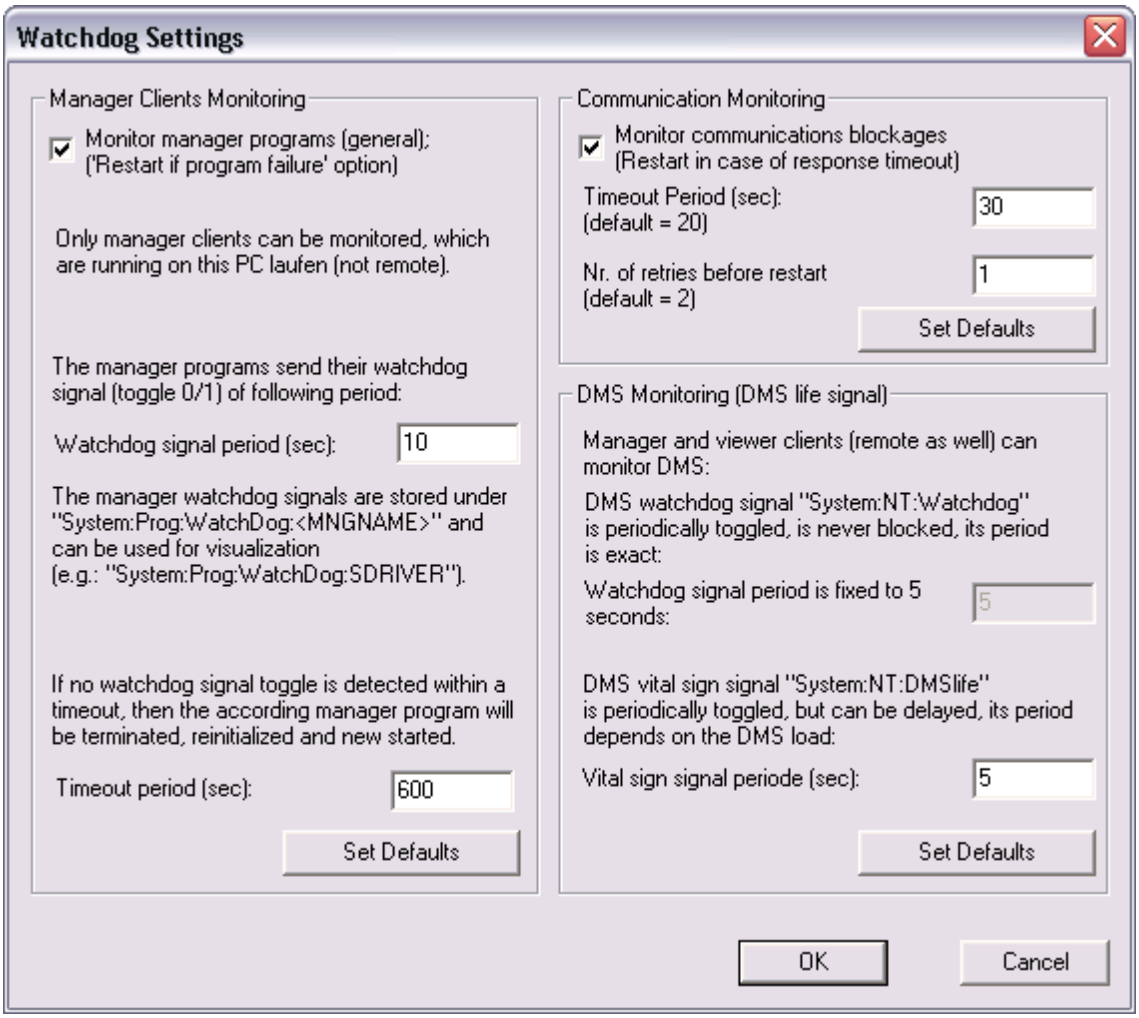
Here the Windows keys can be disabled with the "**Lock Windows Keys**" button (or reactivated with "**Activate Windows Keys**") so that the user does not have access to the Windows taskbar and the start menu.
Since this causes changes in the system registry, Windows must be restarted each time:



The registry entries are made in [HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Keyboard Layout] *and are contained in the following registry files:* enable_winkeys.reg *and* disable_winkeys.reg, *which are located in the default directory "C:\Visi.Plus\cfg\cfg".*

7.2.10.8 Watchdog Settings

In the window "**Watchdog Settings**", various settings can can be changed which have an influence Watchdog monitoring of the Visi.Plus modules. See also **Manager Clients Monitoring Log Settings**



Manager Clients Monitoring

Monitoring Manager Programs (General)

If this option is set, **all** communications drivers and utility programs are monitored by the DMS; they exchange a live bit with the DMS. These settings refer to all Visi.Plus modules and any changes are also immediately effective.

The live bit signal is managed in **System:Prog:Watchdog:<NAME>**; it is either switched periodically 0/1 or can also permanently = 1. If it = -1, the respective module is not monitored.

If no live bit from a monitored module is detected during the watchdog timeout period (300 seconds by default), the program is first terminated (if it is still in the task manager list as an active program, it is assumed to have locked up) and then restarted.

All detected terminations/restarts are logged under "**System:Prog:Watchdog:<NAME>:Restarted/Terminated – <date time>**".

If a monitored program is terminated properly (by the user), it is, of course, removed from the list of monitored programs. That means the value is set as follows:**System:Prog:Watchdog:<NAME> = -1**

The same watchdog settings can also be adjusted in the respective monitored modules (cf. Watchdog settings of the Alarm Manager, for example). These are however only based on the respective associated module. This makes it possible to remove individual modules from monitoring.

Communications Monitoring

Timeout in milliseconds for noncritical data (value change messages for HDA trend logging):
In the case of a write buffer overflow (as a consequence of a CPU/DMS overload), the output buffer is cleared once again after the timeout expires, i.e. the DMS is unblocked at the latest after this time.

Timeout in seconds for critical data (system messages, alarms, etc.):
In the case of a write buffer overflow (as a consequence of a CPU/DMS overload), the output buffer is cleared once again after the timeout expires, i.e. the DMS is unblocked at the latest after this time.

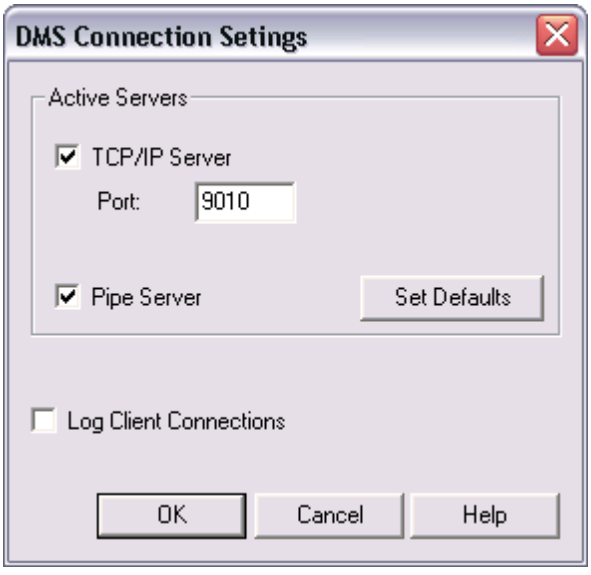
DMS Monitoring

The DMS has its own live bit **System:NT:Watchdog**, which is periodically switched 0/1 and can be used by remote clients (GE, AlmView, PrtView) or other third party programs for DMS monitoring.

7.2.10.9 Connection Settings

By default, the DMS uses port 9010.

In the window "**DMS Connection Settings**", various settings can can be changed, which have an influence on the communication of the DMS.



TCP/IP Server
Must be activated, otherwise the DMS cannot communicate with the remote modules (pChart, SetDMSVal, etc.) via TCP/IP.
The **default port for DMS is 9010** (cf. TCP/IP port for PDBS is 9011).

Pipe Server
Must be activated, otherwise the DMS cannot communicate with any Visi.Plus modules. This option is implemented mainly just for debugging and testing purposes.

Advanced System Settings
Definition of the output queues or of how to communicate with the SDriver asynchronously. These options should be left at their default settings and only serve the purpose of optimisation in highly complex systems.

Log Connections to Clients
If this option is activated, all connections (connection establishment and disconnection) are logged. Only important for debugging and testing purposes.

By default, always use the pipe connections. They are much faster and allow communication over the widest range of network types.
TCP/IP Servers should only be used if the telegrams must be forwarded through firewalls and/or over routers.

If you activate both servers, the pipe connection is established by default.

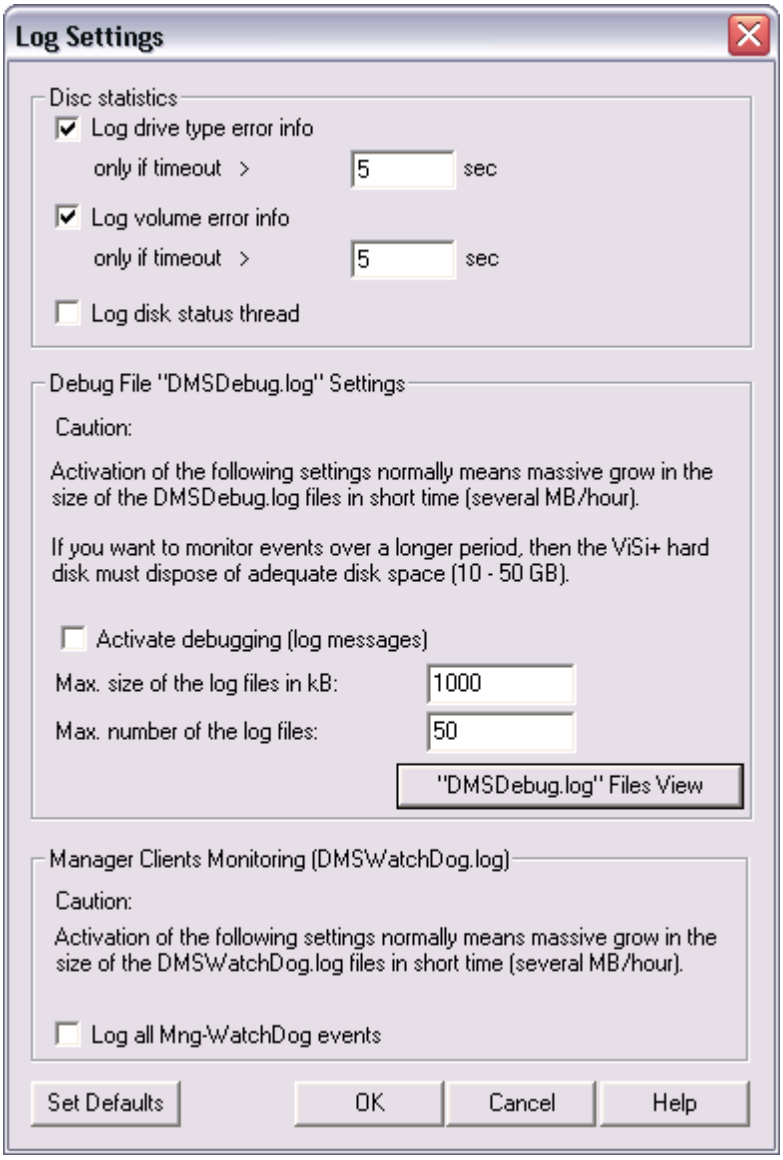
7.2.10.10 Log Settings

In the "Log **Settings**" window, various settings which have an influence on the various logs and debug options for the DMS can be changed.

There are three DMS logs:

- **DMS.log**
  Default log for DMS start/exit: lists all Visi.Plus module versions and licence information on each start-up as well as all incoming communications data requirements and errors.
- **DMSWatchDog.log**
  Special log for Visi.Plus module monitoring (Watchdog) as well as DMS capacity utilisation.
- **DMSDebug.log**
  Special log for DMS structure write and read accesses.

All log files are managed as a FiFo buffer. The size of the individual log files can be defined (recommended: max. 1 MB).
The number of files can also be defined. If more than the defined number of files are present, the oldest files are automatically deleted.



Disk Statistics
Settings for the logging of drive and volume errors to the default **DMS.log.** The DMS regularly scans all drives (including network drives) in order to update the data on the disks to the DMS. This can take a very long time over networks. The timeout times are definable.

Debug file "DMSDebug.log" options
This option only serves for troubleshooting and should be deactivated in normal operation. If this option is activated, all changes in value of the DMS structure are **logged** with a timestamp resolution of milliseconds to DMSDebug.log. Essentially, it is a hard copy of the Debug window.

Example:

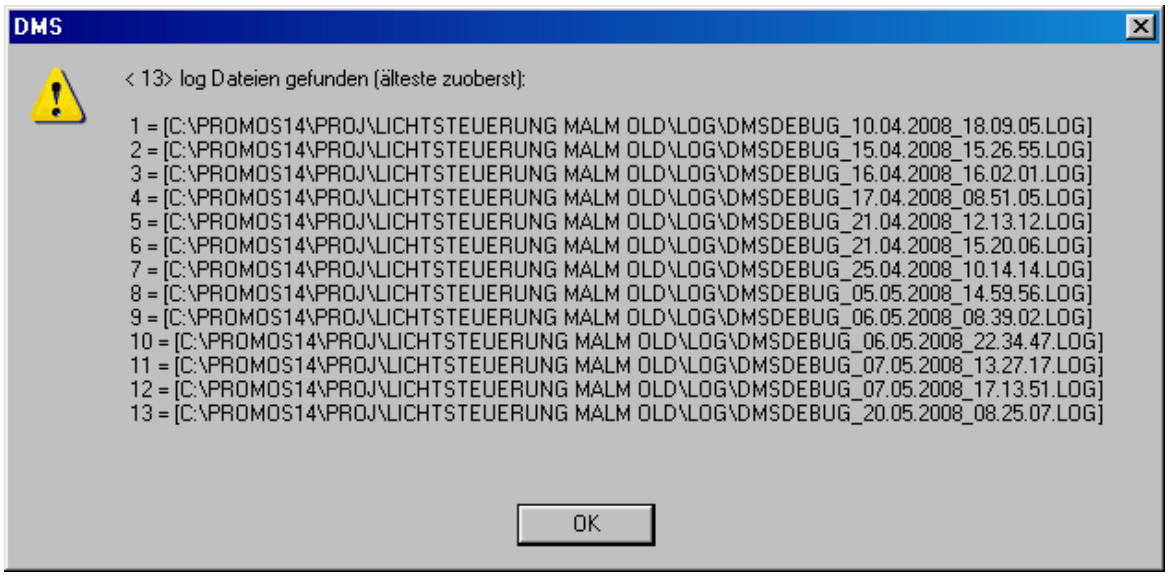
```
20.11.06 13:35:08 --------------- Start Debug ----------------
13:36:00.026   Open pipe conn. [1232]
13:36:00.076   System:Prog:PDBS_UP - READ    : System:License:Productname Value [STR]: Saia
13:36:00.086   System:Prog:PDBS_UP - READ    : System:License:User Value [STR]: MST Systemt
13:36:00.086   System:Prog:PDBS_UP - READ    : System:License:Location Value [STR]: Interna
13:36:00.096   System:Prog:PDBS_UP - READ    : System:License:Copyright  Value [STR]: Saia-
```

```
13:36:00.527   Open pipe conn. [1640]
13:36:00.617   PDBS BCKRD  - READ TYPE       : System:BackupPath Value [STR]:  (type: RW Re
13:36:00.627   PDBS BCKRD  - WRITE               : System:BackupPath Value [STR]: )
13:36:00.627   PDBS BCKRD  - CLOSE PIPE (DMS) : 1640
13:36:00.687   Close pipe conn[1640] [] (msg.message_id = ID_CLOSEPIPE)
13:36:00.707   BrokenConnection[1232] [] (Connection closed)
13:36:00.717   BrokenConnection[1232] [] (Connection closed)
13:36:00.727   System:Prog:PDBS_UP - CLOSE PIPE (DMS) : 1232
13:36:00.777   BrokenConnection[1232] [] (Connection closed)
```

This log option is independent of the status (activated/deactivated) of the **Debug**  icon or the "**Display Value Changes**" command.

Depending on project configuration, the DMSDebug.log becomes full very quickly. With the adjustable max. file size, as well as the number of backup files, the logs can be optimised.
If DMSDebug.log becomes full, meaning its max. size is reached, it is renamed as a backup file (e.g. DMSDebug_05.05.2008_14.59.56.log) and a new version of DMSDebug.log is created.

"DMSDebug.log" Files View
This function shows all versions of existing DMSDebug log files.
Example:



Manager Clients Monitoring (DMSWatchDog.log)

This option is only of interest for troubleshooting; if it is activated, watchdog status changes are logged in **DMSWatchDog.log** for all monitored Visi.Plus modules.

DMS capacity utilisation analysis
Another important entry is logged in DMSWatchDog.log, from which the DMS capacity utilisation can be seen: every 5 seconds the access availability of the DMS structure is logged:

```
"==>WatchDog Timer: CriticalSection busy, no WD decr."
```

Individual entries for different times (not in directly consecutive 5 sec. intervals) have no relevant meaning.

In the following example, an **average load** is shown:

```
.................
25.01.08 09:18:15 ==>WatchDog Timer: CriticalSection busy, no WD decr.
25.01.08 09:32:24 ==>WatchDog Timer: CriticalSection busy, no WD decr.
25.01.08 09:33:53 ==>WatchDog Timer: CriticalSection busy, no WD decr.
25.01.08 09:39:59 ==>WatchDog Timer: CriticalSection busy, no WD decr.
25.01.08 09:47:16 ==>WatchDog Timer: CriticalSection busy, no WD decr.
25.01.08 09:48:30 ==>WatchDog Timer: CriticalSection busy, no WD decr.
.................
```

On the other hand, these messages come every 5 seconds without interruption, e.g.:

```
.................
08.02.08 09:33:42 ==>WatchDog Timer: CriticalSection busy, no WD decr.
08.02.08 09:33:47 ==>WatchDog Timer: CriticalSection busy, no WD decr.
08.02.08 09:33:52 ==>WatchDog Timer: CriticalSection busy, no WD decr.
08.02.08 09:33:57 ==>WatchDog Timer: CriticalSection busy, no WD decr.
08.02.08 09:34:02 ==>WatchDog Timer: CriticalSection busy, no WD decr.
08.02.08 09:34:07 ==>WatchDog Timer: CriticalSection busy, no WD decr.
08.02.08 09:34:12 ==>WatchDog Timer: CriticalSection busy, no WD decr.
08.02.08 09:34:17 ==>WatchDog Timer: CriticalSection busy, no WD decr.
08.02.08 09:34:22 ==>WatchDog Timer: CriticalSection busy, no WD decr.
08.02.08 09:34:27 ==>WatchDog Timer: CriticalSection busy, no WD decr.
.................,
```

this means a maximum DMS capacity utilisation during this period of time.

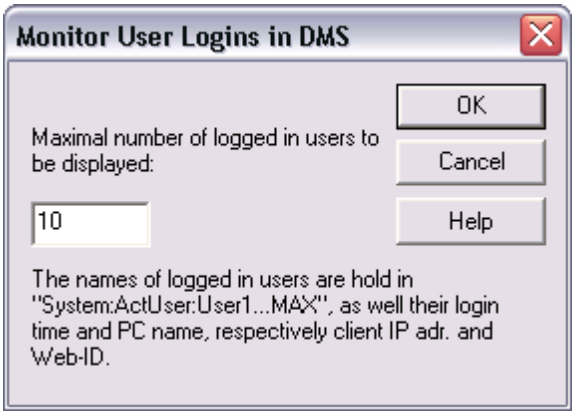
If the max. capacity utilisation periods continue for more than 10–20 minutes, this is practically synonymous with a **DMS blockade**, whose causes should be analysed without fail!

Data Point Type Monitoring

The data type of a data point normally never changes. Nevertheless, it is possible, for example, that the OPC client may determine that the data types do not match and then make the adjustment itself. Data type changes of this sort can be logged.

7.2.10.11 Login Monitoring

Here you can specify the maximum number of users whose logins can be registered while the DMS is running:



The login data for up to the maximum number of users is stored in the DMS (see DMS system data points):

User name in	**System:ActUser:User<no.>**
Login time in	**System:ActUser:User<no.>:LoginTime**
PC name/IP adr. in	**System:ActUser:User<no.>:PC**

where **<no.>** is a number from **1** up to the **max. number** of specified users.
The data where <no.> = 1 concerns the user who has been logged in for the longest; the higher the <no.>, the more recent the login time.

This allows the display of all users currently logged in to the system (e.g. in a list in the GE).

7.2.10.12 Operating Language

This setting is used to select the operating language. The new selection only takes effect after restarting the entire Visi.Plus system. The language setting applies for all Visi.Plus modules.

7.2.10.13 Encoding

Setting the language code pages. Is primarily used for Asiatic languages.

7.2.10.14 Select standard language

An operating language can be defined for every user. If no language is defined, the language defined here is used.

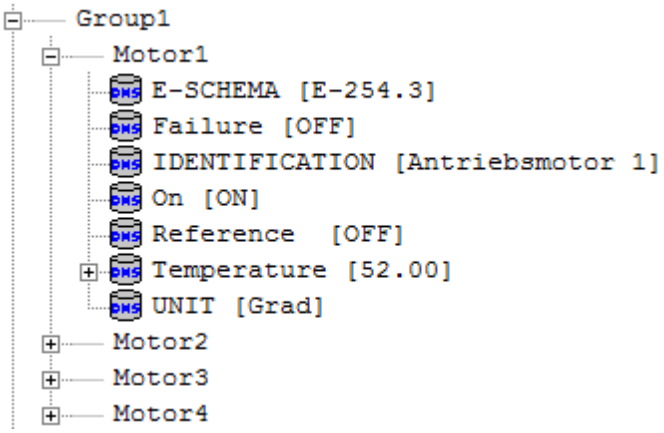
7.2.10.15 Value simulation (Group1)

It is sometimes helpful to have simulated values available without the PLC, e.g. to test graphical elements.

By activating **Value Simulation**, Visi.Plus uses the computer to simulate a series of continuously changing values (time-controlled).

The signal values changing at different intervals can be found as motor objects **Motor1-4** in **Group1**.


If **Group1** is not visible, click on the **System Objects** button.

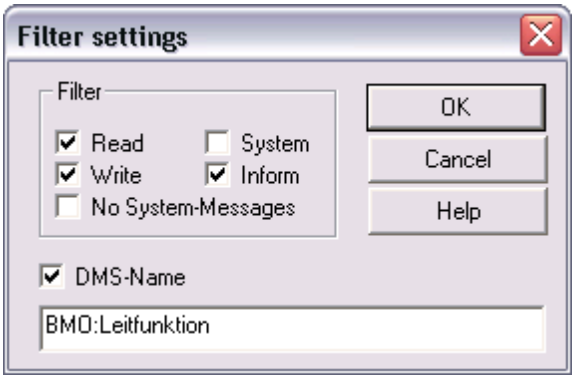


Motor 1 and 2 are simulated, motor 3 and 4 are static.

The simulated values should be deactivated in active systems for performance reasons.

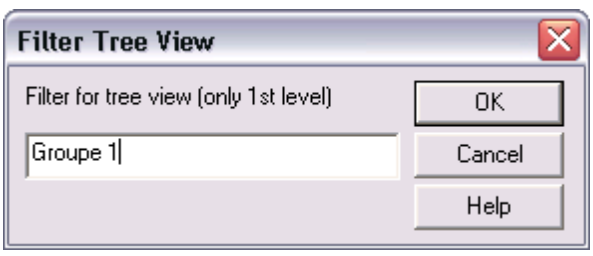
7.2.10.16 Filter Settings

The filter settings enable filter settings for the debug output. Only value changes that can fulfil the filter criteria are shown.



The filter settings defined in the preceding image are only based on the DMS Debug Display. Only the messages with checkboxes marked in the **Filter** section are shown. By entering a DMS name, you can have only the debug messages for a predefined object displayed.
By clicking the **<OK>** button the settings are adopted.

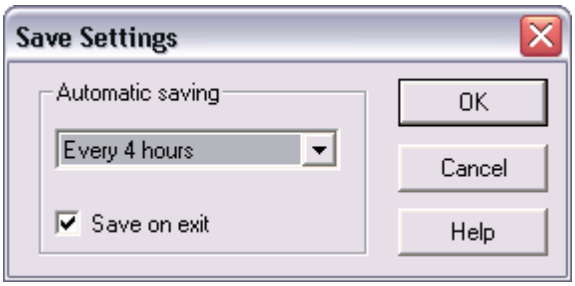
7.2.10.17 Tree View Filter



With this filter setting, only the project data that begins with the indicated name is shown in the tree structure. For very large projects, this can be very helpful so that millions of data points do not have to be displayed.

7.2.10.18 Backup Settings

DMS backups can be automatically scheduled and managed via the Backup Settings. The data in the DMS are only kept in RAM and therefore should be saved regularly. In active systems, a daily backup is useful; any more loads the system unnecessarily.



Automatic Backup:
By clicking on the dropdown menu a desired time for backing up the DMS can be selected.

Backup on Exit:
By marking the checkbox the DMS is automatically saved on each exit.

*By automatically saving the DMS a backup is generated. The backup data is located in the project directory under the path "**\cfg**". The filenames are **promos.bak** and **bmo.bak**.*

The DMS can recognise automatically if larger data volumes have changed (e.g. new data points inserted or many text changes). If a preset limit is exceeded, a backup is performed automatically.

7.2.10.18.1 Restore Backup

*In order to restore the backup, the files **promos.dms** and **bmo.dms** must be renamed (e.g. promos.neu and bmo.neu). Then the backup files **promos.bak** and **bmo.bak** are renamed to **promos.dms** and **bmo.dms**.*

7.2.11 View menu

The "**View**" menu presents different options for modifying the appearance of the DMS.

7.2.11.1 Update

The entire DMS tree structure is rebuilt and displayed.

7.2.11.2 System Object View

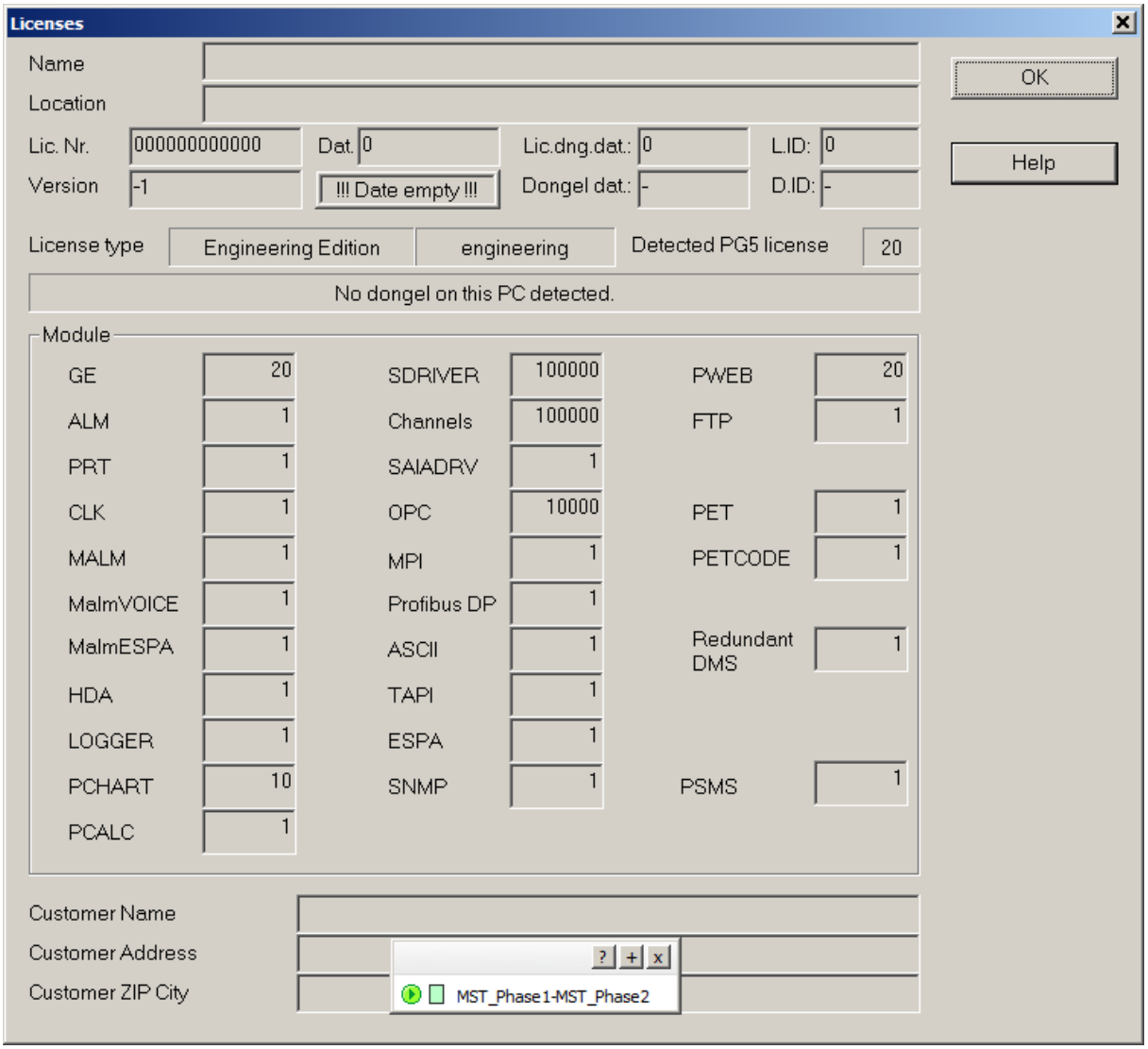
 If the command "**System Object View**" (project data) or the **System Objects** icon in the toolbar is activated, the data which is actually being used is displayed.

7.2.11.3 Template object view

 If the command "**Template Object View**" (**Operating Resources View**) or the **Template objects** icon in the toolbar are pressed, all template objects are shown.

7.2.11.4 Licences

All purchased licence data is shown in the **Licence Display** window.



7.2.11.5 Toolbar

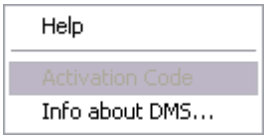
This option switches the toolbar on or off.

7.2.11.6 Status bar

This option switches the status bar on or off.

7.2.12 Help menu

The Visi.Plus online help and current information about the DMS version are found in the "**?**" menu.



7.2.12.1 Help Topics

The Visi.Plus online help can be accessed through the menu item "Help Topics", via the corresponding Help button or by pressing the **<F1>** key.

7.2.12.2 Activation code

See also chapter Installation activation code.

When Visi.Plus is running in demo mode (if there is either no licence or hardware dongle present), activation codes can be entered in this menu. They enable the full functionality of Visi.Plus for a 90-day trial period.

7.2.12.3 About DMS

Here information about the version of the DMS currently in use can be found. The version is important for any support queries or in the forum (see appendix for addresses).

We recommend our internet forum for support queries. This offers two advantages:

- *Help is received more quickly (including outside office hours).*
- *Other users benefit from the entries.*

Version numbers (e.g. 1.6.7.90):

Main number:	1.6
Setup number:	7
Code number:	90

7.3 The PDBS long-term database (pdbs.exe)

The PDBS module is the database which manages the values on the hard drive.
The PDBS stores all data in the current project directory in the folder **dat**.
The database is optimised for data with a timestamp (historical data, alarm data, etc.).
Therefore the PDBS is approx. 100x faster than using ODBC (e.g. Access).
The database also allows the export of data to other databases with the "**pChart**" module and "**AsciiExport**". Then historical data, for example, can be evaluated in a spreadsheet.
The program also offers a debug feature, which shows the data transfer on the screen. This allows you to track every data transfer of the PDBS.

In order for values to be shown in the PDBS, the Log Manager (PrtMng module), the Alarm Manager (AlmMng module) or the Historical Database (HDAMng module) must be started (yellow, dark red or violet dot on the Windows taskbar).

7.3.1 Starting the program

The PDBS is automatically opened every time the Visi.Plus system starts up. The module normally runs as a background process. This is visible as a green ring icon on the Windows taskbar (bottom right at the edge of the screen).

In order to access the module, click with the **right mouse button** on the "**green dot**" and select the command **Display**. The window for the PDBS is displayed.

The PDBS program can, depending on PC performance, process up to more than 1,500 messages per second (debug mode switched off).
An active virus scanner can significantly influence this performance. If the system is not connected to the Internet and no users install any programs, we recommend deactivating the virus scanner (or to explicitly exclude the DAT directory).

7.3.2 Data storage

A lifetime can be defined for the data for every data point (LIFETIME). This time refers solely to the time over which the data remain stored in the current project directory.
The default value is 31 days. This is actually sufficient in all known cases. For data-intensive systems (e.g. electrical current measurement systems with a lot of counter data), the lifetime should be reduced (e.g. seven days). This can prevent the individual files in the DAT directory from becoming too large (and consequently, the access too slow).
If a backup is defined (see Data backup), the lifetime plays a subordinate role for the user because when scrolling in the historical data, a switch to the backup directory occurs automatically. The user won't notice the lifetime at all (in practice, access on the same PC is a few milliseconds slower and via the network, a bit slower yet, but hardly noticeable.)
The backup can be configured for any drive (including network drives). The backup directory can also be backed up with any commercially available backup software because, normally, Visi.Plus does not access these data (exceptions: users analyse older data pools or a year-long or multi-year analysis is run using pCalc). All access is read only.

It is recommended that the backup directory not be defined in the current project directory, because the data volume for a project backup can become very large (several gigabytes).

7.3.3 General information for communication between the modules

All communication between the individual Vis.Plus modules takes place exclusively through Intertask communication. The DMS provides two different interfaces:

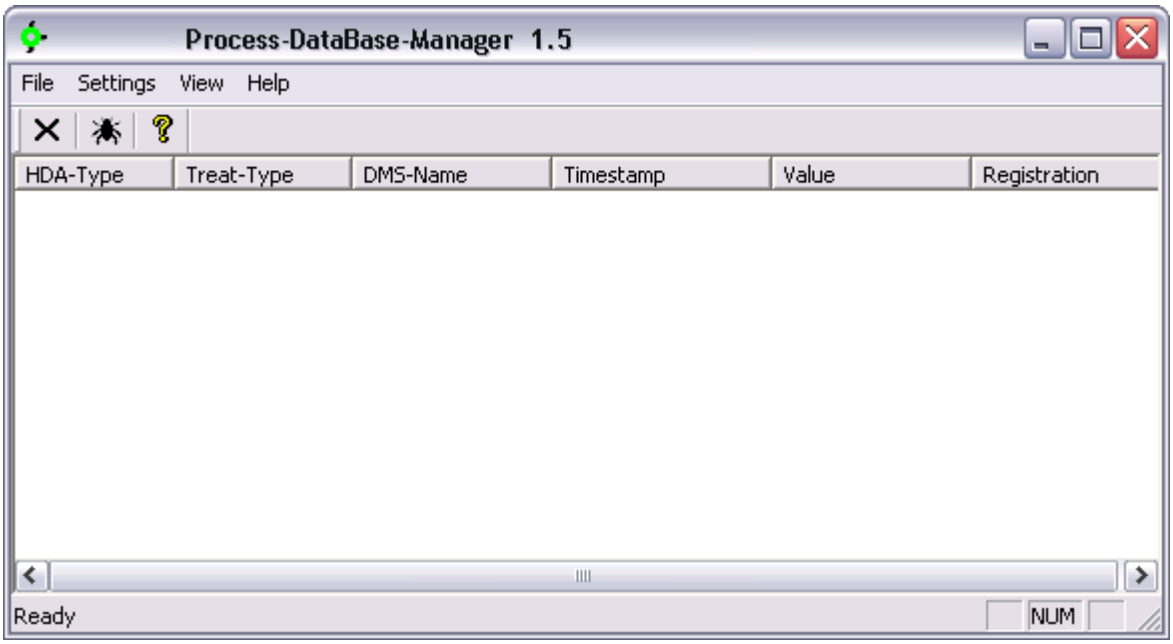
- Communication through pipes
- Communication through sockets (TCP/IP)

In both the DMS and PDBS, two different communications servers are implemented which provide access to the respective service after a connection is established.

The DMS/PDBS programs generate an independent communications thread for each connection, which responds to all of a program's queries. The DMS/PDBS programs cannot establish connection on their own. With registered data points, the values are automatically transmitted to the client (event-oriented).

For information on establishing communication: see the chapter DMS Connection Settings.

7.3.4 The PDBS user interface



Here are the window contents of the long-term database (PDBS).

1 **Title bar**
Contains the name of the current window

2 **Menu bar**
The menus are sorted according to basic functions. Functions for configuring the PDBS are found under the "**Settings**" menu. Under "**View**", the view of elements on the screen can be defined.

3 **Toolbar**
Displays icons for some important commands and functions that can be selected in **PDBS**

4 **Heading Line**
Names the columns of your data points

5 **Debug View**

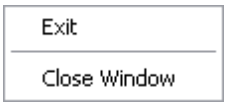
All registered data is displayed here sequentially.

Status Bar

Displays the module (program) status and the meaning of the buttons in the toolbar.

7.3.5 File menu

Various options for exiting and closing the PDBS are available in the "**File**" menu.



7.3.5.1 Exit

In order to exit the **PDBS** module, all modules registered with the PDBS must first also be exited. The PDBS normally never needs to be exited because the PDBS is automatically run down when the DMS is exited.

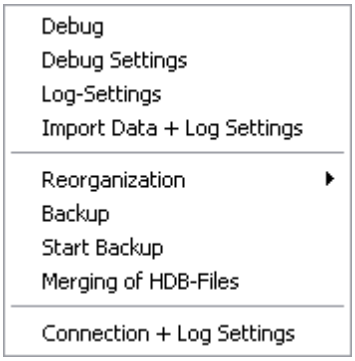
 As long as programs are still registered with the PDBS, it cannot be exited! A corresponding message will notify you of this.

7.3.5.2 Close Window

 Closes the control window of the PDBS. This does not exit the PDBS module; it is only minimised to the taskbar.

7.3.6 Settings menu

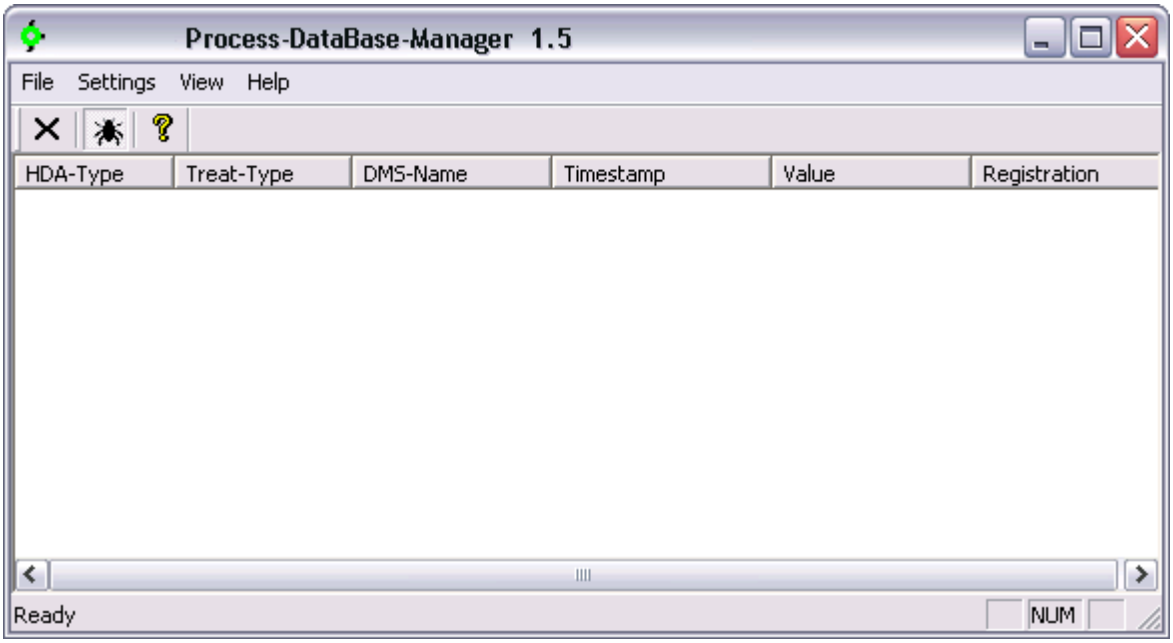
The "**Settings**" menu offer various options for settings, value displays and backing up the PDBS.



7.3.6.1 Debug

 In order to display on screen all the messages registered by the PDBS, the command **Debug** must be selected or the **Debug** icon clicked on the toolbar. Now it is possible to track each data transfer of the PDBS on screen.

 *In order for values to be shown in the PDBS, the Log Manager (PrtMng module), the Alarm Manager (AlmMng module) or the Historical Database (HDAMng module) must be started (yellow or violet dot on the Windows taskbar).*



The following describes the columns of the **heading line**.

HDA Type
Indicates what command type was used to register the event.

Data Request
Data is requested.

HDA Handler
Data is exchanged.

HDA Main
Data is read in on startup.

Data Import
Data is imported.

Treat Type
Indicates which command type was executed.

Create
Data is created.

Append
Data is appended.

Read
Data is read.

Lifetime Check
Comparison has occured.

Create Path
A new path is created.

Open File
A file is opened.

Close File
A file is closed.

Is File Open
Queries whether a file is open.

Get Size:
A file is queried as to its size.

Move:
Data is read into another location.

Move Next:
Data is read into the next entry.

Move Prev.:
Data is read into the previous entry.

Set DMS Filter:
A filter is set in the DMS.

Set Text Filter:
A text filter is set.

Clear Filter:
A filter is deleted.

Move Time:
It is set to an absolute time.

DMS Name
Indicates DMS names or log names.

Timestamp
Displays the current date and time.

Value
Displays the current value of a data point.

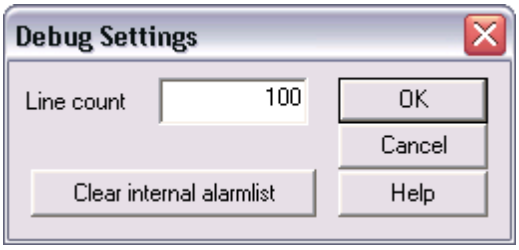
Logging:
Column shows the actions currently being performed.

Cyclical Save:
The save operation is performed cyclically.

New Data:
New data is created.

New File:
A new file is created.

7.3.6.2 Debug Settings

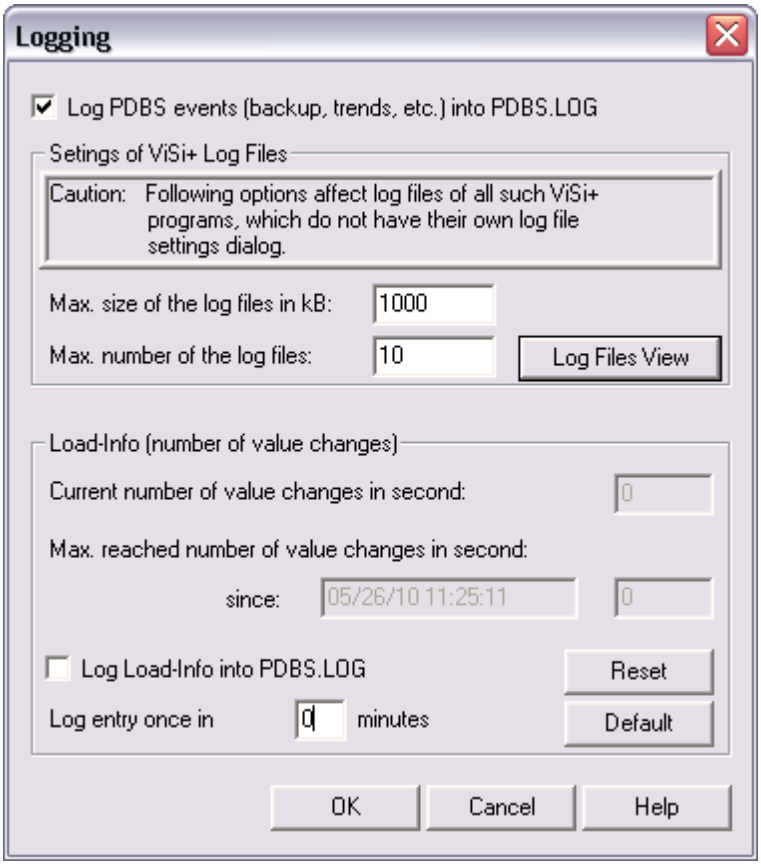


Debug Lines: Here the maximum number of messages is indicated that should be displayed in the Debug window.

<Delete internal alarm list> deletes all alarms which are still present in the internal buffer. The Alarm Manager forwards the alarms to the PDBS. The Alarm View, in turn, reads the alarms from the PDBS. If communication is then interrupted and an alarm is acknowledged on the PLC during this interruption, the alarm cannot be acknowledged in the PDBS.

7.3.6.3 Log Settings

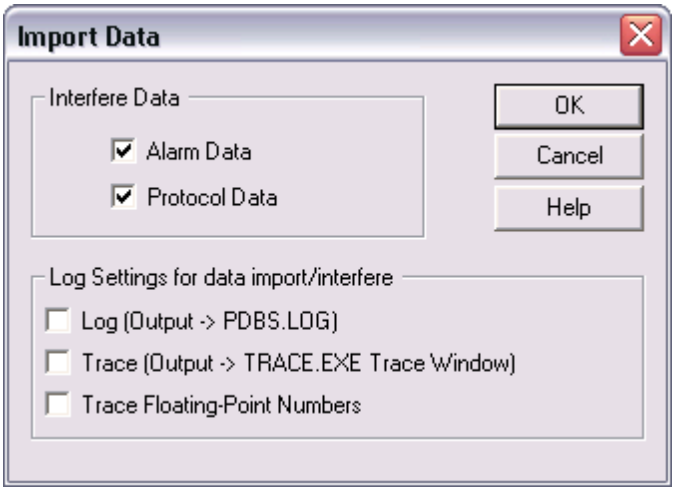
Various PDBC functions can be output in a log file.



These settings are only used for debug purposes and troubleshooting.

7.3.6.4 Import Data and Log Settings

Here you can configure additional debug settings. In addition, in the "**Merge Data**" section, you can configure whether alarms or logs from an ASCII file should be processed in the PDBS.



Merging data allows you to import external data (e.g. via modem connections). The importing itself occurs by means of a PDBS system call (program function call) and cannot be performed manually.

7.3.6.5 Reorganise

This reorganises all data created by the PDBS.

As part of this the following functions are executed.

- Test of whether meaningful date/time data is present. If not, the data set is deleted.
- Test of whether the data is ordered by date/time. If not, the whole file is resorted.

It is not usually necessary to carry out reorganisation. However, after a power failure or a problem with the hard drive, we recommend that this function be executed.
The function can be executed while the program is running normally.

HDB files in the DAT directory
The data of the last 31 days are reorganised. Normally, this is sufficient in the event of problems.

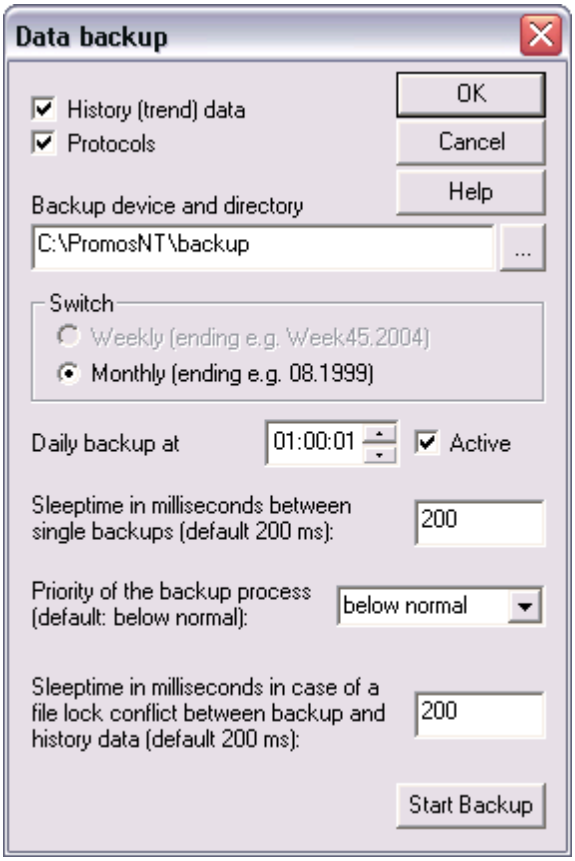
HDB files in the backup directory
All backup data are reorganised. This can take up to several hours (depending on the size of the backup).

All HDB files
The data in the DAT directory and the data in the backup directory are reorganised.

7.3.6.6 Data backup

In the Backup dialog, you can specify how and where data should be backed up on the hard drive.

A lifetime is defined for the trend data and the logs in the PET (normally 31 days).
After expiration of the lifetime, the data is overwritten (function of a ring buffer)!
In order to back up data permanently, backup files must be continuously generated.
For the user, access to the backup data is transparent, meaning they will not notice that the data is being fetched from backup storage (unless it is contained on network drives or other external data carriers).
The lifetime setting should not be increased, otherwise the files may become very large.



Trend data and/or
Logs determine from which data a backup should be generated.

Backup Drive and Directory holds the path of the storage location in which the backup files should be saved. The path can be selected by clicking on the button to the right of the input field. It is recommended that the backup directory not be defined in the current project directory, because otherwise a project backup can become very large (several gigabytes).

Split all Logs
The log data are managed in the same manner as the historical data (lifetime, 31 days). If the option is not active, all log data is stored in the same file (the file can become very large).

Daily Backup At, defines the time at which the backup should start.

Active activates daily backup.

Sleep time is the interval between the backup of individual files. This means that the system is not required to operate at 100% of full capacity during backup.

Priority
System priority of the backup task. A high priority means that the backup is processed quickly, but it is possible that data could be lost because they cannot be processed. With a lower priority, the backup runs more slowly, but hardly impacts the current processes. Recommendation: "under normal".

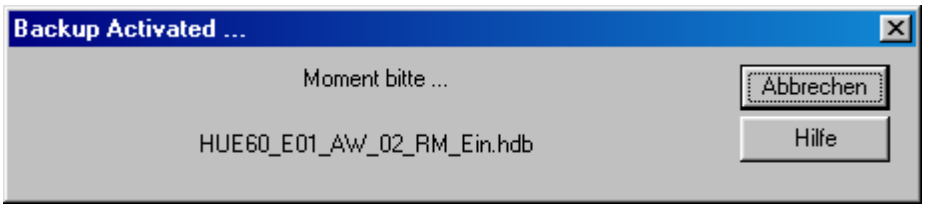
Sleep Time
In the event of a conflict (HDA data are currently being saved), the backup waits for the time set and then attempts once again to access the data.

Start Backup
The backup can be started at any time using this button.

7.3.6.7 Starting backup

The backup process described above is started manually (if activated).

A dialog window, which displays the file currently being backed up, appears.
The backup process can be stopped (interrupted) at any time.



Due to the load placed on the system, it is recommended that backup be performed overnight.

7.3.6.8 Inserting HDB files

HDB files can be inserted in the corresponding current HDB project files, whereby the insertion point takes the time of creation into account.

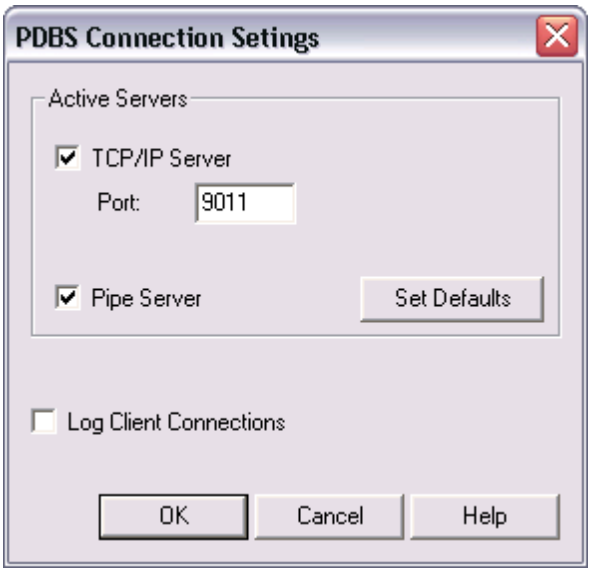
These functions can be used, for example, when data from several projects must be analysed together.

7.3.6.9 Connection and Log Settings

The PDBS allows access via so-called socket connections.

In the window "**PDBS Connection Settings**", various settings can can be changed, which have an influence on the communication of the PDBS.

By default, the PDBS uses port 9011.



TCP/IP Server
Must be activated, otherwise the DMS cannot communicate with the remote modules (pChart, SetDMSVal, etc.) via TCP/IP.
The **default port for the PDBS is 9011** (cf. TCP/IP port for the DMS is 9010).

Pipe Server
Must be activated, otherwise the DMS cannot communicate with any Visi.Plus modules. This option is implemented mainly for debugging and testing purposes.

Log Connections to Clients
If this option is activated, all connections (connection establishment and disconnection) are logged. Only important for debugging and testing purposes.

By default, always use the pipe connections. They are much faster and allow communication over the widest range of network types.
TCP/IP Servers should only be used if the telegrams must be forwarded through firewalls and/or over routers.

If you activate both servers, the pipe connection is established by default.

7.3.7 View menu

In the View menu, you will find the display settings for the PDBS.

7.3.7.1 Toolbar

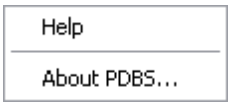
Switches the toolbar on or off.

7.3.7.2 Status bar

Switches the status bar on or off.

7.3.8 "?"/Help menu

The Visi.Plus online help and current information about the PDBS version are found under the "**?**" menu bar.



7.3.8.1 Help Topics

The Visi.Plus online help can be accessed through the "**Help Topics**" menu, via the corresponding **<Help>** button or by pressing the **<F1>** key.

7.3.8.2 About the PDBS

Here you will find the version number of the PDBS, among other things. It is important and must be specified for any support queries!

We recommend our internet forum for support queries.This offers two advantages:

- *Help is received more quickly (including after office hours)*
- *Other users benefit from the entries*

Saia Visi.Plus

Utility programs

Chapter 8

8 Utility programs

8.1 Summary

This chapter contains instructions for the various utilities of Visi.Plus. Utilities are modules that make all types of alarms, as well as logging and time switch functions, available from the computer, etc.

- **Alarm Management (AlmMng.exe)**
  How alarms are inserted and registered
- **Remote Alarm Management (MalmMng.exe)**
  How remote alarms are configured, inserted and set off
- **Historical Data Acquisition (HDAMng.exe)**
  How trend data is registered and saved historically
- **Logging (prtmng.exe)**
  How logs are defined, inserted and registered
- **The Time Switch Program (clkmng.exe)**
  How switch times can be managed
- **The Web Server (PWEB.exe)**
  How data can be viewed over the internet
- **The FTP Server (pFTP.exe)**
  How data can be transmitted over a File Transfer Protocol server
- **The Logger (Logger.exe)**
  How data can be cyclically logged
- **Alarm logging via SMS (pSMS.exe)**
  How alarms that are sent via SMS are inserted and registered.

8.2 The Alarm Manager (AlmMng.exe)

At startup, the Alarm Manager registers in the DMS all the data points which require monitoring. The DMS in turn reports all changes in the values of the data points to the Alarm Manager. Each message from the DMS is compared with the defined limit values for the alarms. If a value is exceeded or not reached the Alarm Manager inserts an entry into the PDBS.

8.2.1 Alarm management

The alarm management program logs and manages all fault messages entered in the PET in the column **Alarm**.
For Visi.Plus, alarm management is shared between two programs:

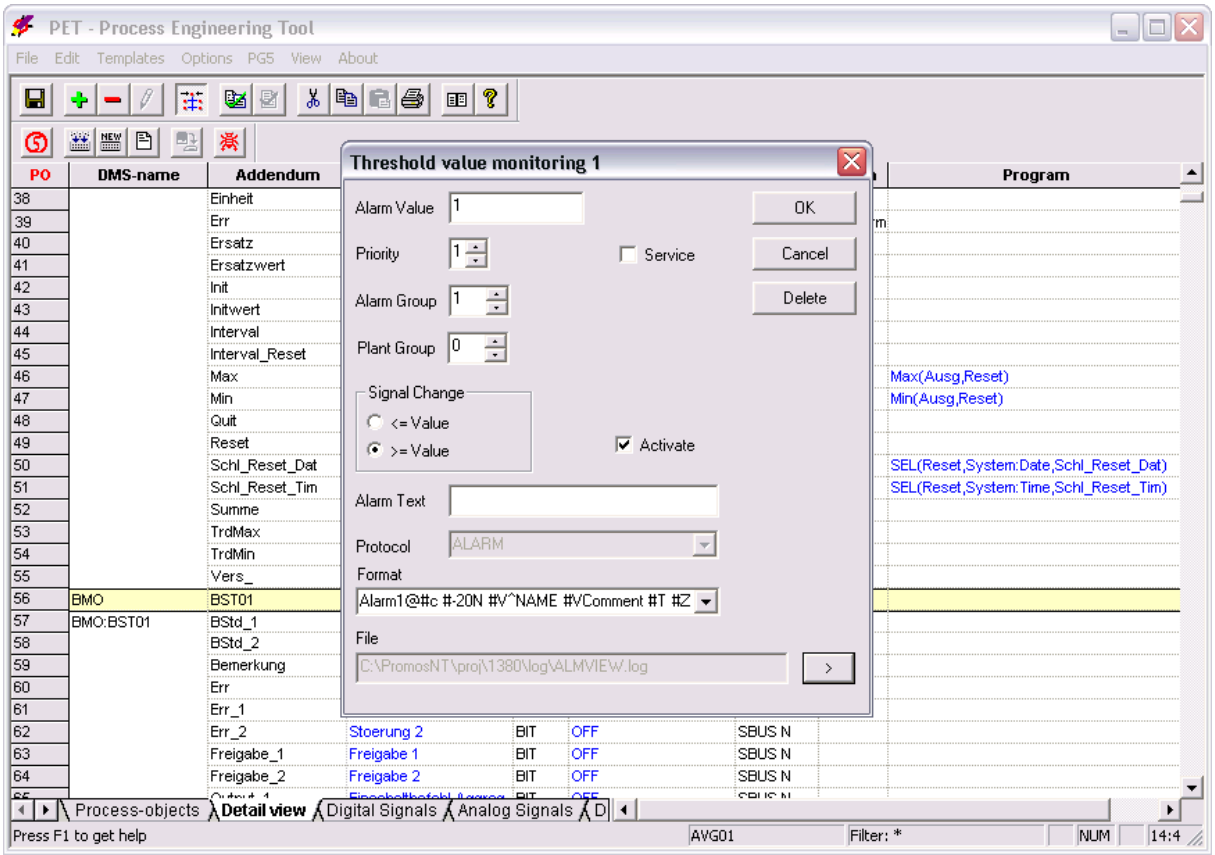
- **Alarm Manager (AlmMng)**
  The Alarm Manager is located on the computer where the DMS is installed. It monitors all alarm limit values. If an alarm occurs, the Alarm Manager reports it to the Visi.Plus database (PDBS). The database, in turn, saves the data to storage.

- **Alarm Viewer (AlmView)**
  The Alarm Viewer lists all alarms which occur and provides the opportunity to acknowledge them. More information about the Alarm Viewer can be found in the chapter The Alarm Viewer.

8.2.2 Defining an alarm

An alarm is entered in a template object attribute. To do so, proceed as follows:

- Open the PET (tip: at least the DMS must be started in advance)
-  Select the template object view 
- Search for the data point and click on the same line in the column "**Alarm**". The window **Limit value monitoring 1** appears.

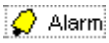


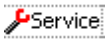
Explanation of the fields of the current window:

Limit value
Triggers an alarm when the indicated value is exceeded/not reached.

Priority
of the alarm (1 is the highest priority).

Maintenance
When activated, the **File** section is enabled and a directory can be specified where the maintenance file (log file) will be stored.

The symbol 

becomes the symbol: 

Alarm group
Here an alarm group can be assigned. In doing so, there is the possibility of assigning all alarms of a station to the same alarm group, thus generating collective alarms.

System group
When defining the user in pUser, system levels can be determined. The system levels, together with the system group to be determined here, serve as a filter in the Alarm Viewer. An alarm is then only shown to the user if the system level corresponds to the system group of the alarm. For example, if Alarm group 2 has been specified for the alarm, it will only be shown to the logged-in user if System level 2 is available to them. If the value 0 is entered, the alarm can be viewed by any user.

Signal transmission
Defines whether an alarm should be triggered when the limit value is exceeded or not reached.

Activated
When activated, monitoring of the data point is enabled.

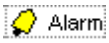
Alarm text
Enables the input of the desired alarm text, which can be displayed in the Alarm Viewer.

Log
In the current version of Visi.Plus, all alarms are logged in the standard, predefined alarm log. No other log can be selected.

Format
Applied by selecting a "predefined format" (see PRTFormat).


The Alarm is saved in the PDBS Database. The whole expression can't be longer than 140 characters.

File
With maintenance alarms, a file can be indicated for the alarm, which contains the information for the maintenance to be carried out. This can be called up in Alarm Viewer.


All attributes (e.g. measurement values) displaying the symbol 

are monitored (when activated).

All attributes (e.g. measurement values) displaying the symbol Service are maintained (logged) ***in addition to*** the monitoring of the corresponding directory.

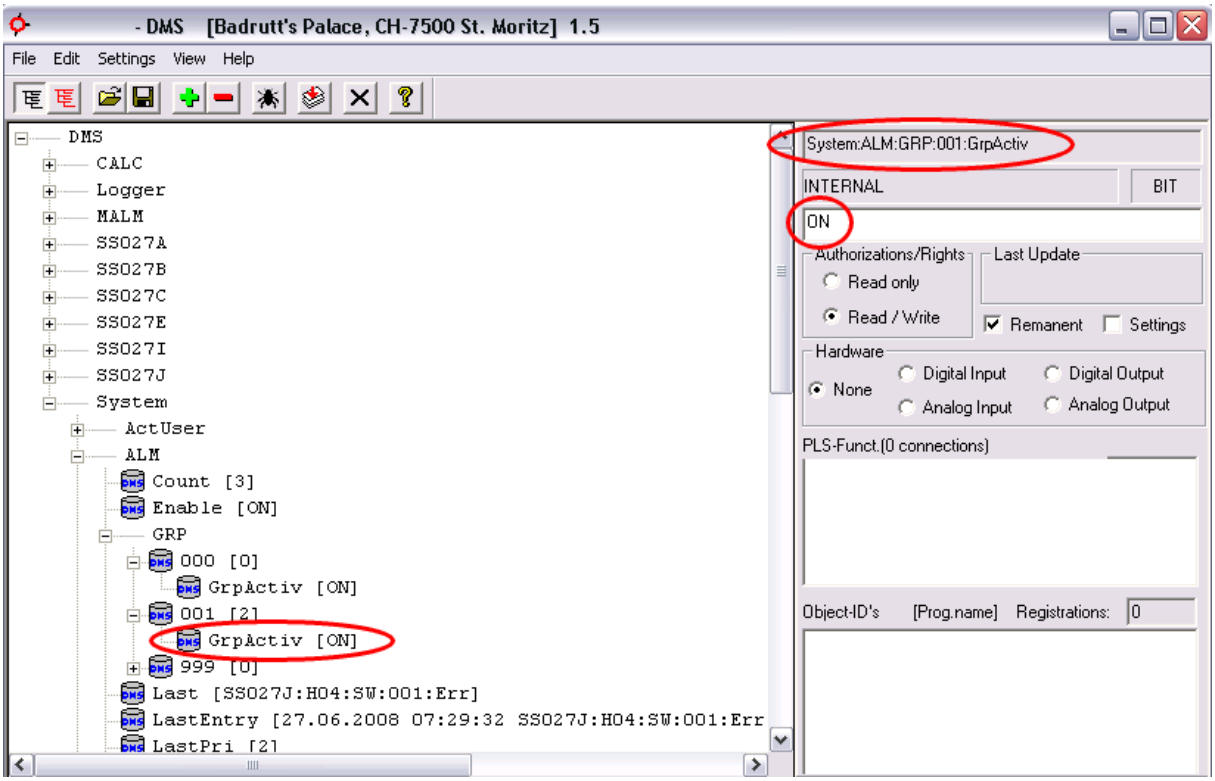
8.2.3 Deactivating alarms (temporarily)

When conducting various types of service work on the system (e.g. motors, pumps, etc.), temporary deactivation of the relevant alarms (or even all of the system's alarms) is advantageous.

For this purpose you can visualise the following data points (e.g. on button object):

- Deactivate all system alarms: **System:ALM:Enable** set to **OFF** (=0)
- Deactivate individual alarm group(s): e.g. set **System:ALM:GRP:001:GrpActiv** to **OFF** (=0)

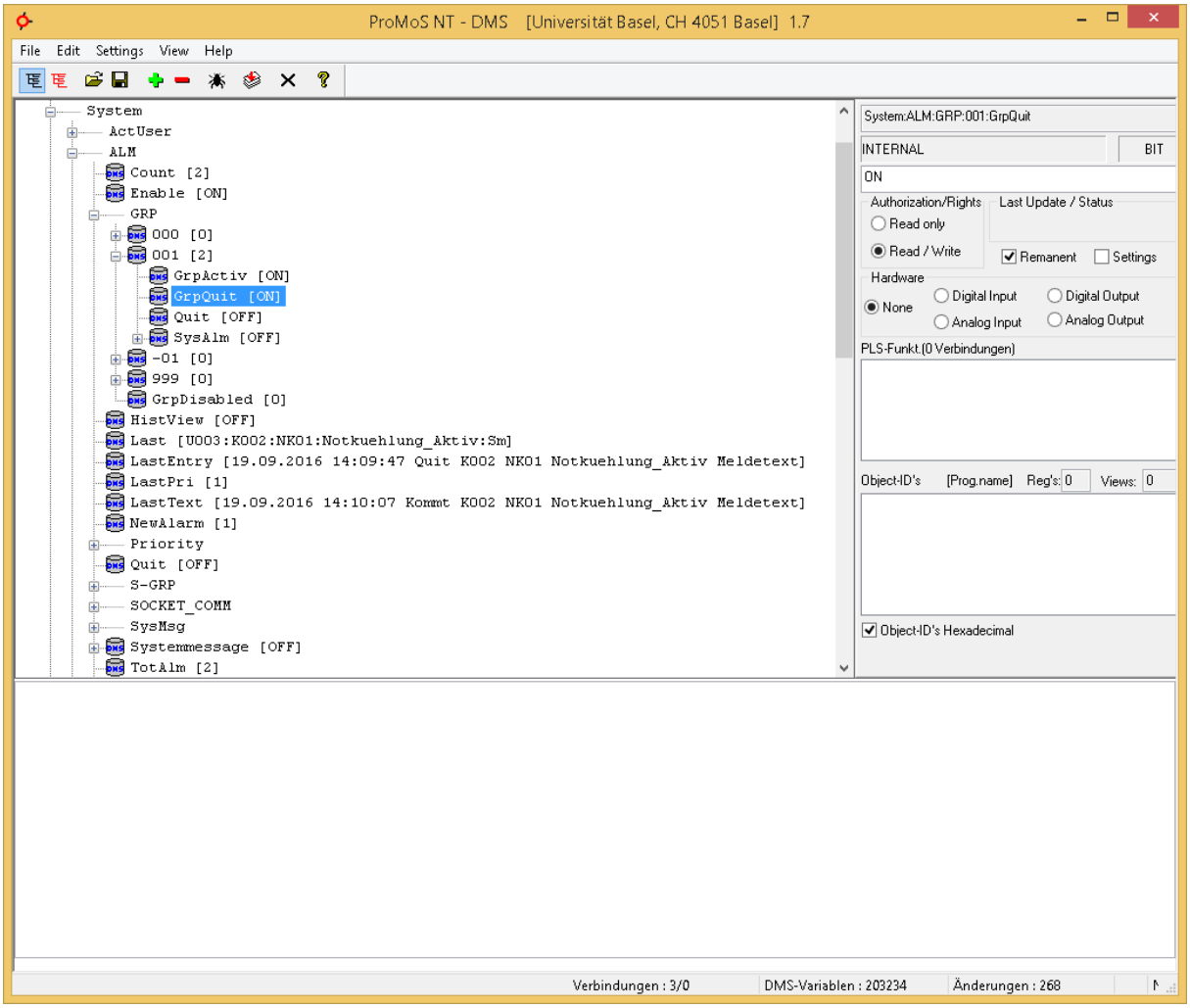
Normally values are set for these data points (ON = true) and the alarms are relayed; if they are set to OFF = false (as above), the alarms are only displayed (**IGNORED status**).



8.2.4 Alarm group

It is possible to group alarms from a zone, building, controller, etc. Each alarm has alarm groups and collective alarm groups to enable this. It is then possible to deactivate the alarms in a group, such as during servicing or commissioning work. The alarms are suppressed. The alarm group and the collective alarm group form an OR connection. If an alarm data point is assigned to a deactivated group the alarm is suppressed.

Alarm and collective alarm groups can be deactivated or acknowledged. If a group is deactivated, the existing alarms are removed from the viewer. The entries are still visible in the historic alarms.
If the group is reactivated, the alarms are tracked again. Any due status changes are automatically updated.



The alarm groups and collective alarm groups are all stored in the DMS under System:Alm:Grp or System:Alm:S-GRP. The structure is identical for every group. Each group has the following data points:

GrpActive: determines whether or not the group is activated. If this data point is switched to 'Off', alarms from this group will no longer be displayed or transmitted via the AlmMng.
GrpAck: If this data point is set to 'Off', all pending alarms in this group are acknowledged.
SysAlm: This is the alarm data point of the group. If an alarm group is deactivated, it is possible to trigger an alarm if desired! It is triggered if the flag GrpActive is switched to 'Off'.
Ack: This is the acknowledgement flag of SysAlm.

It is possible to define 999 alarm groups and 999 collective alarm groups.

8.2.5 Alarmdatapoints

A range of data points related to the alarms are stored in the DMS under System:Alm.

System:Alm:Enable:
This flag can be used to globally deactivate the alarms. No alarms will be transmitted if the data point is set to OFF (=0).
The values of this data point are normally set (ON = true) and the alarms are transmitted. If the values are set to OFF=false (as above), the alarms are ignored (**Status IGNORED**).

HistView:
AlarmViewer is being updated. This data point is set or reset by the system, and should not be changed manually.

Last:
The last coming alarm data point is stored here.

Last:Entry:
The alarm text of the last changed alarm is stored here. All status changes are considered.

LastPri:
The priority of the most recently changed alarm is stored here.

LastText:
The alarm text of the last incoming alarm is stored here. Only a status change to incoming is considered.

Priority:
Here, all pending alarms (incoming and acknowledged) are counted per priority under **APCount**, with all unacknowledged alarms counted under **APNewAlarm**.

Count:
Counter for all pending alarms. Alarms with incoming and acknowledged status are counted together.

NewAlarm:
Counter for all unacknowledged alarms. Alarms with incoming and outgoing status are counted together.

TotAlm:
Counter for all alarms together. Alarms with the status incoming, outgoing and acknowledged are counted together in this data point. The data point is intended for visualisation in GE.

SysAlm:
This is used by the system if a driver has not yet started.

UpdAlmView:
AlarmViewer is being updated. This data point is used by the system, and should not be changed manually.

View:
This data point is used by the web, and should not be changed manually.

8.2.6 Start behaviour

The start behaviour of AlarmMng can be configured. Different settings are available in the cfg file:

USE_HISTORY:
If USE_HISTORY=0, the historical alarms are ignored and the state of the DMS is adopted. The historical state of the alarm before the system start is ignored.

IGNORE_QUIT:
If active, the quit flag is ignored at startup.

WAIT_FOR_DRIVER:
Display of the corresponding alarms is only activated when the driver has started.

TIME_SYSTEMMESSAGE:
This is a time specification in minutes. This setting only has an effect when WAIT_FOR_DRIVER is active. If an alarm is active but the driver has not yet started, an alarm message is displayed in the viewer after the set time. The message is deactivated with 0.

Alarms that are not displayed can be deleted in AlmMng under File-> Statup alarms.

Example 1: (standard setting)

USE_HISTORY=0
IGNORE_QUIT=0
WAIT_FOR_DRIVER=0
TIME_SYSTEMMESSAGE=0

	State in AlarmViewer before downloading	State after starting.
Alarm 1	incoming	incoming
Alarm 2	outgoing	-
Alarm 3	quit	quit

Outgoing alarms are not displayed again! The time stamp is set by the system start. As a result, 2 incoming alarms appear in the historical view of alarm 1 and 3 outgoing alarms for alarm 3.

Example 2:

USE_HISTORY=0
IGNORE_QUIT=1
WAIT_FOR_DRIVER=0
TIME_SYSTEMMESSAGE=0

	State in AlarmViewer before downloading	State after startup.
Alarm 1	incoming	incoming
Alarm 2	outgoing	-
Alarm 3	quit	incoming

Acknowledged alarms are displayed as incoming again and outgoing alarms disappear.

Example 3:

USE_HISTORY=1
IGNORE_QUIT=1
WAIT_FOR_DRIVER=0
TIME_SYSTEMMESSAGE=0

	State in AlarmViewer before downloading	State after startup.
Alarm 1	incoming	incoming
Alarm 2	outgoing	-
Alarm 3	quit	quit

Outgoing alarms are not displayed again. The time stamps of incoming and acknowledged alarms remain.

Example 4:

USE_HISTORY=1
IGNORE_QUIT=0
WAIT_FOR_DRIVER=0
TIME_SYSTEMMESSAGE=0

	State in AlarmViewer before downloading	State after startup.
Alarm 1	incoming	incoming
Alarm 2	outgoing	-
Alarm 3	quit	quit

Outgoing alarms are not displayed again. The time stamps of incoming and acknowledged alarms remain.

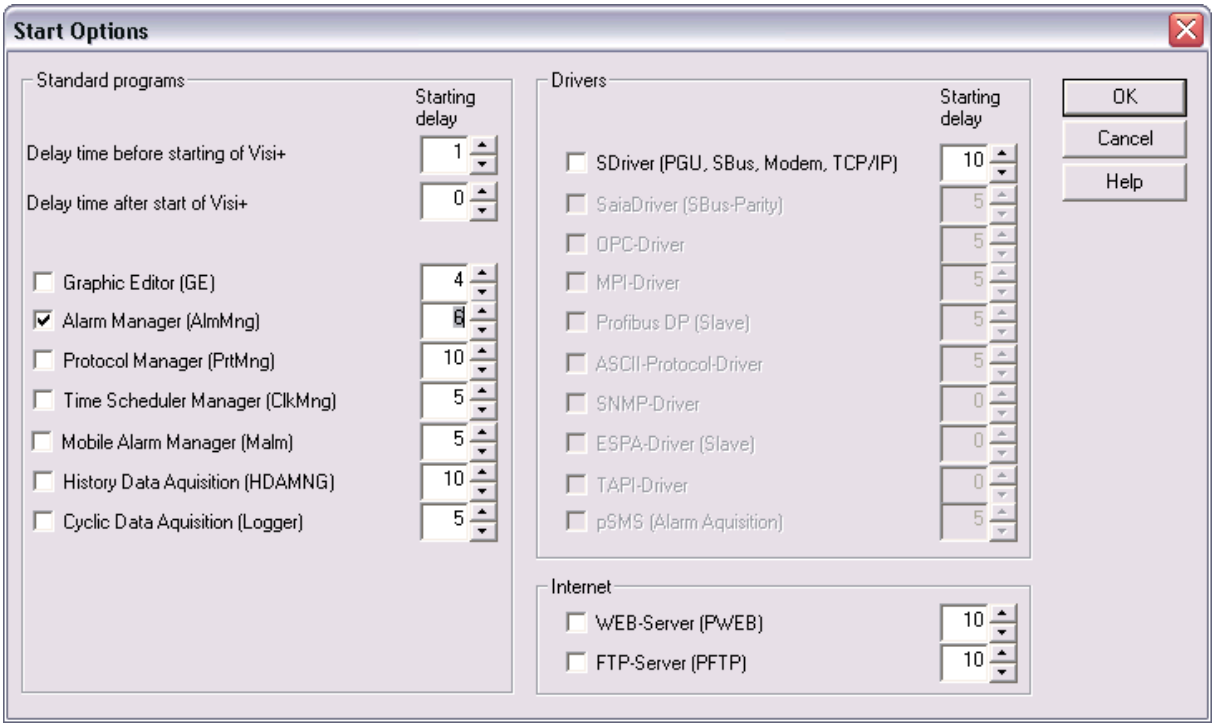
8.2.7 Starting the Alarm Manager (AlmMng.exe)

The **DMS** module **must** already be started. The **Alarm Manager** can be started at any time, either separately like any other program under Windows or automatically when the project is opened (Visi.Plus).

If it should be started when Visi.Plus is started, proceed as follows:

Start the **Project Manager** module and click on the **[Start Options]** button.

In the open **Start Options** window, mark the checkbox Alarm Manager (AlmMng) in the **Standard Programs** section. The next time the project is opened, **Alarm Manager** will automatically be started up along with it, with the indicated time delay.



The module normally runs as a background process and is displayed, while running, on the windows taskbar as a small clock (bottom right corner of the screen).

8.2.8 Alarm Manager (AlmMng.exe) control window

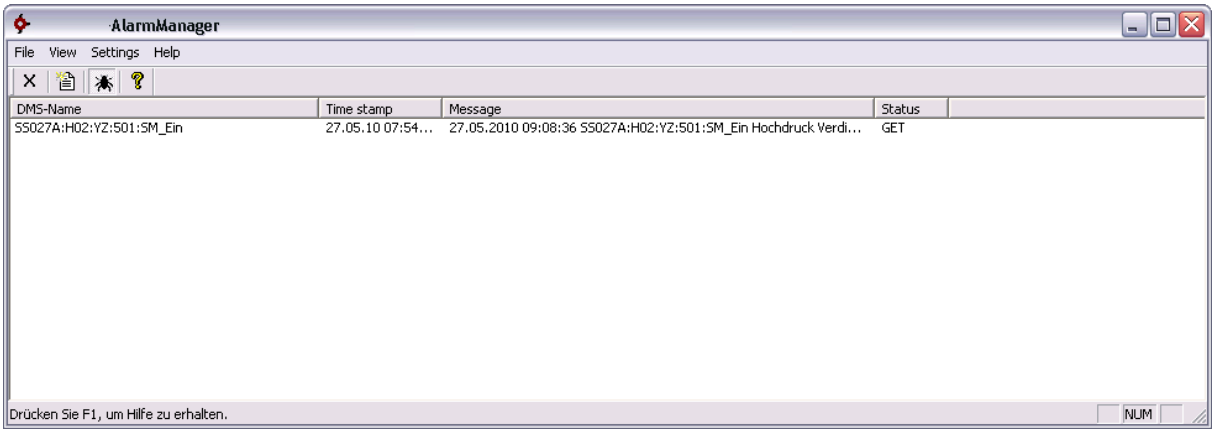
The Alarm Manager window is opened as described in the following:

Double-click on the **AlmMng icon**

or

click on the icon in the Windows taskbar with the right mouse button and choose the command "**Display**".

If the Alarm Manager is not running, any alarms that occur will not be registered. Alarms will only be monitored if the Alarm Manager is running!



Debug or 

If you wish to display all registered messages on the screen in **AlmMng**, select the command "**Debug**". This means that any alarms that occur are tracked on screen.

Column meanings:

DMS name
Shows the AKS code for the message.

Timestamp
Indicates the date and the time of the event.

Message
Shows the alarm log text. The chapter PRTFormat describes how alarm logs are defined.

Status
Indicates the current status of the alarm:
- incoming
- outgoing
- ack.

8.2.9 Exporting alarms via ASCII file in real time

An additional data interface of alarms in the form of ASCII files can be defined manually via an **AlmMng.cfg** configuration file (by default under <proj>\projectname\cfg).

The alarms are written to the respective file in tabular form similarly to the output in **AlarmViewer** windows.

Configuration
The data interface is activated with an entry in the **AlmMng.cfg** "FILE_COM" section:
[FILE_COM]
"Filename" ASCII filename, default "Alarme.log" (if this entry is missing).
The filename can contain the entire path; if the name does not have a path statement, the project log folder is used by default, e.g. c:\Visi.Plus\<proj>\log\

"Update=60" If > 0: Periods in seconds for data export of current alarms to an Alarme.log file. Minimum size is 30 seconds.
If this key entry is missing or is **=0**, the alarm export through this data interface is **deactivated**.

"Delimiter=;" : Separating character for alarm lines:
"Alarmtext<;>Status<;>Prio<;>DMSName"

These settings can be changed online; however, in order for them to be re-initialised, the menu command "File/Update from DMS" must be selected.

Example:

```
[FILE_COM]
Filename=AlmExport.log
Delimiter=;
Update=30
```

If an alarm comes in, goes out or is acknowledged, the file AlmExport.log is updated in the folder <proj>\log within the next 30 seconds.

Output format
Each alarm corresponds to a line. The following information is separated by the *delimiter character*:

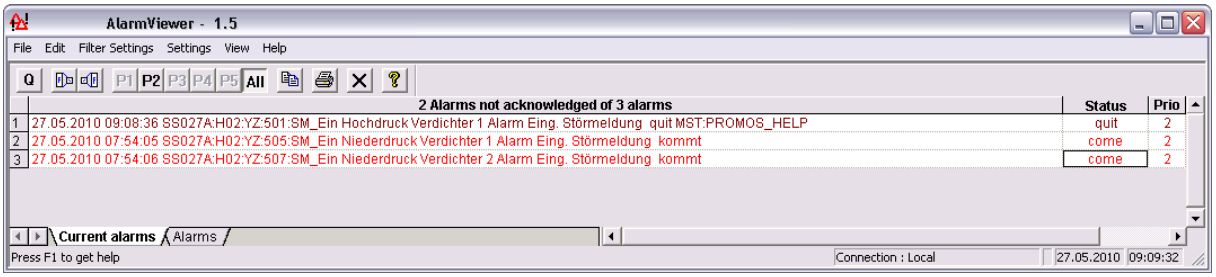
Alarmtext;Status;Priority;DMS name

The alarm text format is given in accordance with the PET definition; it corresponds precisely to the output in AlarmViewer.

Example:

```
26.05.2008 16:16:06 MST:MT:502:Err       Room sensor Sensor break  incoming;incoming;2;MST:MT:5
26.05.2008 16:16:07 MST:LG:001:Release Control Lamp Group 1 Bit (re)set ON  ;incoming;1;MST:LG
26.05.2008 16:16:08 UV07.01:L50:MT01:GW_HE_Err Room temperature L50 Pool technology upper thres
26.05.2008 16:16:08 UV07.01:S08:MT04:GW_HE_Err Temperature 3 Boiler BG02 S08 Upper threshold re
26.05.2008 17:20:27 HUE60:E01:AW:01:RMP_Err Pump Heating Process Acknowledgment  Ack.;ack.;1;HU
```

Corresponding **AlarmViewer** output window:



Currently the necessary settings must still be made manually in a text editor (e.g.

Notepad); menu-supported adjustment is planned for the next version.

The alarm file is not locked against read-only access. If no change occurs in alarm status in the update period, the file is not updated either. The entire update procedure is executed in a single thread, so that no locks or "trailers" occur when writing large files.

8.2.10 System alarms

System alarms define all alarms that are created by Visi.Plus.
This can be an alarm belonging to a driver or database, for example.

Driver alarms
The drivers generate a system alarm if a connection to the corresponding controller or field device fails.
The system alarms are all allocated to the alarm group 990-998 and can be deactivated according to the group. The alarms are all always activated as standard.
It is possible to filter out the system alarms in the alarm viewer. See the settings section of the AlarmViewer for more information.

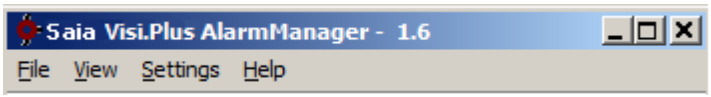
For information about the system alarms of the individual drivers, please refer to the corresponding sections.

PCDDriver system alarms
BACDriver system alarms
OPCDriver system alarms
ESPADriver system alarms
SNMPDriver system alarms

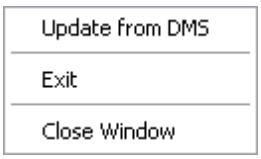
Alarm groups and collective alarm groups also generate system alarms if a corresponding group is suppressed. This helps visualise that a group is still deactivated. See also the section on alarm groups.

8.2.11 Alarm Manager menus

The Alarm Manager has 4 different menus:



8.2.11.1 File menu



Update from DMS
By selecting this command, all data (analogue values, digital values, etc.) are re-imported from the DMS (Data Management System of Visi.Plus) and updated in **AlmMng**.
Both the ***incoming*** and ***outgoing*** alarms are displayed if an update from the DMS is generated.

Changes in the DMS and in the PET (insertion of a new alarm) are not automatically imported into AlmMng. The alarms are only re-imported after restarting the **AlmMng** or by using the command "**Update from DMS**" .

Exit
Before the application can be exited, all modules registered with **AlmMng** must first be closed.

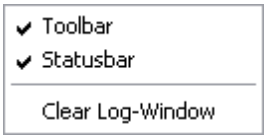
As long as programs are registered with **AlmMng**, **AlmMng** cannot be exited. In any case, a corresponding message will notify you of this.

Close

This menu item closes the control window of **AlmMng**. This does not terminate the program **AlmMng**; it is only minimised to the taskbar (dark red spot).

8.2.11.2 View menu

In the "**View**" menu, the **toolbar** and/or **status bar** can be switched on or off.



Toolbar
Switches the toolbar on or off.

 Closes the AlarmMng

 Deletes the current log window

 Debug button. If the debug button is activated, the incoming and outgoing alarms are listed in the log window.

 Info button. Displays the current version of the Alarm Manager as well as the licence number.

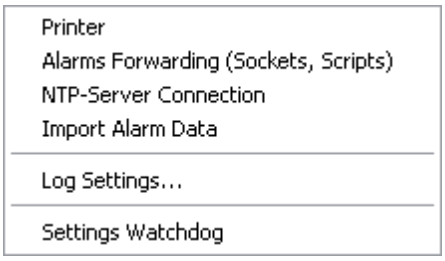
Status bar
Switches the status bar on or off.

Delete log window

This menu item deletes the main window (log list) of the **Alarm Manager**. This has no influence on the functionality of the **Alarm Manager**.

8.2.11.3 Settings menu

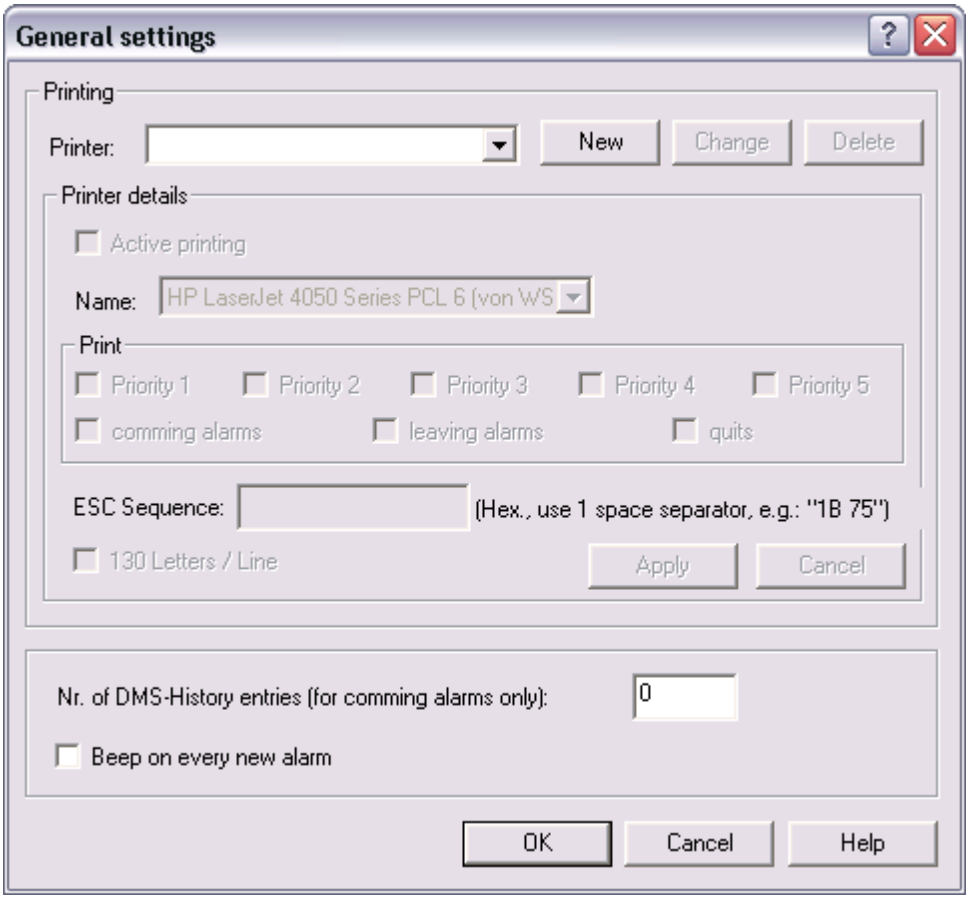
In the "**Settings**" menu, different possibilities are available for printing and importing.



8.2.11.3.1 Printer

Alarms can be output to multiple printers. The active printers are listed in the **<Printer>** combo box. Different priorities and alarm statuses (incoming/outgoing/acknowledged) can be assigned to each printer.

In the **Spontaneous Print** section, the printer should be indicated which is to be used for any alarms which occur. First, set up the printer by pressing the **<New>** button. In the **Printer Details** section, the respective printer and the various alarm priorities can now be selected and a port can be defined. The **<Change>** button enables the print settings to be changed if necessary. The <Delete> button deletes the settings for the printer indicated under **Printer** and removes it from the list.



Printer Details section

Activate printing
If the checkbox is marked, alarms which occur are printed directly to the printer indicated.

Name
Indicates the printer name or names that are installed on your system and can be used.

Priority 1 - 5
Here you can indicate which priority should trigger a spontaneous print (e.g. if the **Priority 1** checkbox is marked, alarms which occur are printed immediately upon occurrence with Priority = 1).

130 characters per line
When activated, 130 characters per line are output. 80 characters is the standard setting.

ESC sequence
This is a sequence of special characters that should send a control command to the printer. Normally an ESC sequence begins with an escape character.

Example:
<esc>right.
You can gather valid ESC sequences for your printer from its manual.

The ESC sequence is entered hexadecimally, whereas the ESC sequences are to be separated by a blank space. For example, "0F 0B" with an IBM printer means:

condensed print and tab before each line.

Number of DMS History Entries
Defines the maximum number of messages that the Alarm Manager should display (possible values: 0–24).

Audible Message for New Alarm
Activates an audible beep tone with a new alarm.

 Alarms with a priority higher than 4 cannot be printed.

 Alarms can be output to various printers.
(e.g. if you have multiple alarms with different priorities and want them printed at different locations)

 Start-up behaviour:
Outgoing alarms are not displayed on program start-up.
All alarms, including **outgoing** alarms are shown with the option **Update from DMS**.

8.2.11.3.2 Alarm forwarding (sockets, scripts)

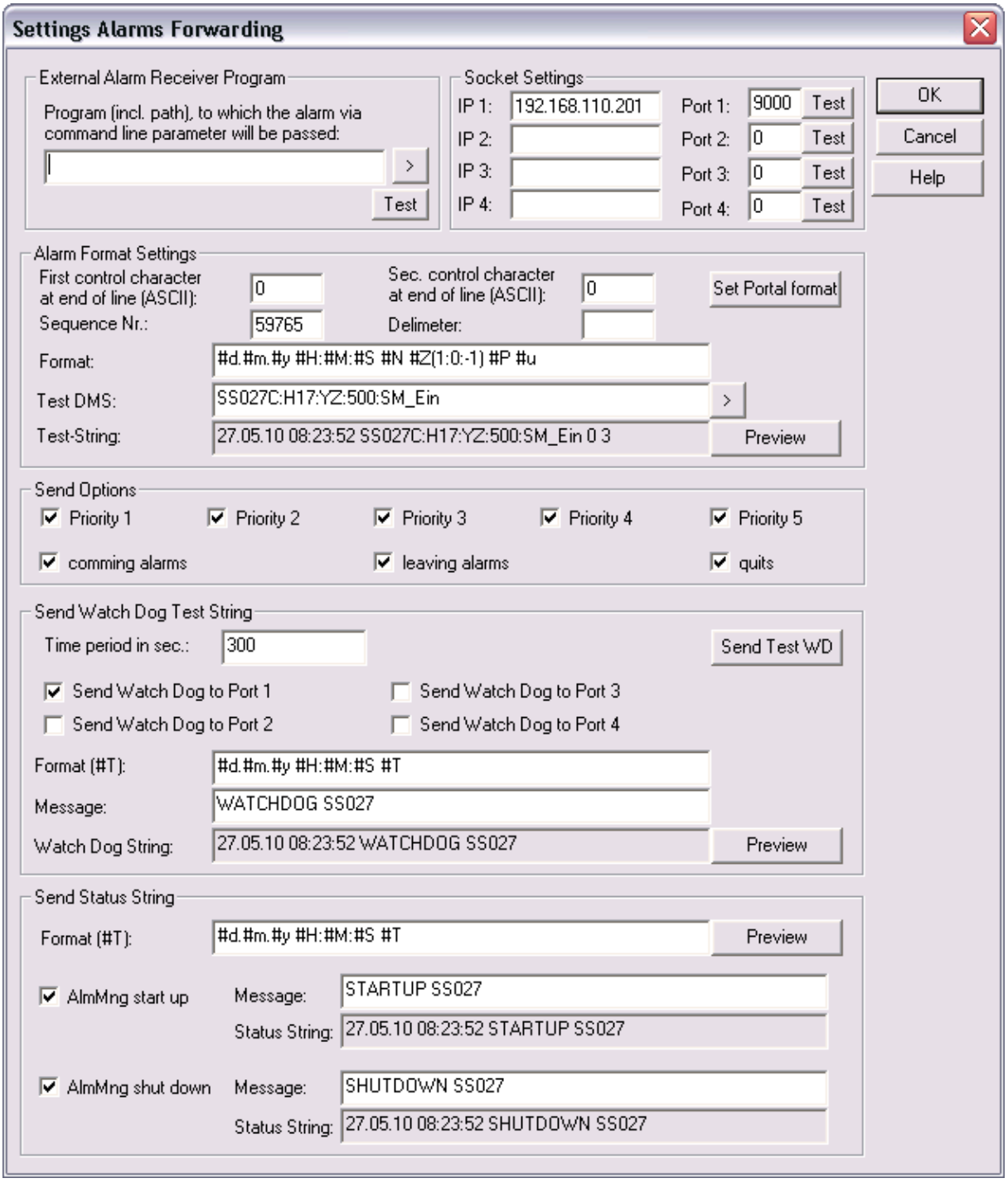
Alarms can be forwarded through **sockets** (TCP/IP) and/or to an **external alarm recipient program** (via command line parameter).

- **Sockets**: if an alarm is to be transmitted to a second computer via a socket connection, the socket connection (the IP address and at least one port under 'Socket Settings') and the alarm telegram to be sent must be defined here.
  At the recipient station (second computer) an application must be opened to log and display the alarms (e.g. GISDriver or AlmSocketMng from Visi.Plus). The transmission format can also be modified for third party products (e.g. alarm systems).
- **External alarm recipient program:** forwarding is activated if the input field "**Program (incl. data path)...**" includes a valid link to an executable program (exe, com or bat). The alarm text is transmitted as a command line parameter in quotation marks.

The alarms to be forwarded have their own format definition (independent of the PET), see the **Alarm Format Configuration** section further down.

Additional criteria for forwarding are the set priorities (1-5) and alarm types (incoming, outgoing, acknowledgment), see section **Transmission Options** further down.

Actuation of the **<Save>** button saves the settings that were made. The **<Cancel>** button closes the window without applying the settings.
If <OK> is pressed, the window closes and the configuration is applied.



Section: External Alarm Recipient Program

Program (incl. data path) to which the alarm is transmitted via command line parameter

Here, a link to an executable program (exe, com or bat) must be entered.
Alternatively, it can be selected with the browser (executable programs: exe, com or bat).
The alarm text is transmitted to this program as a command line parameter in quotation marks.
The quotation marks are necessary, because the alarm text may contain blank spaces.

Test
This button starts the external alarm recipient program with the alarm text from the **Test String** input field as a command line parameter.

The functionality can be demonstrated in the following **example**. A simple batch program 'exampleAlm.bat' writes the alarm text as a line in the console window.

exampleAlm.bat

```
echo off
echo Alarmtest
ECHO %1
PAUSE
```

The batch program is called up by AlmMng in the following way:

```
C:\Documents and settings\exampleAlm.bat "21.08.2008 17:03:15Room Temperature Receipt Se
```

Section: Socket Settings

IP address
The IP address to which the alarm should be sent can be entered here.

Port 1 - 4
Any port number to be used can be entered here. Alarms can be sent to a maximum of four different ports.

The application started on the recipient station must be set to the appropriate port that you would like to use here.

Test Port 1 – 4
Pressing the button <Test Port 1> to <Test Port 4> enables testing of the connection to the selected port.

Section: Alarm Format Configuration

First/second control character
The control character is defined in ASCII format here. It should be at the end of the message line.
(e.g. first character = 99
e.g. second character = 99
Now there is a **cc** at the end of the alarm message)
Example: 13 10 sends CR and LF

Sequence No.
Enter a value here that the counter should begin with for the sequence, e.g. 001 (then the next alarm sent to the socket connection receives the sequence 002, the following alarm 003, etc.) The sequence counter is increased by one for every telegram. The sequence number ends at 999,999, or begins again at zero after this number. The sequence number can be defined with the placeholder #Q.

Separator
A separator to be used can be indicated here, e.g. # or ; etc.
The delimiter can be defined in the format string as #D.

Format
See also the chapter Formatdefinitionen (PrtFormat.exe).

The desired format for the message to be transmitted can be entered here.
For example, the same formats are applicable as for the alarm definitions.
(e.g. #c / #T)

#H	Current hour (current time)
#M	Current minute (current time)
#S	Current second (current time)
#d	Current day (date)
#m	Current month (date)
#y	Current year (date - two digits)
#Y	Current year (date - four digits)
#C	Comment (in the DMS as comment under the alarm data point)
#N	DMS name
#v	Current analogue value
#z	Current digital value
#Z	Current alarm status (incoming, outgoing, ack.), text must be included in brackets (e.g. (in:out:ack.))
#u	User Name
#V	Current DMS value (a higher level can be indicated with ^)
#T	Alarm text (defined in the Alarm Definition dialog)
#F	Special format for Saia PCD®(date/time) e.g. #F(HH:MM)
#t	Special format for Saia PCD® (date/time).
#E	External text (can be indicated for e-mails with *)
#Q	Sequence number (0 - 999999)
#P	Alarm priority
#R	Alarm group
#D	Delimiter

* An external text file must be saved in the project directory in a subdirectory with the name "\TXT". For each DMS data point, there must be one file whose filename consists of the DMS name with the ending .TXT. Colons in the DMS name are replaced by _ (e.g. WL027_H04_AB_001_Strg.txt). If this file is not found, the program searches for the higher-ranking filename (according to the DMS structure). Example: WL027_H04_AB_001.txt.
If the file is not found, a file with the name "Default.txt" is searched for and, if present, its data content is output.

Test DMS
A DMS data point can be selected with the '**>**' button.

Test String
The Test DMS data point can be formated with the set format and displayed with the <**Preview**> button.

Section: Transmission options

Priority 1-5
When activated, only alarms of the *respective priority* are forwarded.

Incoming Alarms
When activated, *incoming alarms* are forwarded.

Outgoing Alarms

When activated, *outgoing alarms* are forwarded.

Acknowledgments
When activated, *acknowledgements* are forwarded.

Section: Send WatchDog Test string

With WatchDog, for example, you can test in predefined intervals whether the socket connection with the respective port for the second computer is still present or has problems. With this function, a "test alarm" can be sent at regular intervals in order to monitor the communications connection.

Time periods in sec.
Here you can indicate the time periods in seconds for the intervals at which a WatchDog test string should be sent to the second computer.

Send Watch Dog
When activated, a Watch Dog at **Port 1-4** is sent to the respective port number.

Format (#T)
Enter the format for the WatchDog here.

Message
The desired message for the WatchDog can be entered here.

Watch Dog String
Here you can see the resulting WatchDog message after pressing **<Preview>**.

Send Test WD
Pressing this button enables testing of the WatchDog connection to the specified port.

Section: Send Status String
(If AlmMng is **started** or **exited**)

Format (#T)
The desired format of the message to be sent can be defined here.

*The entered format applies for **both** message definitions (up and down).*

Message
The desired message can be entered here.

Status String
Shows the string of the message for checking purposes.

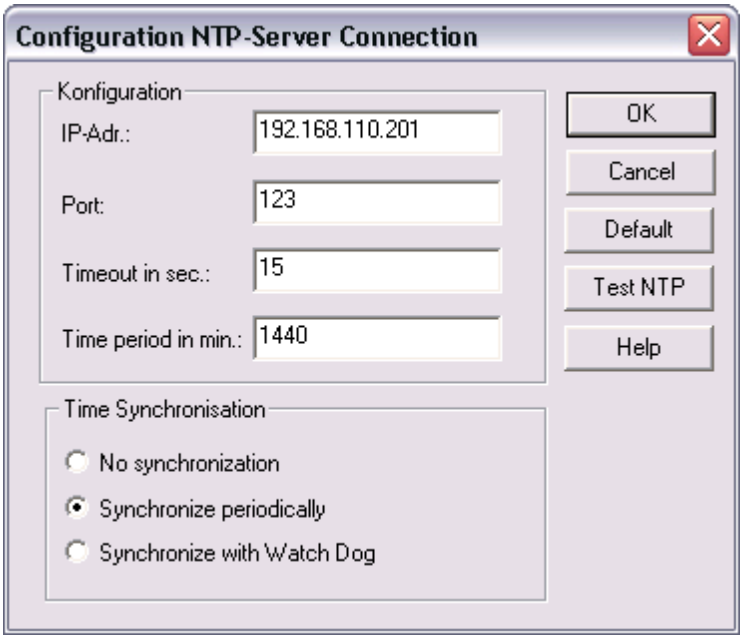
AlmMng up/down
When activated, a message, as defined under **Message**, is set when starting AlmMng.

Preview

Pressing the **<Preview>** button displays a preview of the defined message in the **String** field.

8.2.11.3.3 NTP Server Connection

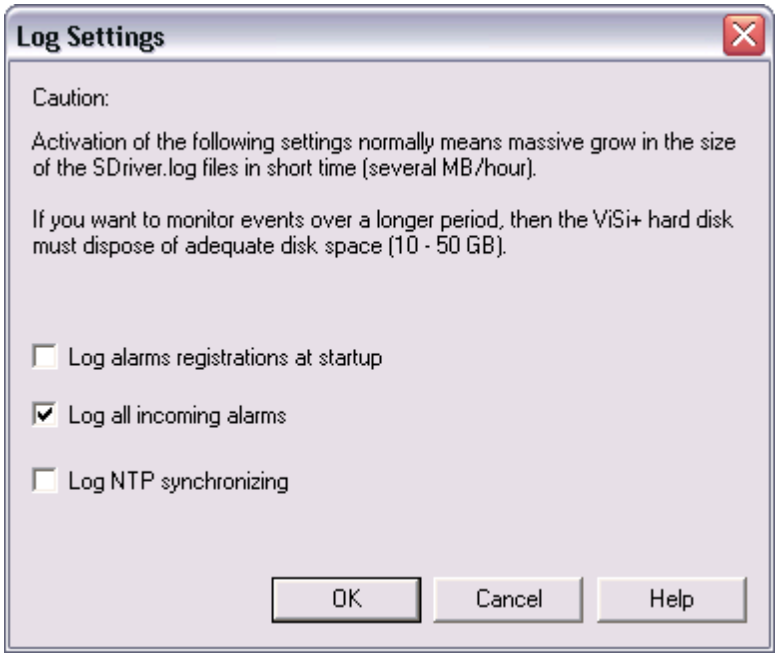
This menu item allows the installed NTP client to be configured. Thus the time on the PC can be synchronised with an NTP server.



IP address	Address of the NTP server
Port	Port of the NTP server (123 is the default)
Timeout	Timeout on connection with the NTP server
Time periods	Cyclical periods of time for synchronising the time (in minutes)
No synchronisation	Synchronisation is deactivated
Periodic synchronisation	Cyclical synchronisation (according to periods of time)
WatchDog synchronisation	Synchronisation with WatchDog

8.2.11.3.4 Log Options

The following functions are available for testing and debugging purposes:



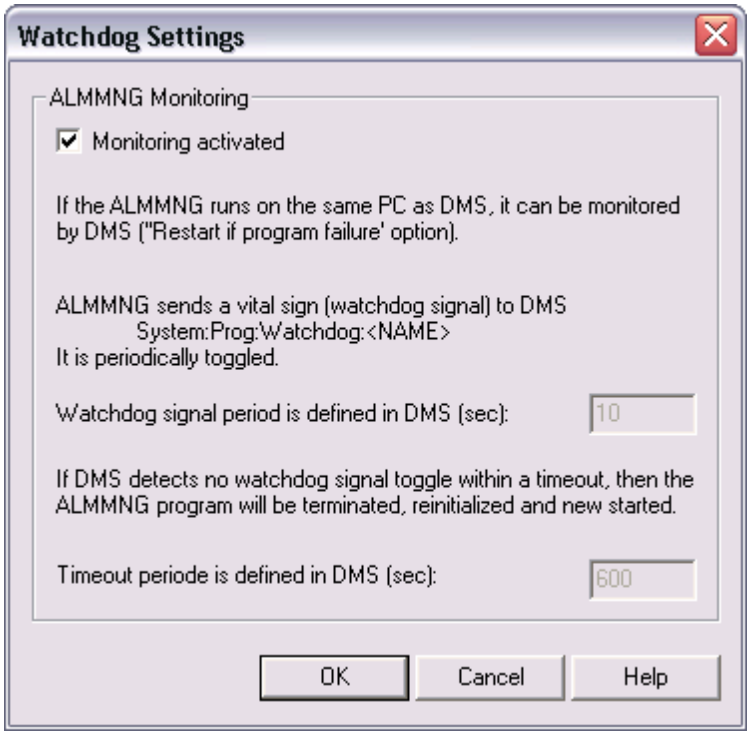
<Log alarm registrations on start-up>
Only effective after restarting AlmMng! All registered alarms are listed in detail in **AlmMng.log** when AlmMng is started.

<Log all incoming alarms>
More messages are written in greater detail to the log file **AlmMng.log** for the alarm processes which occur.

<Log NTP synchronisations>
Records the synchronisation messages of the NTP server connection in the **AlmMng.log** file.

8.2.11.3.5 Watchdog Settings

The **AlmMng** manager can be monitored by the Data Management System (DMS). **AlmMng** and the DMS exchange a live bit upon activation of the function.



8.2.11.4 Help menu

The "**?**" menu contains the Visi.Plus online help and current information on the **AlmMng** version.

Help Topics
The Visi.Plus online help is accessed through the **"Help topics"** menu item via the **<Help>** menu button or by pressing the **<F1>** key.

About AlmMng
Display the current version of **AlmMng** and its developer. The version must always be given in case of any support queries!

Should support queries become necessary, the internet forum is preferential to contact via telephone or e-mail. This offers two advantages: First, help is available more quickly; and secondly, other users also benefit from the entries.

8.3 The MALM Manager (MalmMng.exe)

The **MAlm** Manager is used to administer any remote alarm processes that occur.

MAlmMng consists of two modules, the **MAlm Configurator** and the **MAlm Manager**. They offer the possibility of transmitting faults and messages via modem/internet or via mobile.

In the **MAlm Configurator** you can define how the remote alarm should be transmitted **(MalmConfig.exe).**

The **MAlm Manager** manages the alarms, shows which remote alarms occur and which have already been transmitted. **(MalmMng.exe)**

There are various alarm possibilities (notification types) available to choose from. Multiple services at the same time are also possible:

- Alarm via e-mail (internet, modem)
- Alarm via SMS (Short Message Service): SMS-UCP protocol for Switzerland and SMS-TAP protocol for Austria
- Alarm via pager (telepage)
- Alarm via Cityruf (special technology from Germany)

*Whether a data point will trigger a remote alarm or not is defined in the column **MAlarm**. If the column is not visible, it can be switched on in the "**View**" menu.*
Procedure: Click on the respective data point in the column, and select the alarm group previously configured in MAlmConfig and the alarm format.

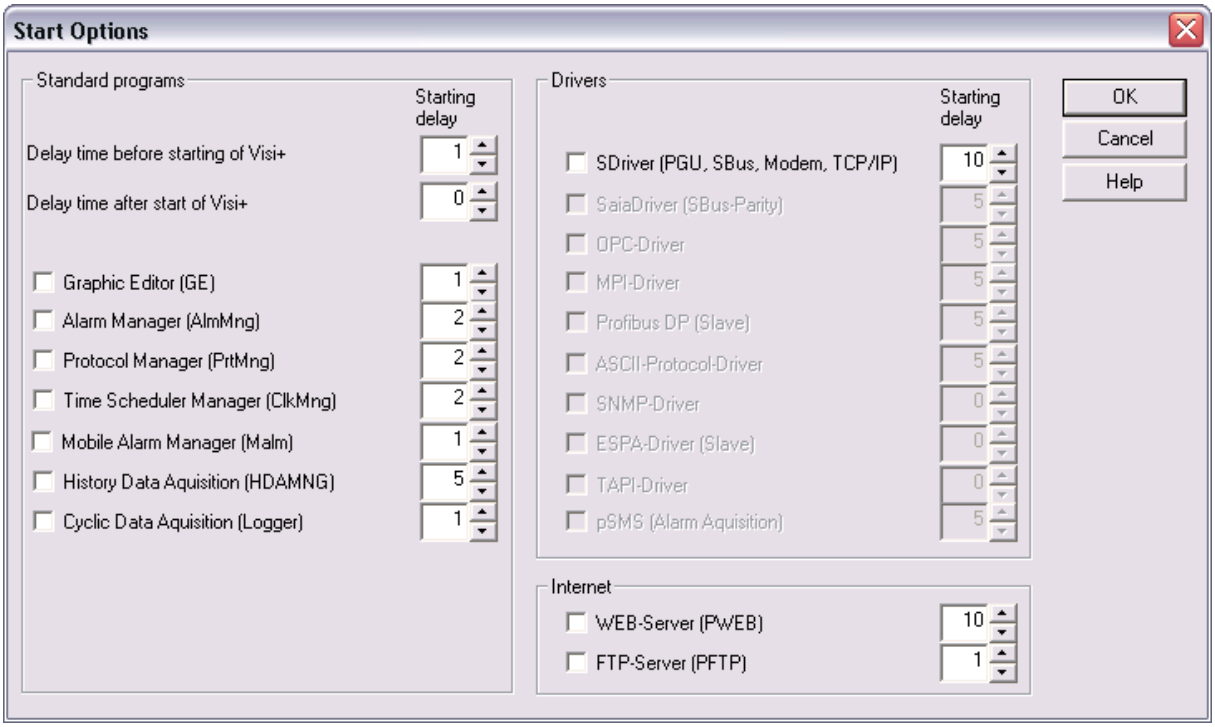
New: in Version 1.5 and higher the following improvements are included:

1. New notification types:
   - ESPA external systems, e.g. staff paging systems, telephone exchanges, etc.
   - Voice: Telephone voice output with the possibility of acknowledging the alarm immediately
   - Spool mode: Notification via file system to an external spool program, e.g. SMS server.

8.3.1 Starting the MAlm Manager program

In order to be able to start the **MAlmMng** module when a project is opened, the appropriate checkbox must be marked in the **Start Options**.

To do this, start the **ProjectCfg** module and click on the **<Start Options>** button.



Select **Remote Alarm Manager (MAlmMng)** in the window which opens.

Confirm with **<OK>.**

The **MAlm** Manager is now started automatically the next time a project is opened.

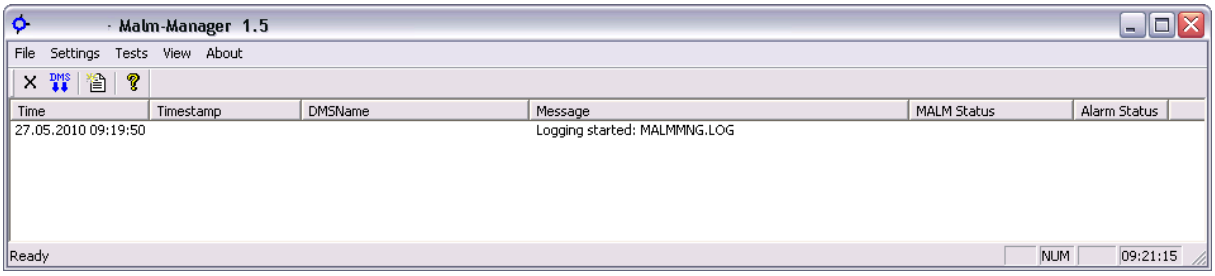
The module normally runs as a background process and is displayed as a blue dot on the Windows taskbar.

If **MAlm** Manager is not started, the remote messages which occur are not transmitted!

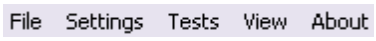
8.3.2 The MAlm Manager main window

Opening the main window:

Click on the blue **MalmMng** icon with the right mouse button and select the "**Display**" command.

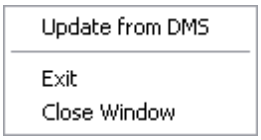


8.3.3 The MAlm Manager menus



8.3.3.1 File menu

Functions for DMS Update and closing and exiting the module are available in the "**File**" menu.



Update from DMS 

This menu item forces a reconfiguration of the **MAlm** Manager.

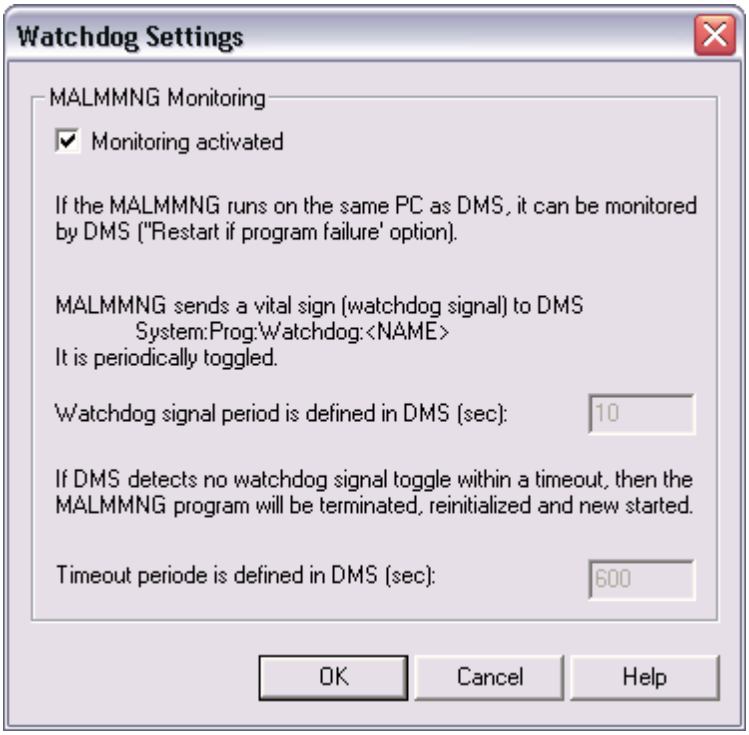
Exit
Exits the **MAlm** Manager.

Close 
This menu item closes the control window of the **MAlm** Manager. This does not exit the **MAlm** manager module; it is only minimised to the taskbar.

8.3.3.2 Settings menu

The following functions are available in the "**Settings**" menu:

Watchdog Settings
The **MAlm** Manager can be monitored by the Data Management System (DMS).The **MAlm** Manager and the DMS exchange a live bit upon activation of the function.



Log Options

The following functions are available for testing and debugging purposes:

<Log Startup: Registration of the Alarms>

Is only effective after restarting MalmMng! All registered alarms are listed in detail in **MalmMng.log** when MalmMng is started.

<Log Details: Relay Alarms>

More messages are written in greater detail to the log file **MalmMng.log** for the remote alarm processes which occur.

Delete Log Window 

This menu item deletes the main window (log list) of the **MAlm** manager. This has no influence on the functionality of the **MAlm** Manager.

8.3.3.3 Test menu

The functions in the "**Test**" menu are used to test or simulate the individual remote alarm processes and their output devices.

A Test dialog is currently only available for the Voice service.

8.3.3.4 View menu

Functions relating to the appearance of the program are available in the "**View**" menu.

Toolbar
Switches the toolbar on or off.

Status bar
Switches the status bar on or off.

8.3.3.5 Help menu

Help
Opens the help system of the MAlarm Configurator.

About MAlarm...
Displays the current version of **MAlmMng**. The version must always be given in case of any support queries!

Should support queries become necessary, the internet forum is preferential to contact via telephone or e-mail. This offers two advantages: First, help is available more quickly; and secondly, other users also benefit from the entries.

8.3.4 Alarm counter

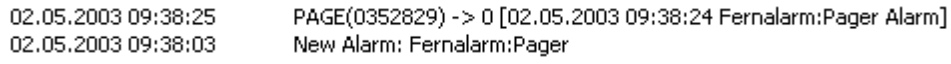
When any alarm is triggered, the corresponding counter is incremented (DMS type DWU - double word unsigned):

- System:MALM:Count:SMS for SMS and Cityruf alarms
- System:MALM:Count:PAGER for pager alarms
- System:MALM:Count:MAIL for email alarms
- System:MALM:Count:ESPA for ESPA alarms
- System:MALM:Count:VOICE for voice alarms
- System:MALM:Count:SPOOL for spool alarms

All alarm counters can be reset via the data point "System:MALM:Count:Reset" (type bit, transition 0 -> 1).

8.3.5 MalmMng error handling

The **MAlmMng** module has error handling.**MAlmMng** writes error messages to the **Log window** and into the **MalmMng.log** file as a numerical code.



The numerical code is separated from the text by an arrow. In the image above, an example of a pager message that was sent with the **Error code 0** is shown. The message shows a successfully transmitted pager message.

Each of the types of transmission: e-mail, SMS, pager and Cityruf has its own numerical code for error messages, which are described in the following.

8.3.5.1 E-mail error messages

Code 0	SUCCESS	E-mail successfully sent
Code 1	ERROR	The process ended with an error (operation failed)

Communications errors

Code 2	CONNECT_FAILED	The connection to the mail server was terminated with an error
Code 3	CONNECT_REJECTED	
Code 4	CONNECT_TERMINATED	
Code 5	CONNECT_TIMEOUT	
Code 6	NOCONNECTION	
Code 7	NAME_LOOKUP_FAILED	There is no connection
Code 8	DATAPORT_FAILED	
Code 9	ACCEPT_FAILED	

Server errors

Code 10	SVR_REQUEST_DENIED	
Code 11	SVR_NOT_SUPPORTED	
Code 12	SVR_NO_RESPONSE	There is no answer from the server
Code 13	SVR_ACCESS_DENIED	
Code 14	SVR_DATA_CONNECT_ FAILED	

Socket errors

Code 15	NOT_OPEN
Code 16	SOCK_ALREADY_OPEN
Code 17	SOCK_CREATE_FAILED
Code 18	SOCK_BIND_FAILED
Code 19	SOCK_CONNECT_FAILED

Code 20 SOCK_TIMEOUT
Code 21 SOCK_RECEIVE_ERROR
Code 22 SOCK_SEND_ERROR
Code 23 SOCK_LISTEN_ERROR
Code 24 CLIENT_RESET
Code 25 SERVER_RESET

File errors
Code 26 FILE_TYPE_ERROR
Code 27 FILE_OPEN_ERROR
Code 28 FILE_CREATE_ERROR
Code 29 FILE_READ_ERROR
Code 30 FILE_WRITE_ERROR
Code 31 FILE_CLOSE_ERROR
Code 32 FILE_ERROR
Code 33 FILE_FORMAT_ERROR
Code 34 FILE_TMP_NAME_FAILED

Buffer error
Code 35 BUFFER_TOO_SHORT
Code 36 NULL_PARAM

Response error
Code 37 INVALID_RESPONSE
Code 38 NO_RESPONSE

Index error
Code 39 INDEX_OUTOFRANGE

User validation errors
Code 40 USER_ERROR
Code 41 PASSWORD_ERROR Password errors

Message errors
Code 42 MESSAGE
Code 43 INVALID_FORMAT
Code 44 FILE_NOT_MIME

URL errors
Code 45 URL

Command errors
Code 46 INVALID_COMMAND
Code 47 MAIL_FAILED
Code 48 RETR_FAILED
Code 49 RETR_FAILED

Code 50	LIST_FAILED	
Code 51	STOR_FAILED	
Code 52	USER_FAILED	
Code 53	DATA_FAILED	HELLO command was rejected or not responded to by the server
Code 54	HELLO_FAILED	
Code 55	PASS_FAILED	
Code 56	STAT_FAILED	
Code 57	TOP_FAILED	
Code 58	UIDL_FAILED	
Code 59	DELE_FAILED	
Code 60	XOVER_FAILED	
Code 61	RSET_FAILED	
Code 62	USER_NA	
Code 63	PASS_NA	
Code 64	ACCT_NA	
Code 65	RNFR_NA	
Code 66	RNTO_NA	
Code 67	RCPT_FAILED	
Code 68	NNTP_BAD_ARTICLE	
Code 69	NNTP_NOPOSTING	
Code 70	NNTP_POST_FAILED	
Code 71	NNTP_AUTHINFO_USER_	
Code 72	FAILED	
Code 73	NNTP_AUTHINFO_PASS_F AILED	
	XOVER_COMMAND_FAILE D	

Message errors

Code 74	MSG_OPEN_FAILED
Code 75	MSG_CLOSE_FAILED
Code 76	MSG_WRITE_LINE_FAILED
Code 77	MSG_READ_LINE_FAILED
Code 78	MSG_NO_ATTACHMENTS
Code 79	MSG_BODY_TOO_BIG
Code 80	MSG_ATTACHMENT_ADD_FAILED

Data source errors

Code 81	DS_OPEN_FAILED
Code 82	DS_CLOSE_FAILED
Code 83	DS_WRITE_FAILED

Encoding errors

Code 84	ENCODING_INVALID_CHAR
Code 85	ENCODING_LINE_TOO_LONG

IMAP4 errors

Code 86	LOGIN_FAILED
Code 87	NOOP_FAILED
Code 88	UNKNOWN_COMMAND

Code 89 UNKNOWN_RESPONSE
Code 90 AUTH_OR_SELECTED_STATE_REQUIRED
Code 91 SELECTED_STATE_REQUIRED

RAS Errors

Code 92 RAS_LOAD_ERROR
Code 93 RAS_DIAL_ERROR
Code 94 RAS_DIALINIT_ERROR
Code 95 RAS_HANDLE_ERROR
Code 96 RAS_ENUM_ERROR
Code 97 RAS_ENTRYNAME_ERROR

Unclassified errors

Code 98 ABORTED
Code 99 BAD_HOSTNAME
Code 100 INVALID_ADDRESS
Code 101 INVALID_ADDRESS_FORMAT
Code 102 USER_TERMINATED
Code 103 SERVER_SET_NAME_FAILED
Code 104 ANS_NOT_FOUND
Code 105 PARAMETER_TOO_LONG
Code 106 PARAMETER_INVALID_VALUE
Code 107 TEMP_FILENAME_FAILED
Code 108 OUT_OF_MEMORY
Code 109 GROUP_INFO_UPDATE_FAILED
Code 110 GROUP_NOT_SELECTED
Code 111 INTERNAL_ERROR
Code 112 ALREADY_IN_USE
Code 113 NO_CURRENT_MSG_SET
Code 114 QUOTE_LINE_IS_EMPTY
Code 115 REST_COMMAND_NOT_SUPPORTED
Code 116 SYSTEM_INFO_LOAD_FAILED
Code 117 USER_INFO_LOAD_FAILED
Code 118 USER_NAME_ALREDY_EXIST
Code 119 MAILBOX_NAME_ALREDY_EXIST
Code 120 LOAD_SECURITY_LIBRARIES_FAILED
Code 121 OPEN_CERTIFICATE_STORE_FAILED
Code 122 FAILED_TO_FIND_CERTIFICATE
Code 123 FAILED_TO_CREATE_SECURITY_CREDENTIALS
Code 124 FAILED_TO_INITIALIZE_SECURITY_CONTEXT
Code 125 FAILED_TO_ACCEPT_SECURITY_CONTEXT
Code 126 FAILED_TO_QUERY_CERTIFICATE
Code 127 HANDSHAKE_FAILED
Code 128 FAILED_TO_GET_CERTIFICATE_CHAIN
Code 129 FAILED_TO_VERIFY_CERTIFICATE_CHAIN
Code 130 FAILED_TO_VERIFY_CERTIFICATE_TRUST
Code 131 FAILED_TO_GET_SECURITY_STREAM_SIZE
Code 132 FAILED_TO_APPLY_CONTROL_TOKEN
Code 133 FAILED_TO_RECEIVE_SECURITY_MESSAGE
Code 134 SECURITY_CONTEXT_EXPIRED
Code 135 FAILED_TO_DECRYPT_SECURITY_MESSAGE
Code 136 FAILED_TO_ENCRYPT_SECURITY_MESSAGE

Code 137 INVALID_CHARS_IN_STRING_PARAM
Code 138 FAILED_TO_GENERATE_SECURITY_KEY
Code 139 FAILED_TO_EXPORT_SECURITY_KEY
Code 140 OPEN_CERTIFICATE_STORE_FIRST
Code 141 FAILED_CREATE_ICENROLL
Code 142 CONNECT_FAIL_NO_SSL_SUPPORT
Code 143 CERTIFICATE_INVALID_DATE
Code 144 IMAP4_TLS_NOT_SUPPORTED
Code 145 SMTP_TLS_NOT_SUPPORTED
Code 146 POP3_TLS_NOT_SUPPORTED
Code 147 UNSUPPORTED_KEY_SIZE
Code 148 UNSUPPORTED_ENCODING_TYPE
Code 149 INVALID_CHARACTER_IN_CHARSET
Code 150 CHARSET_TOO_BIG
Code 151 INVALID_ENCODING_FORMAT

8.3.5.2 SMS error messages (UCP protocol)

Code 0
The SMS was successfully sent.

Code 1
Internal software problem. Your operating system could not allocate the necessary resources. Try again later.

Code 2
The port address could not be found or is being used by another program. Make sure that you have selected the right interface.

Code 3
The call to the SMS message centre could not be placed. Check the SMS message centre number you entered.

Code 4
The modem could not be found. Make sure that the modem is switched on and properly connected to the computer.

Code 5
The modem has determined that there is no dial tone. Make sure that the telephone line is not in use.

Code 6
The SMS message centre is busy. Try again later.

Code 7
There is no response from the location called. Make sure that you have entered a valid SMS message centre number.

Code 8
Modem error. The modem reported an error message. Make sure that the modem is working.

Code 9

The modem is not answering. Make sure that the modem is properly connected to the telephone line and the computer.

Code 10
Internal software timeout. The modem could not establish connection before the software timeout. Try again later.

Code 11
The SMS message centre terminated the connection before all data could be sent. Try again later.

8.3.5.3 SMS error messages (TAP protocol)

Code 0
The SMS was successfully sent.

Code 1
Internal software problem. Your operating system could not allocate the necessary resources. Try again later.

Code 2
The port address could not be found or is being used by another program. Make sure that you have selected the right interface.

Code 3
The call to the SMS message centre could not be placed. Check the SMS message centre number you entered.

Code 4
The modem could not be found. Make sure that the modem is switched on and properly connected to the computer.

Code 5
The modem has determined that there is no dial tone. Make sure that the telephone line is not in use.

Code 6
The SMS message centre is busy. Try again later.

Code 7
There is no response from the location called. Make sure that you have entered a valid SMS message centre number.

Code 8
Modem error. The modem reported an error message. Make sure that the modem is working.

Code 9
The modem is not answering. Make sure that the modem is properly connected to the telephone line and the computer.

Code 10
Internal software timeout. The modem could not establish connection before the software timeout. Try again later.

Code 11
The SMS message centre terminated the connection before all data could be sent. Try again later.

Code 12
Incorrect login to the SMS message centre. The message centre is not accepting the login information.

Code 13
The SMS message centre could not forward the SMS. Possibly invalid recipient number.

Code 14
The SMS was not accepted by the message centre. Check whether invalid characters were sent.

Code 15
The connection was terminated by the SMS message centre. Try again later.

Code 16
The SMS message centre is not receiving any SMSs or is not accepting them.

8.3.5.4 SMS message centre error messages

The error messages listed below are only related to the Swisscom SMS message centre (079 499 89 90). If a different SMS message centre was entered, the error messages will not correspond to the following list in some cases.

Code 101
Checksum error. Check your entry.

Code 102
Syntax error. Check your entry.

Code 103
Unsupported operation. Check your entry.

Code 104
Unauthorised operation. Check your entry.

Code 105
SMS silence is switched on. You cannot currently send messages with this number.

Code 106
Invalid number. Check the number.

Code 107
Incorrect password. Check the password.

Code 108
Incorrect legitimation (for all messages). Check the legitimation.

Code 109

Invalid geographic zone (geographic area). Check the geographic zone.

Code 110
Repeat dialling is not permitted. You cannot switch on repeat dialling.

Code 111
Incorrect legitimation for repeat dialling. Check the legitimation.

Code 112
Change of priority is not permitted. You cannot change priority.

Code 113
Incorrect legitimation for change of priority. Check the legitimation.

Code 114
Urgent message characters are not allowed. You cannot add urgent characters.

Code 115
Incorrect legitimation for urgent message. Check the legitimation.

Code 116
Reverse loading not permitted. You cannot switch on "reverse loading".

Code 117
Incorrect legitimation for "reverse loading". Check the legitimation.

Code 118
Delayed call transmission not permitted. You cannot enter a "delayed message transmission".

Code 119
Invalid new password. Check the new password.

Code 120
Invalid new legitimation. Check the new legitimation.

Code 121
Invalid default text. Check the number of the default text.

Code 122
Invalid time indication. Check the time indication.

Code 123
Message class not supported. You cannot send a message of this class.

Code 124
Message too long (has too many characters). Check the message.

Code 125
Requisite default text is invalid. Check the number of the default text.

Code 126
Message class not compatible with the calling class of this number. This number cannot send this kind of message.

Code 127
Message not found in SMS message centre.

Code 130
Subscriber hang-up.

Code 131
Fax group not supported.

Code 132
Fax message class not supported.

Code 133
The address is already in the list (60 series).

Code 134
The address is not in the list (60 series).

Code 135
Address list full. No new addresses can be added.

Code 136
Response ID already used.

Code 137
Message is being delivered.

Code 138
Message is being forwarded.

8.3.5.5 Pager error messages

Code 0
The pager message was successfully sent.

Code 1
Internal software problem. Your operating system could not allocate the necessary resources. Try again later.

Code 2
The port address could not be found or is being used by another program. Make sure that you have selected the right interface.

Code 3
The call to the pager centre could not be placed. Check the pager centre number you entered.

Code 4
The modem could not be found. Make sure that the modem is switched on and properly connected to the computer.

Code 5
The modem has determined that there is no dial tone. Make sure that the telephone line is not in use.

Code 6

The pager centre is busy. Try again later.

Code 7
There is no response from the location called. Make sure that you have entered a valid pager centre number.

Code 8
Modem error. The modem reported an error message. Make sure that the modem is working.

Code 9
The modem is not answering. Make sure that the modem is properly connected to the telephone line and the computer.

Code 10
Internal software timeout. The modem could not establish connection before the software timeout. Try again later.

Code 11
The pager centre terminated the connection before all data could be sent. Try again later.

8.3.5.6 Pager centre error messages

 The error messages listed below are only related to the All Wireless pager centre (074 0900 108). If a different pager centre was entered, the error messages will not correspond with the following list in some cases.

Code 101
Checksum error. Check your entry.

Code 102
Syntax error. Check your entry.

Code 103
Unsupported operation. Check your entry.

Code 104
Unauthorised operation. Check your entry.

Code 105
Pager silence is turned on. You cannot currently send messages with this number.

Code 106
Invalid number. Check the number: if a user who should be informed by pager is defined, the prefix of the pager (074) may not be entered, because the pager prefix is processed directly at the pager centre. If the pager number is entered with a prefix, the centre reports Error 106.

Code 107
Incorrect password. Check the password.

Code 108
Incorrect legitimation (for all messages). Check the legitimation.

Code 109

Invalid geographic zone (geographic area). Check the geographic zone.

Code 110
Repeat dialling is not permitted. You cannot switch on repeat dialling.

Code 111
Incorrect legitimation for repeat dialling. Check the legitimation.

Code 112
Change of priority is not permitted. You cannot change priority.

Code 113
Incorrect legitimation for change of priority. Check the legitimation.

Code 114
Urgent message characters are not allowed. You cannot add urgent characters.

Code 115
Incorrect legitimation for "urgent message". Check the legitimation.

Code 116
Reverse loading not permitted. You cannot switch on "reverse loading".

Code 117
Incorrect legitimation for "reverse loading". Check the legitimation.

Code 118
"Delayed page transmission" not permitted. You cannot enter a "delayed message transmission".

Code 119
Invalid new password. Check the new password.

Code 120
Invalid new legitimation. Check the new legitimation.

Code 121
Invalid default text. Check the number of the default text.

Code 122
Invalid time indication. Check the time indication.

Code 123
Message class not supported. You cannot send a message of this class.

Code 124
Message too long (has too many characters). Check the message.

Code 125
Requisite default text is invalid. Check the number of the default text.

Code 126
Message class not compatible with the calling class of this number. This number cannot send this kind of message.

Code 150
Network interference or overload. Try again in 30 seconds. Maximum of 12 repeats. After that, please contact technical support (telephone 0800 55 64 64).

Code 151
Incorrect legitimation for default-text user rights. Check the legitimation.

Code 152
No messages saved. No messages saved at the pager centre. If you have subscribed to message storage, messages are stored at the pager centre for 72 hours.

Code 153
Operation partly successful. When paging multiple addresses, not all members received the message. Check your entry.

Code 154
Operation unsuccessful, e.g. the message could not be sent to any of the indicated numbers. Check your entry.

Code 155
System error. The system is currently not working. Try again in 30 seconds. Maximum of 12 repeats. After that, please contact technical support (telephone 0800 55 64 64).

Code 157
Call number is already in this group. Check members of the indicated group.

Code 158
Call number is not a member of this group. Check members of the indicated group.

Code 159
This default text list in invalid. Check the number of the default text list.

Code 161
No authorisation for processing this address list. Check the address list.

Code 162
Default text list is too large. Check the size of the default text list.

Code 163
No authorisation for processing this default text list. Check the number of the default text list.

Code 164
Address list is full. Check the address lists.

Code 165
Invalid group call number. Check the group call number.

Code 166
Operation may only be executed by the caller. Check your entry.

Code 168
Invalid type of call number. Check the type of call number.

Code 169
Call number cannot be included in the list. Check the list.

8.3.5.7 Cityruf error messages

Code 0
The City message was successfully sent.

Code 1
Internal software problem. Your operating system could not allocate the necessary resources. Try again later.

Code 2
The port address could not be found or is being used by another program. Make sure that you have selected the right interface.

Code 3
Internal software problem.

Code 4
The modem could not be found. Make sure that the modem is switched on and properly connected to the computer.

Code 5
The modem has determined that there is no dial tone. Make sure that the telephone line is not in use.

Code 6
The pager centre is busy. Try again later.

Code 7
There is no response from the location called. Make sure that you have entered a valid pager centre number.

Code 8
Modem error. The modem reported an error message. Make sure that the modem is working.

Code 9
The modem is not answering. Make sure that the modem is properly connected to the telephone line and the computer.

Code 10
Internal software timeout. The modem could not establish connection before the software timeout. Try again later.

Code 11
The pager centre terminated the connection before all data could be sent. Try again later.

Code 12
Internal software problem: TAP protocol could not be activated.

Code 13
Internal software problem: SMSC could not send 'go ahead' message.

Code 14
Internal software problem: SMSC did not accept the message.

Code 15
Internal software problem: SMSC interrupted the connection.

Code 16
Internal software problem: SMSC ignored the message.

8.3.5.8 Voice error messages

Code 0
The voice message was successfully sent.

Code 1
Internal software problem. Your operating system could not allocate the necessary resources. Try again later.

Code 2
User input timeout: the telephone user must confirm each voice message by pressing a key.This did not happen within a prescribed length of time (see MallmCfg menu, Settings, Voice Alarm: max. time exceeded for telephone pick-up detection).

Code 3
The telephone is busy. Try again later.

Code 4
Call timeout: the telephone user has not terminated, hung up, acknowledged or confirmed the call in the prescribed time (see MalmCfg menu, Settings, Voice Alarm: max. time exceeded for entire length of telephone conversation).

Code 5
The modem is not defined or is not defined as an appropriate type (not data voice capable).

Code 6
The call could not be initialised; check in the Control Panel that the the Unimodem full duplexer is selected as the sound controller in the Device Manager. Make sure that the modem is working.

Code 7
The call could not be placed.

Code 8
Acknowledgment error: despite the correct ack. code, the alarm could not be acknowledged (DMS error).

8.3.5.9 Spool mode error messages

Code 0
The alarm message was successfully sent via file system.

Code 1
Spool template file does not exist.

Code 2
Spool template file could not be copied.

Code 3
Spool alarm file could not be modified.

Code 4
Spool alarm file could not be copied.

8.4 Historical Data Acquisition (HDAMng.exe)

HDAMng (Historical Data Acquisition) is the data acquisition module of Visi.Plus. With HDAMng it is possible to save histories of process data and variables on a hard disk.

With **HDAMng** the data of sensors and actuators can be read from the DMS and saved to the hard disk with the help of PDBS. If trend data is to be logged, it is mandatory

for the HDAMng as well as the long-term database to be running. Modules which read data from the PDBS, such as pChart, GE or AlmView, do not need the HDAMng.

Which data, when and in what intervals it should be acquired by **HDAMng** are defined in the PET under the column "**Trend**". **HDAMng** stores this in the PDBS database (on the hard disk).

The stored data can be shown and evaluated in process diagrams with the help of the GE through so-called trend graphs.

How a trend graph is created is described in the chapter Operating programs (Graphical Editor).

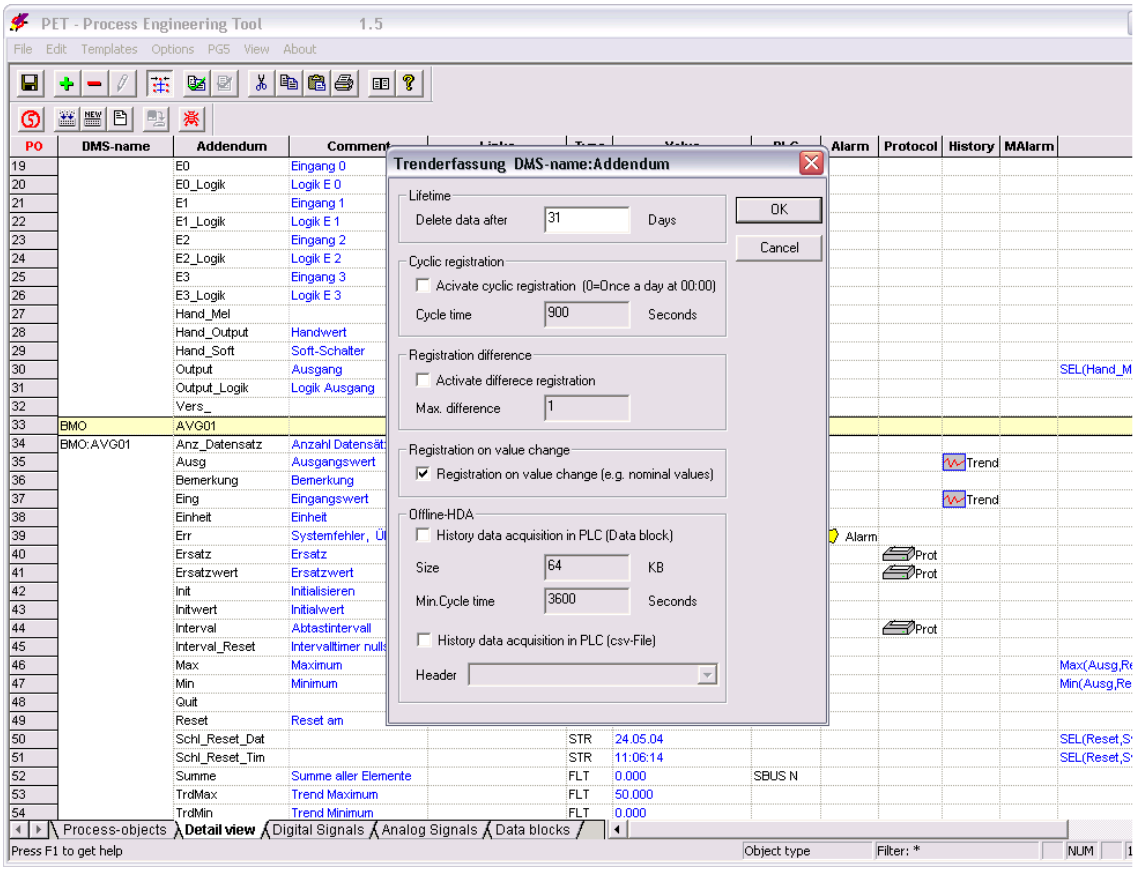
If trend data is to be exported for another use, the module "pChart.exe" can be used for this purpose (see chapter Operating programs).

8.4.1 Inserting a trend in a template object attribute

If you have not yet done so, you must first start up the **PET**.

In order to integrate a trend into a template object attribute (data point), first select the **Template Object View** and then click on the line of the respective data point in the**Trend** column.

*If the "**Trend**" column is not shown, it can be switched on under "**View** > **Trend Column**".*



Trend Lifetime
The lifetime of the data in days is entered under **Time**. In the example the data is saved for 31 days. Afterwards, is is continually overwritten, i.e.:

First logging of the data: 1st day, 2nd day, ... 31st day
then, the 1st day is overwritten, the 2nd day is overwritten, etc.

It is not usually logical to save more than 31 days, because the data is permanently stored in the backup. Access is completely transparent for the user.

In order to permanently save data, use the backup function in under PDBS Menu Settings > Backup (see the corresponding chapter in PDBS).

Cycle Acquisition
Activates/deactivates the intervals at which the data is logged, indicated in seconds, under cycle time.

Acquisition on deviation
Activates/deactivates the logging of the value indicated under **Maximum Deviation**. The data is saved when the **Maximum Deviation** is exceeded.

Acquisition on Change
Activates/deactivates saving with a change of the status or value of the attribute. (e.g. target value and/or digital outputs)

Offline HDA

See corresponding chapter Offline data logging (with code generation)

All attributes for which the trend symbol is displayed are logged by **HDAMng**.

The previously described procedure for trend acquisition in the template objects is advantageous because the trend functions entered for the data points are not automatically inserted (inherited) into all system objects based on this VLO (BMO).

Visi.Plus also allows trend data to be logged in the system objects. However, this poses the risk that these trend data points can be lost during a later change of the VLO and therefore manually adding them again must be considered. Therefore we urgently recommend making all changes directly in the VLO.

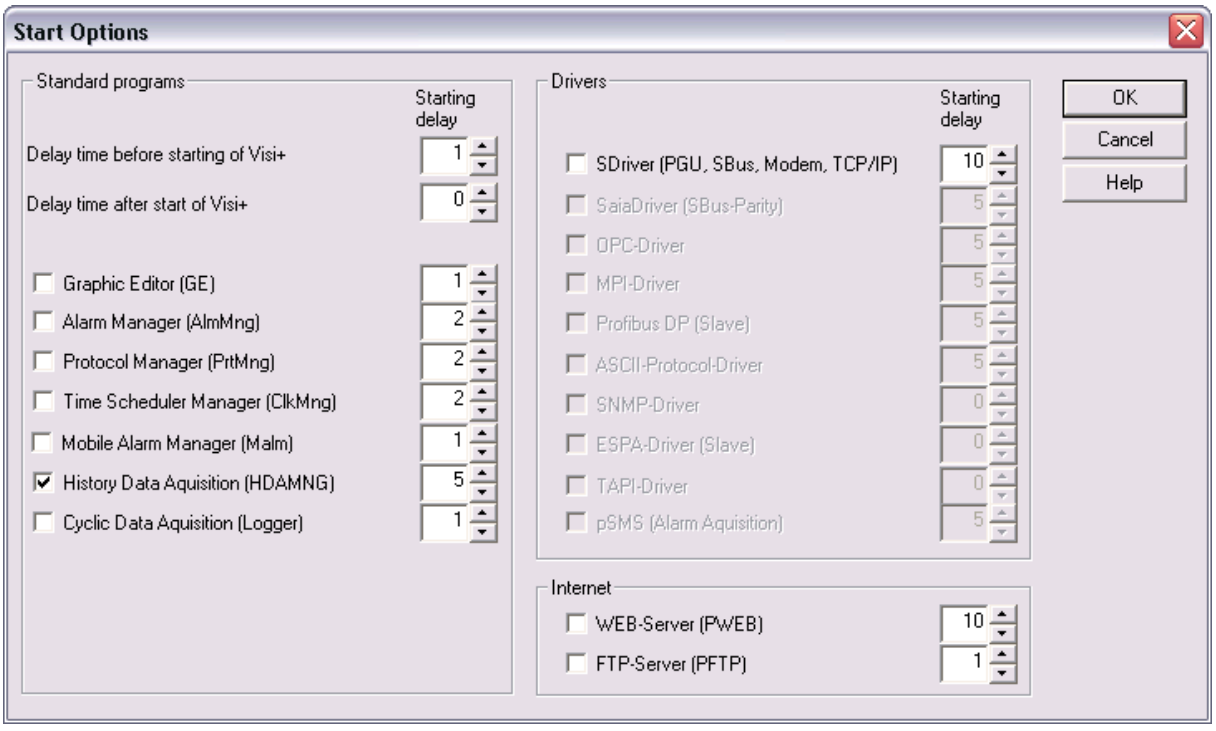
8.4.2 Starting the program HDAMng

As soon as the DMS is loaded, **HDAMng** can be started like any other program by clicking on its filename (HDAMng.exe) in Windows in the "**bin**" directory.

It also makes sense to start the PLC driver beforehand.

This can be done more conveniently via a user-created shortcut on the desktop or via the **Project Manager** module each time Visi.Plus is started. To do so, proceed as follows:

1. Start the **Project Manager module.**
2. Access the menu item or press the **<Start Options>** button.
3. Mark the checkbox **Historical Data Logging (HDAMng)**.



4. HDAMng automatically starts up on the next project start.

The module normally runs as a background process and displayed on the Windows taskbar (bottom right corner of the screen) as a pink ring .

If the program HDAMng is not started, the accruing historical data is not saved!

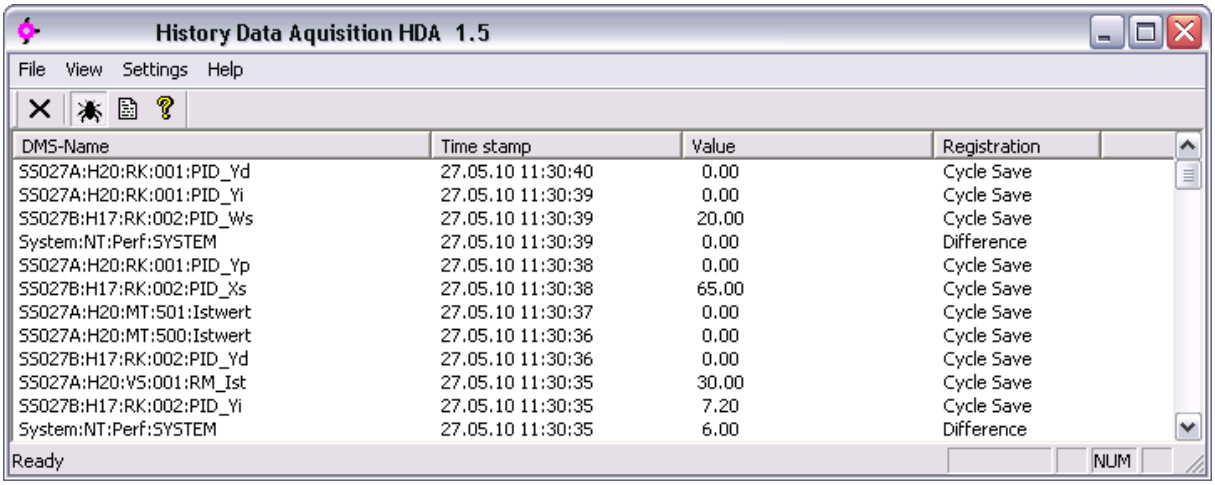
8.4.3 The HDAMng main window

The control window of the HDAMng can be opened in two ways. Either by

double-clicking on the HDAMng icon

or

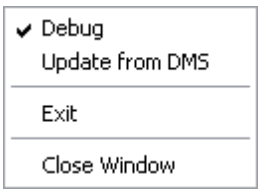
1. clicking with the **right mouse button** on the HDAMng icon on the windows taskbar .
2. Select the "**Display**" command.



8.4.4 The HDAMng menus

8.4.4.1 File menu

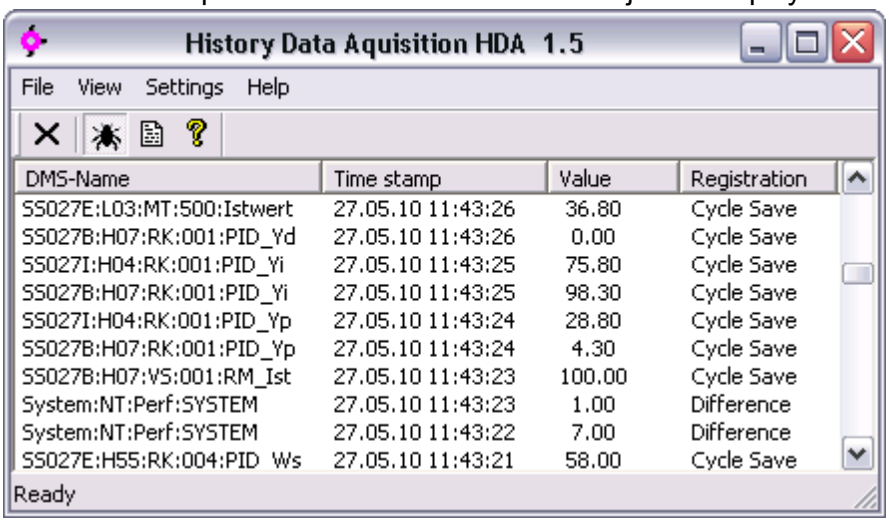
This menu provides functions for exiting and closing the HDAMng module and for debugging the data.



Debug or 

If you wish to display all messages recorded in **HDAMng** on screen, select the "**Debug**" command. By doing so, each data transfer of the PDBS can be tracked on the screen.

The current respective **Actual value** for each object is displayed with a time stamp.



Each modified data point is displayed as follows:

- **DMS name**
  Respective AKS name of the trend data point.
- **Time stamp**
  Date and time.
- **Value**
  current numerical value
- **Acquisition type**
  Reason for acquisition. Here, the following acquisition types apply:
  - **New data**
    When data is acquired for the first time, **New data** is indicated as the acquisition type.
  - **Cycle save**
    If data is acquired cyclically, this takes place through the **Cycle save** acquisition type.
  - **Change**
    The data is acquired with each change.
  - **Difference**
    Data is acquired in the case of significant deviations (adjustable). The measurement interval is thereby shortened in order for the signal response to be precisely registered.
  - **Draw Connect**
    Data Query

Update from DMS
If this command is issued, all data (actual values, limit values, etc.) is reimported from the DMS (Data Management System of Visi.Plus) and updated in the **HDAMng**.

Exit
The **HDAMng** is exited.

Close or 
This menu item closes the main window of the **HDAMng**. This does not quit the program; it is only minimised to the taskbar.

8.4.4.2 View menu

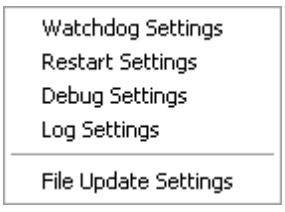
In the menu "**View**" functions are available for modifying the appearance of the **HDAMng**.

Toolbar
Switches the toolbar on or off.

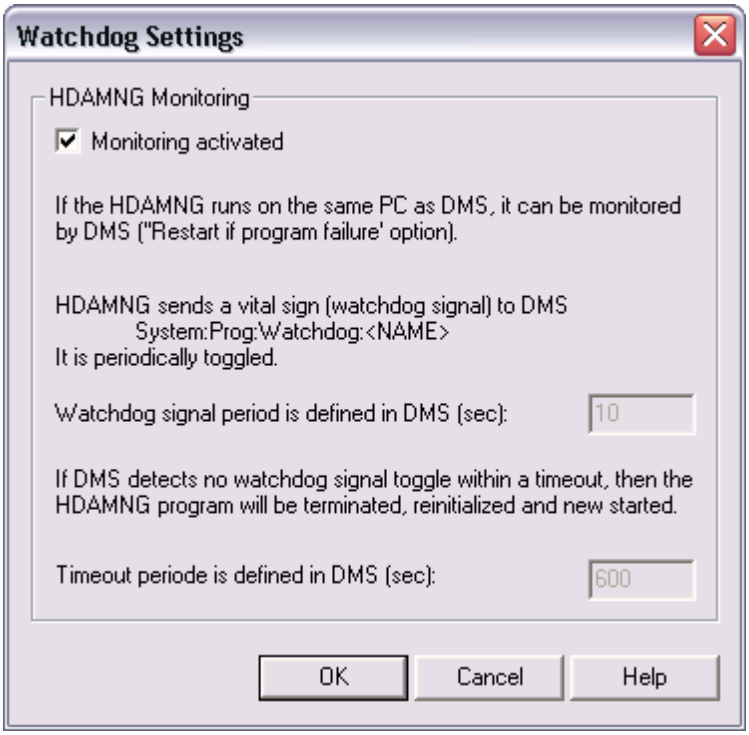
Status bar
Switches the status bar on or off.

8.4.4.3 Settings menu

In the **"Settings"** menu the "**Debug Settings**" and the "**File Update Settings**" of the HDAMng can be specified.



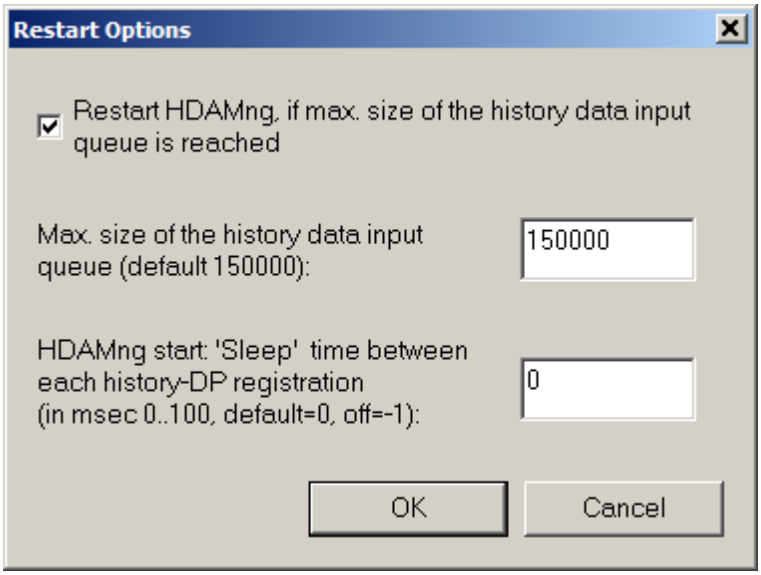
8.4.4.3.1 Watchdog Settings



Monitoring of the AlmMng can be activated through this dialog. With monitoring activated, AlmMng sends a telegram to the DMS at regular intervals. If a telegram does not arrive, the DMS attempts to restart the AlmMng.

8.4.4.3.2 Restart settings

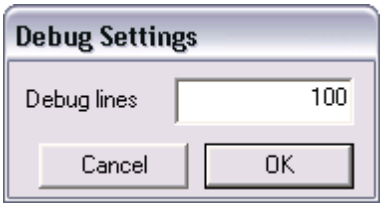
The restart options for the HDAMng can be defined in this setting window. If the system load is very high, it is possible for the HDAMng to become unable to store the data in the PDBS quickly enough. When this happens, the internal buffer which is located in RAM, is filled. A restart can prevent this. Trend data are lost briefly as a result, however, the restart ensures that the RAM is not overloaded unnecessarily, thus negatively affecting other systems.



The buffer size can be defined under Trend Data Input Queue. The default is 150000. Sleep time specifies the pause between individual write accesses. This prevents heavy system loading by the HDAMng. The default here is set to 0. The data items are stored to the PDBS immediately one after the other.

It is recommended that the default settings be used.

8.4.4.3.3 Debug Settings



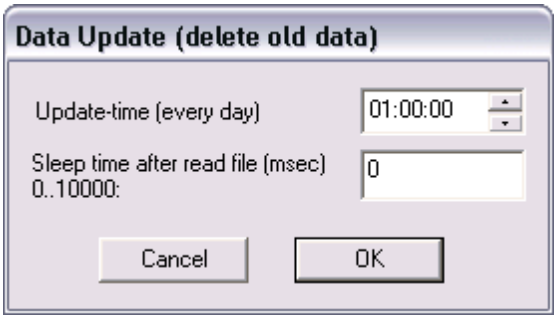
Debug Lines
Defines the maximum number of lines which should be displayed in the Debug window.

8.4.4.3.4 Log Setting

Logging can be switched on or off here and the write cycles can be specified. Logging is useful for problem analysis, but is otherwise not normally used. Consequently, it is recommended that the log function be switched of to reduce system loading.

8.4.4.3.5 File Update Setting

Here you can set when the files should be checked for old data. When the time is reached, the program **HDAMng** sends the **PDBS** the command to delete the older data.



Under Sleep time it is possible to specify how long after the processing of a file the HDAMng should wait before the old data of the next file is deleted. For very large systems, the file update can take a very long time. Using a pause can prevent the system loading from becoming an overload due to the HDAMng.

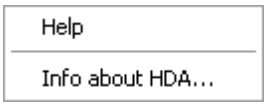
Normally, data that is older than 31 days is deleted from the current database. Older data can be retrieved at any time from the backup files (as long as this is activated in the PDBS) without the user even needing to know.

The databases are also rewritten individually in order to minimise fragmentation of the hard disk.

*The backup function of the PDBS under "**Settings > Backup**" in the menu saves the data to the hard disk. In another case all data whose lifetime has expired is overwritten.*

8.4.4.4 Help menu

The Visi.Plus help and current information about the **HDAMng** database are found in the **"Help"** menu.



Help Topics
The Visi.Plus online help can be accessed through the **"Help Topics"** menu item, via the corresponding **<Help>** button or by pressing the key **<F1>**.

About HDAMng
Displays the current version of **HDAMng**. The version must always be given in case of any support queries!

Should support queries become necessary, the internet forum is preferential to contact via telephone or e-mail. This offers two advantages: First, help is available more quickly; and secondly, other users also benefit from the entries.

8.5 The Log Manager (PRTMng.exe)

With the logging module, any event can be saved and displayed in the Visi.Plus database.

All signals registered in Visi.Plus, such as events and alarms, can be logged and further processed.

The logging format can be generated with the Log Manager module.

How logs and their formats are defined is described in the chapter Format definitions.

Multiple logs can be defined. One of these logs can be assigned to each event.

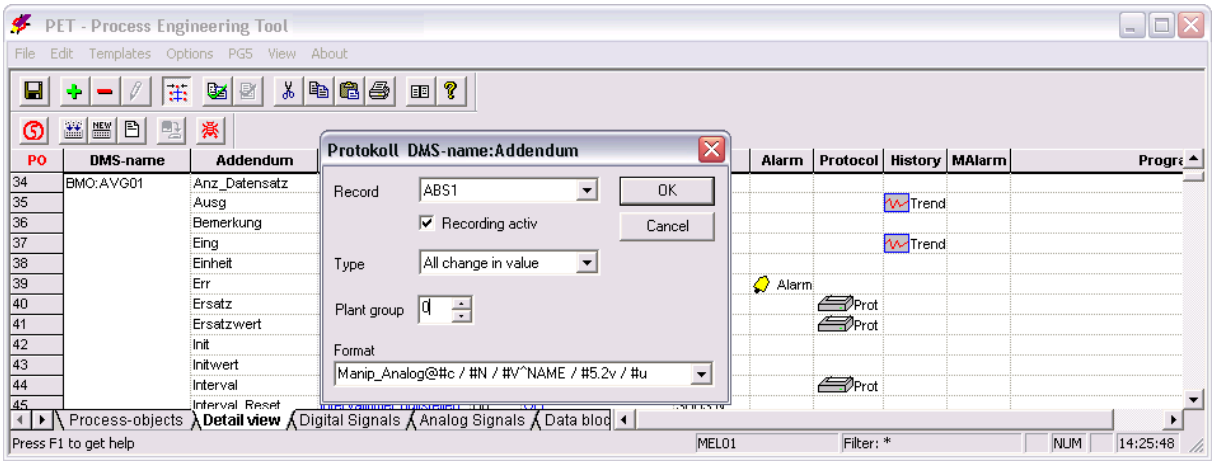
8.5.1 Inserting a log into a template object attribute

Before logs and their formats can be used, they must be defined. The description for this can be found in the **Auxiliary programs chapter** (prtformat.exe).

Then the **PET** should be started.

In order to insert a **log** into a template object attribute (data point), first select the **Template Object View** and then click on the line of the respective data point in the **Log** column.

If the **Log** column is not shown, it can be switched on under "**View** > **Log Column**".



The input window shown here, "**Log ...**" opens.

Log
By clicking on the dropdown menu, previously generated logs are listed and can be selected accordingly.

Logging Active
If this checkbox is marked, the template object attribute is logged.

Type
Determines which event the attribute should be logged for.

System group
When defining the user in pUser, system levels can be determined. The system levels, together with the system group to be determined here, serve as a filter in the Log Viewer. Then log is only shown to the user if the system level corresponds to the system group of the log. For example, if System Group 2 has been defined for the log, the log entry will only be shown to the logged-in user if System Level 2 is available to them. If the value 0 is entered, the log entry can be viewed by any user.

Format
By clicking on the dropdown menu, previously generated formats are listed and can be selected accordingly.

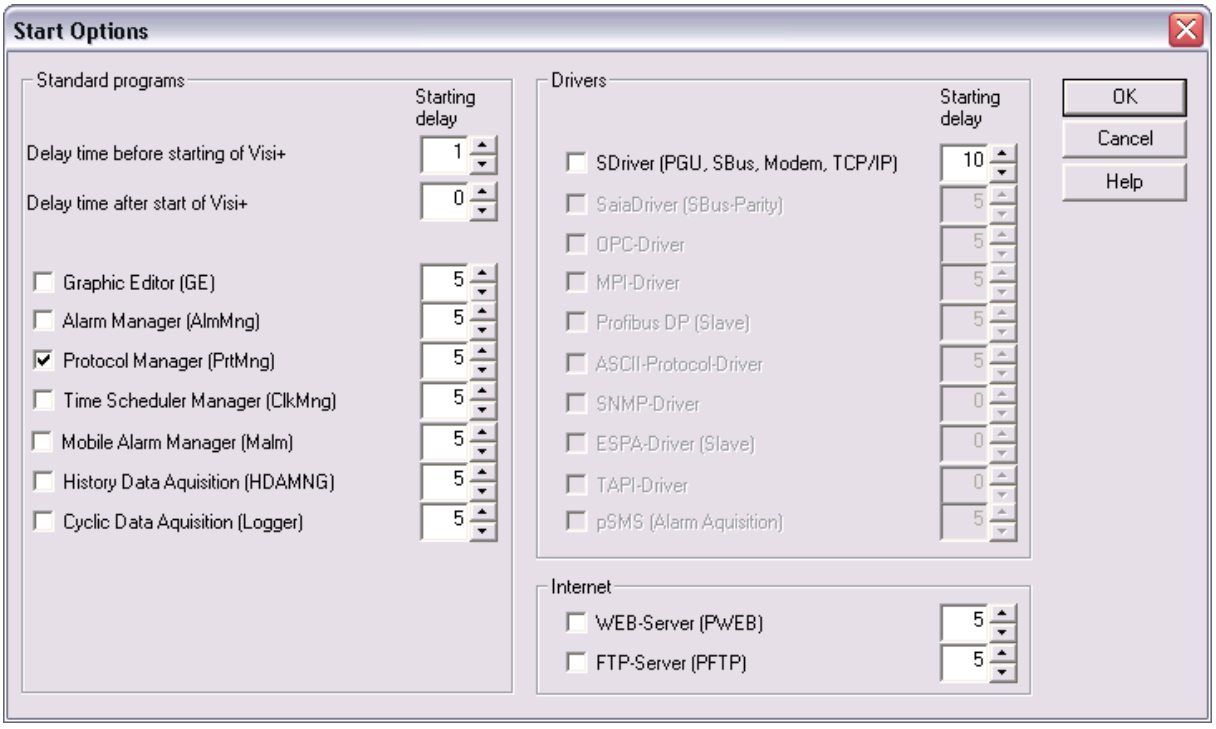
All attributes for which the log symbol Prot *is shown in the PET are logged.*

8.5.2 Starting PRTMng

As soon as the DMS is loaded, PRTMng can be started like any other program by clicking on its filename (PRTMng.exe) in Windows in the "**bin**" directory.

This can be done more conveniently using a user-created shortcut on the desktop or through the **Project Manager** module. Automatic startup each time Visi.Plus is started can be activated in the **Project Manager**. To do so, proceed as follows:

1. Start the **Project Manager** module.
2. Access the menu item or press the **[Start Options]** button.
3. Activate the Log Manager (PRTMng) checkbox.



4. The **Log Manager** starts up automatically the next time a project is opened.

The module normally runs as a background process and is displayed on the Windows taskbar (bottom right at the edge of the screen) as a yellow dot .

If the program PRTmng is not started, the events which occur will not be logged!

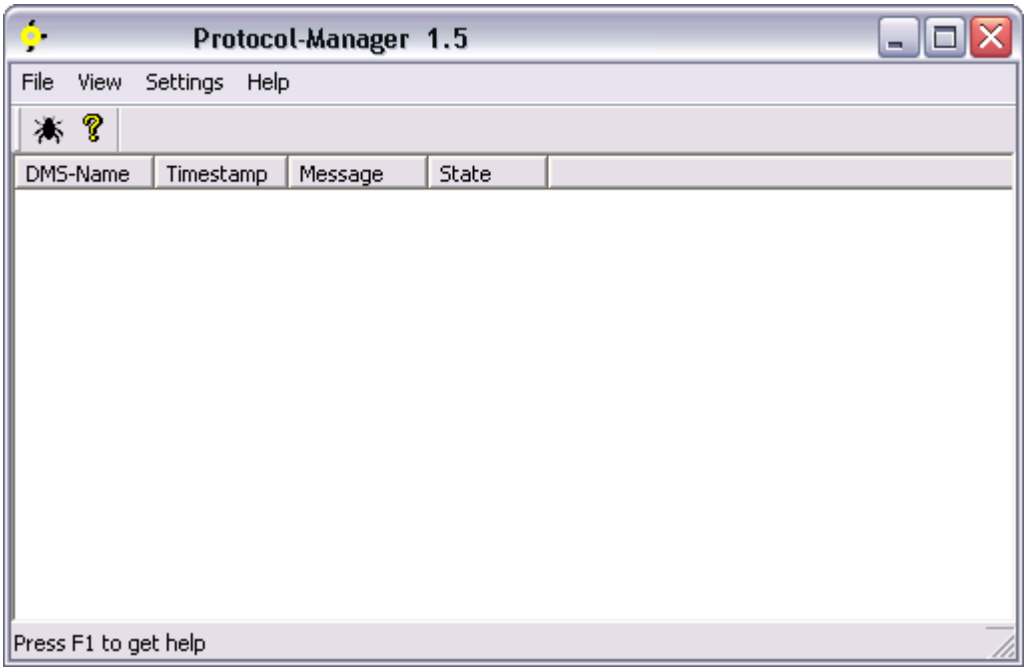
8.5.3 The PRTMng main window

The control window of the PRTMng can be opened in two ways. Either by

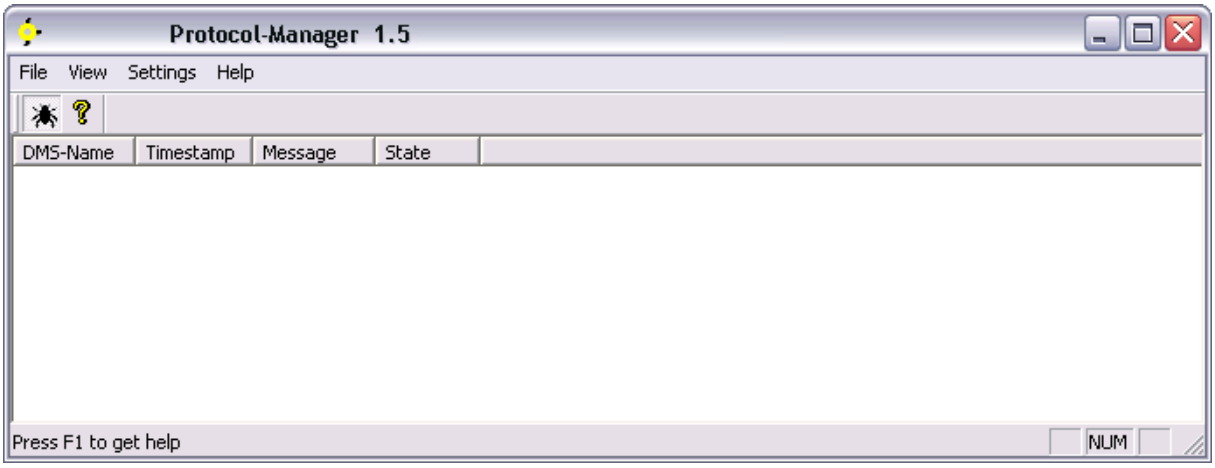
double-clicking on the PRTMng icon

or

1. clicking with the **right mouse button** on the PRTMng icon on the Windows taskbar.
2. Select the "**Display**" command.



With the debug function switched on, the logged events are shown.



Column meanings:

DMS name
Displays the name of the logged event.

Timestamp
Displays the date and time.

Message
The message is shown here on the basis of the log format defined in the PET. The log for the first message above would appear as follows:

#c / #z (On:Off) / #N / #u

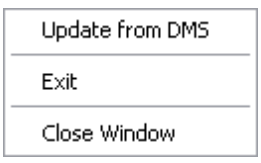
Status
Shows either **OK** or one of the error message numbers used internally by Windows.

How logs and their formats are defined is described in the chapter PRTFormat.exe.

8.5.4 The PRTMng menus

8.5.4.1 File menu

In the **"File"** menu there are options for closing and exiting PRTMng and for updating values.



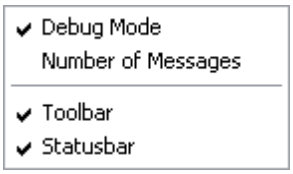
Update from DMS
With this command, all data (actual values, limit values, etc.) is imported from the DMS (the Data Management System of Visi.Plus) and updated in the PRTMng.

Exit
Exits the application PRTMng.

Close
This menu item closes the control window of PRTMng. It does not exit the program; it is only minimised to the taskbar.

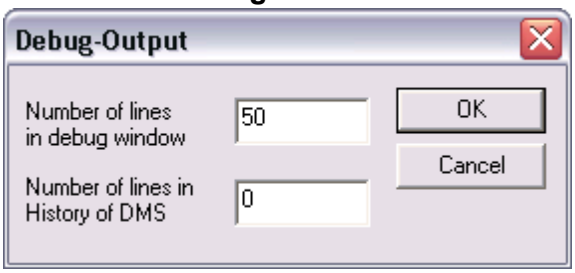
8.5.4.2 View menu

In the **"View"** menu there are options for modifying the appearance of PRTMng.



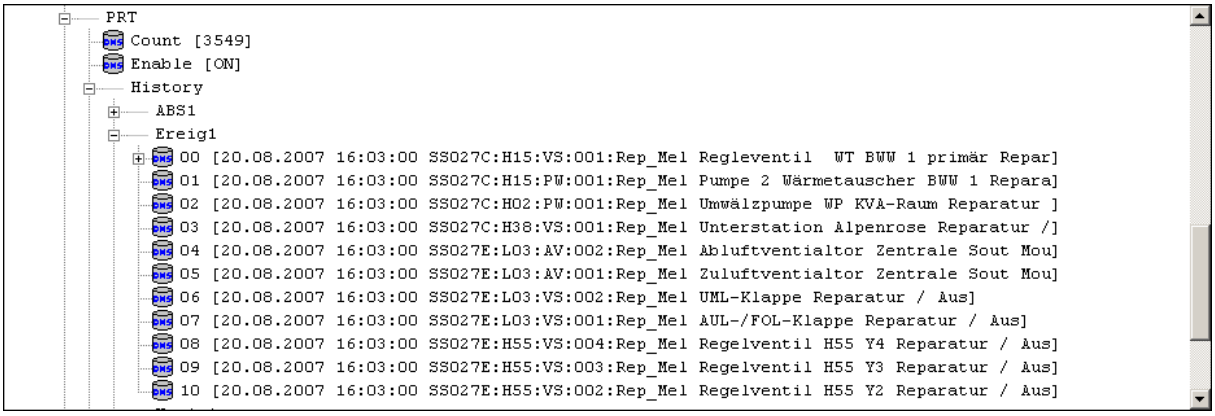
Debug Display or
In order to output all events registered by PRTMng on screen, select the "**Debug Display**" command. By doing so, any logging of the PRTMng can be tracked on the screen.

Number of Messages



Number of Messages in the Debug Window
Here you can indicates the maximum number of messages that should be displayed in the Debug window.

History in the DMS
Indicates how many messages should be entered in the DMS as data points. In doing so, a small history can be created right in the graphical system.

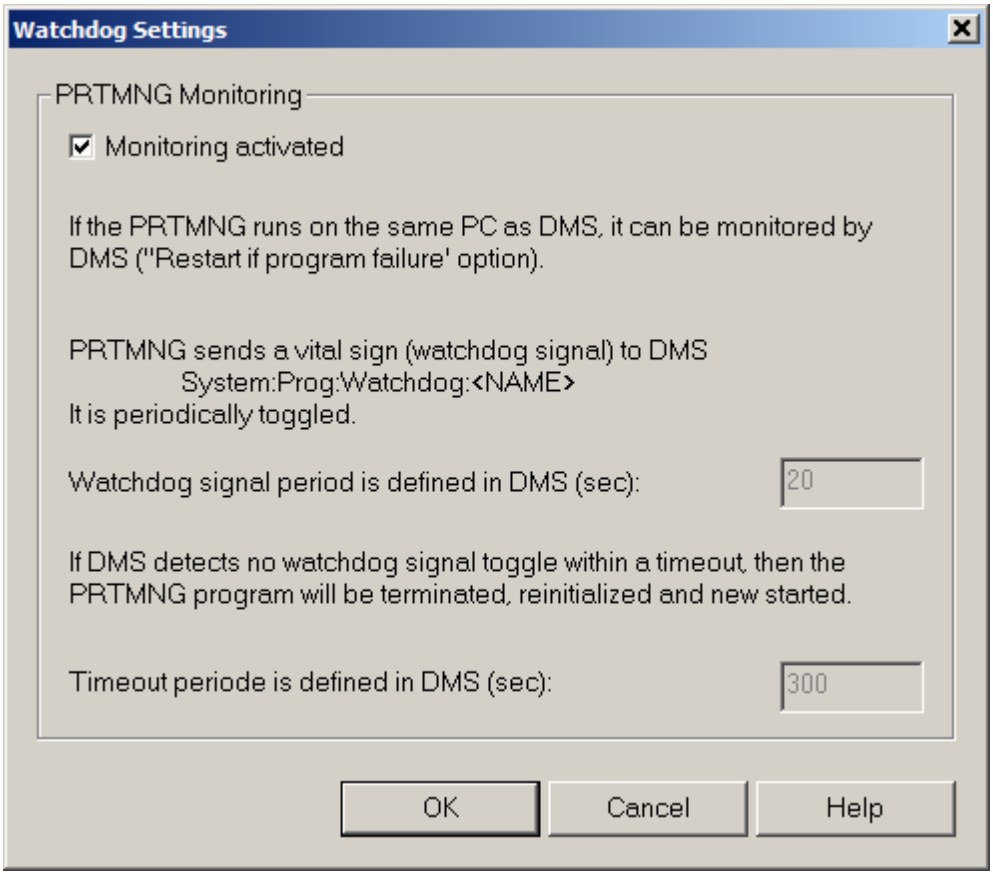


Therefore a list of the latest log entries can be created in the GE. The list is managed in the DMS as a FIFO buffer, meaning the oldest entry is deleted with a new log entry. The size of the FIFO buffer can be specified.
A history is maintained for each log, so that multiple log lists can also be maintained in the GE.

Toolbar
Switches the toolbar on or off.

Status bar
Switches the status bar on or off.

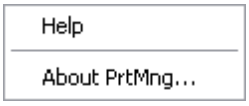
8.5.4.3 Settings



The watchdog can be activated or deactivated in the **"Settings"** menu. If the PrtMng does not report to the DMS for a certain period of time, the DMS restarts the PrtMng. This is only possible if the PrtMng and the DMS are running on the same PC.

8.5.4.4 Help menu

The Visi.Plus online help and current information about the Log Manager (PRTMng) are found in the **"Help"** menu.



Help Topics
The Visi.Plus online help can be accessed through the **"Help Topics"** menu, via the corresponding **<Help>** button or by pressing the **<F1>** key.

About PRTMng
Displays the current version of PRTMng. The version must always be given in case of any support queries!

Should support queries become necessary, the internet forum is preferential to contact via telephone or e-mail. This offers two advantages: First, help is available more

quickly; and secondly, other users also benefit from the entries.

8.6 Time switch functions (CLK)

The Time Switch Program manages and logs all occurring switching processes.

The time switch functions of Visi.Plus are divided into two programs:

1. **Time Switch Manager (CLKMng)**
   The **Clock Manager** is installed on the computer where the DMS runs. It executes all switching processes. If a switching process occurs, the **Clock Manager** reports it to the Visi.Plus database (DMS).

2. **Time Switch Program (CLKCfg)**
   The Time Switch Program manages and lists all configured switching processes. Switching times can be edited with the **Time Switch Program**. You can find more information about **CLKCfg** in the chapter "The Time Switch Program (CLKCfg)".

*Switching processes executed with **CLKMng** pose a risk.*
*If the computer crashes for any reason whatsoever, the switching processes defined in **CLKMng** can no longer be executed!*
*Therefore we recommend executing **important switch time functions** on the **PLC**.*

This is also ideally suited for backup and/or file transfer automation.

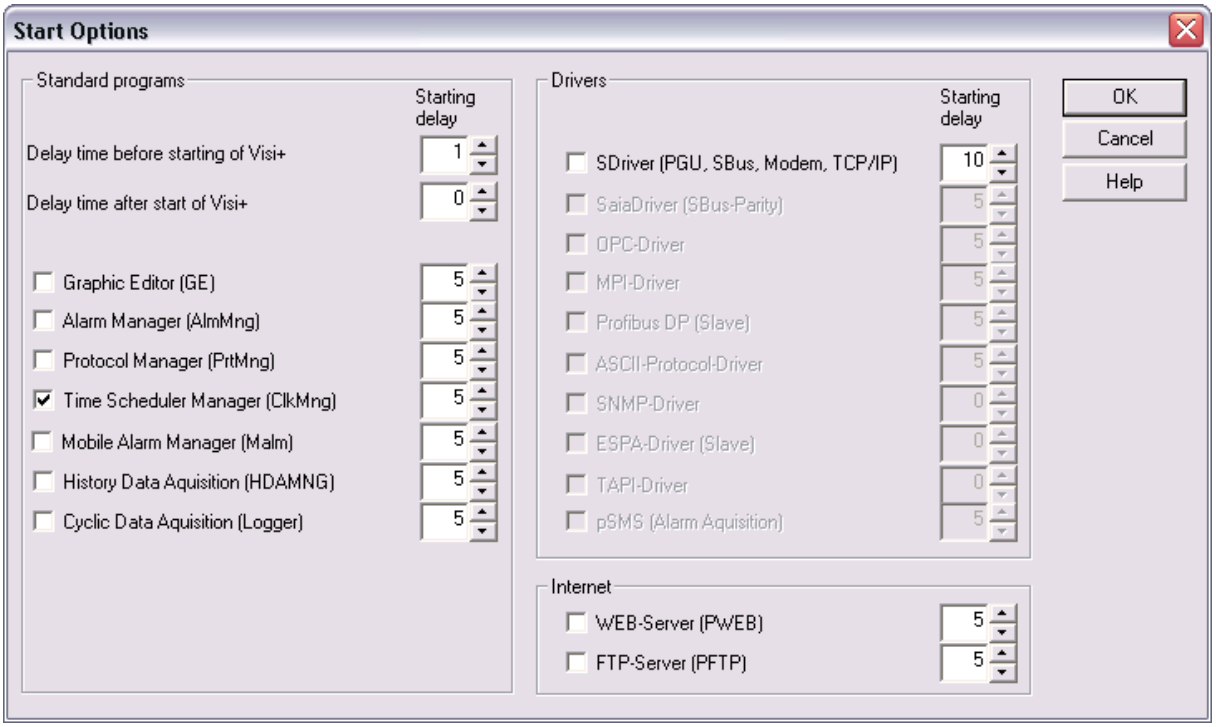
8.6.1 The Time Switch Manager (CLKMng.exe)

The **DMS** and **PDBS** modules must already be started. The **Time Switch Manager (CLKMng)** can be started at any time, either separately like any other program in Windows in the "**\bin**" directory or automatically during the startup of the project (Visi.Plus start options).

If it should be started at the same time as Visi.Plus is started, proceed as follows:

Start the **Project Manager** module and click the **<Start Options>** button.

In the open **Start Options** window, mark the checkbox **Time Switch Manager (CLKMng)** in the Default Programs area. The next time the project starts, **CLKMng** will automatically be started up along with it.



The module normally runs as a background process and is displayed on the Windows taskbar, while it is running, as a small clock (bottom right corner of the screen).

If the Time Switch Manager is not running, the switching processes are not executed.

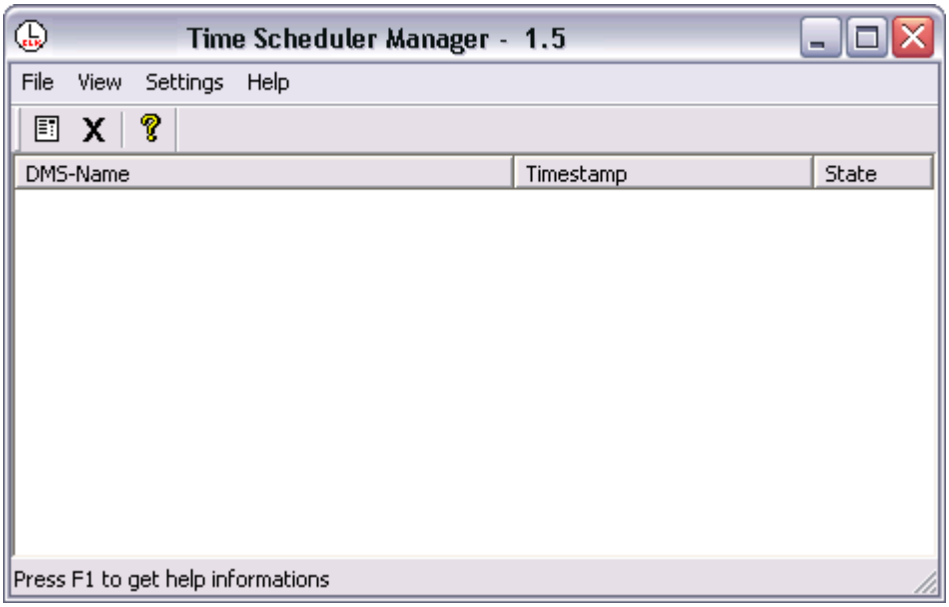
8.6.1.1 The Time Switch Manager main window

The **Time Switch Manager (CLKMng)** can be opened as described in the following:

Double-click on the **CLKMng icon**

or

click on the **CLKMng icon** in the Windows taskbar with the **right mouse button** and choose the "**Display**" command.



Column meanings:

DMS name
Shows the AKS code of the switched element.

Timestamp
Shows the switch time of the element.

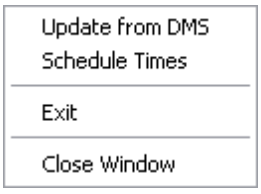
Status
Shows the switch status of the element.

In the preceding image, switch processes that the Time Switch Manager reported to the Time Switch Program and the DMS were logged.

8.6.1.2 Time Switch Manager menus

8.6.1.2.1 File menu

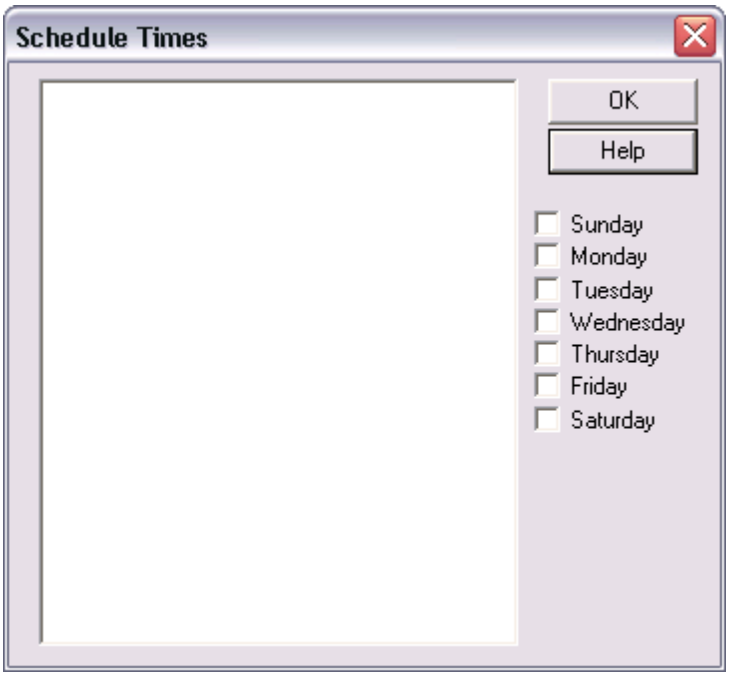
Three options are available in the "**File**" menu: Update values, i.e. "**Update from DMS**", "**Exit**" the Time Switch Manager and "**Close**" the Time Switch Manager main window.



Update from DMS

By selecting this command, all time switch processes (analogue values, digital values, etc.) are reimported from the DMS (Data Management System of Visi.Plus) and updated in the Time Switch Manager. Changes in the DMS are not imported automatically. The switch processes are only imported after restarting CLKMng or by using the command "**Update from DMS**" .

Switch Times
With the command "**Switch times**", CLKMng offers the option of displaying all switch processes defined in the DMS, their respective switch on/off time and the days of the week. The **Switch Times dialog window opens.** A scheduled process can be selected here with the mouse. On the left side you can see which day the process is to be executed on.



Exit
In order to exit the application, all modules registered with CLKMng must first be exited.

Close
This menu item closes the main window of CLKMng. This does not exit the program CLKMng; it is only minimised to the taskbar.

8.6.1.2.2 View menu

You have the option of changing the appearance of the time switch manager in the "**View**" menu.

Toolbar
Switches the toolbar on or off.

Status bar

Switches the status bar on or off.

8.6.1.2.3 Help menu

The Visi.Plus online help and current information about the time switch manager version are found under the **"?" menu**.

Help Topics
The Visi.Plus online help can be accessed through the **"Help Topics"** menu, via the corresponding **<Help>** button or by pressing the **<F1>** key.

About CLKMng
This indicates the version number of the time switch manager, among other things. The version must always be given in case of any support queries!

We recommend our internet forum for support queries. This offers two advantages: First of all, it may provide help more quickly (and beyond regular office hours), and secondly, other users can benefit from the entries.

8.7 The WebAccess (pWA.exe)

8.7.1 Introduction

WebAccess (pWA) makes services for viewing / operating GE images / alarms and logs available via a browser.
The latest technologies such as HTML5 and SVG are used in the process.
The pWA is a new web server and must be licensed separately. Missing licences or the development licence is shown by a watermark in the browser.

The pWeb licence is not automatically carried over.

Please note that the latest browser versions must be used:

- Firefox from version 24
- Chrome from version 29
- IE from version 11

Other browsers have not been tested.

See also:

- http://caniuse.com/
- http://html5readiness.com/
- http://html5test.com/

The pWA functions are carried over from the GE. The images are then converted. Naturally, the behaviour in the process must be identical.
However, the fact that the technology used is different from that of the GE means that we will never be able to carry everything over in identical form. We will never be able to carry over certain functions. Details of all known problems as regards different behaviour to that of the GE are listed in the following chapter.

The pWA can be called up via various operating systems. It does not matter whether the browser calls up the pWA via an OS, Windows or a Linux operating system. However, certain devices restrict the browser properties.
For example, the detailed images for a template object are always opened in a new tab on an iPad. This cannot be avoided by setting the pWA accordingly. It is therefore important to check end devices before using them.

It is also recommended that you use the same browsers. Entry windows look slightly different in all browsers. For an end customer, it can be troublesome if the entry windows look different on each touchscreen.

The window size is carried over from the GE and cannot be adjusted. However, a zoom factor can be set in all browsers. This is saved when the application is exited. However, this setting applies globally for one browser.

8.7.2 Differences compared to the GE

The following GE functions are not supported by the pWA.

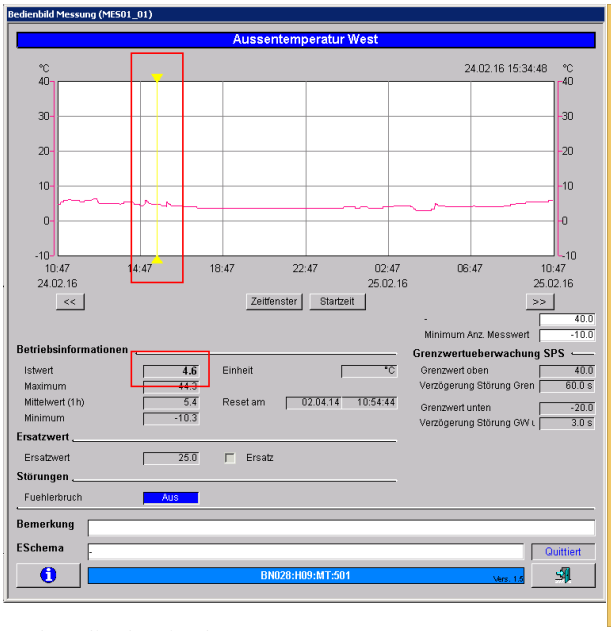
Ruler function:
The ruler function is not supported by the pWA. See also the Ruler chapter for the GE

Close image:
If an image is opened in full screen format in the GE without closing the old image, this image is still available in the background. In the pWeb, it is opened in a new tab if the old image was not closed and the new image is opened in full screen format.

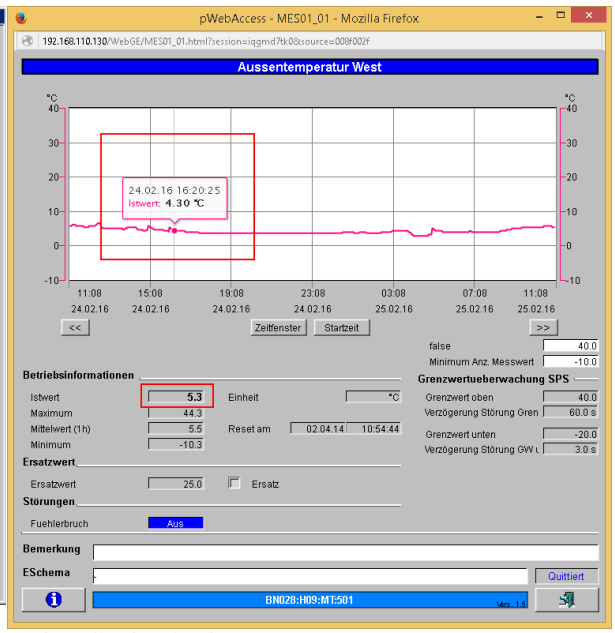
Call up external program:
This does not work in the pWA. However, the following programs can be called up by Visi.Plus:
- AlarmViewer
- ProtocolViewer
- pList
- pdf call-ups
- Internet links

Zoom function for the trend:
Zooming works, but only statically. This means that the normal view will be shown again if the time function is changed.

Value display for the trend curve:
In the pWA, the value display in the trend images is shown in the image itself. In the GE, the actual value is changed according to the ruler. This is no longer the case in the pWA. As a comparison, the two images which make up a detailed image:



Value display in the GE



Value display in the pWA

Resolution if Zoom is active:
If zoom is active in pWA Trend the resolution is smaller than in GE. The zoom function don't get new data as GE doing. If you need a better resolution for data you need to change the time domain.

Value changes for overlapping elements:
If two elements overlap and both should execute a function when activated, only the element on top will take effect.

Image call-up via a drop down list:
This is supported by the GE but not by the pWA.

Links and layers:
These are not shown by the pWA.

Special characters in image names:
These can cause problems when GE2XML is executed.

Text alignment:
The following text alignments do not match the GE:
All 90° and 270° alignments which are not centred in the middle do not match the GE. This applies to both text and button elements.

Alignments:
The alignments for check boxes and radio boxes do not match the GE.

\n for texts in the check box and radio button:

This results in the texts no longer being displayed.

Fill patterns and line attributes:
Certain ones do not match 100%.

Polygon:
A change of frame colour from 2 and from N is not supported.

Radio button and check box:
A change of text colour from 2 and from N is not supported.

Master page:
Elements on the master page are in the background. This means that all page elements are one level higher. This affects combination fields in particular. When expanded, these can be shown in the background. You are therefore advised not to place combination fields on the master page.

Image sub-directories:
Sub-directories are supported. Sub-directories in sub-directories have never been tested and are therefore not officially supported.
Having the same image files in sub-directories should be avoided. The behaviour has not been defined. We strongly advise you not to store back-ups in the same directory.

Dynamic Size changes for Buttons, Text or Editfields:
This is not supported for this Elements.

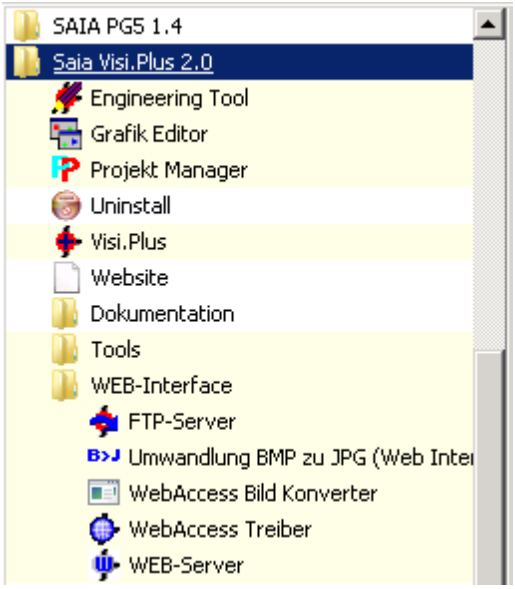
8.7.3 Start and conversion

Unlike with the other Visi.Plus modules, the application files for the pWA are in <Installation Path>/SaaS rather than <Installation Path>/bin

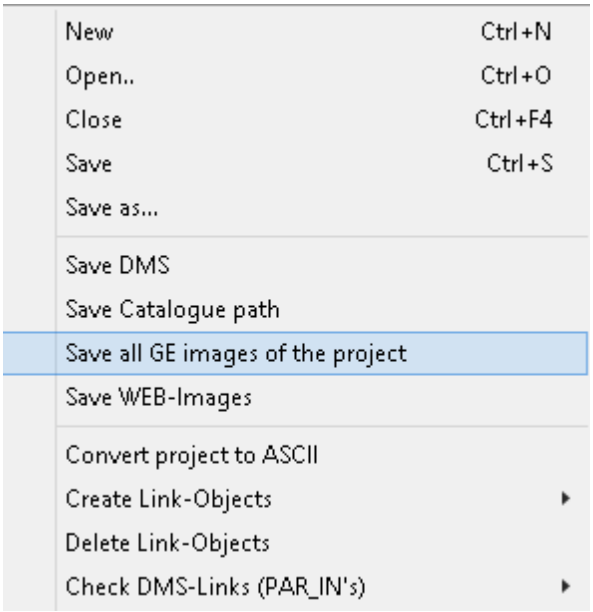
A link is provided in the P1.7 directory under Mager-Driver.

In the start menu, the following entries are made under WEB Interface:

- WebAccess Image Converter
- WebAccess Driver



Nach erfolgreicher Installation müssen als erstes die GE-Bilder konvertiert werden. Es wird empfohlen zuerst jeweils alle Bilder im GE im aktuellen Format zu speichern damit der pWA auch die aktuellen Bildinformationen im neusten Format hat. Dazu den GE Starten und im Edit-Modus unter Datei das Menü



This will open all GE images and save them in the current format.
The pWA can then be started. The application can be executed in the SaaS directory under pWebAccess.exe. pWebAccess checks the GE images and converts them if a new image is available or an image has been changed. However, at least the DMS should be started first.

Conversion can also be carried out manually. To do this, select "WebAccess Image Converter" from the start menu (the DMS or remote DMS must be in operation at this point in time!). The tool can also be found under **GE2XML.exe** in the SaaS folder.
Any errors and warnings can be viewed in the log file "\Visi.Plus\proj\{project name}\scr \GE2XML.log".
If an error message regarding missing libraries (MS) appears, "Microsoft Visual C++ 2010 Redistributable Package" must also be installed.
While WebAccess is operating, all GE images are monitored, i.e. an image which is newly

saved by the GE is automatically converted. From the Setup 1.7.104.x, these libraries are supplied as standard.

WebAccess is configured as standard for operation on Port 80. You should ensure that no other program uses this port.
If another port is needed, this can be specified accordingly in the entry "commServer openTCP web %{serverIp} **80** multi" in the script "\Visi.Plus\SaaS\scripts\WebServer.Script".

You must ensure that any active firewalls allow access to Port 80!

The pWA can operate alongside the pWeb if a different port is used.

WebAccess can now be started via "Start | Visi.Plus| WEB Interface | WebAccess Driver". You will find the application file pWebAccess.exe under /SaaS/ in the installation folder.

After the start, the following symbol will appear in the symbol bar:

Right-clicking on the symbol will bring up the following menu:

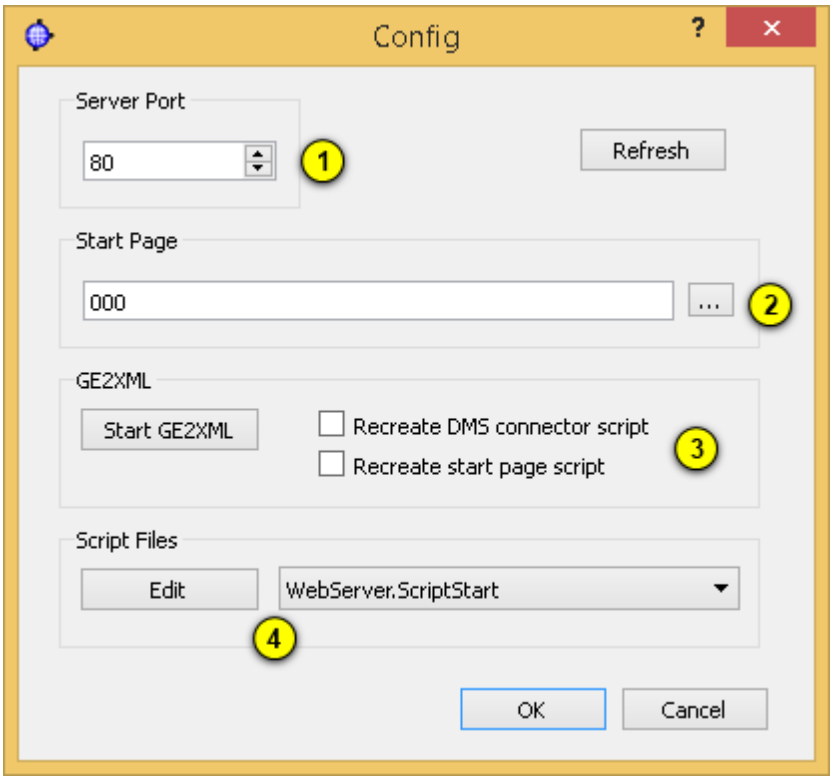
8.7.3.1 Start with Project

If this is enabled, the pWA will automatically be included in the start configuration for the project. As a result, the pWA will automatically start when the project starts.

In the project configurator, the pWA can be selected under the heading Internet in the start options.

8.7.3.2 Configure

The following configuration menu will appear:



1 Port setting: Here, you can define the port under which the pWA runs. The Windows firewall may need to be configured accordingly. With the Refresh button, the port is read in again from the corresponding Config file.

2 Start page: Here, you can define the start page from the GE images. The start page can differ from that of the GE.

3 GE2XML settings:

- Recreate DMS connector script: As a result, the connection settings are reloaded by the GE. If the check box is not active, the settings will not be overwritten. Consequently, the GE and pWA connections to the DMS can theoretically be different.
- Recreate start page script: If this is active, the start page will be carried over from the GE when GE2XML is executed.

The button "Start GE2XML" converts the GE images and carries over the settings from the GE according to the check boxes.

4 Here, you can open the configuration files:

- WebServer.Script
- WebServer.ScriptStart
- WebServer.ScriptEnd
- DMSConnector.Script
- WebServer.ScripStart

- ProjectStart.js

WebServer.ScriptEnde
Settings for monitoring from the DMS. If ComponenStartet.cmd is used instead of pWebAccess (cmd console instead of tray icon), the second line must be commented out.

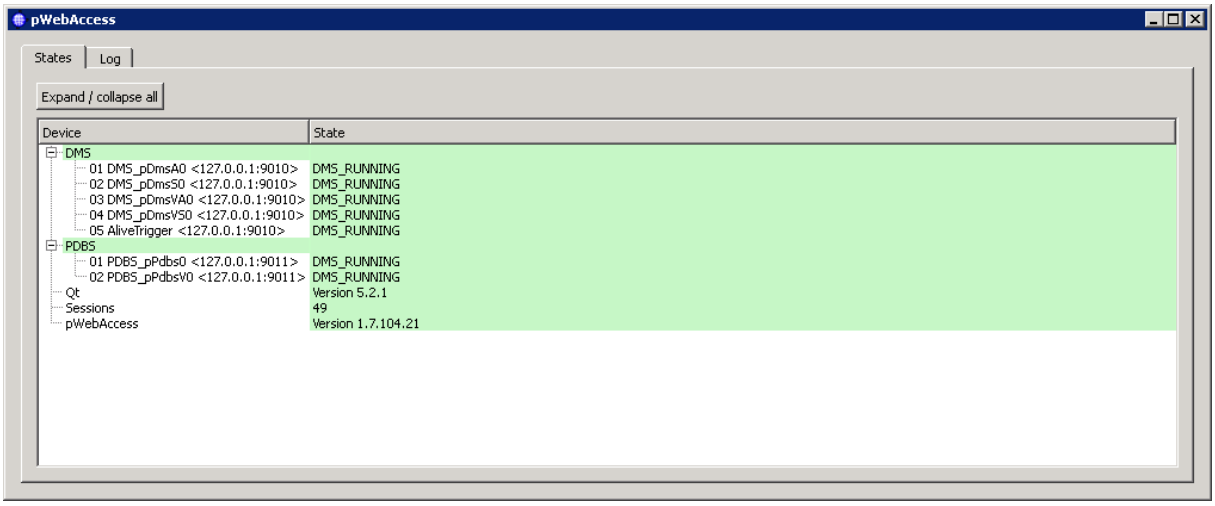
DMSConnector.Script
The connection settings for the various DMS are present here.

WebServer.ScriptStart
The port settings for the pWA are defined in this file. Port 80 is defined as standard.

ProjectStart.js
The start page for the pWA is defined here.

8.7.3.3 State

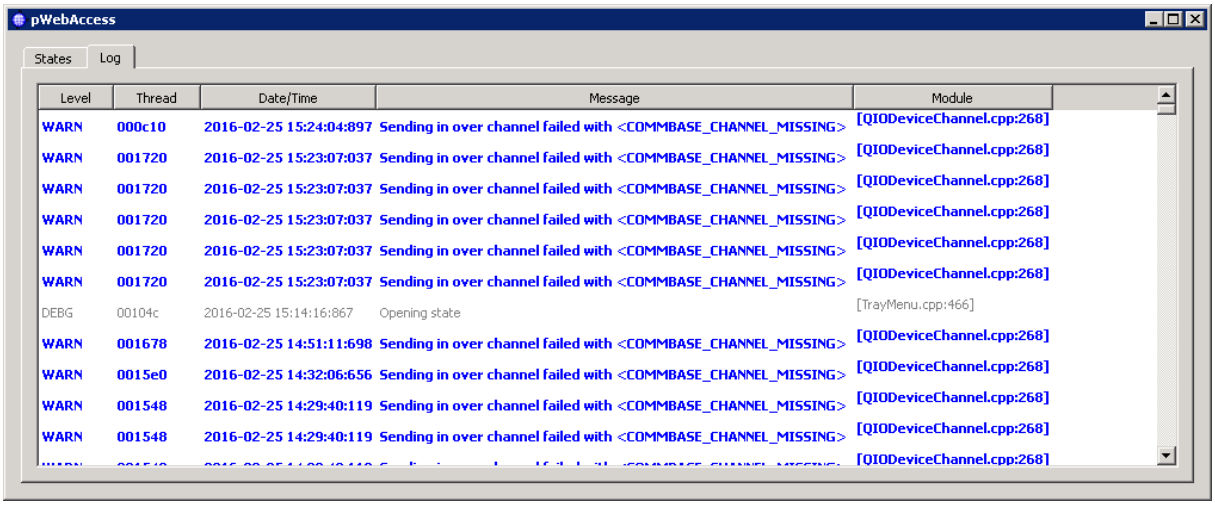
If this menu is opened, the following menu will then appear:



States: The connection and version information is provided here. The following is listed:

DMS/ PDBS:	The connections between pWeb and DMS are listed. 5 DMS and 2 PDBS connections are available per DMS.
Qt:	Qt version number used
Session:	Sessions available
pWebAccs:	Version of the pWA used

The various events are listed under the Log tab:



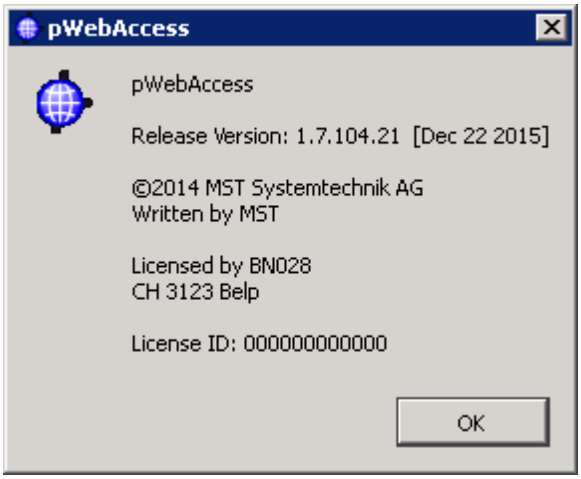
8.7.3.4 Terminate

This button terminates the pWA.

The pWA is currently not terminated with pStopt! This is due to Multi-DMS connections which are possible with the pWA. However, connections to the DMS will automatically be established by the pWA when the DMS starts.

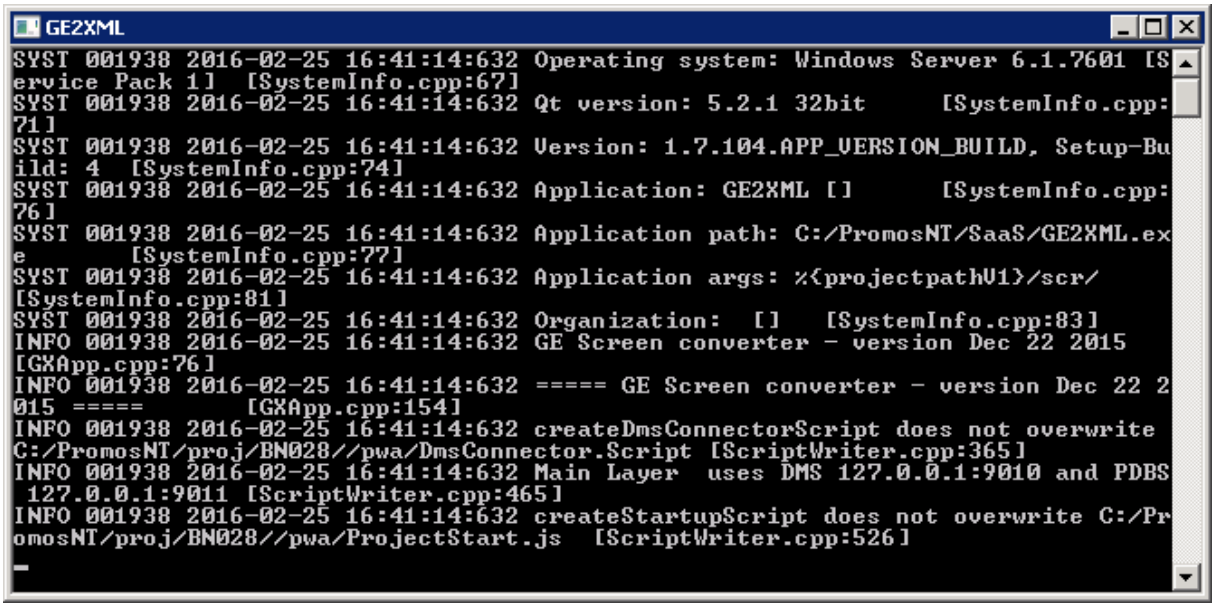
8.7.3.5 About

This button opens the pWA's About menu.



8.7.4 GE2XML

The images are converted for the pWA with the GE2XML.exe script.
In the process, project information such as the start page, connection settings or pUser user data are carried over from the GE.
In order to start GE2XML, at least the DMS and PDBS must be started.



If the images are converted, a new folder with the designation ***pWA*** will automatically follow in the project.

The converted GE images are saved as *.xges*.
There is also an ***Images*** sub-folder with the icon images for the project. Transparent images are saved as .png.

In the Log directory for the project, corresponding log files regarding the converted images are saved:

ge2xml.log *Summary of the conversion of all images.*
GE2XML-<Image Name>.log *Information regarding an individual GE image.*

The most common error messages are those relating to missing data points or different data types of an object in the image for the template project.

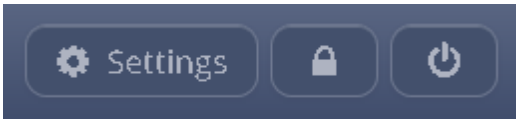
Following conversion, you are advised to view the ge2xml.log log file and deal with any warnings and errors.

If the pWA is running, changed images will automatically be converted when they are saved.

8.7.5 Applications

In order to make operation easier, all applications have similar operating elements.

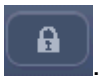
General operating elements (top, right):



- Settings

Opens the Settings dialogue for the relevant application.

- Logon / Logoff

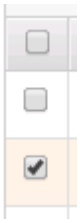
  Logging on/off is possible via the button  .

  The button will be shown as follows after a successful logon:  .

- Close application

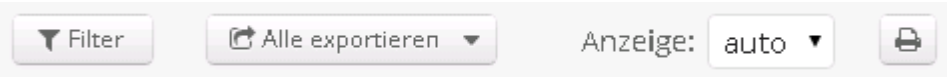
  The application is closed via the button .  .

Select table lines (table, left):



Here, individual table lines can be selected or deselected. Using the table head, all lines shown can be selected or deselected.

Table operating elements (table head, right):



- Filter
  Opens the Filter dialogue for the relevant application. An active filter is highlighted in colour: : 

  .

- Export all
  All available data will be exported. You have the option of carrying out a general export (always the same format) or an export in accordance with the columns shown.

- Display
  You can set the number of lines to be displayed here. Please note that certain browsers may take a long time to display large quantities of data.

- Print
  Allows you to print the page currently displayed.
  If you wish, you can include a logo of your own (top right of the printout) with the file "C:\Visi.Plus\SaaS\www\images\Print_Logo.png".

Table pages operating elements (bottom, right):

These allow you to navigate through the pages.

8.7.5.1 Alarm Viewer

With the Alarm Viewer, the alarms/maintenance jobs generated by the Visi.Plus can be displayed and observed.

You also have various options for filtering current and historical alarms/maintenance jobs.

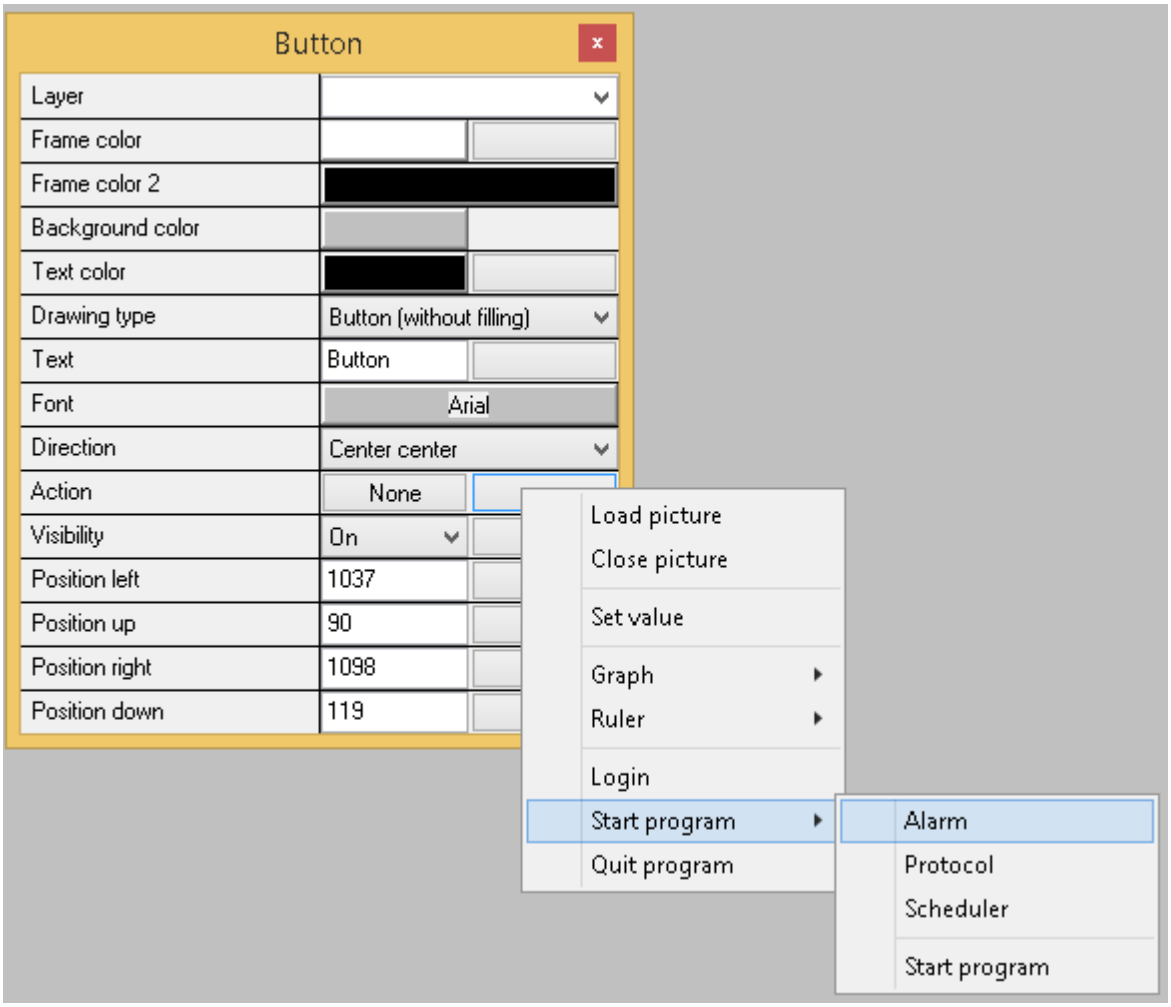
The list is updated within three seconds of an alarm state changing (depending on the client computer's computing power).

A maximum of 5,000 alarms are retrieved from the database (PDBS). In MultiPDBS mode, the number of alarms retrieved per PDBS is reduced accordingly (e.g. five parallel PDBS in operation -> a maximum of 1,000 alarms are retrieved from each PDBS).

In order to display alarms outside these limits, appropriate filters can be used.

8.7.5.1.1 Start

The call-up from the GE is performed via the following action:



The following options are available as parameters:

- Filter state - "/Z="
  E.g.: "/Z=1"
  where 1 means coming, 0 means acknowledged and -1 means going

- Filter DMS name - "/DMS="
  E.g.: "/DMS=033EA:033E:"

- Filter alarm text - "/TXT="
  E.g.: "/TXT=D116"

- Filter priority - "/PRI="
  E.g.: "/PRI=1"

- Filter alarm group - "/GRP="
  E.g.: "/GRP=15"

- Lines to be displayed - "/LIN="
  E.g.: "/LIN=30"

- Start with historical display - "/H"

8.7.5.1.2 Settings

The display can be managed in the settings.

The settings apply to the observer's browser in each case (they are saved in a cookie).

- Table columns
  Here, you can select the table columns that are to be displayed. On the left hand side, you can also manage the order(, moves the entry up/down).

- Various settings

  - *Text display on button*
  This is used for the columns "Alarm image" and "E-scheme", example:
  On: 

, off:: 

  - *Automatically update historical alarms*
  If this option is enabled, historical alarms will be updated in the event of changes. This may significantly impair display performance or cause problems when searching.

8.7.5.1.3 Filterung

Via the Filter dialogue, you can specify various arguments for searching:

The filter set applies to the current display page in each case.
The "Priority" filter is not available for maintenance jobs.

The wildcards "*" and "?" can be used for text filters ("Alarm" and "DMS Name"). "*" signifies any characters and character length and "?" any character in this position.
The DMS name can also be selected via a selection dialogue.

Examples:

"L20:MT:08:" produces for example: "090Va:090V:**L20:MT:08:**1_GW_HE_Mel" and "090Va:090V:**L20:MT:08:**FBr_Err"

"061U:DI:1?:00" produces for example: "061Ub:**061U:DI:1**6**:00**:SM_Err" and "061Ub:**061U:DI:1**4**:00**:SM_Err"

"061U:DI:*:00" produces for example: "061Ub:**061U:DI:**16**:00**:SM_Err", "061Ub:**061U:DI:**14**:00**:SM_Err" and "061Ub:**061U:DI:**17**:00**:SM_Err"

All available options are shown on the right for the filters "State", "Site Group", "Alarm Group" and "Priority".

8.7.5.1.4 Export

The button at the head of the table exports all currently available data sets in CSV format.
If lines were selected, these can be exported using the button at the base of the table.

The following template applies for the file name that is to be saved: "{selected tab}_{current date}_{current time}.csv", e.g. "Alarme_20131108_144927.csv".

The export for all columns always takes place in the following format:

Title in the first line:
pc;created;dms_name;text;state;alarm_group;site_group;priority;e_scheme;

Data in the following lines, where the following applies for "state":
1 = coming
2 = leaving
3 = acknowledged

Example:
pc;created;dms_name;text;state;alarm_group;site_group;priority;e_scheme;
"Z6003";30.01.2014 15:54:30;"160Eb:160E:L00:AK:16:RM_Offen_Err";"ZUL BSK Zone Geschoss V 7.1 Keine Rückmeldung Offen test geht";2;1;0;1;"398Y7";
"Z6003";30.01.2014 15:54:30;"160Eb:160E:L00:AK:16:RM_Offen_Err";"ZUL BSK Zone Geschoss V 7.1 Keine Rückmeldung Offen test geht";2;1;0;1;"398Y7";
"Z6003";30.01.2014 15:54:05;"160Eb:160E:L00:AK:14:RM_Zu_Err";"ABL BSK Zone Geschoss A 3000.2 Keine Rückmeldung Zu test geht";2;1;0;1;"397Y7";
"Z6003";30.01.2014 15:54:05;"160Eb:160E:L00:AK:14:RM_Zu_Err";"ABL BSK Zone Geschoss A 3000.2 Keine Rückmeldung Zu test geht";2;1;0;1;"397Y7";

When the visible columns are exported, the columns configured for viewing are exported accordingly.

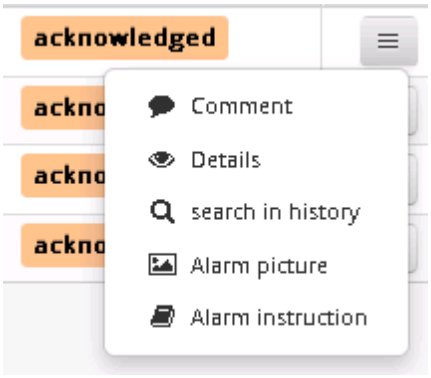
Example:
"Date / Time";"DMS Name";"Alarm Text";"State";
30.01.2014 15:54:30;"160Eb:160E:L00:AK:16:RM_Offen_Err";"ZUL BSK Zone Geschoss V 7.1 Keine Rückmeldung Offen test geht";"geht";
30.01.2014 15:54:30;"160Eb:160E:L00:AK:16:RM_Offen_Err";"ZUL BSK Zone Geschoss V 7.1 Keine Rückmeldung Offen test geht";"geht";
30.01.2014 15:54:05;"160Eb:160E:L00:AK:14:RM_Zu_Err";"ABL BSK Zone Geschoss A 3000.2 Keine Rückmeldung Zu test geht";"geht";

The state is output here in the clear text.

8.7.5.1.5 Acknowledgement

Upcoming alarms can be acknowledged in the following ways:

- Via the  button in the "Acknowledge" column
- Via the "Acknowledge" function under the "Options" button:



- Via the 

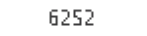
button for all selected alarms

Important:
Depending on the server setting in the script "\Visi.Plus\SaaS\Webserver.Script", configuration "alarmServer setAcknowledgeWithLogin [yes|no]", upcoming alarms can only be acknowledged in logged in mode.
A login is required as standard.

8.7.5.1.6 Electro Scheme

The Alarm Viewer provides the option of displaying Electro Scheme (incl. navigation to the reference object).
If the EScheme column is enabled, the following displays are possible:

There is no entry in the "EScheme" data point.



A reference is given in the "EScheme" data point but no corresponding scheme was found.

or 
The corresponding scheme will now be called up and displayed (can also be selected via the Options button).
The recorded reference (in this case "162B6") will be passed on to the display program as a search parameter.
However, the search will only work with Acrobat Reader (integrated into the browser).

The requirements for displaying the scheme are:

- The "EScheme" data point in the corresponding object must contain a valid reference
- Electro Scheme is on the server in the project folder ".../doc/escheme/" (as PDF file).

The following naming details must be observed:

a) In the "EScheme" data point, a "*File Name:*" was given first, e.g. "**ES_U502:** 120X0"
- The exact file name will be searched for as the first priority (in this case: **ES_U502**.pdf),
- A file name which contains the configured term will be searched for as the second priority.
-> E.g. the file "Electroscheme_**ES_U502**_2013.pdf" will be found

b) The (first) file name which contains an AKS part (with at least 4 characters) will be searched for.
E.g. AKS: "**530Ub**:530U:DI:24:04:EScheme"
-> "110324_1_2029 ASM NORD1_**530Ub**_.pdf" will be found
or AKS: "530Ub:**530U**:DI:24:04:EScheme"
-> "110324_1_2029 ASM NORD1_**530U**_.pdf" will be found

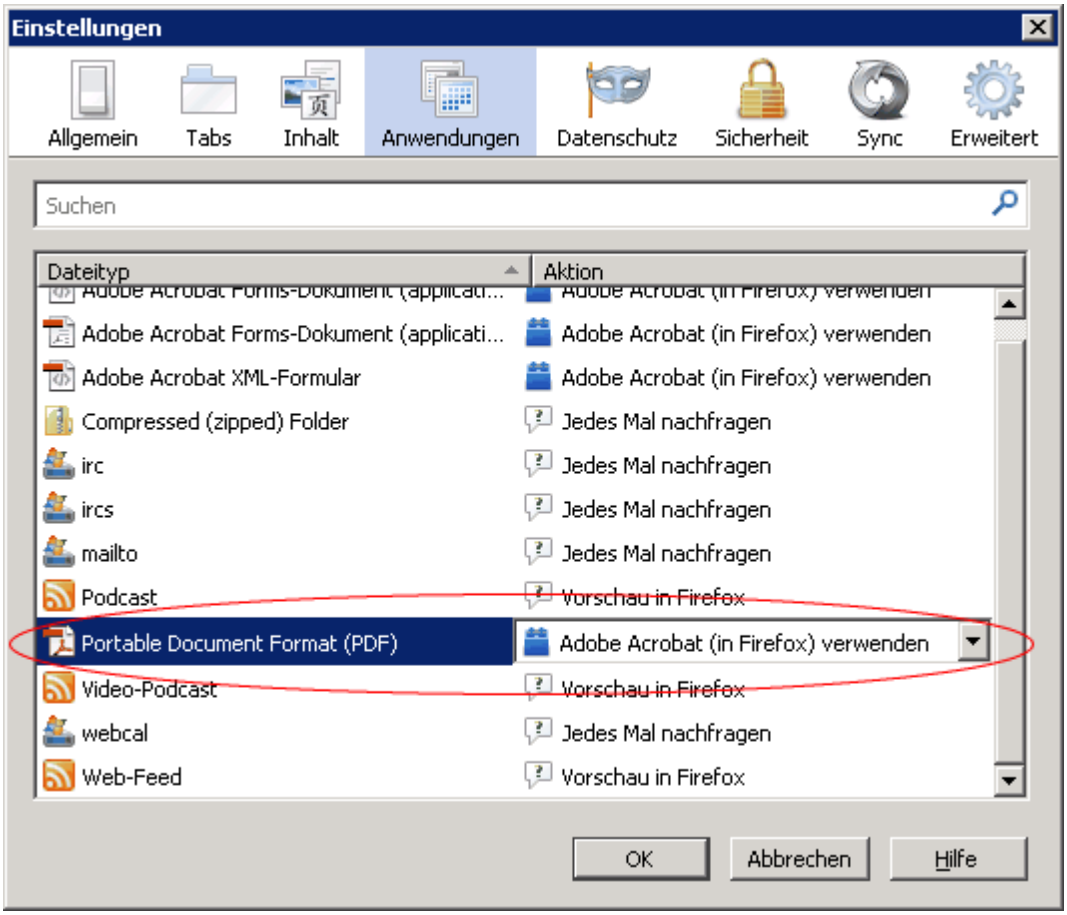
8.7.5.1.6.1 Browser setting

Adobe Acrobat Reader must always be installed.

(See also: : http://helpx.adobe.com/acrobat/kb/pdf-browser-plugin-configuration.html)

Settings in Firefox:

The "Tools | Options" menu takes you to the following configuration window (the display may vary slightly depending on the version of Firefox used):



Under "Portable Document Format (PDF)", "Use Adobe Acrobat (in Firefox)" must be set. Other plug-ins may be available – these must be tested in each individual case. Setting use "...(in Firefox)..." is important.

(see also https://support.mozilla.org/de/kb/PDF-Dateien-in-Firefox-oeffnen)

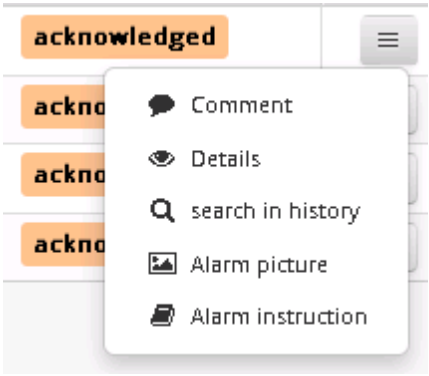
Settings in Chrome:

The following configuration window will appear when you enter the URL "chrome://plugins/":

"Chrome PDF Viewer" must be disabled and "Adobe Reader" enabled here:

8.7.5.1.7 Options

Via the Options button, various functions can be executed. The relevant functions are only shown if they are available:



- Acknowledge
  Acknowledgment of the corresponding alarm.

- Alarm image
  With this function, a switchover to the corresponding image takes place.
  The name of the image must be recorded in the DMS in the corresponding alarm data point ("ALM:Screen").

- Commentary
  A dialogue window in which any texts can be entered is opened.
  The texts are saved on the server and can therefore be viewed / edited by any user.
  If a commentary has not yet been provided for this data point, a text with the date and user name (if logged in) will be shown.

- Details
  Window with details of the entry.

- Search in history
  The alarm data point is searched for in the history and the relevant entries are displayed.

- Instructions (alarm / maintenance)
  The relevant instruction is opened in a dialogue or window.
  Files for display can be saved on the server in the project directory ".../rtf/". Pure text files (.txt) are displayed in a dialogue, and a new window is opened for other file types.
  The name of the file to be displayed must be recorded in the DMS in the corresponding alarm data point:
  For alarms in "..:ALM:File"
  For maintenance jobs in "...:ALM:Alarm1:ExtAlmText"

8.7.5.2 Protocol Viewer

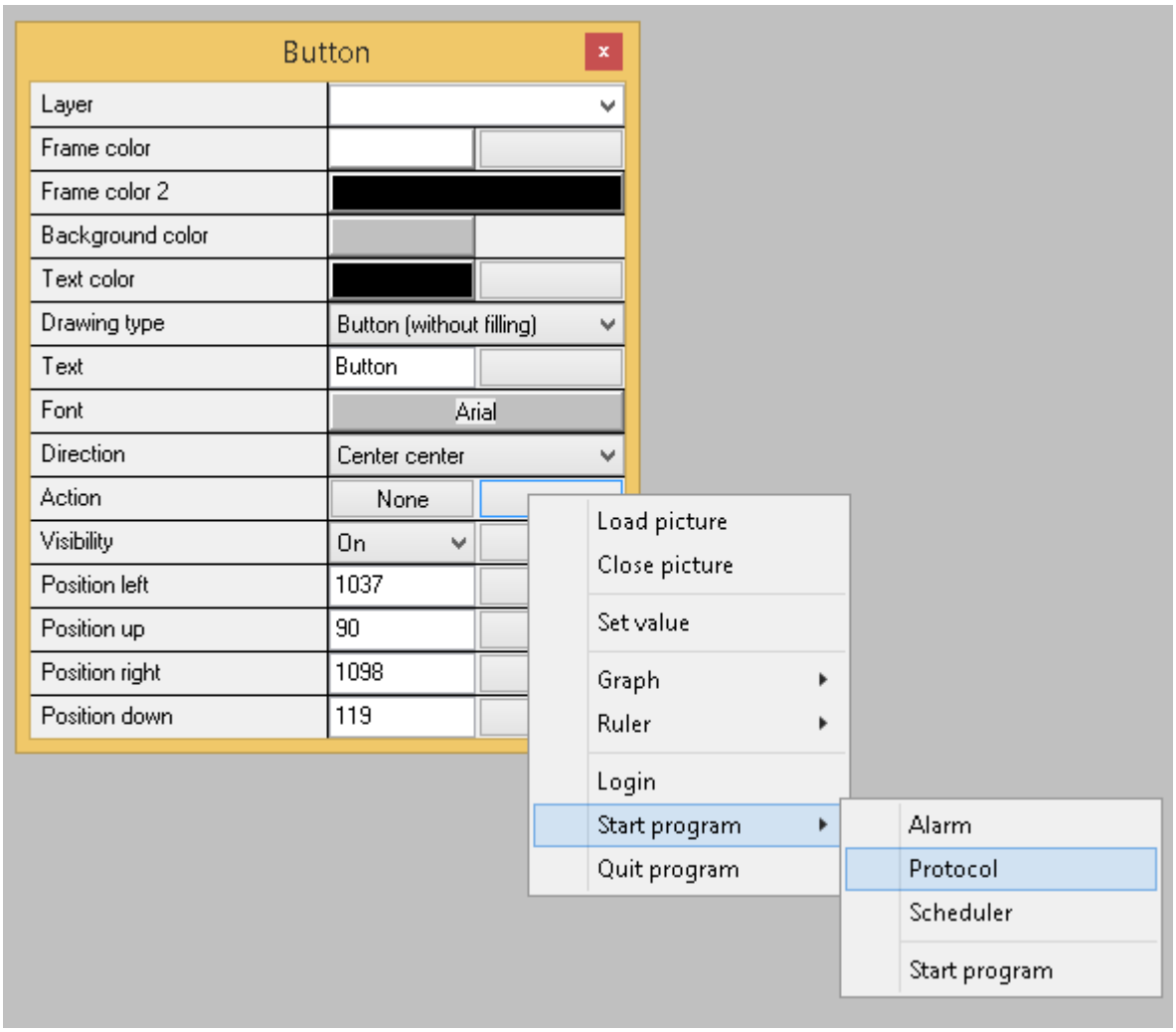
With the Protocol Viewer, recorded protocols are visualized:

A maximum of 5,000 messages are retrieved from the database (PDBS). In MultiPDBS mode, the number of messages retrieved per PDBS is reduced accordingly (e.g. five parallel PDBS in operation -> a maximum of 1,000 messages are retrieved from each PDBS).

In order to display protocols outside these limits, appropriate filters can be used.

8.7.5.2.1 Start

The call-up from the GE is performed via the following action:



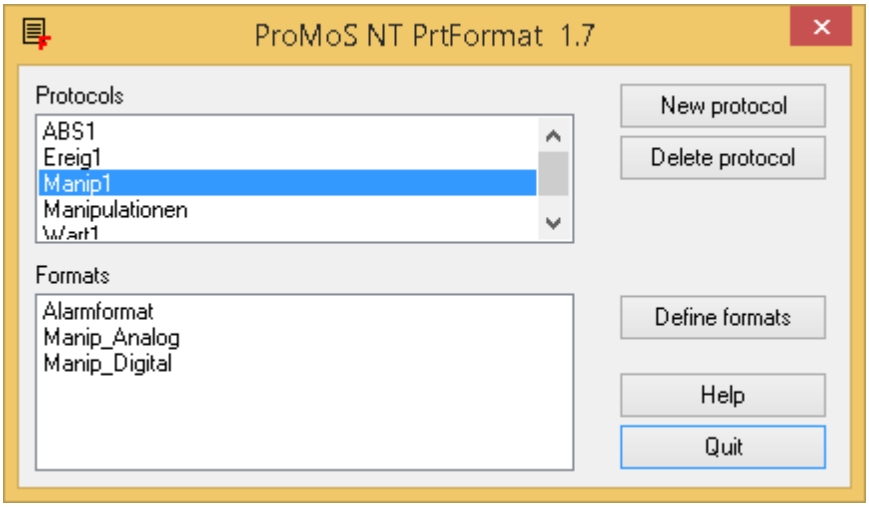
The following options are available as parameters:

- Protocol to be displayed
  E.g.: "Login"
  Multiple protocols can also be given, e.g.: "LOGIN ALARM"

- Filter DMS name - "/DMS="
  E.g.: "/DMS=033EA:033E:"

- Filter text - "/TXT="
  E.g.: "/TXT=D116"

If no protocol to be displayed was given as a parameter, all protocols set via "PrtFormat" will be displayed.



8.7.5.2.2 Settings

The display can be managed in the settings.

The settings apply to the observer's browser in each case (they are saved in a cookie).

- Table columns
  Here, you can select the table columns that are to be displayed. On the left hand side, you can also manage the order (, moves the entry up/down).

- Various settings

  - *Automatically update data*
  If this option is enabled, the databases will be checked for new messages every 30 seconds.

8.7.5.2.3 Filtering

Via the Filter dialogue, you can specify various arguments for searching:

The filter set applies to the current display page in each case.

The wildcards "*" and "?" can be used for text filters ("DMS Name" and "Text"). "*" signifies any characters and character length and "?" any character in this position.
The DMS name can also be selected via a selection dialogue.

See also the examples for the AlarmViewer .

8.7.5.2.4 Export

All currently filtered data sets are exported in CSV format.

The following template applies for the file name that is to be saved: "{selected tab}_{current date}_{current time}.csv", e.g. "ABS1_20131108_144927.csv".

The export for all columns always takes place in the following format:

Title in the first line:
`pc;created;dms_name;text;`

Example:
```
pc;created;dms_name;text;
"Z6003";04.11.2013 16:29:08;"510Ua:510U:L25:AK:03:ABS_Ein";"ERG-Klappe Ein";
"Z6003";04.11.2013 16:28:22;"510Ua:510U:L25:AK:02:ABS_Ein";"FOL-Klappe Ein";
"Z6003";04.11.2013 16:27:36;"510Ua:510U:L25:AK:01:ABS_Ein";"AUL-Klappe Ein";
"Z6003";04.11.2013 16:26:38;"510Ua:510U:L25:VE:03:ABS_Ein";"LKU-Ventil Ein";
"Z6003";04.11.2013 16:25:40;"510Ua:510U:L25:VE:02:ABS_Ein";"LE-NW Ventil Ein";
```

When the visible columns are exported, the columns configured for viewing are exported accordingly.

Example:
```
"Date / Time";"DMS Name";"Text";
04.11.2013 16:29:08;"510Ua:510U:L25:AK:03:ABS_Ein";"ERG-Klappe Ein";
04.11.2013 16:28:22;"510Ua:510U:L25:AK:02:ABS_Ein";"FOL-Klappe Ein";
04.11.2013 16:27:36;"510Ua:510U:L25:AK:01:ABS_Ein";"AUL-Klappe Ein";
```

8.7.5.3 DataList

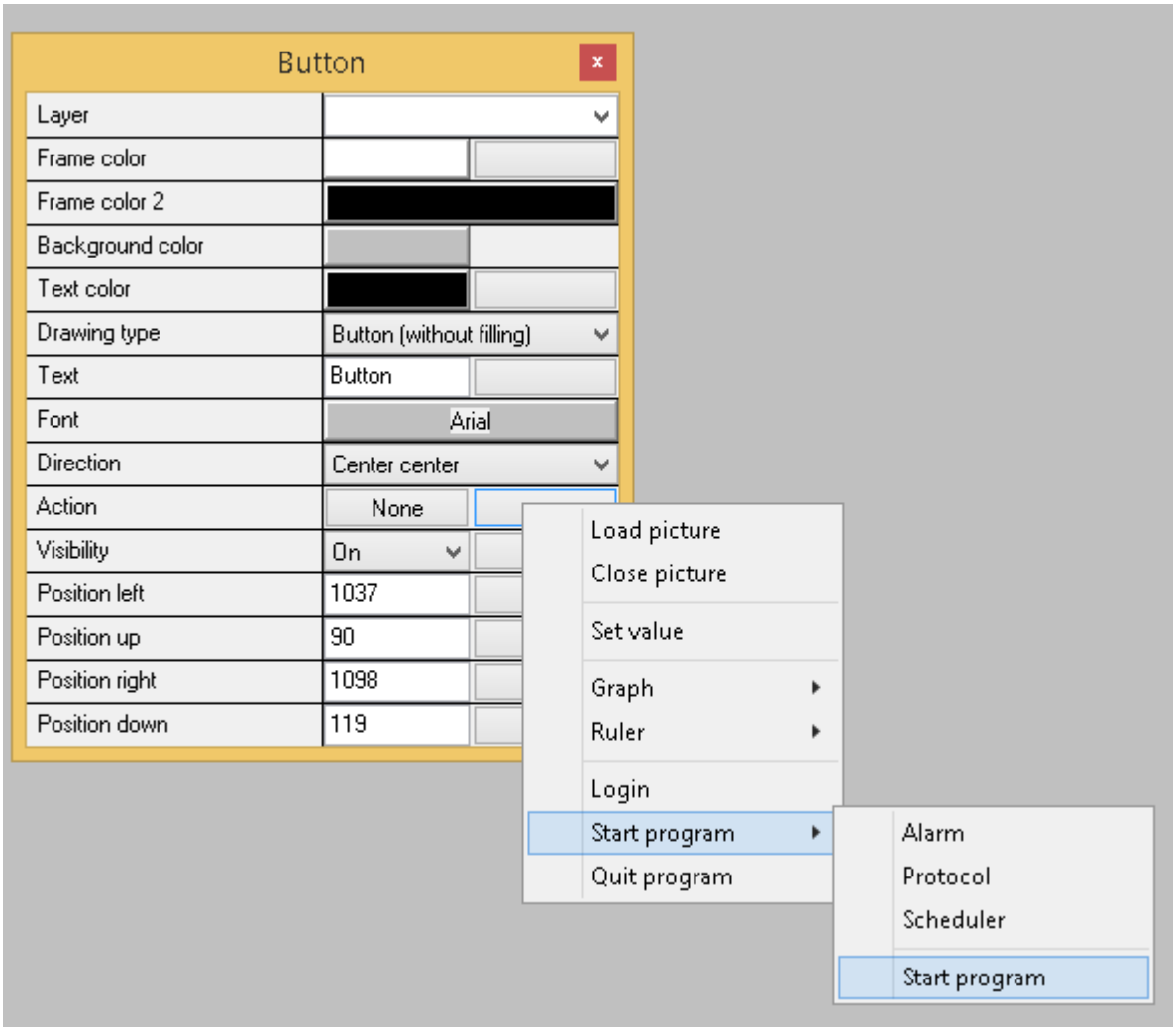
With DataList, values can be visualized and changed.

Multiple values can only be changed from a specific user level.

As standard, this is from Level 8 (Level 8 or a higher level is permitted). The level can be configured accordingly in the script "\Visi.Plus\SaaS\scripts\WebServer.Script" in the entry "dataListServer setAccessLevelMultiModify **8**".

8.7.5.3.1 Start

The call-up from the GE is performed via the following action:



As the "program to be started", either "**pList**" or "**DataList**" can be specified.

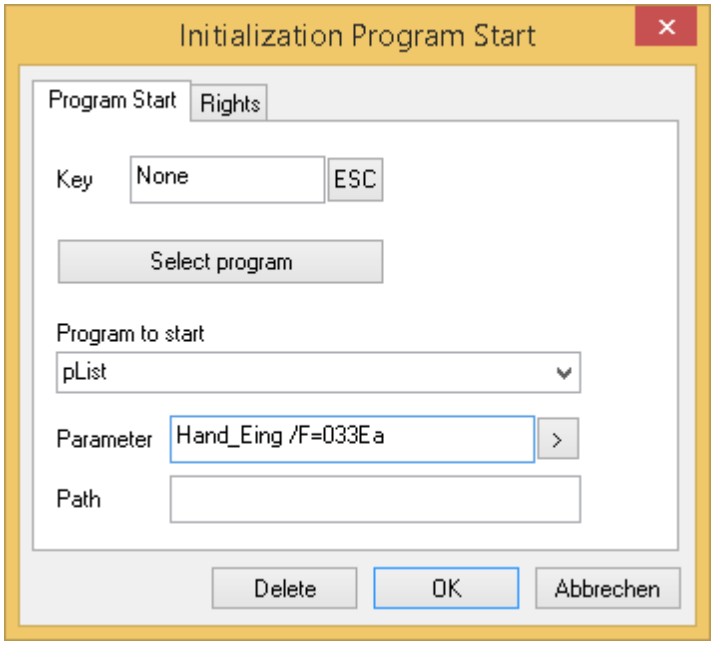
The following options are available as parameters:

- Load existing template - "/T="
  E.g.: "/T=Hand_Eing1"
  Multiple templates can also be specified.

- DMS addition
  E.g. "Hand_Eing"

- Site filter - "/F="
  "/F=033Ea"

Example of a configuration in the GE:



8.7.5.3.2 Settings

The display can be managed in the settings.

The settings apply to the observer's browser in each case (they are saved in a cookie).

- Table columns
  Here, you can select the table columns that are to be displayed. On the left hand side, you can also manage the order (, moves the entry up/down).

- Manage templates
  These manipulations are carried out on the server and therefore influence each instance of access from any PC or browser.

- Various settings

  - *Automatically update data*
  If this option is enabled, the values will be updated every 2 seconds.
  Please note: Any sorting and/or filtering of values will not be taken into account!

  - *Ignore SYSTEM*
  Entries in the DMS that begin with "System:" will not be taken into account.

8.7.5.3.3 Filtering

Via the Filter dialogue, you can specify various arguments for searching:

The filter set is saved in the corresponding template and applies equally to all instances of

browser access.

The wildcards "*" and "?" can be used for text filters ("DMS Name" and "Text"). "*" signifies any characters and character length and "?" any character in this position.
The DMS name can also be selected via a selection dialogue.
See also the examples for the Alarm Viewer.

8.7.5.3.4 Export

All currently filtered data sets are exported in CSV format.
If lines were selected, these can be exported using the button at the base of the table.

The following template applies for the file name that is to be saved: "{selected tab}_{current date}_{current time}.csv", e.g. "Hand_Eing_20131108_144927.csv".

The export for all columns always takes place in the following format:

Title in the first line:

```
pc;dms_name;name;object;add_column;value;value_type;
```

Example:

```
pc;dms_name;name;object;add_column;value;value_type;
"Z6003";"033Ea:033E:LU12:AK:01:Hand_Eing";"AUL-Klappe 1";"VEN02";"";"F.Null";"STR";
"Z6003";"033Ea:033E:LU12:AK:02:Hand_Eing";"FOL-Klappe 1";"VEN02";"";"F.Null";"STR";
"Z6003";"033Ea:033E:LU12:AK:03:Hand_Eing";"ABL-Klappe Raum D100
+102";"VEN02";"";"F.Null";"STR";
"Z6003";"033Ea:033E:LU12:AK:04:Hand_Eing";"ABL-Klappe Raum
D114";"VEN02";"";"adfsdF.Null";"STR";
"Z6003";"033Ea:033E:LU12:AK:05:Hand_Eing";"ABL-Klappe Raum
D116";"VEN02";"";"666";"STR";
```

When the visible columns are exported, the columns configured for viewing are exported accordingly.

Example:

```
"PC";"DMS Name";"Bezeichnung";"BMO";"Typ";"Wert";
"Z6003";"033Ea:033E:LU12:AK:01:Hand_Eing";"AUL-Klappe 1";"VEN02";"STR";"F.Null";
"Z6003";"033Ea:033E:LU12:AK:02:Hand_Eing";"FOL-Klappe 1";"VEN02";"STR";"F.Null";
"Z6003";"033Ea:033E:LU12:AK:03:Hand_Eing";"ABL-Klappe Raum D100
+102";"VEN02";"STR";"F.Null";
"Z6003";"033Ea:033E:LU12:AK:04:Hand_Eing";"ABL-Klappe Raum
D114";"VEN02";"STR";"adfsdF.Null";
"Z6003";"033Ea:033E:LU12:AK:05:Hand_Eing";"ABL-Klappe Raum D116";"VEN02";"STR";"666";
```

8.7.5.4 Integrate PDF in the pWA.

Integrate PDF in the pWA.

Pdf files can be saved and called up via a browser. However, the browser must allow the reading of pdf files (this can often be installed as an add-on):

1st variant:

Action: Image change

Initialising image change (example IP): "http://10.6.40.5/doc/test.pdf"

- A DMS data point can also be given (in which "http://10.6.40.5/doc/test.pdf" is then saved)
- Reinitialisation can also be carried out

2nd variant:

E.g. "PROJECT FOLDER\doc\test.pdf" can be configured in the GE as follows:

Start program

Program to be started: C:\Program Files\Internet Explorer\iexplore.exe
Parameter (incl. "): "http://10.6.40.5/doc/test.pdf"

→ The display in the GE only works if pWebAccess is running!

3rd variant:

The following change is necessary:

In "C:\Visi.Plus\SaaS\config\applications.cfg", add a line at the very bottom:

AcroRd32;popup;/doc/%1

This modification is needed for pWA up to and including 1.7.2.20. It will be included in the following setup.

In the GE, a "Start program" action (e.g. on a button) can be set up:

- Program to be started:
C:\Program Files (x86)\Adobe\Reader 10.0\Reader\AcroRd32.exe

("AcroRd32" is sufficient if it is only to work in the pWA, the entire path is needed for GE)

Parameter:
test.pdf

- Path:
C:\Visi.Plus\proj\{PROJECT}\doc\

The corresponding documents (in this case: "test.pdf") must now be saved in "C:\Visi.Plus\proj\{PROJECT}\doc\".

8.7.6 Fault analysis

Various log recordings are available for fault analysis:

Web server:
\Visi.Plus\proj\{project name}\log\pWebAccess.log

GE2XML:
\Visi.Plus\proj\{project name}\log\GE2XML.log (for complete conversions, last conversion recorded in each case)

\Visi.Plus\proj\{project name}\log\{screen name}.log (for conversions of individual images)

Common problems
Data points are not found by GE2XML.
Depending on the element type, part of the script may be created (e.g. a text is displayed as a string by default - this is acceptable in the majority of cases).

Analysis
Executing GE2XML creates a "GE2XML-log" log file in the {project}/logs/ directory. A new version is written each time that GE2XML starts.
Because this file usually contains numerous warnings and errors, it is difficult to carry out an analysis using the entries.
When individual images are changed in the GE, they are monitored by the pWebAccess as soon as they are saved and converted individually using GE2XML. In the process, an individual log file (e.g. "GE2XML-ZEA46_01.log") is created (log is newly created or attached).

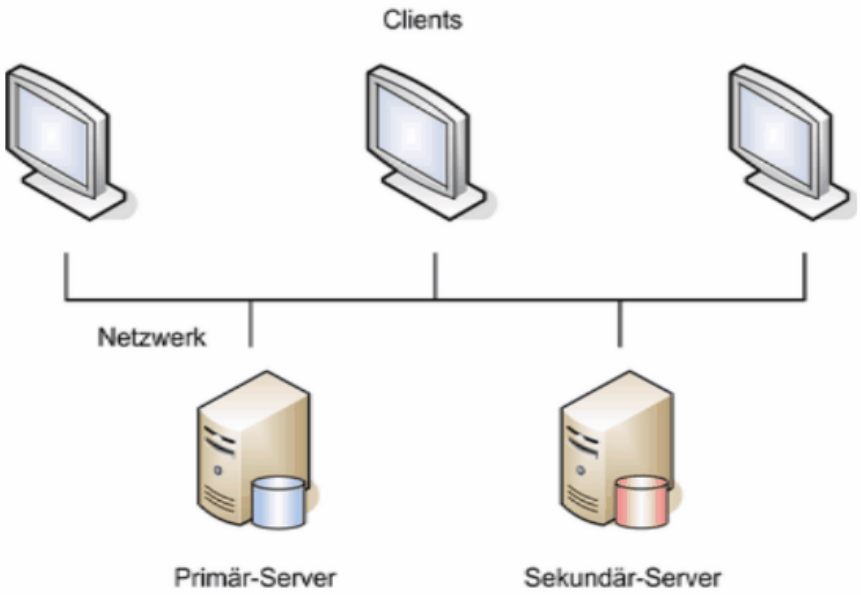
Procedure:
- Delete all "GE2XML-*" in the log directory so that only the up-to-date entries are visible.
- Save the individual "problem" image in the GE (non-initialised image for template images!).
- After a few seconds, the corresponding log will be created ("GE2XML-IMAGE NAME.log").
- In this log, the problems relating to this image can now be examined in detail.

- Carry out corresponding modifications until
===== GE Screen converter finished with 0 errors and 0 warnings. =====

8.7.7 Appendix

8.7.7.1 Hot Standby

Using the following description, a so-called "hot standby" system can be provided on the client system.

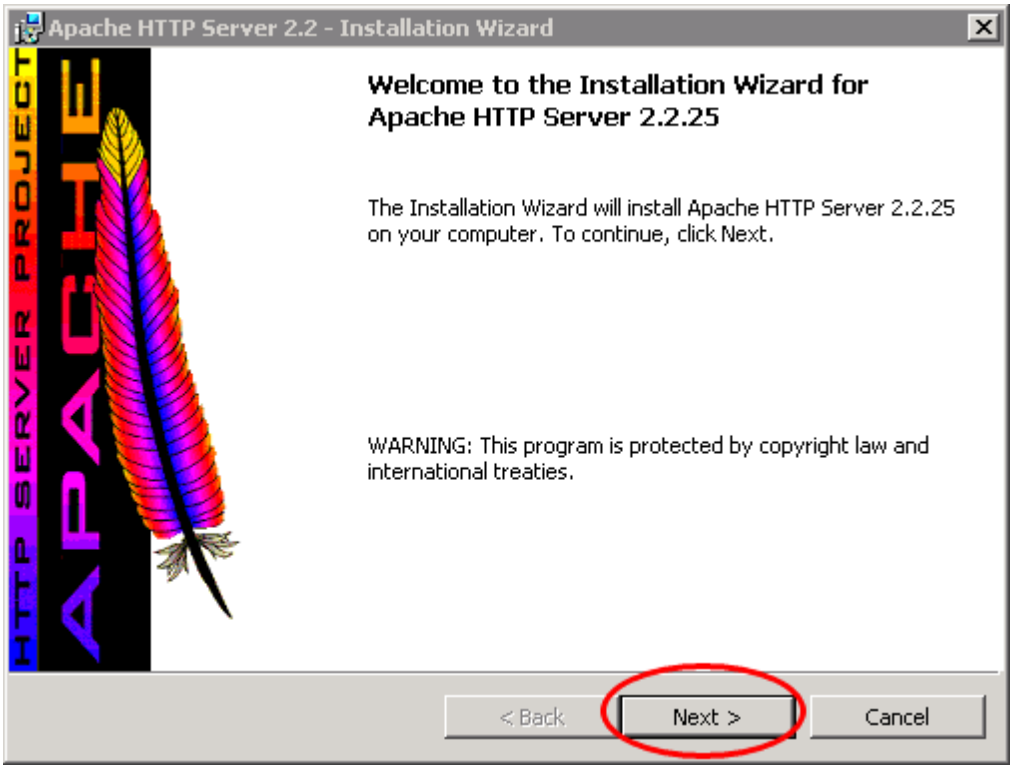


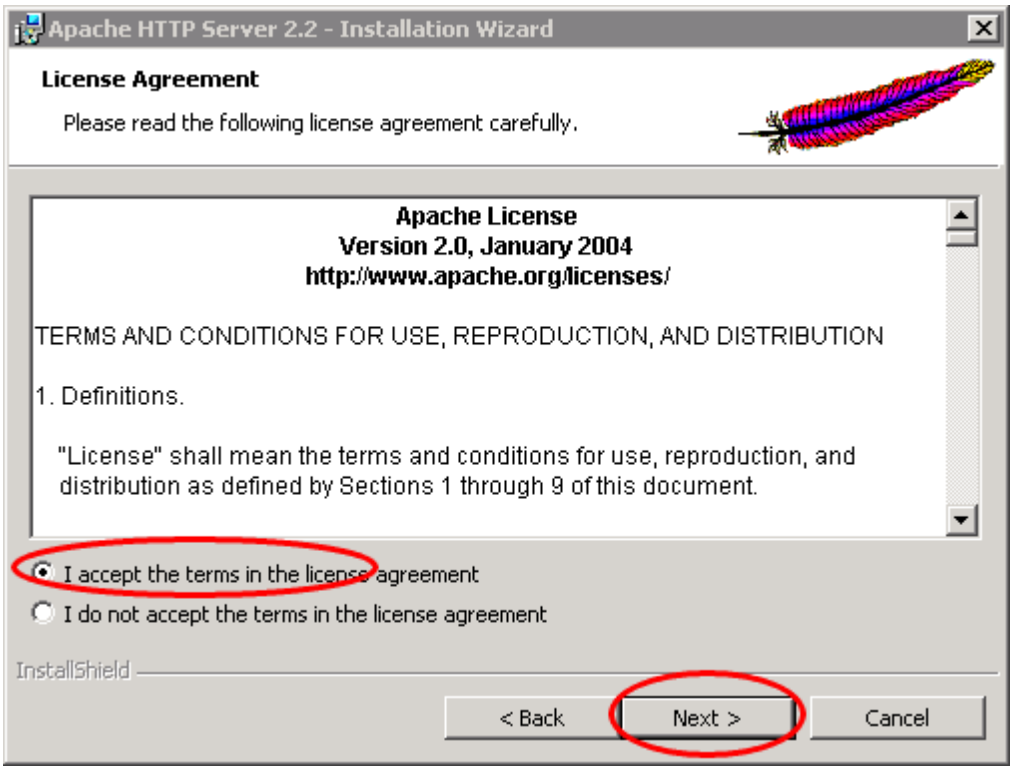
In the process, the client generally communicates with the primary server. If this cannot be contacted for 60 seconds, a switchover to the secondary server will take place. Every two minutes, an attempt will be made to re-establish the connection to the primary server. If this is successful, a switchover back to the primary sever will take place.

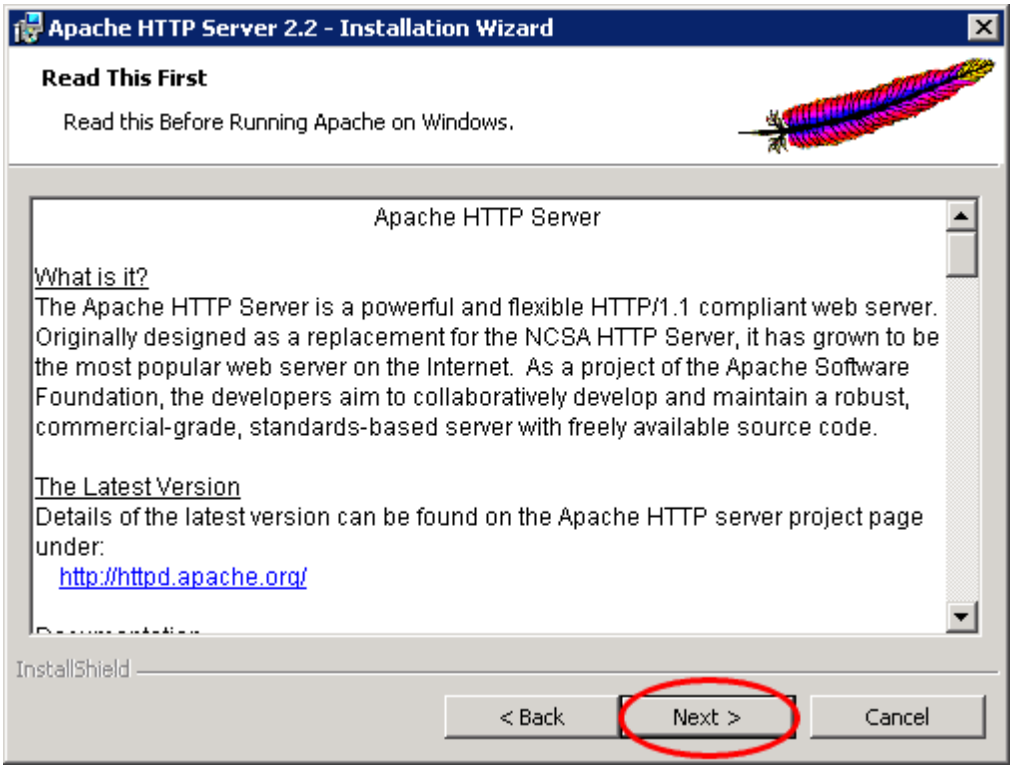
Requirement:
- WebAccess from Version 1.7.1.24

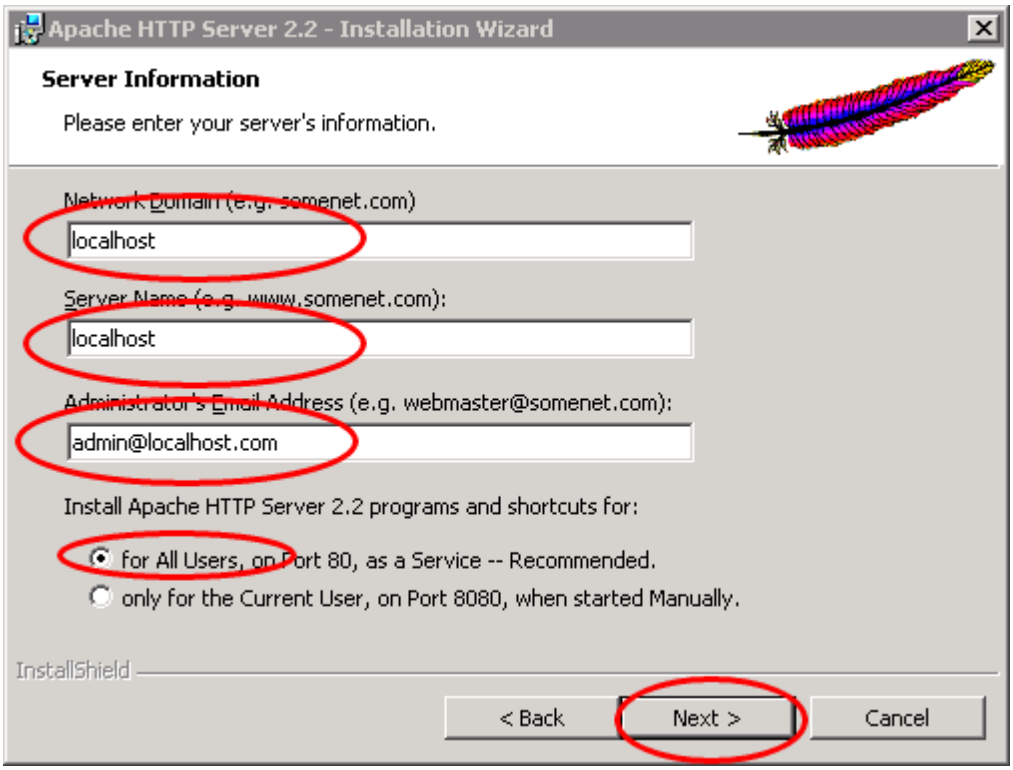
Installation:

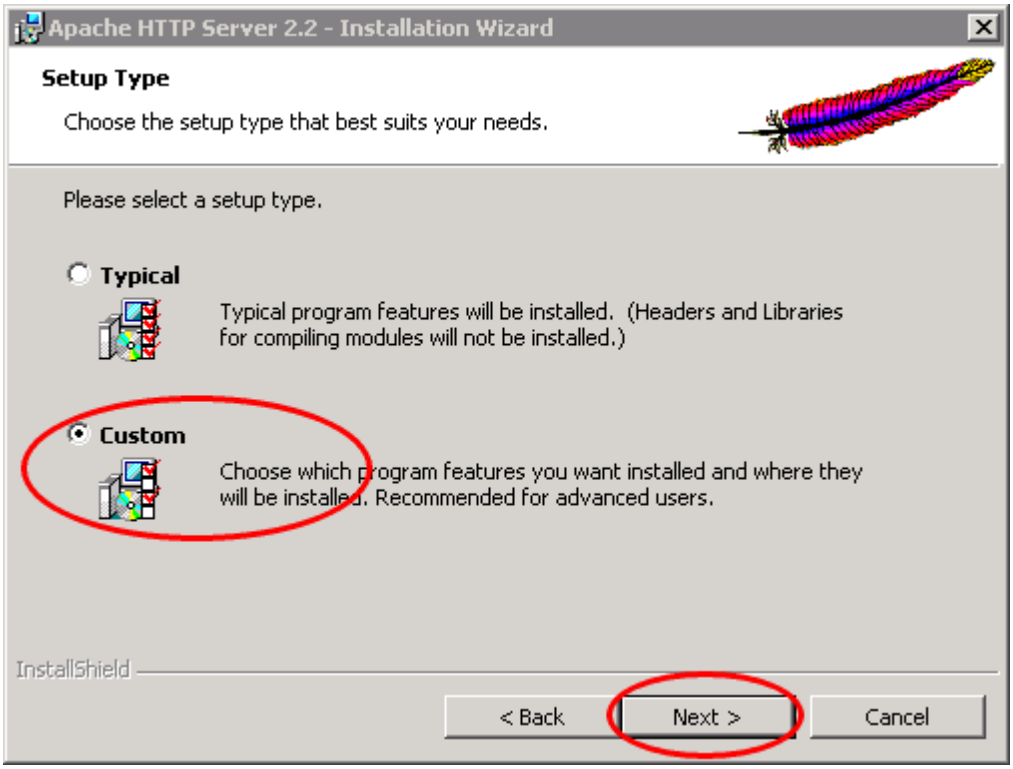
Using Apache2 installation, e.g. with "httpd-2.2.25-win32-x86-no_ssl.msi" from "http://httpd.apache.org/download.cgi" and carry out the following steps:

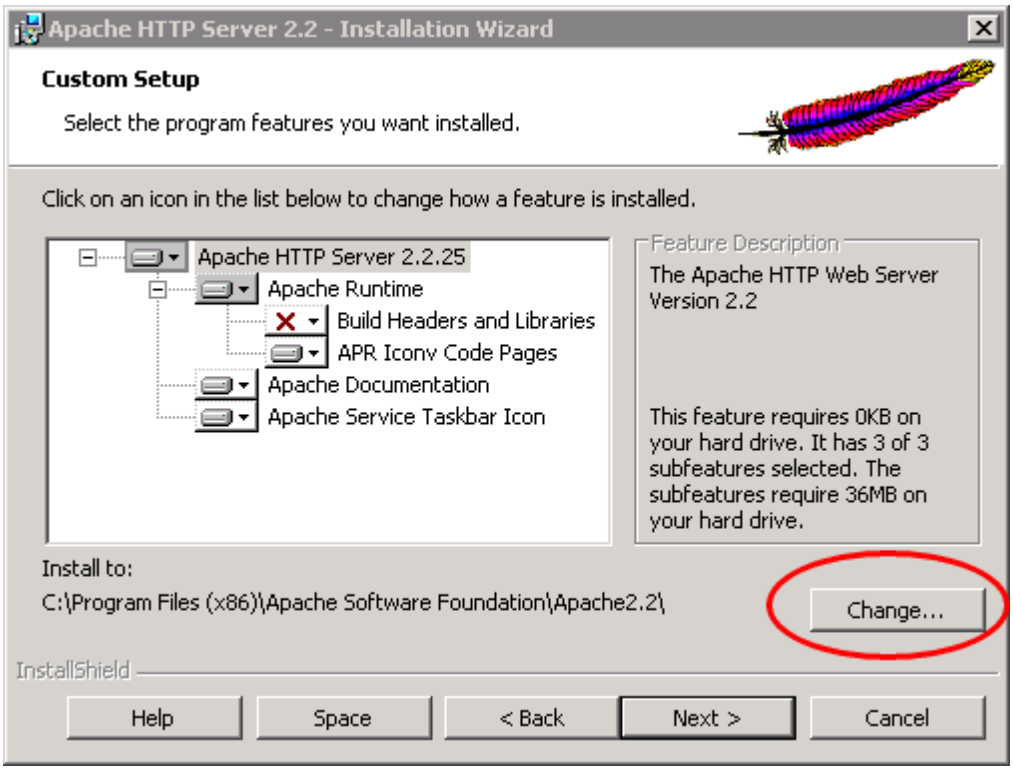


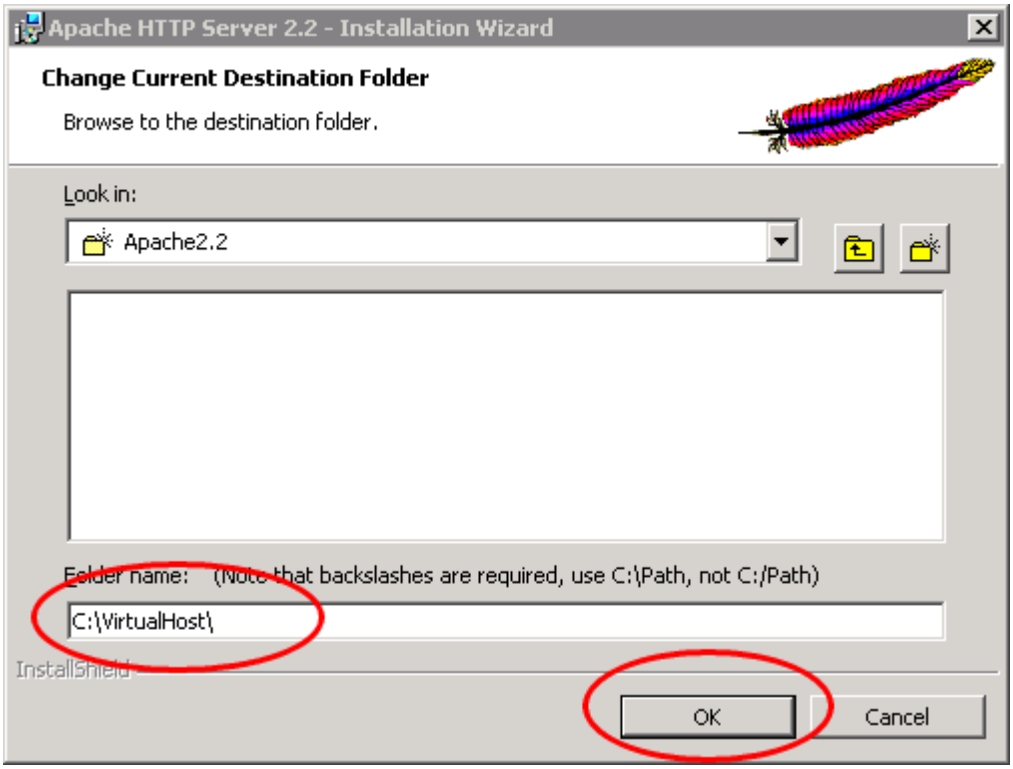




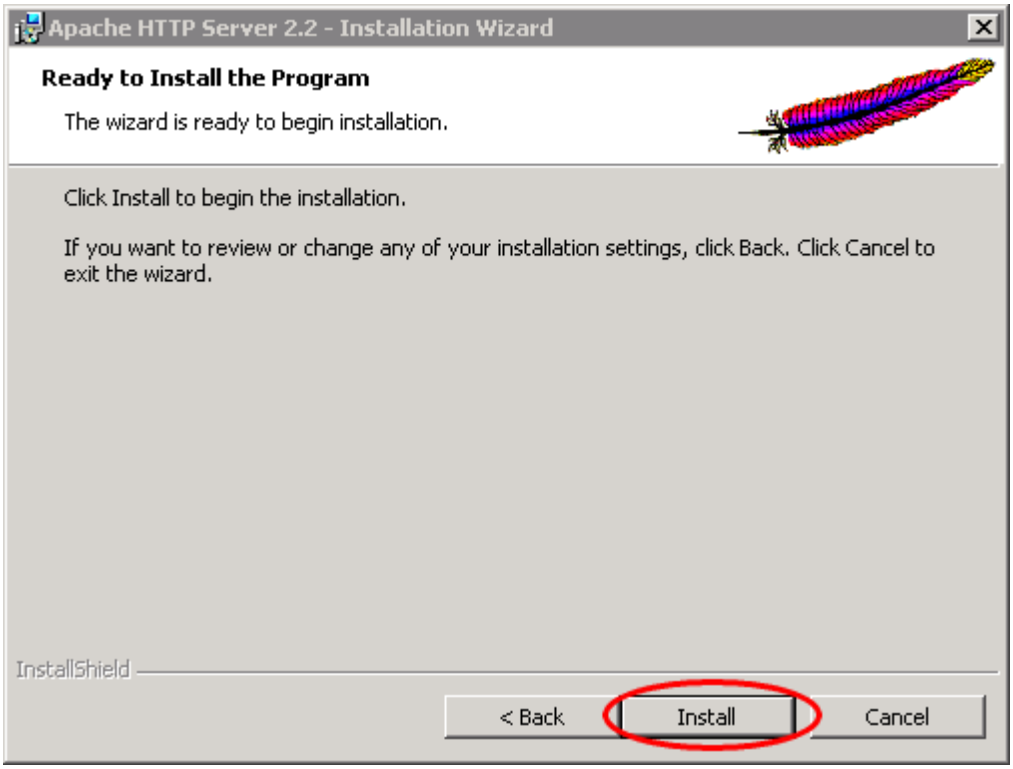


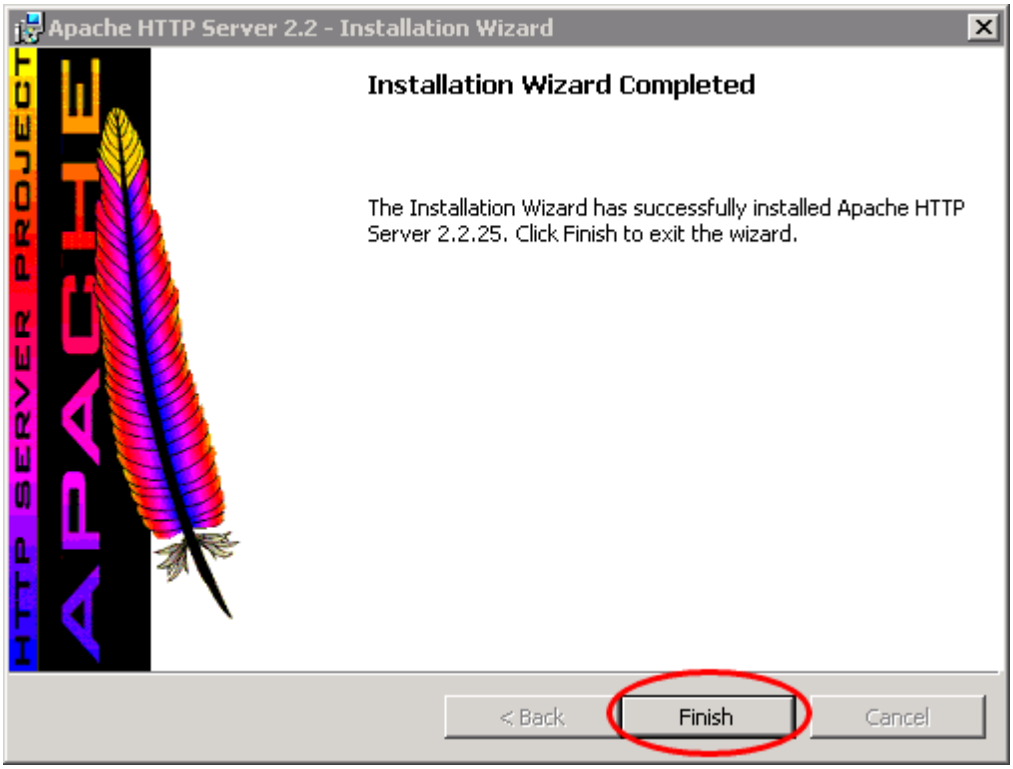






Then "Next >" on the previous screen.





After the installation, the tray icon should appear:

In "C:\VirtualHost\conf" the file "httpd.conf" must now be modified with the following content:

```
#
# This is the main Apache HTTP server configuration file.
#

ServerRoot "C:/VirtualHost"
Listen 80

LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_balancer_module modules/mod_proxy_balancer.so
LoadModule proxy_http_module modules/mod_proxy_http.so

ErrorLog "logs/error.log"
LogLevel warn

#
# Virtual Hosts
#

<VirtualHost *:80>
       ProxyRequests Off
       <Proxy *>
              AddDefaultCharset off
       </Proxy>
       ProxyPreserveHost On

       <Proxy balancer://mycluster>
              # Primary server
              BalancerMember http://192.168.10.175:80 connectiontimeout=60 retry=120
              # Secondary server
              BalancerMember http://10.0.0.165:80 connectiontimeout=60 retry=120 status
```

```
            ProxySet lbmethod=byrequests
        </Proxy>

        ProxyPass / balancer://mycluster/
        ProxyPassReverse / balancer://mycluster/
</VirtualHost>
```

Under "# Primary server" the IP of the main server and under "# Secondary server(s)" the IP of the secondary server should be entered.

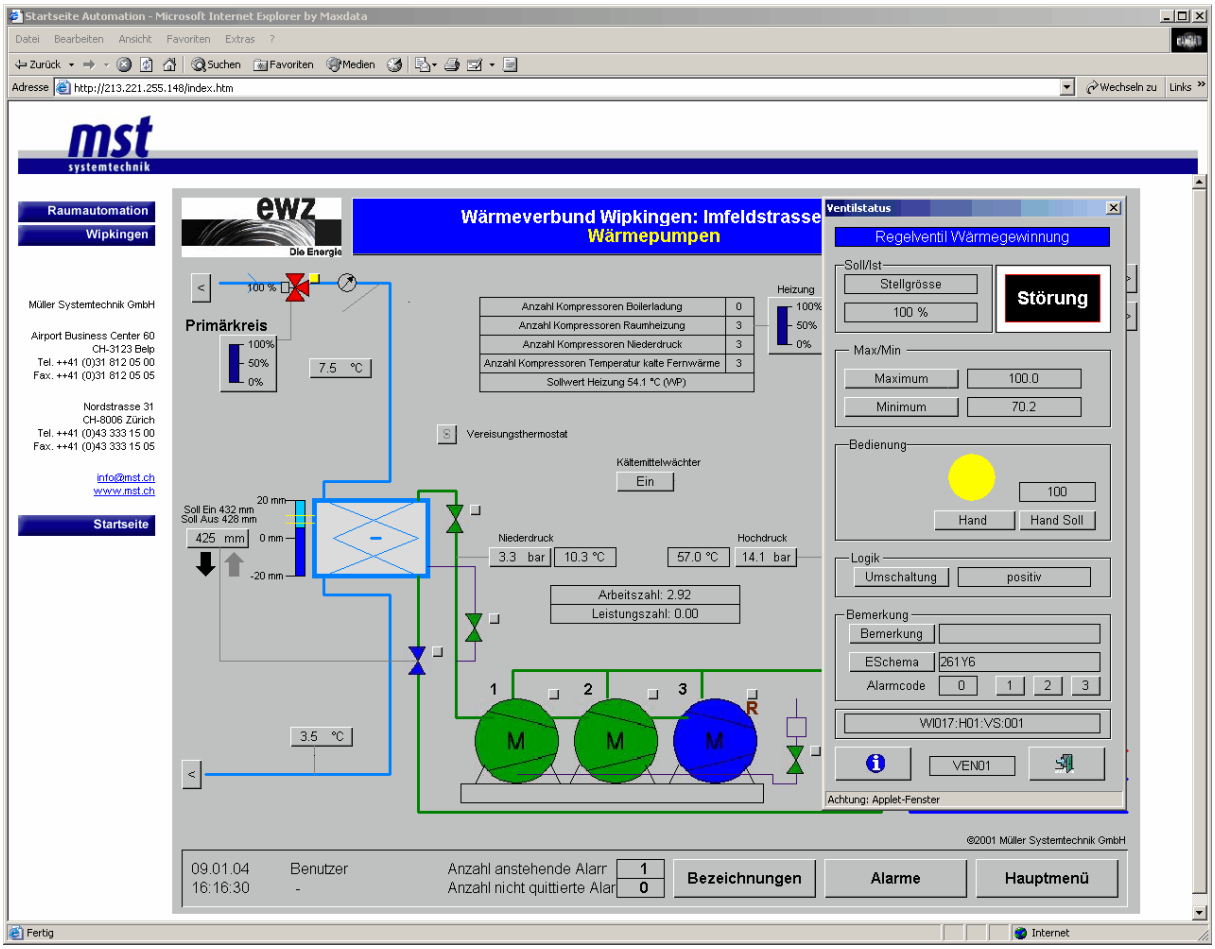
Finally, the Apache server should be restarted via the tray icon  .

8.7.7.2 Print Logo

For AlarmViewer, PrtViewer and DataList it is possible to add a Logo on the right top side of the document. For this in the folder <installation-path>/Visi.Plus/\SaaS\www\images you need to add your Logo as png with the Name "Print_Logo.png".
The Logo will be printed out now with the specific command in your Browser.

8.8 The WebServer (pWEB.exe)

The visualisation images are created with the help of the Graphical Editor. When these images are saved, an image file is automatically created in the WEB directory. This file can be interpreted directly by the WebServer (pWEB) and displayed in the browser.



The WebServer logs all data points with the DMS and manages all changes in value as well as all user entries.
Depending on the network and image complexity, the image change times range from one second (100 Mbit network) to 15 seconds (ISDN) for the initial image build. Subsequent image changes are significantly faster, because the graphics have already been stored in the browser's memory.
Multiple users can access the WebServer simultaneously.

8.8.1 Minimum requirements

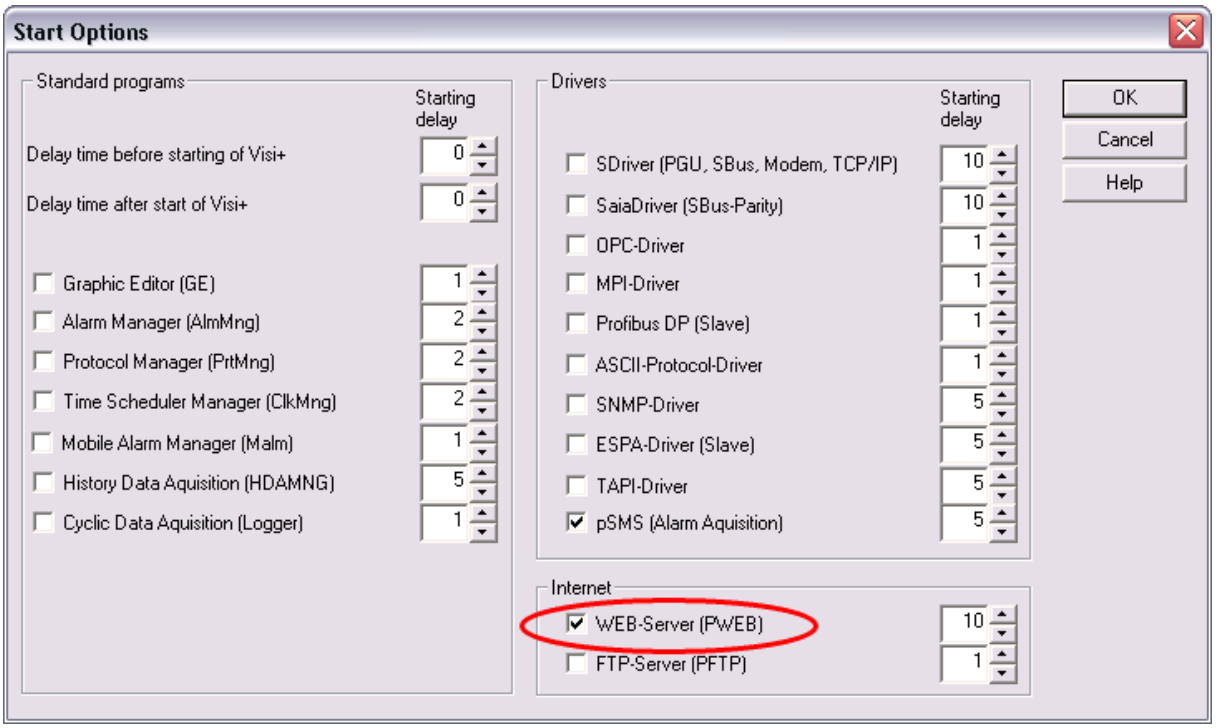
- Internet Explorer 6.0 or higher
- Java applets allowed in Internet Explorer
- Sun Java allowed in Internet Explorer
  Exception: if the Sun Java Plugin configuration is set, Sun Java does not have to be explicitly selected in IE (the plugin does this itself)

The following JAVA VMs are supported:
- Microsoft VM *(will no longer be supported in the future)*
- SUN VM (Version 6 or higher, JRE 1.6)

8.8.2 Starting the program WebServer (pWEB.exe)

In order to be able to start the **pWEB** module during project startup, the appropriate checkbox must be marked in the **Start Options**.

To do this, start the **Project Manager** module and click on the **<Start Options>** button.



Activate **WebServer (pWEB)** in the open window.

Confirm with **<OK>**

WebServer is now automatically started up the next time a project is opened.

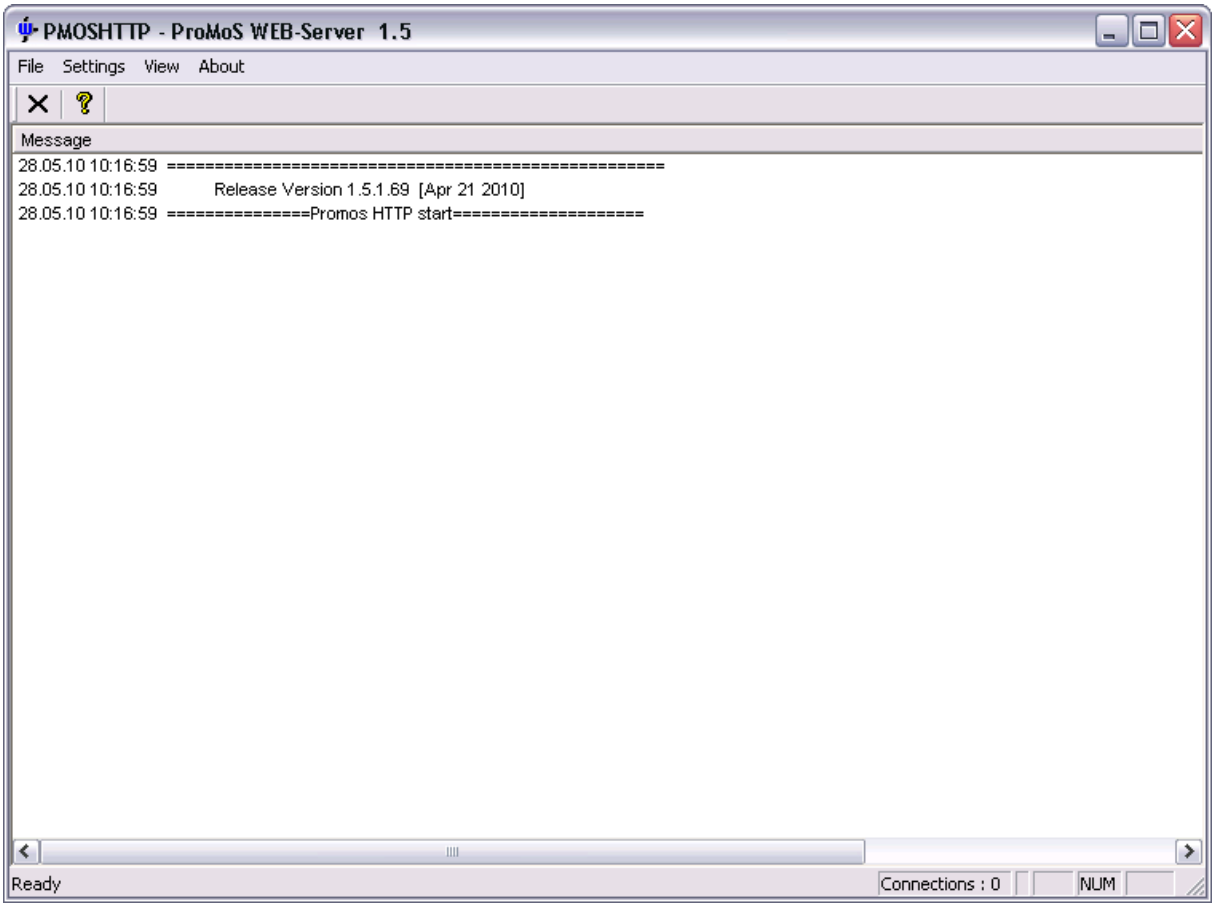
The module normally runs as a background process and is displayed as a large W on the Windows taskbar.

If the **WebServer** is not started up, no WEB access can take place.

8.8.3 The WebServer main window

Open main window.

Click on the **WEB** icon with the right mouse button and choose the "**Display**" command.

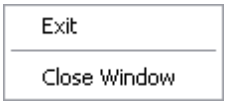


All accesses are logged! This means that you can trace which computer logged into the system and when this took place at any point in time.

8.8.4 The WEBServer menus

8.8.4.1 File menu

Functions available for exiting the module are provided in the "**File**" menu.

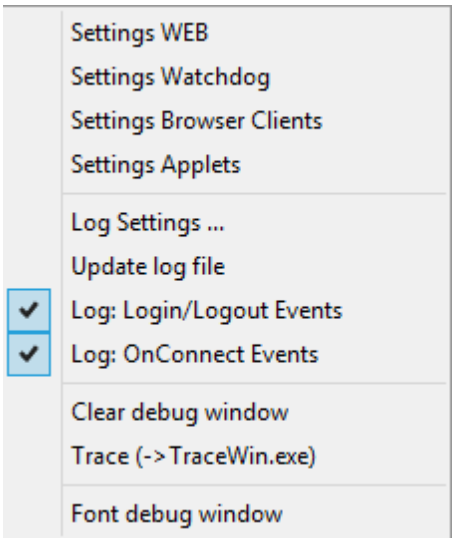


Exit
Exits the **WEB** Server.

Close Window
This menu item closes the control window of pWEB. This does not exit the pWEB program; it is only minimised to the taskbar.

8.8.4.2 Settings menu

The following functions are available in the "**Settings**" menu:
- Web Server Settings
- Watchdog Settings
- Various Log and Debug Settings



Log Login/Logout Events
With this menu item activated, login/logout processes in pWEB are logged.

Log OnConnect Events
With this menu item activated, connection processes in pWEB are logged.

Delete Debug Window
Deletes the debug messages in the WebServer.

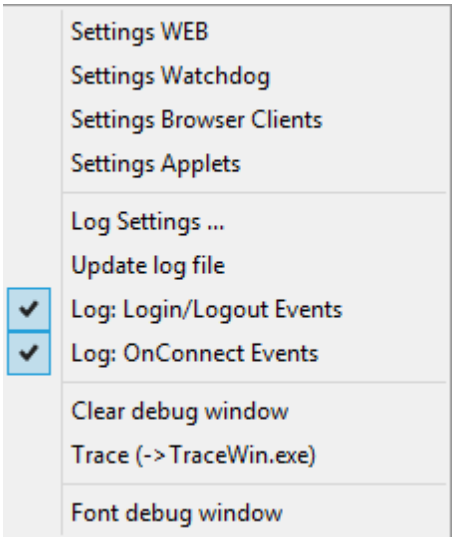
Trace (->TraceWin.exe)
With this menu item activated, additional network processes are output in the debug window (error analysis with separate program).

Log Debug Messages...
With this menu item activated, the debug messages are logged to a file.

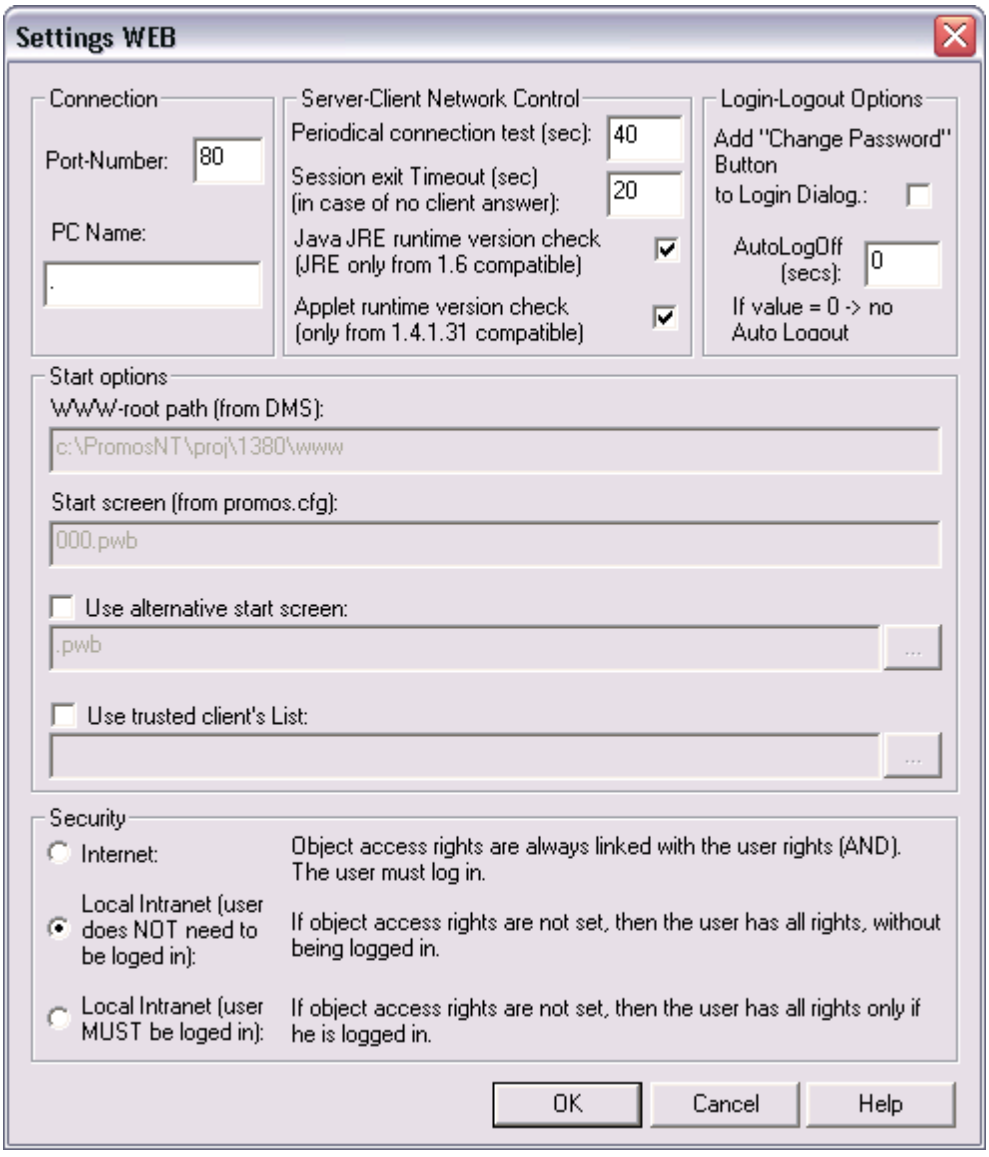
Debug Window Font
The font in which the debug messages are displayed can be selected.

8.8.4.2.1 WEB Settings

WEB Server settings functions are available in the "**Settings**" menu.



WEB settings



Various settings for the WebServer can be configured here.

Connection:
Port Number
Port through which communications should be made (default value: 80).

PC Name:
Entry of a PC name (on which the DMS is running) on any network. If the WebServer has been started locally, a "**.**" (dot) is entered as the name.

Server/Client Network Check:

Periodic Connection Test (sec.):
Entry of a length of time after which the WebServer tests whether a connection is still present.

Session Timeout (sec.):
If there is no answer from the client during the indicated time, the WebServer terminates the connection.

Login Dialog Options

The checkbox **Additional "Change password" Button in Login Dialog** adds an additional button to the login window, which enables the user password to be changed.

AutoLogoff (sec.): Entry of a time after which the WebServer will log off the current session if no interaction (mouse or key action) has taken place. If the value = 0, then the auto-logoff function is deactivated. This setting applies for all users.

Start Options:
In this section, different startup settings can be viewed or executed.

WWW Root Path (from DMS) (read-only):
Specify where the **\www** project folder is located. All images and libraries required by the WebServer are located in the www folder.

Start Screen (from promos.cfg) (read-only):
If nothing else is defined under Use Alternative Start Screen, the WebServer adopts the start screen of the project.

Use Alternative Start Screen:
If the project start screen should not be opened, an alternative start screen can be selected here (click on the button to the right of the input field).

Use Trusted Clients List:
Allows you to enter authorised or trustworthy clients (PCs) for the selected project.

An example of a configuration file, "WebClients.cfg", for trustworthy clients is located in the folder "C:\Visi.Plus\cfg\web\cfg" of the default Visi.Plus installation directory:

```
; Comment: the following addresses of clients for WebServer connections
; are valued as "trustworthy". All other clients are redirected to the page,
; specified under the keyword "default".

default= menu.htm    ; must be located in the project folder \www.
```

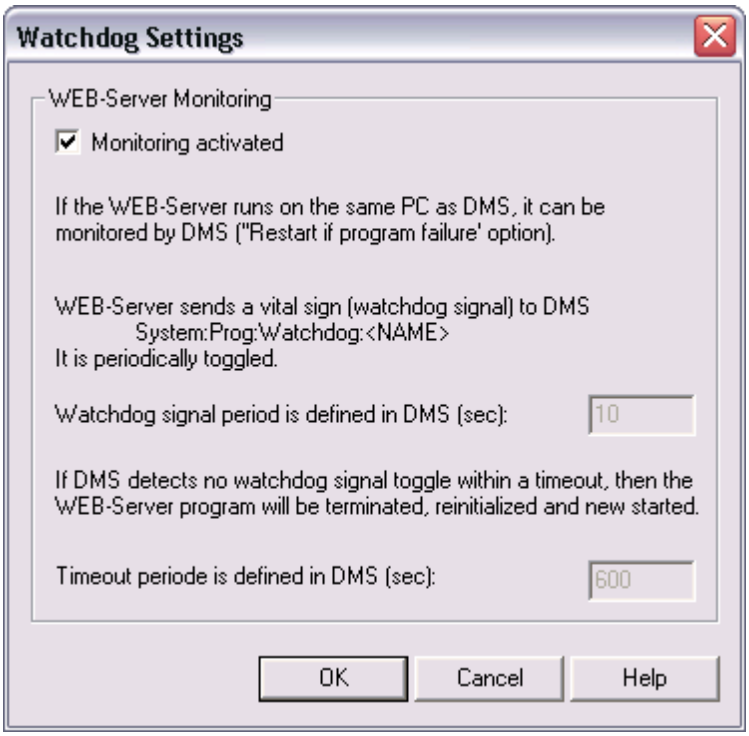
```
; wildcards '*' can be used in the form: '10.0.0.*' or '192.34.*.*' etc.

; 152.34.40.145      ; commented out....
;10.0.0.29           ; my client

; next group
; 172.54.38.*        ; all...
; 172.54.44.100      ; my client
```

Security:
The radio button can be set to **Internet** or **Local Intranet**. The difference is the following: a user must always log in with access from the **Internet**. With access from the **Local Intranet** this is not mandatory, depending on user definition.

8.8.4.2.2 Watchdog Settings

Watchdog Settings

The **WebServer** can be monitored by the Data Management System (DMS). **WebServer** and the DMS exchange a live bit upon activation of the function.

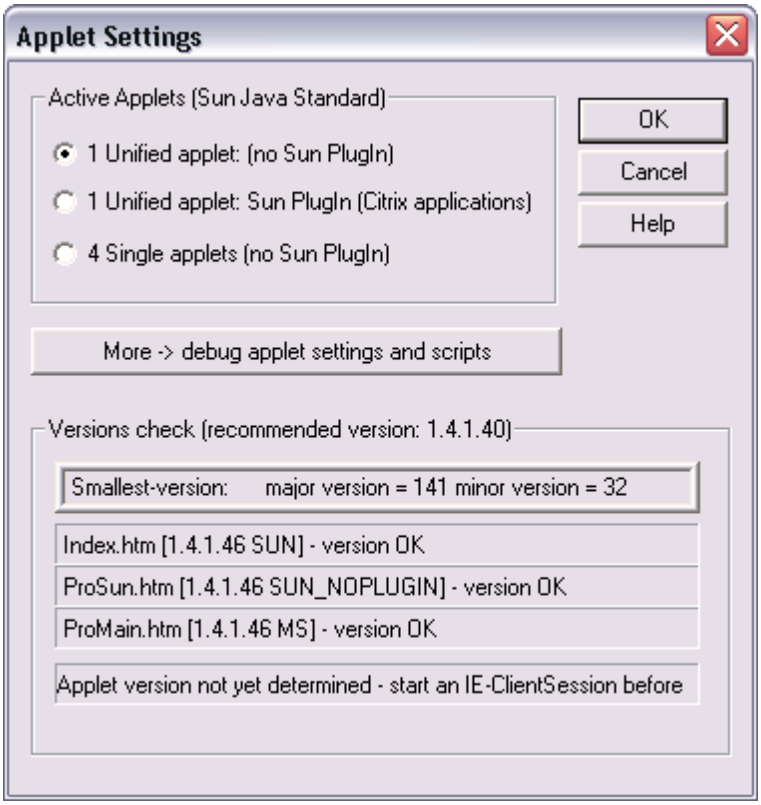


8.8.4.2.3 Applet settings

For the web browser on the client PC to be able to communicate properly with the WebServer, Java Runtime Environment (JRE) from Sun Microsystems must be installed on the client PC. It is recommended that you use the most recent version.

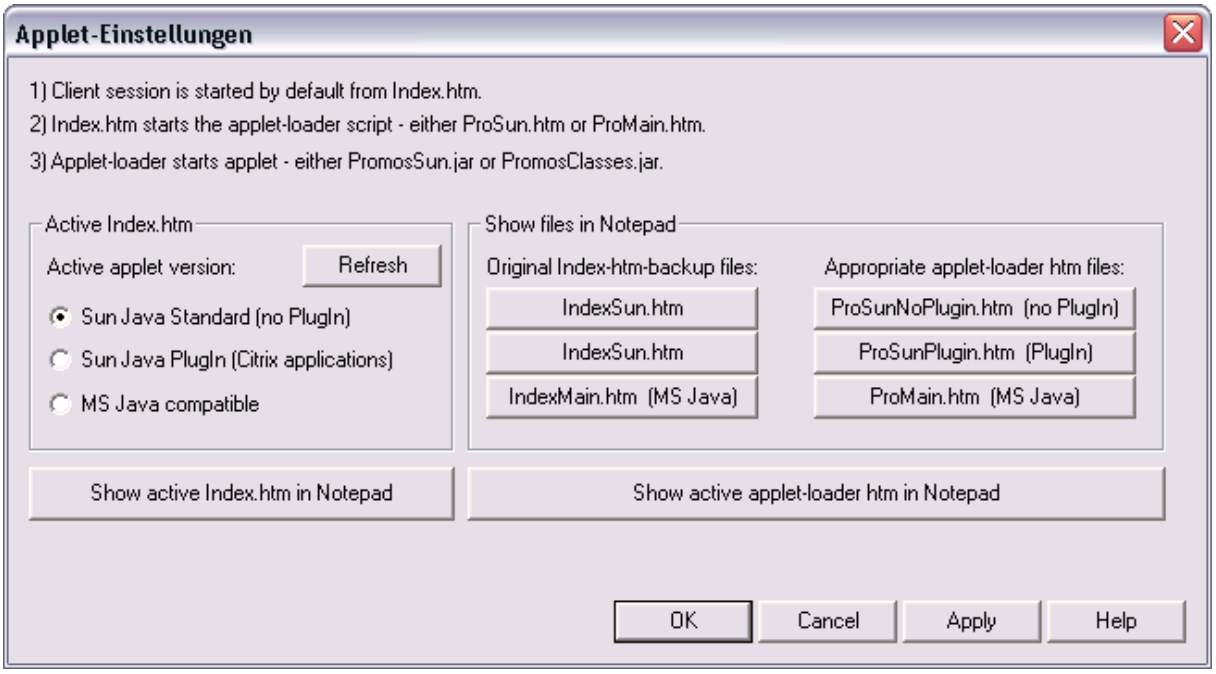
The WebServer will only work with JRE version 6 or higher.

It is possible configure the following applet settings:



- Sun Java Standard: this is the default setting for all browsers.
- Sun Java Plugin: for all clients equipped with a Citrix system (terminal server). For the WebServer to be able to communicate with its applet requires a browser plugin from Sun Microsystems. This configures the browser automatically for Sun Java.
- MS Java Compatible: this applet is only supported owing to is backwards compatibility with Microsoft Java Virtual Machine, however, its development has been discontinued.

More -> Debug Applet Settings and Scripts
This button opens a new dialog window containing in-depth bug analysis options.



1. It offers the same applet selection options as in the Applet Settings dialog.
2. An applet can be selected and the resulting htm scripts checked without having to leave the dialog.

Method of operation of the applet loading process

1. Starting the client session: by entering the WebServer address in the browser on the client PC, the server requests the script file **Index.htm** by default.
2. Index.htm defines a frame for the applet loader and starts the applet loader script: depending on the configuration, either one of the Sun versions (ProSun.htm) or the MS Java version (ProMain.htm).
3. The applet loader starts either the MS applet (PromosClasses.jar), or the Sun applet (PromosSun.jar). If the applet loader is the ProSun.htm plugin version, the Sun plugin is loaded before the applet.

All configurations have renamed backup copies of Index.htm and the corresponding applet loader. When a configuration is activated, the corresponding files are copied and renamed:

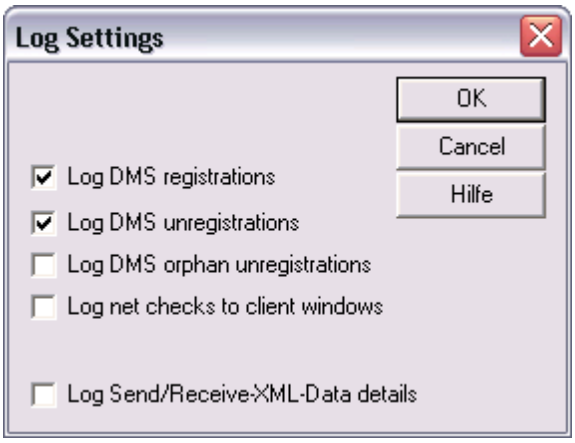
- Sun Java Standard
  IndexSun.htm ➔ Index.htm and ProSunNoPlugin.htm ➔ ProSun.htm
- Sun Java Plugin:
  IndexSun.htm ➔ Index.htm and ProSunPlugin.htm ➔ ProSun.htm
- MS Java Compatible:
  IndexMain.htm ➔ Index.htm

8.8.4.2.4 Log and Debug Settings

Appropriate logs are written to the file **pweb.log** in the project folder **\log**.

Log Settings ...

For debug/analysis purposes and/or troubleshooting, the following log options can be activated in the dialog window:



Log: Login/Logout Events
With this menu item activated, login/logout processes in pWEB are logged.

Log: OnConnect Events
With this menu item activated, connection processes in pWEB are logged.

Delete Debug Window
Deletes the debug messages in the WebServer.

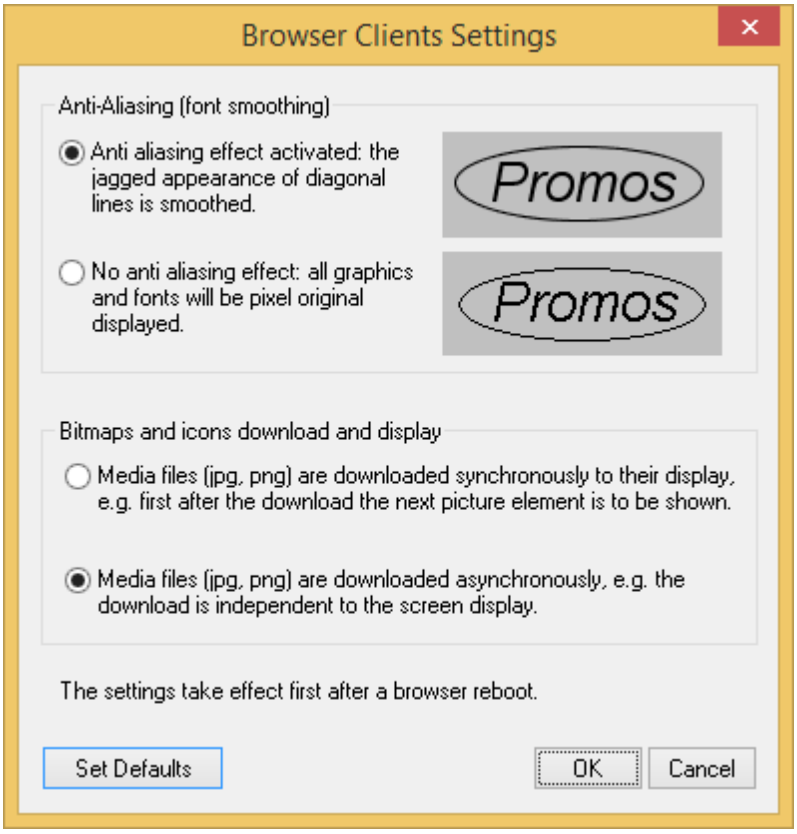
Trace (->Trace.exe)
With this menu item activated, additional network processes are output in the debug window (error analysis with separate **Trace.exe** program).

Log Debug Messages...
With this menu item activated, the debug messages are logged to a file.

Debug Window Font
The font in which the debug messages are displayed can be selected.

8.8.4.2.5 Browser Client Settings

Settings for Aliasing, Image downloads and displayed for Browser are defined in this menu.



By default Anti aliasing and Media files downloaded synchronously is set.

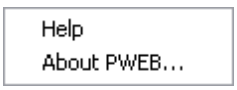
8.8.4.3 View menu

The "**View**" menu allows you to change the appearance of the WEB Server.

Toolbar
Switches the toolbar on or off.

Status bar
Switches the status bar on or off.

8.8.4.4 Help menu



About pWEB...
Displays the current version of the **WEB Server**. The version must always be given in case of any support queries!

Should support queries become necessary, the internet forum is preferential to contact via telephone or e-mail. This offers two advantages: First, help is available more

quickly; and secondly, other users also benefit from the entries.

8.8.5 Required files

The files appearing in the following table are automatically managed by the WebServer. When a new project is generated they are automatically copied from the setup folder "C:\Visi.Plus\cfg\web":

Product	Filename	Saved Location
Client Promos Applet	PromosClasses.jar	C:\Visi.Plus\proj\<ProjName>\www\lib
Client Alarm Viewer Applet	PromosAlarm.jar	C:\Visi.Plus\proj\<ProjName>\www\lib
Client Log Viewer Applet	PromosProt.jar	C:\Visi.Plus\proj\<ProjName>\www\lib
Browser file: Frame definitions for applets	Index.htm	C:\Visi.Plus\proj\<ProjName>\www
Browser file with Promos applet	ProMain.htm	C:\Visi.Plus\proj\<ProjName>\www
Browser file with Alarm Viewer applet	ProAlarm.htm	C:\Visi.Plus\proj\<ProjName>\www
Browser file with Log Viewer applet	ProProt.htm	C:\Visi.Plus\proj\<ProjName>\www
Browser file (no content, serves as a placeholder for the Alarm applet, which is only loaded from the server as needed)	ProFrm1.htm ProFrm2.htm	C:\Visi.Plus\proj\<ProjName>\www
JPG files belonging to browser files	logo_rz.jpg PromosAlarm.jpg PromosProt.jpg	C:\Visi.Plus\proj\<ProjName>\www

Example files

Example of a configuration file "Trusted Client List"	WebClients.cfg	C:\Visi.Plus\proj\< ProjName >\cfg
WEB server configuration file	WebServer.cfg	C:\Visi.Plus\proj\< ProjName >\cfg

Tool program

Automatic conversion of BMP to JPG files	bmp2jpg.exe	C:\Visi.Plus\bin

Image files

JPG files	*.jpg	C:\Visi.Plus\proj\<ProjName>\www\img

Document files
If these files are displayed in GE images, they must also be saved to corresponding \www folders.
See chapter "Displaying documents in the GE/WebServer".

PDF files (Adobe)	*.pdf	C:\Visi.Plus\proj\<ProjName>\www\pdf
DOC files (Office)	*.doc	C:\Visi.Plus\proj\<ProjName>\www\doc
Text files (ASCII Editor, Notepad)	*.txt	C:\Visi.Plus\proj\<ProjName>\www\txt
PPS files (PowerPoint)	*.pps	C:\Visi.Plus\proj\<ProjName>\www\pps
XLS files (Excel)	*.xls	C:\Visi.Plus\proj\<ProjName>\www\xls
HTM/HTML files (web browser)	*.htm or *.html	C:\Visi.Plus\proj\<ProjName>\www\htm or C:\Visi.Plus\proj\<ProjName>\www \html

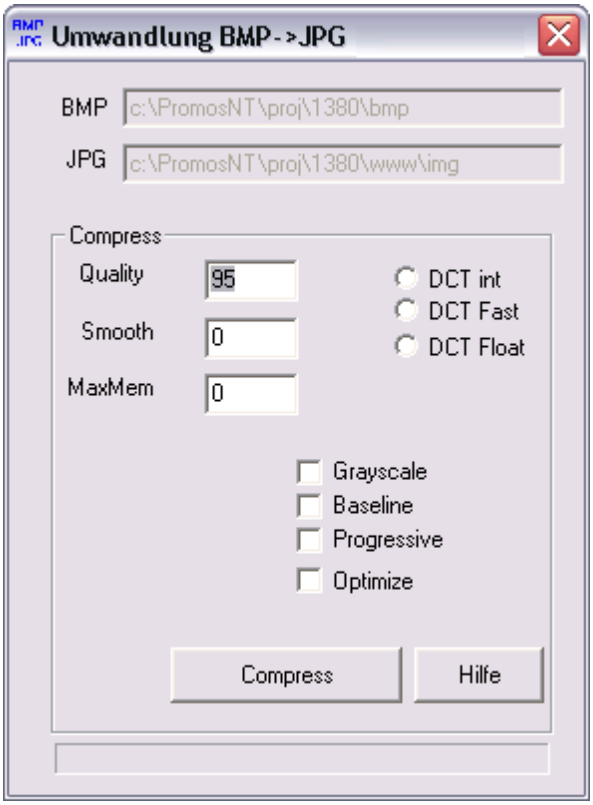
Images saved in the GE

PWB files: individual screen pages	*.pwb	C:\Visi.Plus\proj\<ProjName>\www

Filenames may only consist of characters contained in the US-ASCII character set (no umlauts!).

8.8.6 Creating JPG files

Start the program ***bmp2jpg.exe***: it converts all bmp images to the jpg format and copies them to the folder ..\www\img (this operation must only be performed once during the first installation or when reimporting bitmap files). The bitmaps must have been created with 256 bit colour depth (there are problems with the conversion of 16-bit bitmaps).



All BMP files from the BMP directory are converted to JPG files and copied to the directory www\img.

JPG files in the BMP directory are not automatically copied to the web directory. They must be copied manually.

Subfolder in the bmp-folder are supported in GE but not for pWeb and bmp2jpg.

8.8.7 Technical settings

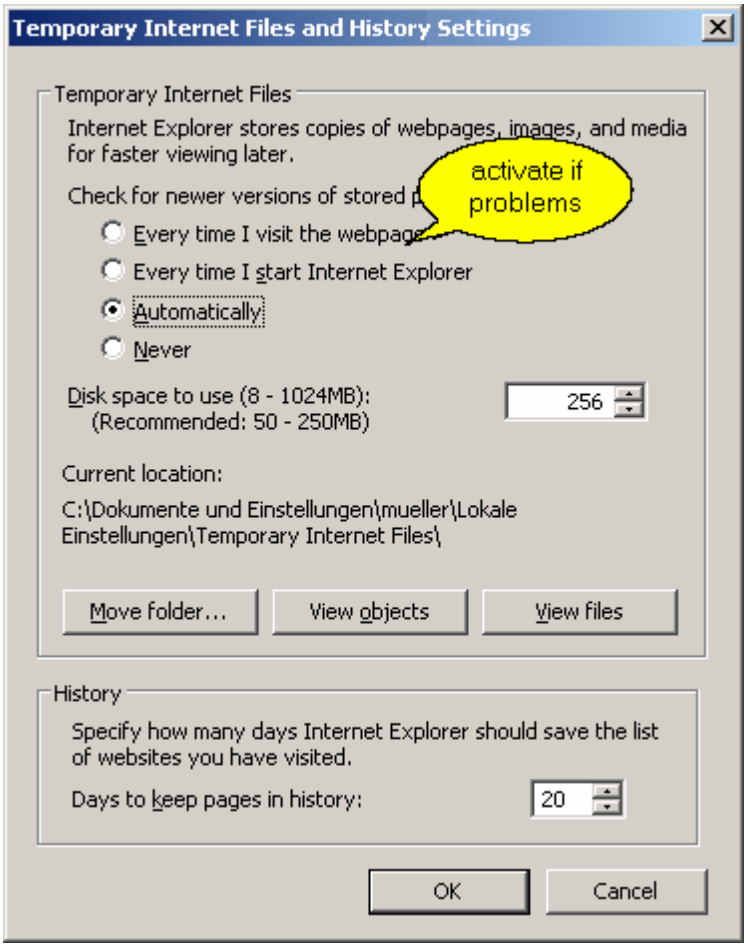
8.8.7.1 Browser settings

The WebServer has been tested with Microsoft's IE 6/7/8, as well as Firefox and Opera. JAVA Virtual Machine (VM) must be installed (see www.sun.de).

It is compatible with Sun as well as Microsoft Java VM, though Microsoft no longer supports the VM. We recommend Java VM from Sun.

The following settings must be configured and/or checked in IE:

Internet Options > Browsing history > Settings > Temporary Internet Files:
Check for newer versions of stored pages:
Select: "Every time I visit the webpage"



8.8.7.2 Proxy Settings

Proxy Settings are not supported with the pWeb.

8.8.7.3 Port Setting

In the WebServer, Settings menu, WEB Settings: in the Connection box, "Port Number".

The default setting is 80.

If a different port is defined, the port must be indicated in Internet Explorer.
Example: http://192.168.1.1:81 (Port 81 is defined here)

8.8.7.4 PC server name settings

WebServer, Settings menu, WEB Settings: in the Connection box, "PC Name". Attention: after any change to the PC name, the WebServer must be restarted.

Inputs:
- WebServer is running on the same PC as the DMS: "." or its own IP address
- Remote connection of the WebServer to DMS, e.g. "\\ldefix" or "10.0.0.33"

8.8.7.5 Configuration file

In the CFG directory a configuration file with the name WebServer.cfg can be defined. The window position of the Login window or the Alarm window, for example, can be defined in this file.

It is also possible to inform the browser to which address a connection should be established (Java applet). In some circumstances, this is necessary for portal solutions (SSL) in order to establish communications.
In the area [PortalServer] the public address of the portal server is entered. In addition, the address for the connection of the applet can also be entered (in the example a local address is entered, for which software adopts the data, encodes it and passes it along to the server).
This option may not be defined without a portal server, because no other communication is possible otherwise.

Example:

```
[LoginWinCoord]
X_topleft=300
Y_topleft=600

[PortalServer]
; Portal IP
SourceIP=172.16.1.10

; Client's Localhost IP
ClientIP=127.0.10.1
```

8.8.7.6 Settings for the trusted client list

WebServer, Settings menu, WEB Settings: in the Start Options box, "Use Trusted Clients List". Enter a CFG (text) file from the project folder \cfg. If this option is selected, a check is performed on client addresses from all incoming connections to see if they are included in this file. If not, this client is redirected to an HTML page, which is entered under "default=".

Example of a Trusted Client List:

```
; Comment: the following addresses of clients for WebServer connections
; are valued as "trustworthy". All other clients are redirected to the page entered,
; under the keyword "default".

default= menu.htm    ; must be located in the project folder \www.

; wildcards '*' can be used in the form: '10.0.0.*' or '192.34.*.*' etc.

; 152.34.40.145      ; commented out....
10.0.0.29            ; my client

; next group
172.54.38.*          ; all...
172.54.44.100        ; my client
```

8.8.7.7 Security

WebServer, Settings menu, WEB Settings: in the Security box, "Internet"/"Intranet".

- Internet: All objects are always linked with user rights (AND), meaning the user must log in in order to receive access rights to objects.
- Intranet: If object rights are not set (i.e. no approvals are selected), the user has all object access rights without having to log on.

8.8.7.8 Login

If the login function is activated using the keyboard (CTRL<L>), the Login window appears by default in the middle of the screen. These Login window coordinates can also be defined in the WebServer.cfg configuration file:

Example:

```
[LoginWinCoord]
X_topleft=300
Y_topleft=200
```

8.8.7.9 Alarm window

The alarm window coordinates can be defined in ProAlarm.htm:

```
<!-- Alarm view window -->
<param name=almwidth value="700">
<param name=almheight value="400">
<param name=almx value="300">
<param name=almy value="20">
```

8.8.8 Request of a specific website

The client can request a specific Web Server homepage by attaching the following parameter to the URL address:

"?pag=<filename>"

e.g.: "http://10.0.0.37/index.htm?pag=stockwerk3"

The client can also access an object directly by appending the <filename> parameter to both reinitialisations (as is also the case with a GE image change):

"?pag=<filename>|Smallest common name|Reinitialisation to>"

e.g.: "http://10.0.0.42/index.htm?pag=UpsSecondaryPage|BMO:UTA|Newawe:U3.5"

8.8.9 Scaling a web page

The Client can scale entire web server pages by attaching the following parameters to the URL address:

"?sca=<factor>"

e.g.: "http://10.0.0.37/index.htm?sca=80"

<factor> is a % number: no change for 100 (=100%), reduction for <100, enlargement for >100. The "?sca" parameter
can be specified along with other URL parameters (e.g. ?pag). The sequence of the parameters has no effect.

8.9 The FTP server (pFTP.exe)

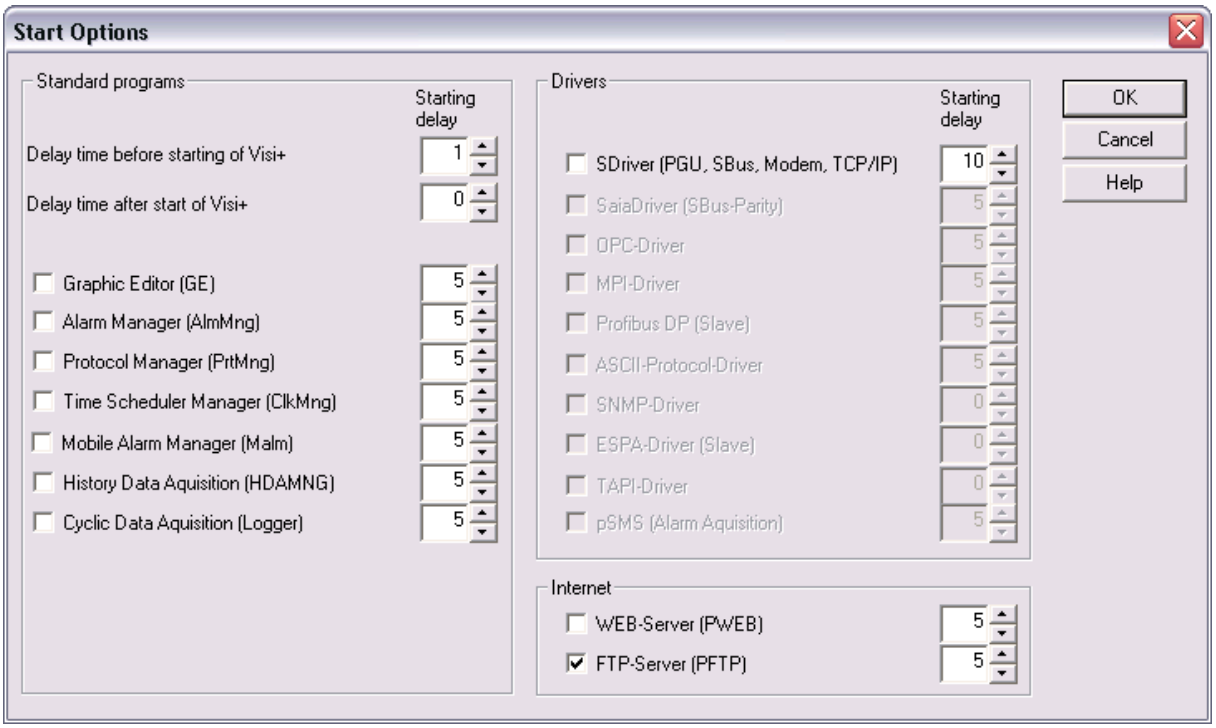
Using the FTP server, a network connection can be established to the system's computer so files can be read and written.

8.9.1 Starting the program pFTP

The FTP Server can be started, like any other program, by clicking on its filename (pFTP.exe) in the Windows directory.

A user-created shortcut on the desktop is even more convenient.
Alternatively, the FTP Server can be started through the **Project Manager** module**,** where the program to be started for each project can be selected. To do so, proceed as follows:

1. Start the **Project Manager** module.
2. Access the menu item or press the **<Start Options>** button.
3. Mark the checkbox **FTP Server (pFTP)**.



4. pFTP is automatically started the next time a project is opened.

The module normally runs as a background process and is displayed on the Windows taskbar (bottom right corner of the screen) as a blue arrow .

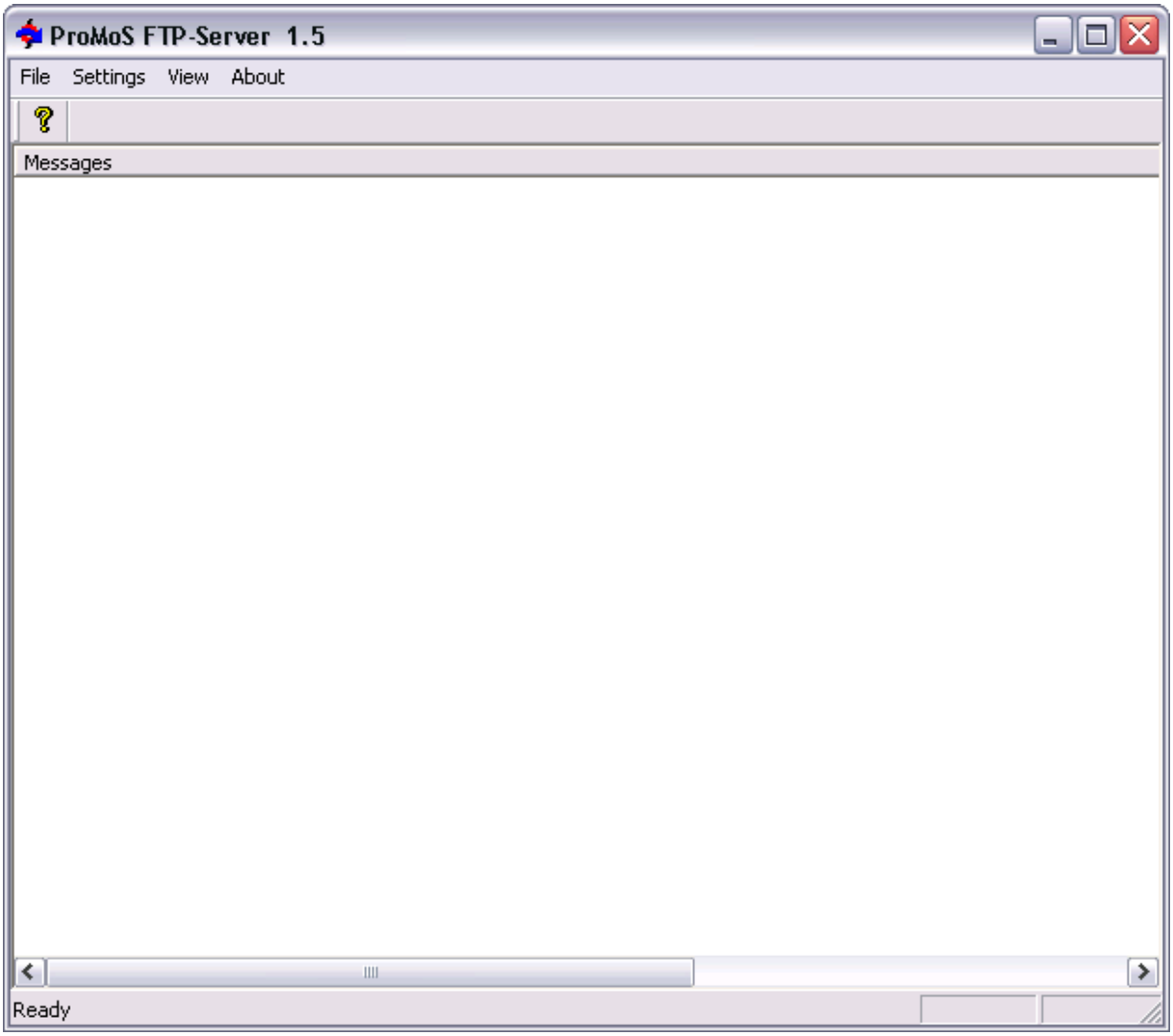
8.9.2 The FTP Server main window

The control window of the FTP Server can be opened two different ways. Either by

double-clicking on the FTP Server icon

or

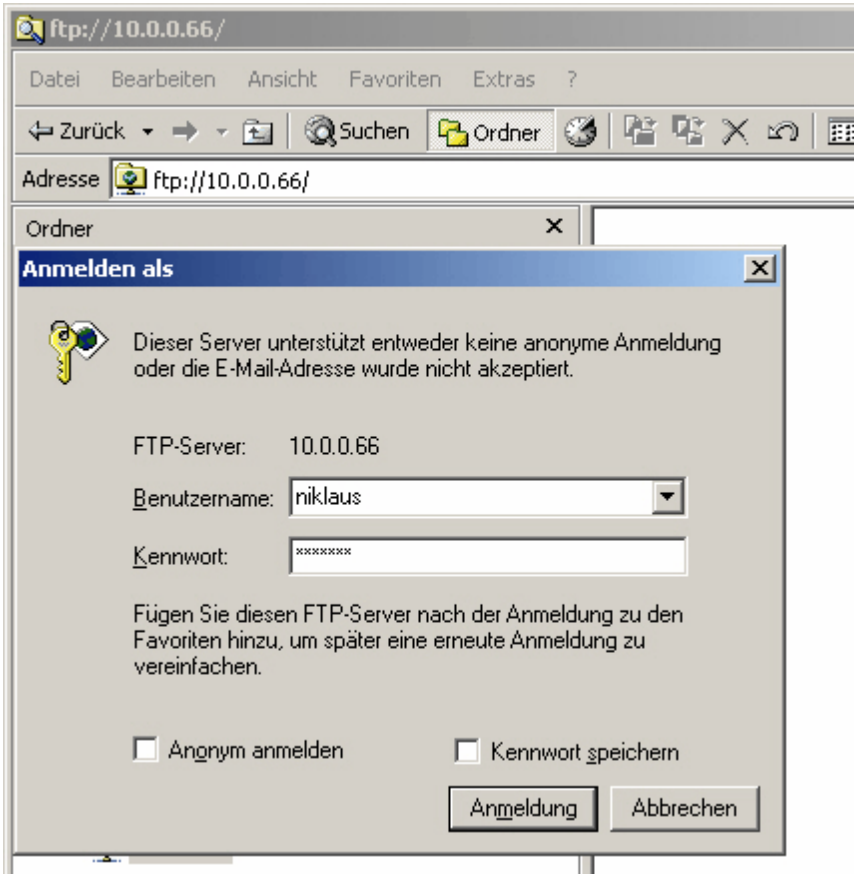
1. clicking with the **right mouse button** on the FTP Server icon on the Windows taskbar.
2. Select the **"Open"** command.



8.9.3 Accessing the FTP Server via Windows Explorer

In principle, all file management programs which can be used to establish an FTP connection can access the FTP Server (e.g. Total Commander).

In Windows Explorer, enter the IP address of the destination computer in the address bar. Then the password prompt automatically appears.

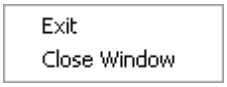


The FTP client in Internet Explorer has not been very well tested in practice. We recommend using dedicated FTP clients (e.g. Total Commander).

8.9.4 FTP Server menus

8.9.4.1 File menu

This menu offers functions for exiting the FTP Server.

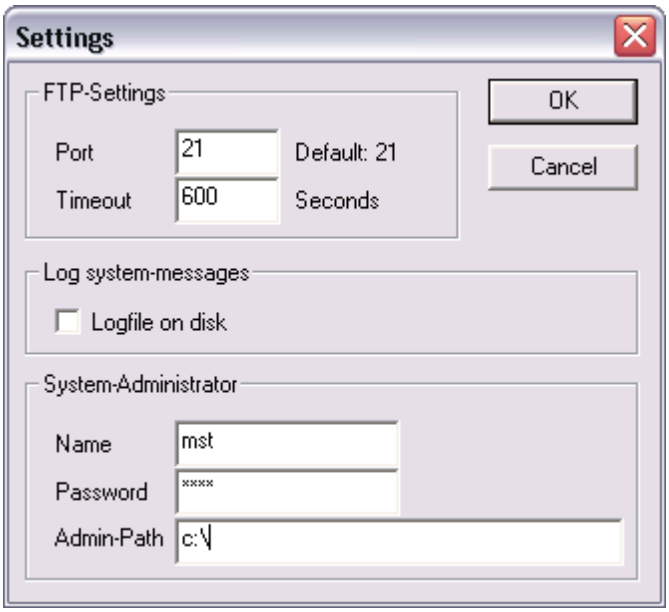


Exit
The FTP Server is exited.

Close
This menu item closes the main window of the FTP Server. It does not exit the program; it is only minimised to the taskbar.

8.9.4.2 Configuration menu

The FTP connection can be configured here.



With **FTP Settings** the port and the timeout can be indicated. After the timeout elapses, the connection is automatically terminated.

If **Log File on Disk** is marked, a log file is generated in the /log folder in the current project directory, in which all debug messages are entered.

In the **System Administrator** section a name and password must be entered. A start path can be indicated under **Admin Path**.

FTP rights can be assigned on a per user basis, and other folders shared, in the User Management module. It is therefore possible to ensure that a user can only edit files in their own directory. The assignment of FTP user rights is described in the ***chapter*** ***User management****.*

8.9.4.3 View menu

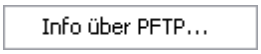
Functions for changing the appearance of the FTP Server are contained in the "**View**" menu.

Toolbar
Switches the toolbar on or off.

Status bar
Switches the status bar on or off.

8.9.4.4 Help menu

The Visi.Plus online help and current information about the FTP Server are found in the **"Help"** menu.



About pFTP
Displays the current version of **FTP Server**. The version must always be given in case of any support queries!

Should support queries become necessary, the internet forum is preferential to contact via telephone or e-mail. This offers two advantages: First, help is available more quickly; and secondly, other users also benefit from the entries.

8.9.5 Supported FTP commands

The FTP Server supports the following commands:

```
USER
PASS
ACCT
CWD
CDUP
SMNT
QUIT
REIN
PORT
PASV
TYPE
STRU
MODE
RETR
STOR
STOU
APPE
ALLO
REST
RNFR
RNTO
ABOR
DELE
RMD
MKD
PWD
LIST
NLST
SITE
SYST
STAT
HELP
AUTH
FEAT
NOOP
```

8.9.6 FTP error management

The possible FTP error messages are listed in the following.

Code 0 UTE_SUCCESS = CUT_SUCCESS
Code 1 UTE_ERROR = CUT_ERROR

Connection errors

Code 2 UTE_CONNECT_FAILED
Code 3 UTE_CONNECT_REJECTED
Code 4 UTE_CONNECT_TERMINATED
Code 5 UTE_CONNECT_TIMEOUT
Code 6 UTE_NOCONNECTION
Code 7 UTE_NAME_LOOKUP_FAILED
Code 8 UTE_DATAPORT_FAILED
Code 9 UTE_ACCEPT_FAILED

Server errors

Code 10 UTE_SVR_REQUEST_DENIED
Code 11 UTE_SVR_NOT_SUPPORTED
Code 12 UTE_SVR_NO_RESPONSE
Code 13 UTE_SVR_ACCESS_DENIED
Code 14 UTE_SVR_DATA_CONNECT_FAILED

Socket errors

Code 15 UTE_SOCK_NOT_OPEN
Code 16 UTE_SOCK_ALREADY_OPEN
Code 17 UTE_SOCK_CREATE_FAILED
Code 18 UTE_SOCK_BIND_FAILED
Code 19 UTE_SOCK_CONNECT_FAILED
Code 20 UTE_SOCK_TIMEOUT
Code 21 UTE_SOCK_RECEIVE_ERROR
Code 22 UTE_SOCK_SEND_ERROR
Code 23 UTE_SOCK_LISTEN_ERROR
Code 24 UTE_CLIENT_RESET
Code 25 UTE_SERVER_RESET

File errors

Code 26 UTE_FILE_TYPE_ERROR
Code 27 UTE_FILE_OPEN_ERROR
Code 28 UTE_FILE_CREATE_ERROR
Code 29 UTE_FILE_READ_ERROR
Code 30 UTE_FILE_WRITE_ERROR
Code 31 UTE_FILE_CLOSE_ERROR
Code 32 UTE_FILE_ERROR
Code 33 UTE_FILE_FORMAT_ERROR
Code 34 UTE_FILE_TMP_NAME_FAILED

Buffer errors

Code 35 UTE_BUFFER_TOO_SHORT

Code 36 UTE_NULL_PARAM

Response errors
Code 37 UTE_INVALID_RESPONSE
Code 38 UTE_NO_RESPONSE

Index errors
Code 39 UTE_INDEX_OUTOFRANGE

User validation errors
Code 40 UTE_USER_ERROR
Code 41 UTE_PASSWORD_ERROR

Message errors
Code 42 UTE_INVALID_MESSAGE
Code 43 UTE_INVALID_FORMAT
Code 44 UTE_FILE_NOT_MIME

URL errors
Code 45 UTE_BAD_URL

Command errors
Code 46 UTE_INVALID_COMMAND
Code 47 UTE_MAIL_FAILED
Code 48 UTE_RETR_FAILED
Code 49 UTE_PORT_FAILED
Code 50 UTE_LIST_FAILED
Code 51 UTE_STOR_FAILED
Code 52 UTE_DATA_FAILED
Code 53 UTE_USER_FAILED
Code 54 UTE_HELLO_FAILED
Code 55 UTE_PASS_FAILED
Code 56 UTE_STAT_FAILED
Code 57 UTE_TOP_FAILED
Code 58 UTE_UIDL_FAILED
Code 59 UTE_DELE_FAILED
Code 60 UTE_RSET_FAILED
Code 61 UTE_XOVER_FAILED
Code 62 UTE_USER_NA
Code 63 UTE_PASS_NA
Code 64 UTE_ACCT_NA
Code 65 UTE_RNFR_NA
Code 66 UTE_RNTO_NA
Code 67 UTE_RCPT_FAILED
Code 68 UTE_NNTP_BAD_ARTICLE
Code 69 UTE_NNTP_NOPOSTING
Code 70 UTE_NNTP_POST_FAILED
Code 71 UTE_NNTP_AUTHINFO_USER_FAILED

Code 72 UTE_NNTP_AUTHINFO_PASS_FAILED
Code 73 UTE_XOVER_COMMAND_FAILED

Message errors
Code 74 UTE_MSG_OPEN_FAILED
Code 75 UTE_MSG_CLOSE_FAILED
Code 76 UTE_MSG_WRITE_LINE_FAILED
Code 77 UTE_MSG_READ_LINE_FAILED
Code 78 UTE_MSG_NO_ATTACHMENTS
Code 79 UTE_MSG_BODY_TOO_BIG
Code 80 UTE_MSG_ATTACHMENT_ADD_FAILED

Data source errors
Code 81 UTE_DS_OPEN_FAILED
Code 82 UTE_DS_CLOSE_FAILED
Code 83 UTE_DS_WRITE_FAILED

Encoding errors
Code 84 UTE_ENCODING_INVALID_CHAR
Code 85 UTE_ENCODING_LINE_TOO_LONG

IMAP4 errors
Code 86 UTE_LOGIN_FAILED
Code 87 UTE_NOOP_FAILED
Code 88 UTE_UNKNOWN_COMMAND
Code 89 UTE_UNKNOWN_RESPONSE
Code 90 UTE_AUTH_OR_SELECTED_STATE_REQUIRED
Code 91 UTE_SELECTED_STATE_REQUIRED

RAS Errors
Code 92 UTE_RAS_LOAD_ERROR
Code 93 UTE_RAS_DIAL_ERROR
Code 94 UTE_RAS_DIALINIT_ERROR
Code 95 UTE_RAS_HANDLE_ERROR
Code 96 UTE_RAS_ENUM_ERROR
Code 97 UTE_RAS_ENTRYNAME_ERROR

Unclassified errors
Code 98 UTE_ABORTED
Code 99 UTE_BAD_HOSTNAME
Code 100 UTE_INVALID_ADDRESS
Code 101 UTE_INVALID_ADDRESS_FORMAT
Code 102 UTE_USER_TERMINATED
Code 103 UTE_ANS_NOT_FOUND
Code 104 UTE_SERVER_SET_NAME_FAILED
Code 105 UTE_PARAMETER_TOO_LONG

Code 106 UTE_PARAMETER_INVALID_VALUE
Code 107 UTE_TEMP_FILENAME_FAILED
Code 108 UTE_OUT_OF_MEMORY
Code 109 UTE_GROUP_INFO_UPDATE_FAILED
Code 110 UTE_GROUP_NOT_SELECTED
Code 111 UTE_INTERNAL_ERROR
Code 112 UTE_ALREADY_IN_USE
Code 113 UTE_NO_CURRENT_MSG_SET
Code 114 UTE_QUOTE_LINE_IS_EMPTY
Code 115 UTE_REST_COMMAND_NOT_SUPPORTED

Unclassified errors

Code 116 UTE_SYSTEM_INFO_LOAD_FAILED
Code 117 UTE_USER_INFO_LOAD_FAILED
Code 118 UTE_USER_NAME_ALREDY_EXIST
Code 119 UTE_MAILBOX_NAME_ALREDY_EXIST
Code 120 UTE_LOAD_SECURITY_LIBRARIES_FAILED
Code 121 UTE_OPEN_CERTIFICATE_STORE_FAILED
Code 122 UTE_FAILED_TO_FIND_CERTIFICATE
Code 123 UTE_FAILED_TO_CREATE_SECURITY_CREDENTIALS
Code 124 UTE_FAILED_TO_INITIALIZE_SECURITY_CONTEXT
Code 125 UTE_FAILED_TO_ACCEPT_SECURITY_CONTEXT
Code 126 UTE_HANDSHAKE_FAILED
Code 127 UTE_FAILED_TO_QUERY_CERTIFICATE
Code 128 UTE_FAILED_TO_GET_CERTIFICATE_CHAIN
Code 129 UTE_FAILED_TO_VERIFY_CERTIFICATE_CHAIN
Code 130 UTE_FAILED_TO_VERIFY_CERTIFICATE_TRUST
Code 131 UTE_FAILED_TO_GET_SECURITY_STREAM_SIZE
Code 132 UTE_FAILED_TO_APPLY_CONTROL_TOKEN
Code 133 UTE_FAILED_TO_RECEIVE_SECURITY_MESSAGE
Code 134 UTE_SECURITY_CONTEXT_EXPIRED
Code 135 UTE_FAILED_TO_DECRYPT_SECURITY_MESSAGE
Code 136 UTE_FAILED_TO_ENCRYPT_SECURITY_MESSAGE
Code 137 UTE_INVALID_CHARS_IN_STRING_PARAM
Code 138 UTE_FAILED_TO_GENERATE_SECURITY_KEY
Code 139 UTE_FAILED_TO_EXPORT_SECURITY_KEY
Code 140 UTE_OPEN_CERTIFICATE_STORE_FIRST
Code 141 UTE_FAILED_CREATE_ICENROLL
Code 142 UTE_CONNECT_FAIL_NO_SSL_SUPPORT
Code 143 UTE_CERTIFICATE_INVALID_DATE
Code 144 UTE_IMAP4_TLS_NOT_SUPPORTED
Code 145 UTE_SMTP_TLS_NOT_SUPPORTED
Code 146 UTE_POP3_TLS_NOT_SUPPORTED
Code 147 UTE_UNSUPPORTED_KEY_SIZE
Code 148 UTE_UNSUPPORTED_ENCODING_TYPE
Code 149 UTE_INVALID_CHARACTER_IN_CHARSET
Code 150 UTE_CHARSET_TOO_BIG
Code 151 UTE_INVALID_ENCODING_FORMAT

8.10 Cyclical data logging (logger.exe)

The Data Logger enables the cyclical logging of data dependent on a DMS value.

The data points defined in a configuration file are saved to an ASCII text format file after the event also defined in the configuration file has occurred. This file can be evaluated with any spreadsheet program.

The **Logger** generates a new file "**name.csv**" each day.
This facilitates access to the stored data.

8.10.1 Configuration of the (Logger)

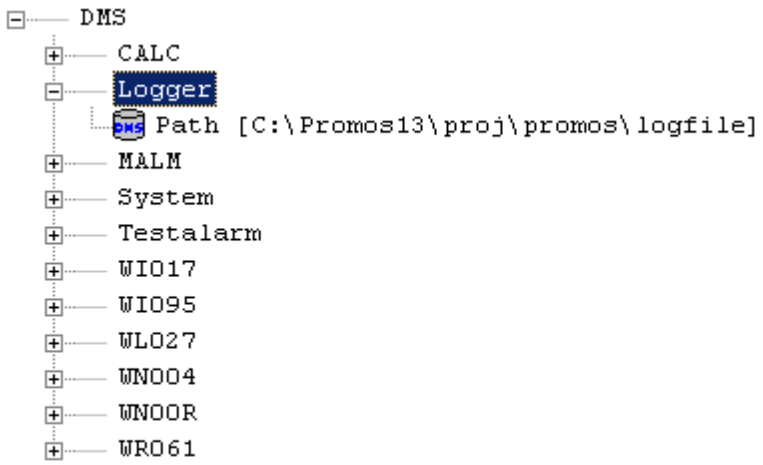
The following steps must be carried out so that the Data Logger can fulfil its task:

- **Save location of the configuration file**
- **Create the configuration file**
- **Start the Logger program**
- If the **Logger** should be triggered by a button from a control screen, the button for this must be created in the GE (see the chapter Operating programs, GE).

8.10.2 Save location of the configuration file

The save location of the file in which the data should be saved is defined with the directory path and filename including extension in the DMS.
This entry in the DMS must still take place manually for the time being.
Working directly in the DMS structure is described in the chapter "**Creating the DMS structure**".

New DMS data point entry "Logger:Path"
First, the data point **Logger** of the type "**NONE**" must be inserted, followed by the data point **Path** of the type "**STR**". The directory path and the filename (in the example "**logfile**") without the extension should be entered as the value of the data point **Path**.

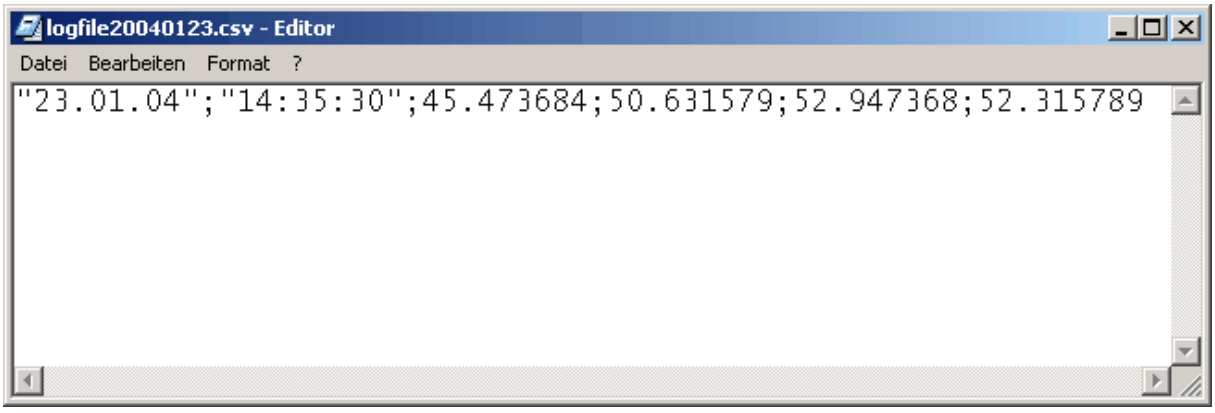


The current date is always attached to the filename (in the example "**logfile**") and saved with the file extension "**.csv**". This means a file with the name **logfile** is saved, for example, as **logfile20040623.csv**.

Advantage of the date appearing in the name:
If changes are made in the file with an ASCII editor and then saved, the file receives the current date from the operating system. The filename is retained, as already explained.

A new file is created each day. This facilitates searching for specific process data.

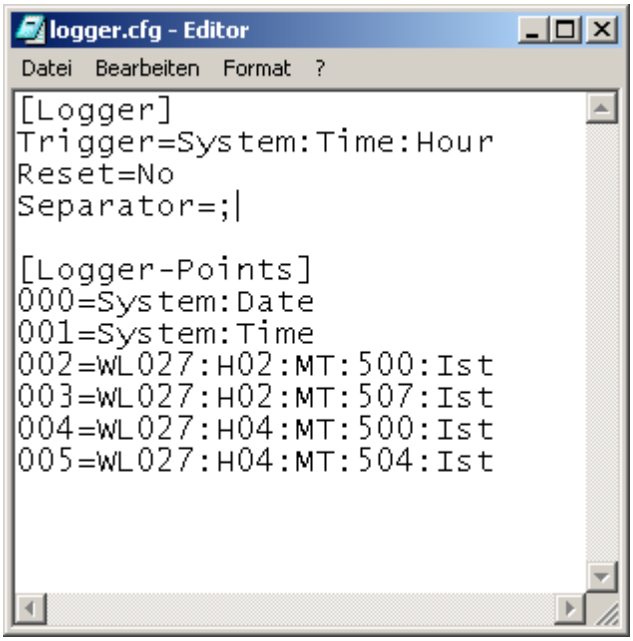
A defined log file could appear as follows:



Character strings (in the image shown above the **date** and **time indication**) are placed in inverted commas. After this, the indicated **current values**, each of which is separated by a semicolon "**;**", are listed.

8.10.3 Create the configuration file

All configurations must be created in a **configuration file**. This must be called **logger.cfg**. This file can be created and edited with any ASCII editor.



[Logger] section
Trigger
Indicates the data point that triggers saving. It is preferably a binary value. It is then possible, for example, to coordinate when a data set should be generated on the PLC. Another possibility would be to save a process diagram every hour through initialisation of the data point **System:Time:Hours** as a trigger.

Reset
If, for example, Visi.Plus should reset the trigger data point in the PLC (a flag) after saving the logger program, **Reset** should be set to **"Yes"**. This means the PLC receives a confirmation when saving is completed.
A reset only makes sense with digital values.

Separator
Determines the separator character that separates the individual data. If this parameter is not indicated, a comma is set as the separator by default.

[Logger-Points] section
In this section a list of all data to be logged is created. The following rules apply:

- Max. 999 data values
- One data value per line
- Progressive three-digit numbered lines, beginning with 000.
  (If the progressive numbering has gaps, **reading of the configuration file is interrupted** with the first number that is **not present).**
- An assignment character **"="** and the DMS data point to be logged must be indicated immediately after the number.

ATTENTION:*The indicated DMS names **must** be present in the DMS!*

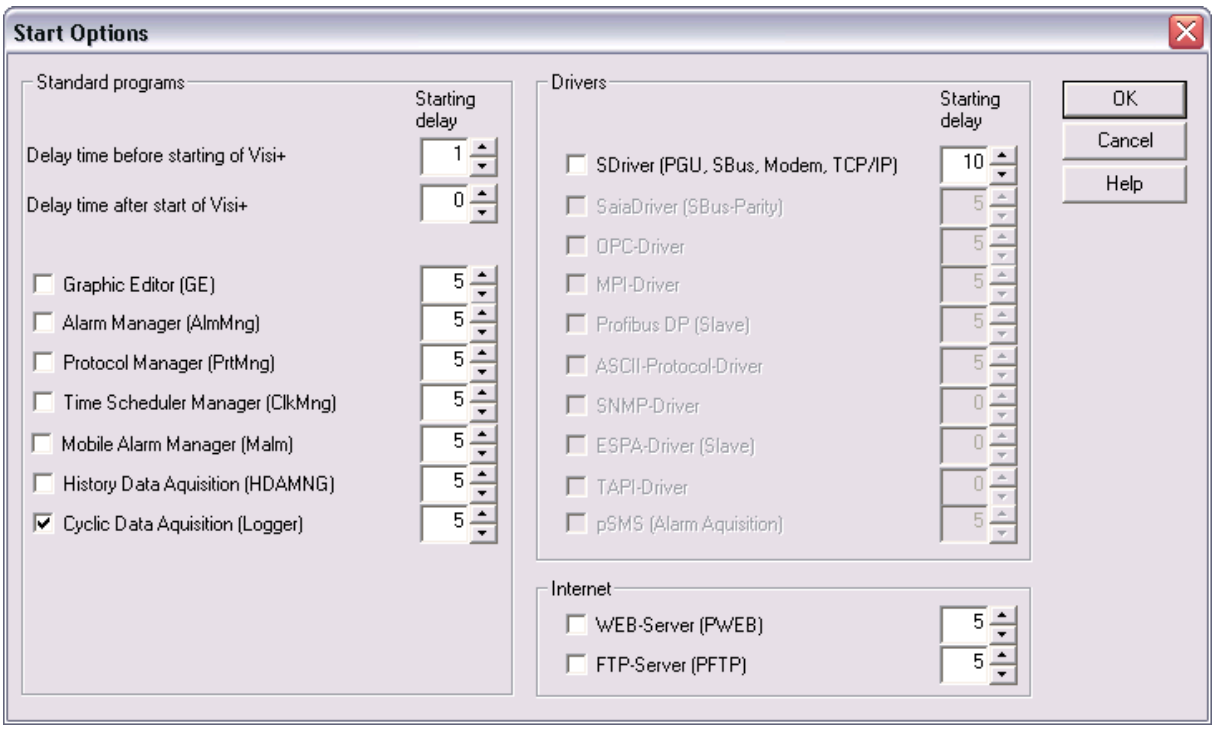
8.10.4 Start the Logger program

Requirement:
The DMS and the PDBS must have been started!

As soon as the DMS is loaded, cyclical data logging can be started, like any other program, by clicking on its filename (logger.exe) in Windows.

This can be done conveniently through a shortcut created on the desktop or each time Visi.Plus starts up via Settings in the **Project Manager** module. To do so, proceed as follows:

1. Start the **Project Manager module.**
2. Select the menu item or the **<Start Options> button.**
3. Mark the checkbox **Cyclical Data Logging (Logger).**
4. The **Logger** starts up automatically the next time a project is opened.



The module normally runs as a background process and is displayed on the Windows taskbar (bottom right corner) as a light blue dot .

If the Logger program is not started, no cyclical data logging is performed!

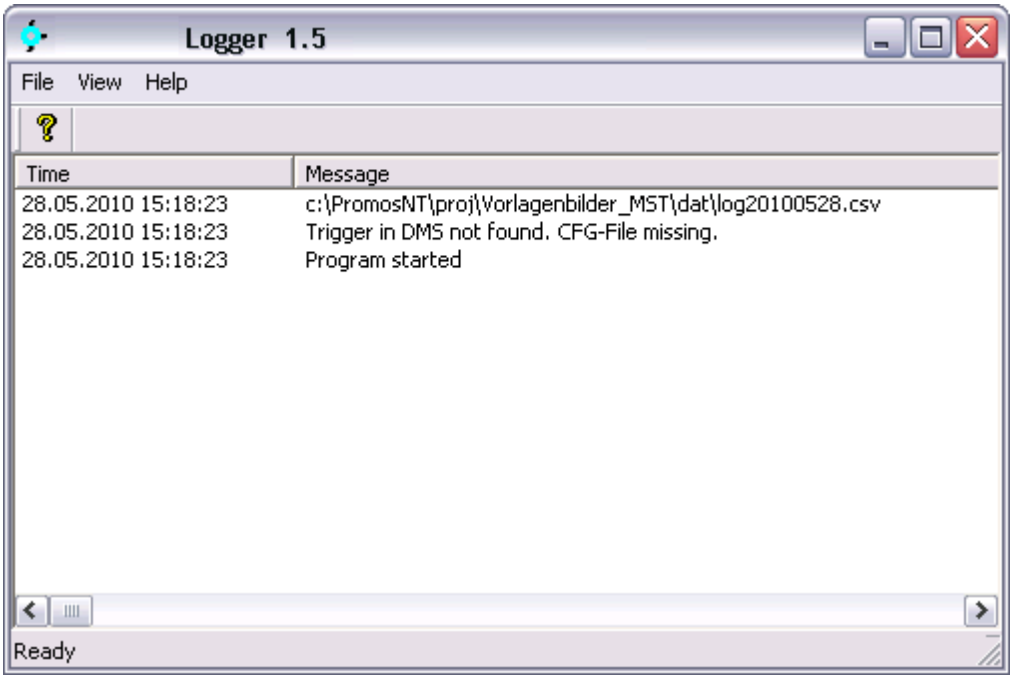
8.10.5 The Logger window

The Logger control window can be opened in two ways.

Either **double-click** on the Logger icon on the Windows taskbar,

or

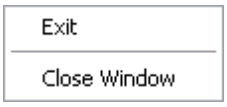
click **on the Logger icon**  with the right mouse button **and then choose the "**Display" command.



8.10.6 The Logger menus

8.10.6.1 File menu

This menu offers functions for exiting and closing the Logger module.



Exit
The **Logger** program is exited.

Close 
Closes the main window of the **Logger**. This does not quit the program; it is only minimised to the taskbar.

8.10.6.2 View menu

In the "**View**" menu, functions are available for modifying the appearance of the **Logger**.

Toolbar
Switches the toolbar on or off.

Status bar
Switches the status bar on or off.

8.10.6.3 Help menu

The Visi.Plus online help and current information about the **Logger** are found in the **"Help"** menu.

About Logger
Displays the current version of the **Logger**. The version must always be given in case of any support queries!

Should support queries become necessary, the internet forum is preferential to contact via telephone or e-mail. This offers two advantages: First, help is available more quickly; and secondly, other users also benefit from the entries.

8.10.7 Logger error handling

The Logger has an error handling capability. It issues messages to the View window of the Logger. The messages are described in the following:

DMS_Connect failed
No connection could be established with the DMS. Check whether the DMS has been started.

Registering "Logger:Trigger" failed
The data point for triggering data acquisition cannot register with the DMS. Check trigger name and whether the DMS is running.

Registering "System:NT:Quit" failed
The data point for exiting the program cannot register with the DMS. Check whether the DMS is running.

Reading "System:Project" failed
Current project cannot be read. Check project settings in the DMS.

Invalid point in list
A data point that is not present in the DMS was found in the list of data to be saved. Check the list of data points and the numbering in the configuration file.

Trigger in DMS not found. CFG file missing
The data point for triggering is not defined. Check configuration file.

Can't open or create Logfile
File cannot be opened. Check the filename including data path in the DMS (data point:**Logger:Path**).

Can't read: XY

Data point XY cannot be read. Check DMS name in the configuration file.

Can't write file
The file cannot be written. Check the data storage medium and/or the data path.

8.11 The calculation tool (pCalc.exe)

Calculation program

8.11.1 Introduction

pCalc allows any number of calculations using data from the DMS and PDBS database.

For example, any value can be used in a calculation with any other value or a constant.

Example:
```
Calculation:Value1 = Group1:Motor1:Temperature * 2.5
```

A new data point "Calculation:Value1" is added to the DMS. When there is any change of the DMS data point "Group:Motor1:Temperature", this value is multiplied by the constant 2.5 and the result is stored in the data point "Calculation:Value1".

The calculations are done on an exclusively event-oriented basis, i.e. the result is only recalculated when a variable in the formula in the DMS is changed.

It is also possible to access historical data. It is possible to access variables when defining dates and times. Historical data are updated hourly.

Example:
```
pCalc:HDA_Data:Value01 = BN028:H09:MT:501:ActualValue@01.02.%y 00:00:00
```

In the example, the value as of 1 February of the current year is accessed. Any specification of date and time can be defined with variables.

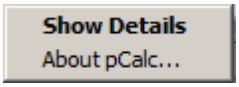
Note: pCalc can only process float values (FLT).

8.11.2 Program outputs

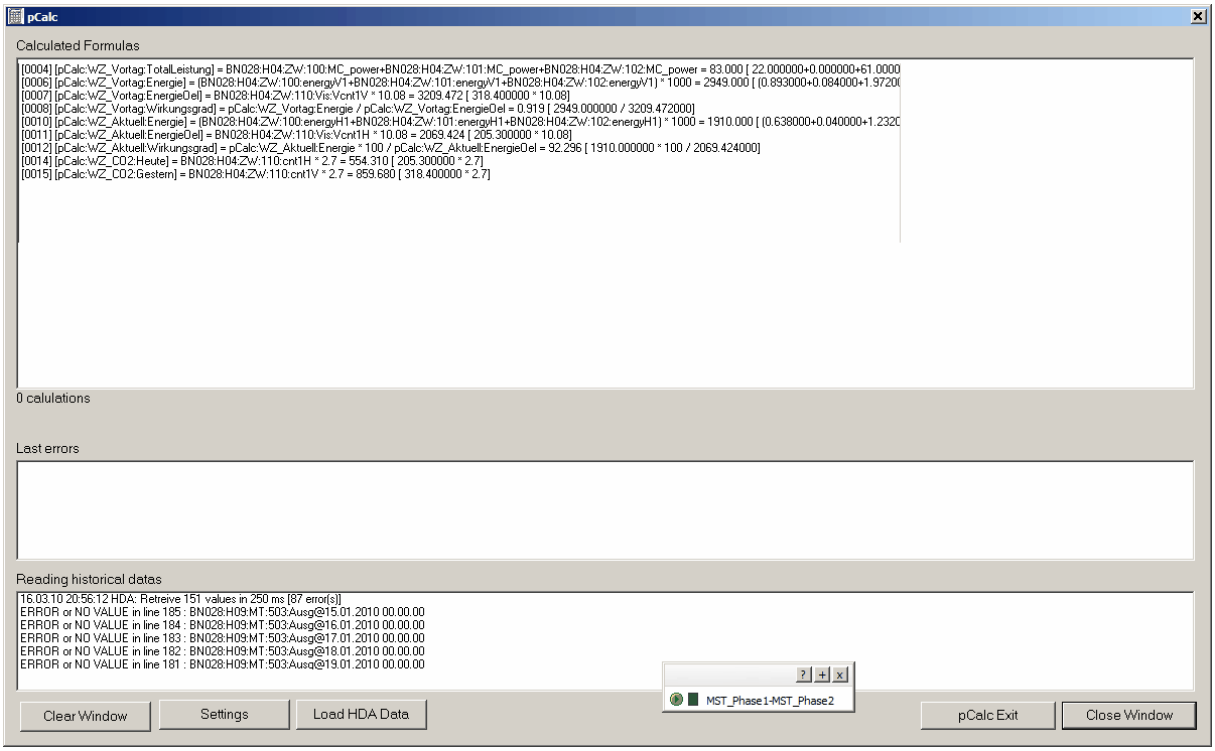
The program normally runs completely in the background.

It is only displayed as a small "computer" icon in the taskbar.

When clicked with the right mouse button, the following menu appears:



The program can be displayed as a window using the menu item "Display details":



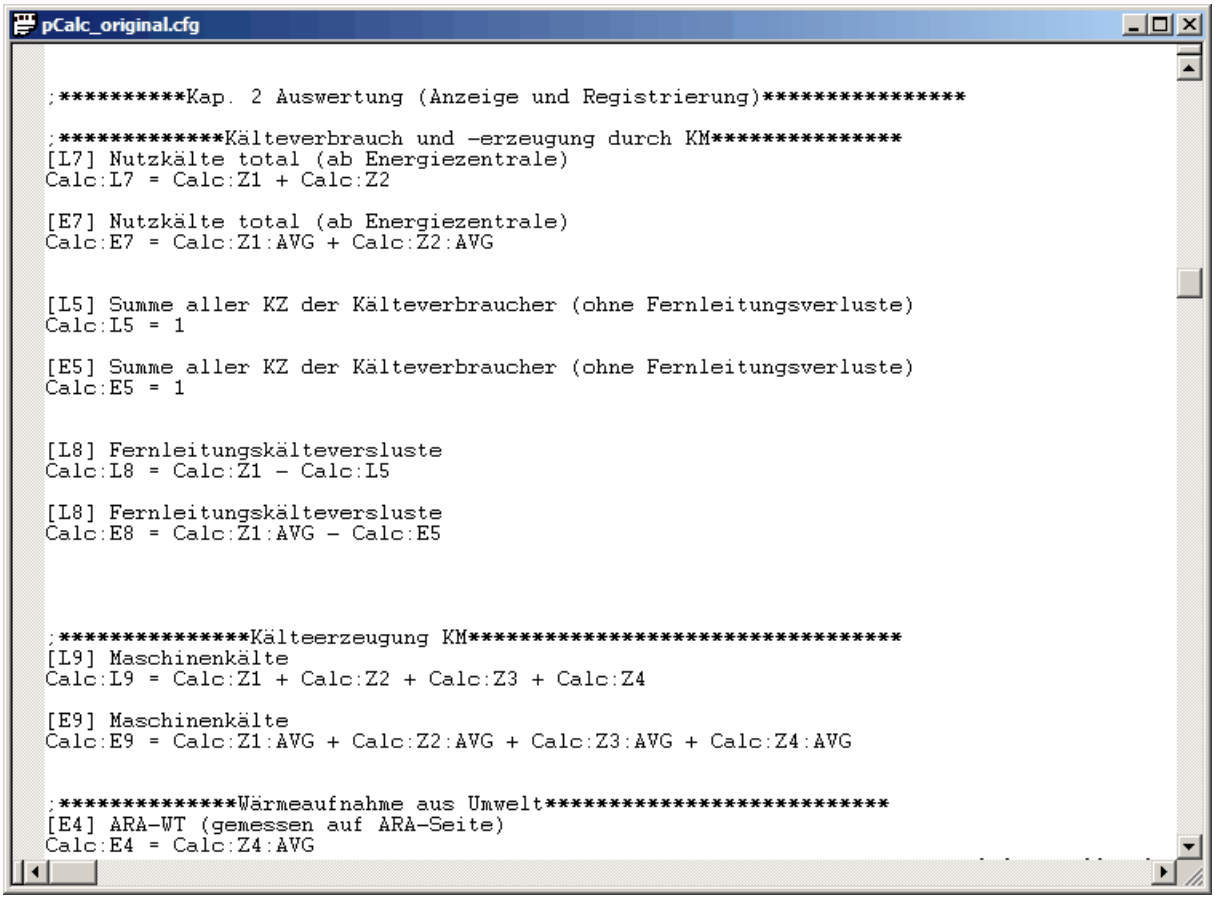
All calculations are output in the upper part of the window and can be monitored there.

Error messages are output in the centre area (see the chapter Error messages).

In the lower part, access operations to historical data are shown.

8.11.3 Configuration

The complete formulas are stored in a configuration file. The file must have the name "pcalc.cfg" and be located in the current project directory under \CFG.



Any number of assignments can be made. The basic rule is:

Result = Formula

Where "Result" must be a data point in the DMS (Format e.g. AAA:BBB:CCC). The formula can be structured in almost any way.

8.11.4 Restrictions

There are some restrictions for the definition and for the formulas.

- A formula must not exceed 1000 characters including assignment
- A maximum of 16 DMS variables per formula can be defined
- No recursions are allowed
- Only FLT variables can be calculated (no boolean operators)
- Limit of calculation are 5000.

If results are saved in DMS variables which are created by pCalc, or are deleted or renamed, the "old" DMS variables are retained. They must be deleted manually in the DMS.

Note: Parentheses must be used to ensure correct computation of more complex formulas.

8.11.5 Constants

Any number of constants can be defined within a formula.

Example:

```
Calc:Const = 4.16
Calc:Const2 = 4.16 * 45 / 44.33
```

Constants within formulas

```
Calc:Result = Calc:Z1 * 3.14159
Calc:Result2 = Calc:Z1 + 589 / (Calc:Z2 * 1.25)
```

8.11.6 Comment

Comments can be inserted in two ways:

- A pure comment in the configuration file
- Comment which is to be applied in the DMS

Pure comment

Every line which begins with a semicolon is ignored by pCalc.

Example:

```
; This is a comment
```

Comment which is applied in the DMS

If a comment is to be applied in the DMS, the text can be inserted before the formula without any special characters such as a semicolon and/or colon.
The last line of the text is always inserted in the DMS as a "Comment". The comment can also be displayed directly in the graphic system.

Example:

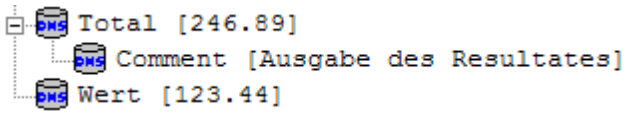
```
Calculation of total consumption
Total:Value = Total:Value1 + Total:Value2 + Total:Value3
```

In this case, the text "Calculation of total consumption" is included directly in the DMS when the pCalc program is started.

Example:

```
Output of the result
Test:Total = Test:Value + Test:Value
```

Inclusion in the DMS:



*Attention:*A comment must not contain a colon (:) or an equal sign (=).

8.11.7 Creating objects

It is possible to have objects created automatically. To do so, the template objects must defined in the configuration file.

Example:

```
;Testobject
OBJECT= MES01
DMSNAME = TT:H02:500
NAME = Test of an object
ActualValue = BN028:H09:MT:501:ActualValue * 2.5
```

All three inputs must always be defined (object type, DMS name and designation).

OBJECT	Template object (BMO)
DMSNAME	DMS name (with no attribute)
NAME	Designation of the new object

Upper and lower case are disregarded.

The designation of the attribute can be defined directly after the object (VLO) has been defined without having to define the complete DMS name. Assignments can only be of the types STR or FLT.

In the example above, the result "ActualValue" refers to the data point "TT:H02:500:ActualValue".

Note: Objects which have been created once are not created again when pCalc is restarted.

8.11.8 Parentheses

Any number of parentheses can be used. The same number of left and right parentheses must be defined. The sequence must conform to the formula.

Example:

```
Calc:Result = ((Calc:Value1 + Calc:Value2) / ((Calc:Value1 - Calc:Value2) * 1.25)
```

8.11.9 Arithmetic operations

The basic arithmetic operations can be used without limit:

- Addition
- Subtraction
- Multiplication
- Division

CAUTION: Be sure to use as many parentheses as possible.
The formula 100/10* 5, for example, will be interpreted incorrectly if no parentheses are used:
- 10* 5 is calculated and afterward 100 / result

The formula must be written as (100/10) * 5.

8.11.9.1 Addition

Any DMS variables may be linked.

Example:

```
Calc:E8 = Calc:Z1:AVG + Calc:E5

Calc:Result = Calc:Z1 + Calc:Z2 + Calc:Z3 + Calc:Z4
```

8.11.9.2 Subtraction

Any DMS variables may be linked.

Example:

```
Calc:E8 = Calc:Z1:AVG - Calc:E5

Calc:Result = Calc:Z1 - Calc:Z2 - Calc:Z3 - Calc:Z4
```

8.11.9.3 Multiplication

Any DMS variables may be linked.

Example:

```
Calc:E8 = Calc:Z1:AVG * Calc:E5

Calc:Result = Calc:Z1 * Calc:Z2 * Calc:Z3 * Calc:Z4 * 3.14159
```

8.11.9.4 Division

Any DMS variables may be linked.

Example:

```
Calc:E8 = Calc:Z1:AVG / Calc:E5

Calc:Result = Calc:Z1 / Calc:Z2 / Calc:Z3 / Calc:Z4
```

8.11.9.5 Special functions

Various special functions are supported, especially trigonometric functions.

ABS(...)	Absolute function	Calculates the absolute value of a variable (e.g.: -123.45 = 123.45)
ACOS(...)	Arc cosine	
ACOSH(...)	Hyperbolic arc cosine	
ASIN(...)	Arc sine	
ASINH(...)	Hyperbolic arc sine	
ATAN(...)	Arc tangent	
ATANH(...)	Hyperbolic arc tangent	
COS(...)	Cosine	
COSH(...)	Hyperbolic cosine	
COT(...)	Cotangent	
FLOOR(...)	Floor	Cuts off decimal places (e.g.: 123.456 = 123.0)
LN(...)	Logarithm	
LOG(...)	Logarithm	
RAD(...)	Radians	
SEC(...)	Secant	
SIN(...)	Sine	
SINH(...)	Hyperbolic sine	
SQRT(...)	Square root	
TAN(...)	Tangent	
TANH(...)	Hyperbolic tangent	

8.11.9.6 Minimum, Maximum

It is also possible to find the smallest or the largest value from a statistical series. To do so, the two functions

- MIN minimum value of several values
- MAX maximum value of several values

are available.

Examples:

```
pCalc:Maximum = max(pCalc:Const, 54.3, Group1:Motor1:Temperature)
pCalc:Minimum = min(Group1:Motor1:Temperature, Group1:Motor2:Temperature,
                    Group1:Motor3:Temperature)
```

A maximum of 16 parameters per definition line is possible. If more than 16 values must be compared, the formula must be extended to several lines.

Attention:
No further calculations are possible within a MAX/MIN function. A parameter must not consist of a formula.

```
pCalc:Maximum = max(pCalc:Const + 44, 54.3, Group1:Motor1:Temperature /
                    Group1:Motor2:Temperature)
```

will not work.

8.11.9.7 Example

Example of a configuration file (calculations for energy analysis)

```
; ------------------------------------------------------------------------------------------------
; Calculation example for pCalc Version 1.6
; ------------------------------------------------------------------------------------------------
This is a comment (is added to the data point as a comment in the DMS (max. 80 characters)
pCalc:WZ_PrevDay:TotalCapacity = BN028:H04:ZW:100:MC_power+BN028:H04:ZW:101:MC_power+
BN028:H04:ZW:102:MC_power

pCalc:WZ_PrevDay:Energy = (BN028:H04:ZW:100:energyV1+BN028:H04:ZW:101:energyV1+
BN028:H04:ZW:102:energyV1) * 1000
pCalc:WZ_PrevDay:EnergOil = BN028:H04:ZW:110:Vis:Vcnt1V * 10.08
pCalc:WZ_PrevDay:Efficiency = pCalc:WZ_PrevDay:Energy / pCalc:WZ_PrevDay:EnergyOil

pCalc:WZ_Current:Energy = (BN028:H04:ZW:100:energyH1+BN028:H04:ZW:101:energyH1+
BN028:H04:ZW:102:energyH1) * 1000
pCalc:WZ_Current:EnergyOil = BN028:H04:ZW:110:Vis:Vcnt1H * 10.08
pCalc:WZ_Current:Efficiency = pCalc:WZ_Current:Energy * 100 / pCalc:WZ_Current:EnergyOil

pCalc:WZ_CO2:Today = BN028:H04:ZW:110:cnt1H * 2.7
pCalc:WZ_CO2:Yesterday = BN028:H04:ZW:110:cnt1V * 2.7

; Temperature values at beginning February (1 to 8 February of the current year)
pCalc:HDA_Data:Value01 = BN028:H09:MT:501:ActualValue@01.02.%y 00:00:00
pCalc:HDA_Data:Value02 = BN028:H09:MT:501:ActualValue@02.02.%y 00:00:00
pCalc:HDA_Data:Value03 = BN028:H09:MT:501:ActualValue@03.02.%y 00:00:00
pCalc:HDA_Data:Value04 = BN028:H09:MT:501:ActualValue@04.02.%y 00:00:00
pCalc:HDA_Data:Value05 = BN028:H09:MT:501:ActualValue@05.02.%y 00:00:00
pCalc:HDA_Data:Value06 = BN028:H09:MT:501:ActualValue@05+1.02.%y 00:00:00
pCalc:HDA_Data:Value07 = BN028:H09:MT:501:ActualValue@05+2.02.%y 00:00:00
pCalc:HDA_Data:Value08 = BN028:H09:MT:501:ActualValue@05+3.02.%y 00:00:00

pCalc:HDA_Data:Oil_LevelToday = BN028:H04:ZW:110:MC_cnt1@%d.%m.%y 00:00:00
pCalc:HDA_Data:Ol_LevelYesterday = BN028:H04:ZW:110:MC_cnt1@%d-1.%m.%y 00:00:00

Oil consumption yesterday
pCalc:HDA_Data:Oil_Yesterday = SUB(BN028:H04:ZW:110:MC_cnt1@%d.%m.%y 00:00:00,
BN028:H04:ZW:110:MC_cnt1@%d-1.%m.%y 00:00:00)
Oil consumption day before yesterday
pCalc:HDA_Data:Oil_DayBeforeYesterday = SUB(BN028:H04:ZW:110:MC_cnt1@%d-1.%m.%y 00:00:00,
BN028:H04:ZW:110:MC_cnt1@%d-2.%m.%y 00:00:00)
Oil consumption two days ago
pCalc:HDA_Data:Oil_2DaysAgo = SUB(BN028:H04:ZW:110:MC_cnt1@%d-2.%m.%y 00:00:00,
BN028:H04:ZW:110:MC_cnt1@%d-3.%m.%y 00:00:00)

;------------------------------------------------------------------------------------------------
;      End of File
;------------------------------------------------------------------------------------------------
```

8.11.9.8 Error messages

The following error messages can be generated:

Mismatched Parens (ERROR [Line x]: Wrong parentheses (-> yz)
Parentheses are incorrect. The number of left parentheses does not equal the number of right parentheses.
Example:

```
Test:Value = ((Test:Value1 * 12.3)
```

Divide By Zero
Division by zero.

Invalid Expression
The formula contains an error.

Missing Operator
An operator is missing.

If you use OBJECT, DMSNAME and NAME you have to define all (Object, DMSName and Name)
This results from an attempt to create an object from a template directly. However, to do this, all three parameters must be defined.
OBJECT = template object (e.g. MES01)
NAME = designation of the object
DMSNAME = DMS name

ERROR [Line x]: Data point not found -> yz
The DMS name defined in the formula cannot be found.

If an error is found (e.g. division by zero), the corresponding DMS value (result) is set to 0.

8.11.9.9 Historical data

It is also possible to access historical data. This is necessary, for example, if an energy consumption value must be calculated.

8.11.9.9.1 Defining the date and time

The date and time must be entered in the following format:
dd.mm.yyyy HH:MM:SS

Wildcards can be used for the time:

%d	for the day
%m	for the month
%y	for the year
%H	for the hours
%M	for the minutes
%S	for the seconds

Example:
Time = 02.01.%y 00:00:00
The wildcard %y will be replaced by the actual year.
Time = 02.01.2010 00:00:00

Constants can be added to or subtracted from the wildcards:

Time = 02.01.%y-1 00:00:00
The wildcard %y-1 will be replaced by the actual year minus one (- 1).
Time = 02.01.2009 00:00:00 (previous year)

Time = 02.%m+1.%y-1 00:00:00"
The wildcards will be set to last year with the next month.
Time = 02.03.2009 00:00:00” (next year)

Examples of format strings:

Value at a certain time
```
HDAVal:Counter:001 = BN028:H04:ZW:110:MC_cnt1@02.01.2010 14:30:00
```

Value at a certain time when using wildcards
```
HDAVal:Counter:002 = BN028:H04:ZW:110:MC_cnt1@02.01.%y 00:00:00
```

Value in previous year
```
HDAVal:Counter:003 = BN028:H04:ZW:110:MC_cnt1@02.01.%y-1 00:00:00
```

Value three months ago
```
HDAVal:Counter:004 = BN028:H04:ZW:110:MC_cnt1@02.%m-3.%y 00:00:00
```

8.11.9.9.2 Calculating counter values

This function allows a simple difference calculation. To calculate counter differences, a special function was implemented SUB(..).

Subtraction of two values:

```
Result = SUB(DMSName@01.%m.%y 00:00:00, DMSName@01.%m-1.%y 00:00:00)
```

corresponds, for example, to a consumption value for the last month.

Examples:

Subtraction of two values:

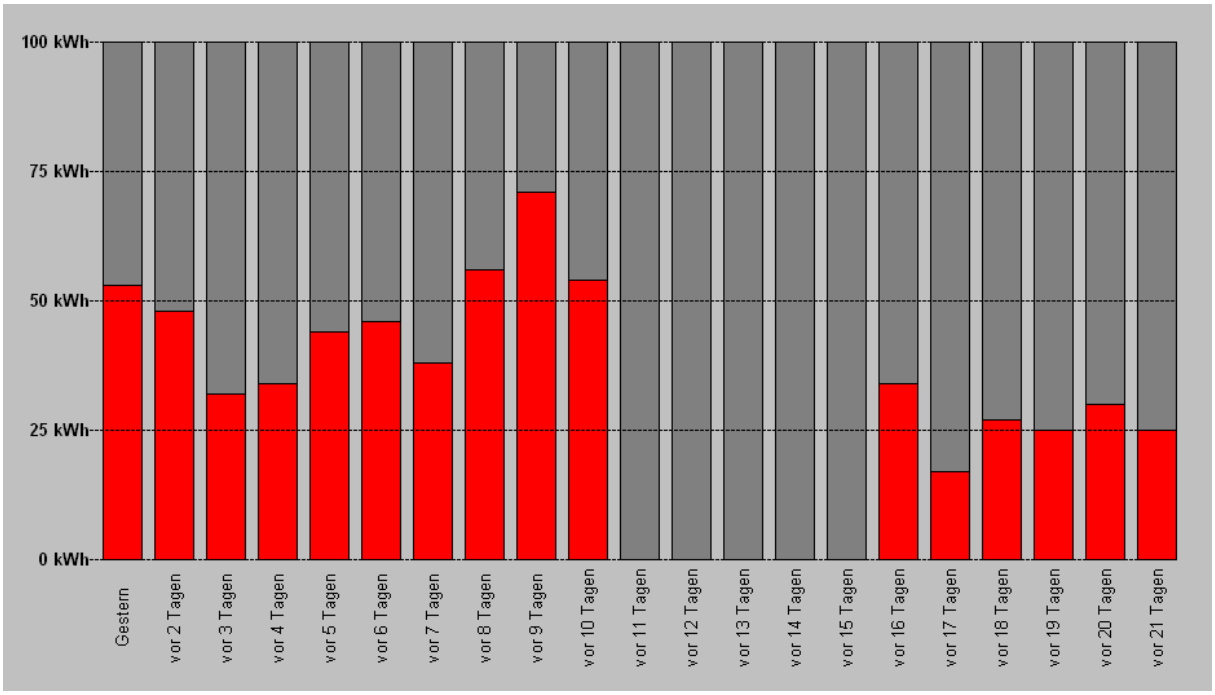
```
HDAVal:Counter:005 = SUB(BN028:H04:ZW:110:MC_cnt1@01.02.%y 00:00:00,
BN028:H04:ZW:110:MC_cnt1@01.01.%y 00:00:00)
```

results in the consumption for January of the current year

8.11.9.9.3 Examples of applications

8.11.9.9.3.1 Bar Charts

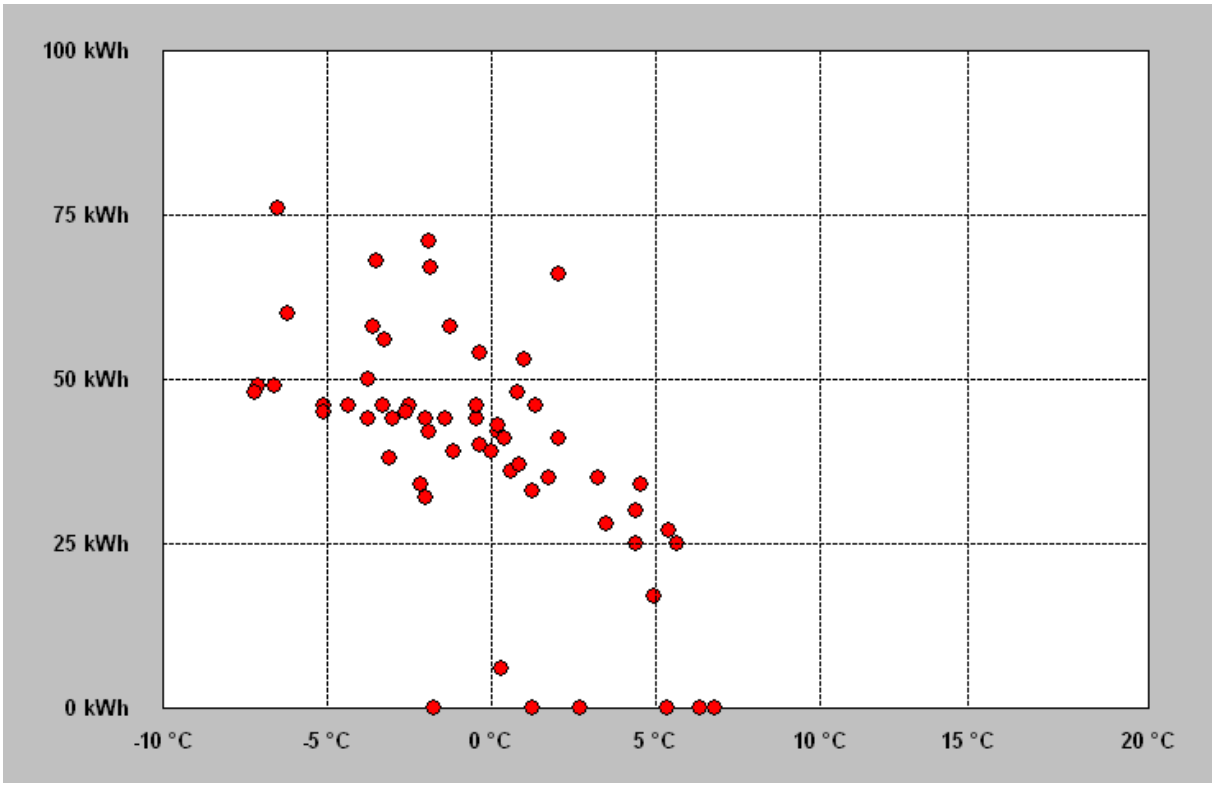
Display of Historical Values




```
; Heating Energy Display (daily)
pCalc:HZ:001 = SUB(BN028:H04:ME:500:MC_energy@%d.%m.%y 00:00:00, BN028:H04:ME:500:MC_energy@%d-1.%m.%y 00:00:00)
pCalc:HZ:002 = SUB(BN028:H04:ME:500:MC_energy@%d-1.%m.%y 00:00:00, BN028:H04:ME:500:MC_energy@%d-2.%m.%y 00:00:00)
pCalc:HZ:003 = SUB(BN028:H04:ME:500:MC_energy@%d-2.%m.%y 00:00:00, BN028:H04:ME:500:MC_energy@%d-3.%m.%y 00:00:00)
pCalc:HZ:004 = SUB(BN028:H04:ME:500:MC_energy@%d-3.%m.%y 00:00:00, BN028:H04:ME:500:MC_energy@%d-4.%m.%y 00:00:00)
pCalc:HZ:005 = SUB(BN028:H04:ME:500:MC_energy@%d-4.%m.%y 00:00:00, BN028:H04:ME:500:MC_energy@%d-5.%m.%y 00:00:00)
pCalc:HZ:006 = SUB(BN028:H04:ME:500:MC_energy@%d-5.%m.%y 00:00:00, BN028:H04:ME:500:MC_energy@%d-6.%m.%y 00:00:00)
pCalc:HZ:007 = SUB(BN028:H04:ME:500:MC_energy@%d-6.%m.%y 00:00:00, BN028:H04:ME:500:MC_energy@%d-7.%m.%y 00:00:00)
pCalc:HZ:008 = SUB(BN028:H04:ME:500:MC_energy@%d-7.%m.%y 00:00:00, BN028:H04:ME:500:MC_energy@%d-8.%m.%y 00:00:00)
pCalc:HZ:009 = SUB(BN028:H04:ME:500:MC_energy@%d-8.%m.%y 00:00:00, BN028:H04:ME:500:MC_energy@%d-9.%m.%y 00:00:00)
pCalc:HZ:010 = SUB(BN028:H04:ME:500:MC_energy@%d-9.%m.%y 00:00:00, BN028:H04:ME:500:MC_energy@%d-10.%m.%y 00:00:00)
pCalc:HZ:011 = SUB(BN028:H04:ME:500:MC_energy@%d-10.%m.%y 00:00:00, BN028:H04:ME:500:MC_energy@%d-11.%m.%y 00:00:00)
pCalc:HZ:012 = SUB(BN028:H04:ME:500:MC_energy@%d-11.%m.%y 00:00:00, BN028:H04:ME:500:MC_energy@%d-12.%m.%y 00:00:00)
pCalc:HZ:013 = SUB(BN028:H04:ME:500:MC_energy@%d-12.%m.%y 00:00:00, BN028:H04:ME:500:MC_energy@%d-13.%m.%y 00:00:00)
pCalc:HZ:014 = SUB(BN028:H04:ME:500:MC_energy@%d-13.%m.%y 00:00:00, BN028:H04:ME:500:MC_energy@%d-14.%m.%y 00:00:00)
pCalc:HZ:015 = SUB(BN028:H04:ME:500:MC_energy@%d-14.%m.%y 00:00:00, BN028:H04:ME:500:MC_energy@%d-15.%m.%y 00:00:00)
pCalc:HZ:016 = SUB(BN028:H04:ME:500:MC_energy@%d-15.%m.%y 00:00:00, BN028:H04:ME:500:MC_energy@%d-16.%m.%y 00:00:00)
pCalc:HZ:017 = SUB(BN028:H04:ME:500:MC_energy@%d-16.%m.%y 00:00:00, BN028:H04:ME:500:MC_energy@%d-17.%m.%y 00:00:00)
pCalc:HZ:018 = SUB(BN028:H04:ME:500:MC_energy@%d-17.%m.%y 00:00:00, BN028:H04:ME:500:MC_energy@%d-18.%m.%y 00:00:00)
pCalc:HZ:019 = SUB(BN028:H04:ME:500:MC_energy@%d-18.%m.%y 00:00:00, BN028:H04:ME:500:MC_energy@%d-19.%m.%y 00:00:00)
pCalc:HZ:020 = SUB(BN028:H04:ME:500:MC_energy@%d-19.%m.%y 00:00:00, BN028:H04:ME:500:MC_energy@%d-20.%m.%y 00:00:00)
pCalc:HZ:021 = SUB(BN028:H04:ME:500:MC_energy@%d-20.%m.%y 00:00:00, BN028:H04:ME:500:MC_energy@%d-21.%m.%y 00:00:00)
pCalc:HZ:022 = SUB(BN028:H04:ME:500:MC_energy@%d-21.%m.%y 00:00:00, BN028:H04:ME:500:MC_energy@%d-22.%m.%y 00:00:00)
...
```

8.11.9.9.3.2 XY Diagram

Display of historical values on an energy / temperature axis.




```
; Heating Energy Display (daily)
pCalc:HZ:001 = SUB(BN028:H04:ME:500:MC_energy@%d.%m.%y 00:00:00, BN028:H04:ME:500:MC_energy@%d-1.%m.%y 00:00:00)
pCalc:HZ:002 = SUB(BN028:H04:ME:500:MC_energy@%d-1.%m.%y 00:00:00, BN028:H04:ME:500:MC_energy@%d-2.%m.%y 00:00:00)
pCalc:HZ:003 = SUB(BN028:H04:ME:500:MC_energy@%d-2.%m.%y 00:00:00, BN028:H04:ME:500:MC_energy@%d-3.%m.%y 00:00:00)
pCalc:HZ:004 = SUB(BN028:H04:ME:500:MC_energy@%d-3.%m.%y 00:00:00, BN028:H04:ME:500:MC_energy@%d-4.%m.%y 00:00:00)
pCalc:HZ:005 = SUB(BN028:H04:ME:500:MC_energy@%d-4.%m.%y 00:00:00, BN028:H04:ME:500:MC_energy@%d-5.%m.%y 00:00:00)
pCalc:HZ:006 = SUB(BN028:H04:ME:500:MC_energy@%d-5.%m.%y 00:00:00, BN028:H04:ME:500:MC_energy@%d-6.%m.%y 00:00:00)
pCalc:HZ:007 = SUB(BN028:H04:ME:500:MC_energy@%d-6.%m.%y 00:00:00, BN028:H04:ME:500:MC_energy@%d-7.%m.%y 00:00:00)
pCalc:HZ:008 = SUB(BN028:H04:ME:500:MC_energy@%d-7.%m.%y 00:00:00, BN028:H04:ME:500:MC_energy@%d-8.%m.%y 00:00:00)
pCalc:HZ:009 = SUB(BN028:H04:ME:500:MC_energy@%d-8.%m.%y 00:00:00, BN028:H04:ME:500:MC_energy@%d-9.%m.%y 00:00:00)
pCalc:HZ:010 = SUB(BN028:H04:ME:500:MC_energy@%d-9.%m.%y 00:00:00, BN028:H04:ME:500:MC_energy@%d-10.%m.%y 00:00:00)
pCalc:HZ:011 = SUB(BN028:H04:ME:500:MC_energy@%d-10.%m.%y 00:00:00, BN028:H04:ME:500:MC_energy@%d-11.%m.%y 00:00:00)
pCalc:HZ:012 = SUB(BN028:H04:ME:500:MC_energy@%d-11.%m.%y 00:00:00, BN028:H04:ME:500:MC_energy@%d-12.%m.%y 00:00:00)
pCalc:HZ:013 = SUB(BN028:H04:ME:500:MC_energy@%d-12.%m.%y 00:00:00, BN028:H04:ME:500:MC_energy@%d-13.%m.%y 00:00:00)
...


; Average Temperatures
pCalc:ATM:001 = BN028:H09:MT:503:Outp@%d.%m.%y 00:00:00
pCalc:ATM:002 = BN028:H09:MT:503:Outp@%d-1.%m.%y 00:00:00
pCalc:ATM:003 = BN028:H09:MT:503:Outp@%d-2.%m.%y 00:00:00
pCalc:ATM:004 = BN028:H09:MT:503:Outp@%d-3.%m.%y 00:00:00
pCalc:ATM:005 = BN028:H09:MT:503:Outp@%d-4.%m.%y 00:00:00
pCalc:ATM:006 = BN028:H09:MT:503:Outp@%d-5.%m.%y 00:00:00
pCalc:ATM:007 = BN028:H09:MT:503:Outp@%d-6.%m.%y 00:00:00
pCalc:ATM:008 = BN028:H09:MT:503:Outp@%d-7.%m.%y 00:00:00
pCalc:ATM:009 = BN028:H09:MT:503:Outp@%d-8.%m.%y 00:00:00
pCalc:ATM:010 = BN028:H09:MT:503:Outp@%d-9.%m.%y 00:00:00
pCalc:ATM:011 = BN028:H09:MT:503:Outp@%d-10.%m.%y 00:00:00
pCalc:ATM:012 = BN028:H09:MT:503:Outp@%d-11.%m.%y 00:00:00
pCalc:ATM:013 = BN028:H09:MT:503:Outp@%d-12.%m.%y 00:00:00
...
```

8.11.10 Special functions

In addition to purely arithmetic functions, special data can also be calculated in pCalc. The result is not stored in the DMS, but rather directly as an HDB file (historical data). Only the last value is stored directly in the DMS.
These functions are usually very CPU intensive and are only carried out once daily.

On initial startup, the data of the last 30 days are processed. Later, only those data which were not calculated are processed. If data are missing for more than 30 days, subsequent calculation is no longer possible.

8.11.10.1 Average calculation

This function is used when an average must be calculated for a single variable. A typical example is the average outdoor temperature per day.

Formula:

```
Average outdoor temperature
CALC:AVG:001:Value = AVG(BN028:H09:MT:501:ActualValue, 24, 900)
```

Parameter 1:
DMS name of the data point for which an average is to be calculated.

Parameter 2:
Time range over which an average is to be calculated. The unit is hours.
In the example above, an average over a period of 24 hours will be calculated. Every 24 hours a value will be stored in the database (HDB file). The stored value always refers to the preceding time period.

Parameter 3:
Number of seconds (sampling interval).
In the example above, a measurement is evaluated every 900 seconds.

8.11.10.2 Calculating hours of operation

Many times, only the current status is stored in the trend database for a unit or motor. The hours of operation are not registered in trend logging because they they would have to be recorded every second during operation. This would generate unnecessary data.

This function makes it possible to calculate the hours of operation per unit of time.

Formula:

```
Fan runtime 1
CALC::001:Value = HOB(BN028:H04:PW:001:RM_On, 1)
```

Parameter 1:
DMS name of the data point for which the hours of operation are to be calculated. The signal must be digital (BIT).

Parameter 2:
Time interval for which the hours of operation are to be calculated. The unit is hours.
In the example above, the hours of operation are calculated for one hour. Every hour, a value is stored in the database (HDB file). The stored value always refers to the preceding time

period.

Note:
HOB stands for **H**ours of **OP**eration.

8.11.10.3 Calculations using hist. data

Almost any calculations can be performed using historical data. The results of the calculations are stored as historical data so that they can be accessed more quickly at a later date. This means that the calculated data can also be analysed with no trouble by means of analysis programs such as pChart.

Formula:
```
CALC:ADD:001:Value = FRM(1, BN028:H30:MT:500:ActualValue +
BN028:H09:MT:501:ActualValue)
```

Parameter 1:
Interval in hours (smaller intervals are not supported because the computing time could block the system).

Parameter 2:
Calculations using the four basic arithmetic operations (addition, subtraction, multiplication, division). The operations can be mixed in any way.
Historical data must exist for all DMS values, otherwise no calculations can be performed.

Note:
All basic arithmetic operations can be performed (+-*/) and any constants can be inserted.
Any number of parentheses () can be used within the formula.

Examples:
```
CALC:ADD:001:Value = FRM(12, (BN028:H30:MT:500:ActualValue * 10 +
BN028:H09:MT:501:ActualValue) / 3.14159)

RES:001:Result = FRM(1, BN028:H04:MV:500:GasVolume * 10.4)

RES:001:Result = FRM(1, (BN028:H09:ME:500:Energy / 1000) * BN028:H09:CT:001:Value +
(BN028:H09:UZ:045:Cnt - BN028:H09:UZ:046:Cnt))
```

Restrictions
Max. 16 DMS variables per formula.
Max. 1000 characters per line.

8.11.10.4 Calculating heating degree days

The degree day number (GTZ, DD) and heating degree days (HDD) are measures for the heating requirements of a building during the heating season. They represent the relationship between room temperature and the outdoor air temperature for the heating days of an assessment period and thus are an aid in determining the heating costs and the heating energy requirements.
The degree day number and the heating degree days can be given using the unit Kd/a (Kelvin * day / year) and therefore have the same dimension as temperature. They are also given with reference to a heating season or a calender month and consequently are of relevance for seasonal variations. There is one value for the long-term climate average and one for the

current weather (meteorological measurement).

Reference values are room temperature and the heating limit. There are two conventions for the reference values:
In accordance with the VDI guideline 2067/DIN 4108 T6, the heating limit is assumed to be 15 °C and the indoor temperature 20 °C; this is specified as GTZ20/15. In Germany, the values determined by the German Meteorological Service provide the basis for the outdoor temperature;
In Austria, Switzerland and Liechtenstein, a heating limit of 12 °C and an indoor temperature of 20 °C are used; this is specified as HDD20/12. The temperature values from ZAMG (Austria) or MeteoSchweiz (Switzerland and Liechtenstein) serve as the reference values here.

As an approximation, the heating degree days can be determined in pCalc based on the measured outdoor temperature.

Formula:
```
CALC:HDD:001:Value = HDD(20, 12, BN028:H09:MT:501:ActualValue)
```

Parameter 1:
The indoor temperature to be used in the calculation (normally 20 °C).

Parameter 2:
The heating limit temperature to be used in the calculation (heating limit). This is country specific (e.g. D: 15°C, CH,AT: 12°C).

Parameter 3:
DMSName of the outdoor sensor value.

8.12 SMS alarm logging (pSMS.exe)

pSMS enables alarms to be sent, received and forwarded to the Alarm Manager via SMS from a PLC.

The received SMS is composed of the telephone number of the PLC that sent the alarm and the SMS text (message). Any PLC can be uniquely identified by the telephone number. The SMS text indicates whether the alarm is incoming or outgoing.

With the DMS configured properly, the corresponding alarm data point is set in the DMS by pSMS. The DMS in turn reports all changes in value of the data point to the Alarm Manager.

The following modems are currently supported:
- TCS MICRO
- Proton GTM
- PhoenixContact GSM/GPRS

The modem must be connected to the serial interface of the PC.

The modules DMS, PDBS and AlmMng must running on the computer.

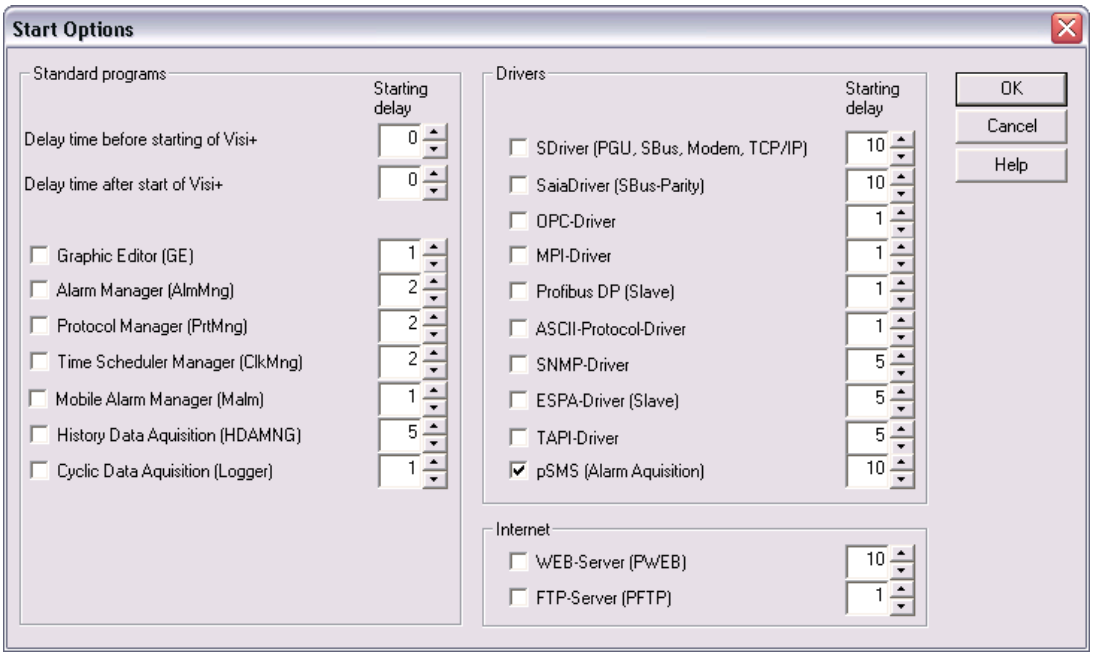
8.12.1 Starting SMS alarm logging (pSMS.exe)

The **DMS** module must have already been started. **pSMS** can either be started separately at any time like any other program in Windows or automatically when a project is started up (Visi.Plus).

If it should be started when Visi.Plus is started, proceed as follows:

Start the **Project Manager** module and click on the **[Start Options]** button.

In the open **Start Options** window, mark the checkbox **pSMS (alarm logging)** in the default programs area. The next time the project is opened, **pSMS** will automatically be started up along with it, with the indicated time delay.



The module normally runs as a background process and is displayed, while running, on the windows taskbar as a small violet dot  (bottom right corner of the screen).

Since incoming alarms from pSMS are forwarded to the Alarm Manager through the DMS, Alarm Manager must also be running on the computer.

*This can also be set under Start Options. To do this, select the **Alarm Manager (AlmMng)** checkbox and set the time delay.*

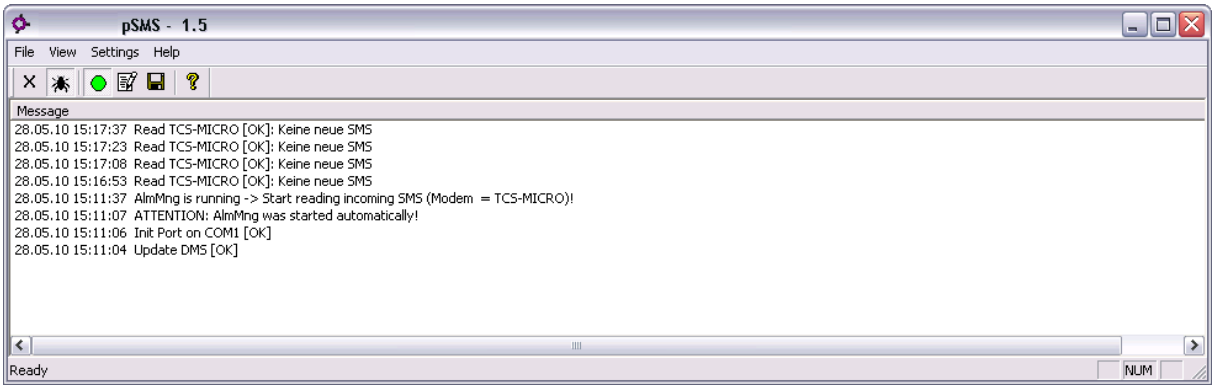
8.12.2 Control window of the SMS Alarm Logging (pSMS.exe) module

The SMS Alarm Logging window is opened as described in the following:

Double-click on the **pSMS icon**

or

click on the icon in the Windows taskbar with the right mouse button and choose the command "**Display**".



The control window is subdivided into 4 areas:

Title bar
Contains the name of the current module.

Menu bar
The menus are sorted according to basic functions.

Toolbar
Displays icons for some important commands and functions available in **pSMS**.

Debug Display (message)

If the (**Debug**) button is activated, all incoming SMSs, errors or other events are displayed here.

Status bar
Indicates the meaning of the button in the toolbar.

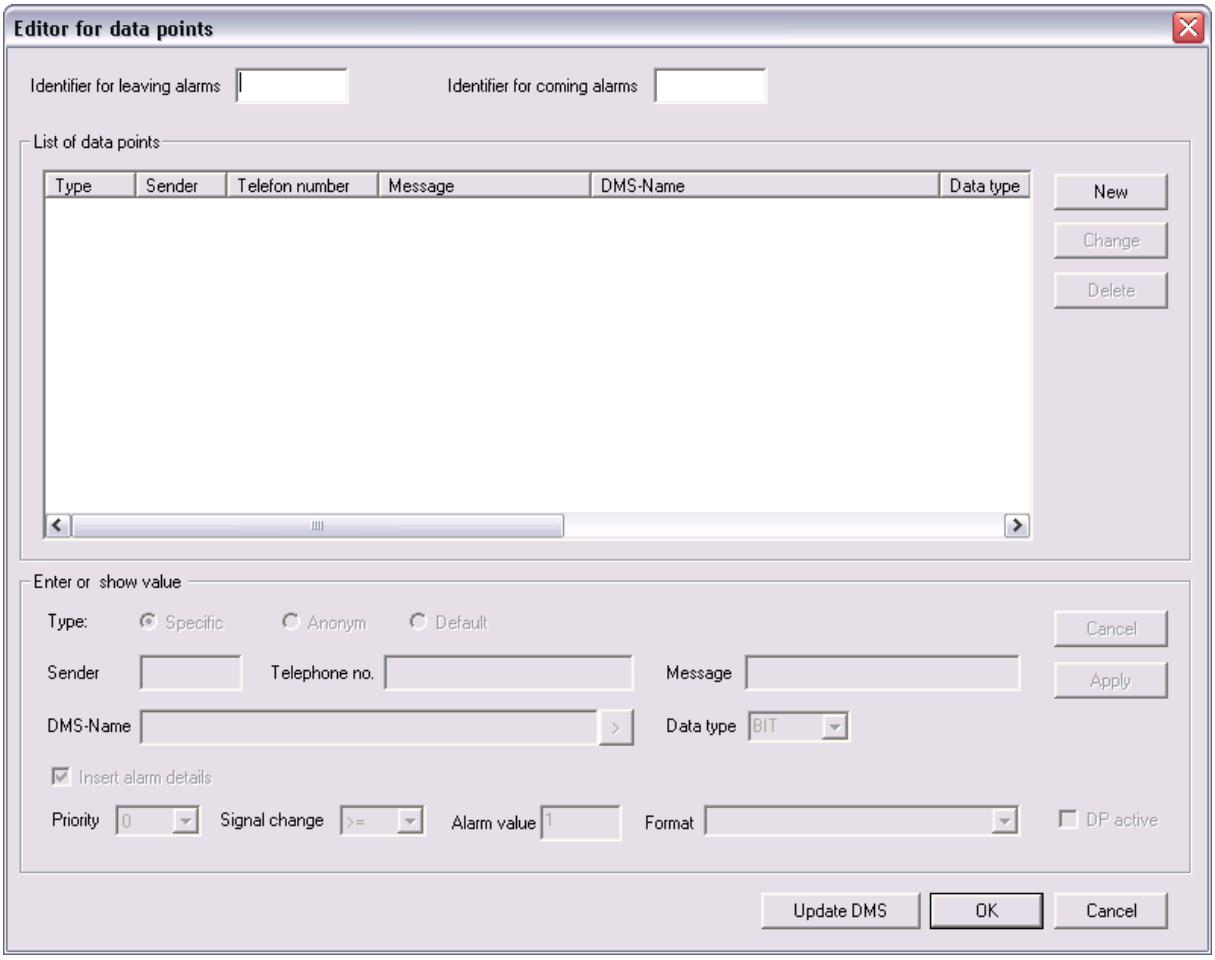
Before we address the individual menu items in greater detail, the following will be explained first:

- how alarms are defined (chapter Defining alarms).
- things to consider during input (chapter Input rules).
- presentation of an example (chapter Example).
- what happens when the alarm data is modified will be explained using the example (chapter Deleting alarms).
- how a sent SMS is evaluated by pSMS will be explained using the example (chapter

Evaluating an SMS).

8.12.3 Defining alarms

pSMS provides an editor for defining alarms. This is opened in the menu via **Settings > Editor for Alarms** or by clicking on the icon  :



To create a new entry:

- Click on the **New button.**
  The fields in the input area and value displays are now enabled.
- Fill in the input fields (the meaning of the fields is explained below).
- Finally, click on the **Adopt.** button.
  The values entered are adopted into the alarm list.

The values in the list can now be changed or deleted. The actions **Change** and **Delete** always refer to the active line (highlighted in blue). In order to activate a new line, click on the appropriate line in the alarm list with the left mouse button. If a line is already active, you can also scroll through the list with the arrow keys.

Explanation of the fields of the current window:

Identifier for Outgoing Alarms
Identifies an outgoing alarm - it is best to enter a number here. If the identifier is present in the sent SMS text, it is an outgoing alarm (for more information see Message input field).

Identifier for Incoming Alarms
Identifies an incoming alarm - it is best to enter a number here. If the identifier is present in the sent SMS text, it is an outgoing alarm (for more information see Message input field).

Alarm Types
The test criteria to be used on receipt of an SMS differentiate between 3 types of alarm (for more information see chapter Evaluating SMS):

Alarm-Type Specific
If this radio button has been selected, the telephone number of the sender (SMS) and the SMS text is checked. If both are found in the alarm list, the value of the indicated data point is changed.

Alarm-Type Anonymous
Only the SMS text is checked in this case. If this is found in the alarm list under anonymous alarms, the value of the indicated data point is changed.
This means that the same alarm can be triggered by various PLCs on the local DMS.

Since no telephone number is necessary, the field Telephone Number as well as the field PLC Number are locked.

Alarm-Type Default
The default alarm is used when a message cannot be assigned to either specific or an anonymous alarm, which is set to the data point entered under DMS name in the DMS.

If this radio button is selected, the input fields PLC Number and Telephone Number are locked.

PLC Number
Identifies a PLC. The input field is only enabled if the radio button Specific is selected.

The numbers must be assigned successively. This means beginning with 1. For example, if the data for 6 PLCs (PLC number 1 - 6) has already been entered and the PLC number 10 is selected for a new PLC, the PLC number is automatically set to 7. If an additional data point is entered for the same PLC, the already entered telephone number is automatically adopted on the basis of the PLC number. In this case the field Telephone Number is locked. If the telephone number is changed, all telephone numbers of the same PLC are adjusted.

Data Point Active
When activated, the monitoring of the data point is switched on (default = activated).

Telephone Number
Telephone number of the PLC, enables the sent alarm to be assigned to the respective data point.

DMS name
Indicates which data point should change the value.

Message
Along with the telephone number, this field allows alarms sent via SMS to be specifically assigned to one data point. The message can be structured in any manner. The structure of the message, a concrete example and what happens with the corresponding data point when an SMS is received (for more information see chapter Evaluating an SMS), can be obtained from the following table. For identification of outgoing alarms, a value of 990 is assigned, and a value of 991 is assigned for incoming alarms.

Structure	Example	Received SMS text	SMS text contains identifier	Value change of the corresponding data point
<SMS text outgoing alarm>/<SMS text incoming alarm>	PRIO0/ PRIO1	PRIO0 PRIO1	no no	Value is set to 0 (outgoing alarm). Value is set to 1 (incoming alarm).
<SMS text>	PRIO	PRIO PRIO990 PRIO991	no yes yes	Value is set to 1 and 30 seconds later set back to 0 (toggling) Value is set to 0 (outgoing alarm). Value is set to 1 (incoming alarm)

Since the message sent is assigned to the data point to be set, the SMS text sent from the PLC for an alarm must be unique (for more information see chapter Input rules). Capital and lower-case letters are differentiated between.

Format
Adopted by selecting a "predefined format" (see chapter Format definitions (PRTFormat.exe), Default = #c #-20N #V^NAME #T).

Priority
of the alarm (1 is the highest priority, default = 0).

Signal transmission
Defines whether an alarm should be transmitted if the limit value is exceeded or not reached (default = ">=").

Limit value
Triggers an alarm when the indicated value is exceeded/not reached.

Explanation of the buttons of the current window

- **In the List of Alarms area:**

  <New>
  Enables the input of alarm data.

  <Change>
  Already entered values can be changed.

  <Delete>
  Deletes the active line. In the process the PLC numbers are automatically adjusted, meaning that if all entries for a PLC are deleted, which reduces the PLC number of the next PLC by 1.

- **In the Input and Value Display area:**

  <Cancel>
  A newly created entry is not adopted into the list.

  <Adopt>
  A newly created entry is adopted into the list. In the process, a check is performed to see whether the newly entered alarm conflicts with alarms already present in the list. A corresponding error message is displayed (see chapter Input rules).

- **Bottom pane:**

  <Update DMS>
  Adopts the values into the DMS. In the process the Alarm Manager is automatically updated, meaning that the alarm data points are registered. The alarm data is filed in the current project directory (subfolder \cfg) in the file pSMS.cfg.

  <OK>
  The dialog is closed. The alarm data is filed in the current project directory (subfolder \cfg) in the file pSMS.cfg.

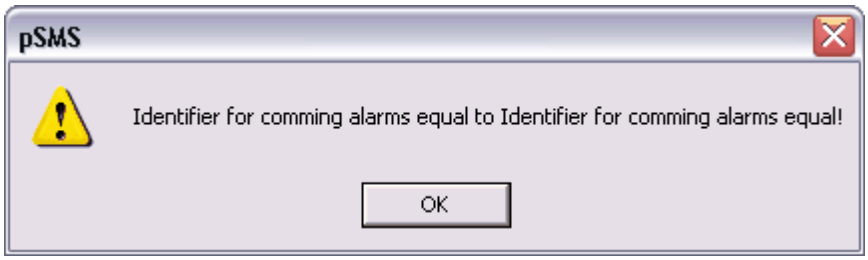
  <Cancel>
  Closes the dialog without saving the alarm data.

8.12.4 Input rules

The following rules apply when entering a new alarm:

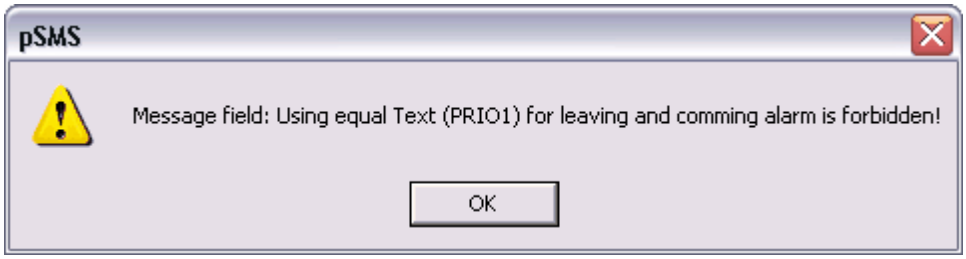
- All fields must be filled, otherwise the following error message is issued:

- There must be text before and after the forward slash in the message input field (e.g. PRIO0/PRIO1). For example, entering /PRIO1 would lead to the following error message:



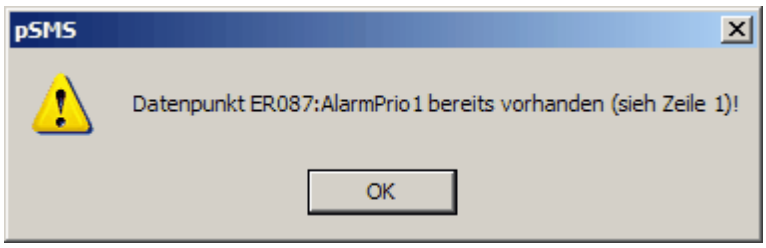
- The text for outgoing alarms must be differentiated from texts for incoming alarms in the message input field (e.g. PRIO0/PRIO1).

  SMS texts for incoming and outgoing alarms must be unique. For example, if the entry "PRIO1/PRIO1" is permitted, pSMS cannot differentiate on receiving an SMS with the text "PRIO1" whether it is an incoming or outgoing alarm. Therefore this entry leads to the following error message:

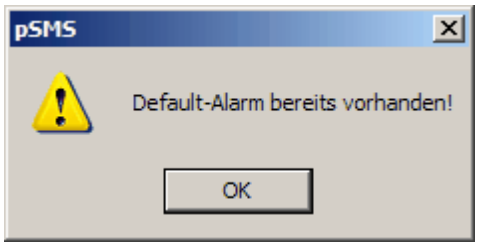


Capital and lower-case letters are differentiated between. It is therefore acceptable to enter "prio1/PRIO1".

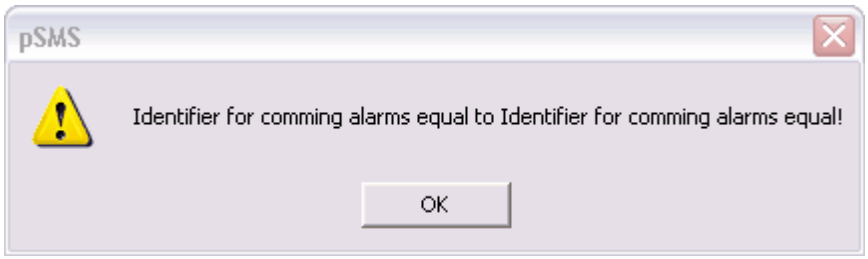
- The DMS name that is entered may not be present in the alarm list, otherwise the following error message is displayed:



- A default alarm may only be defined once. If it is already present in the list, the following error appears:

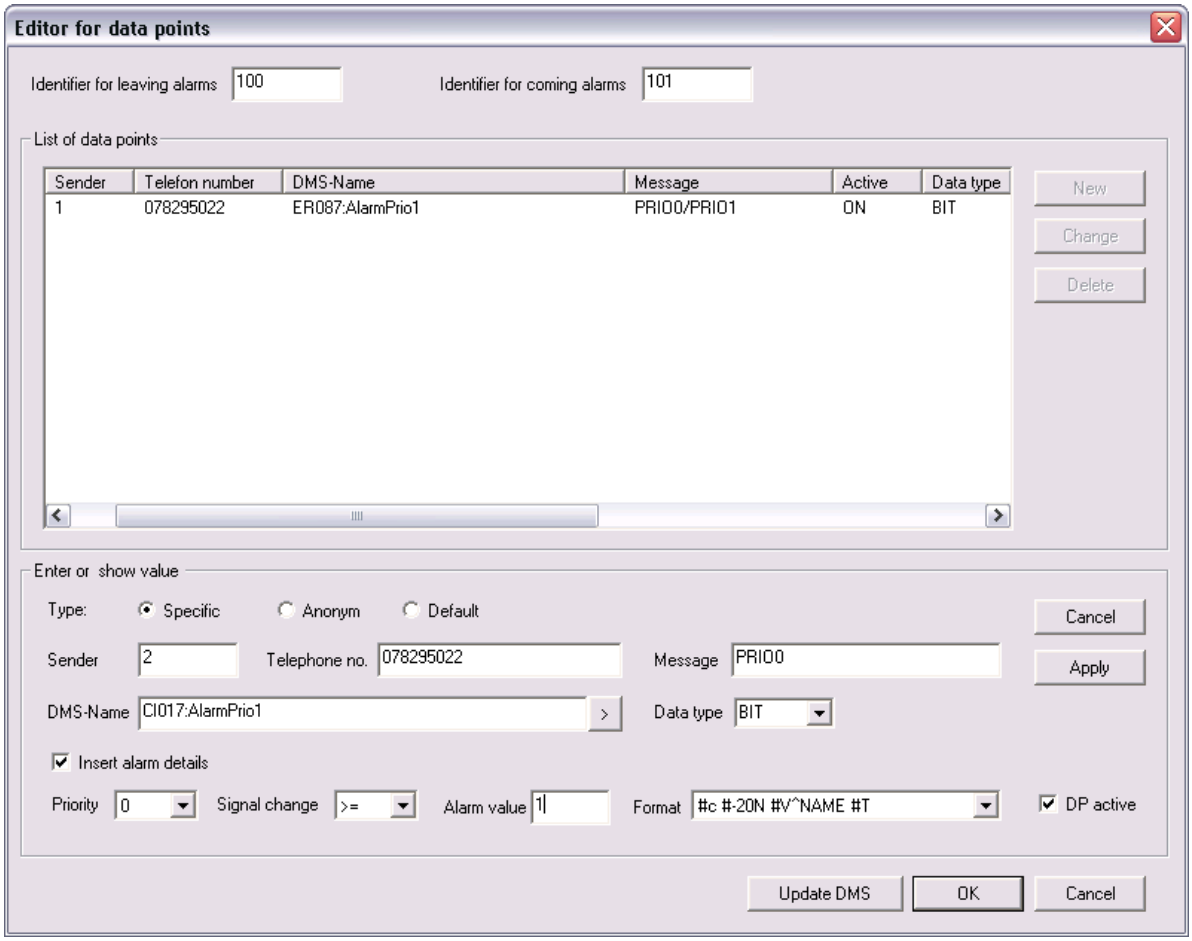


- The identifier for incoming and outgoing alarms must be unique. If both identifiers are the same it leads the the following error report:

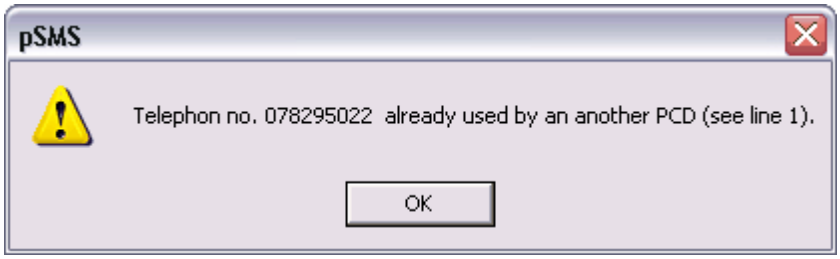


- The telephone number that is entered may not already be in use by another PLC.

  Each PLC is identified by a unique telephone number. This may not be assigned twice to different PLCs. For example, entering "078295022", as shown here:



leads to the following error:

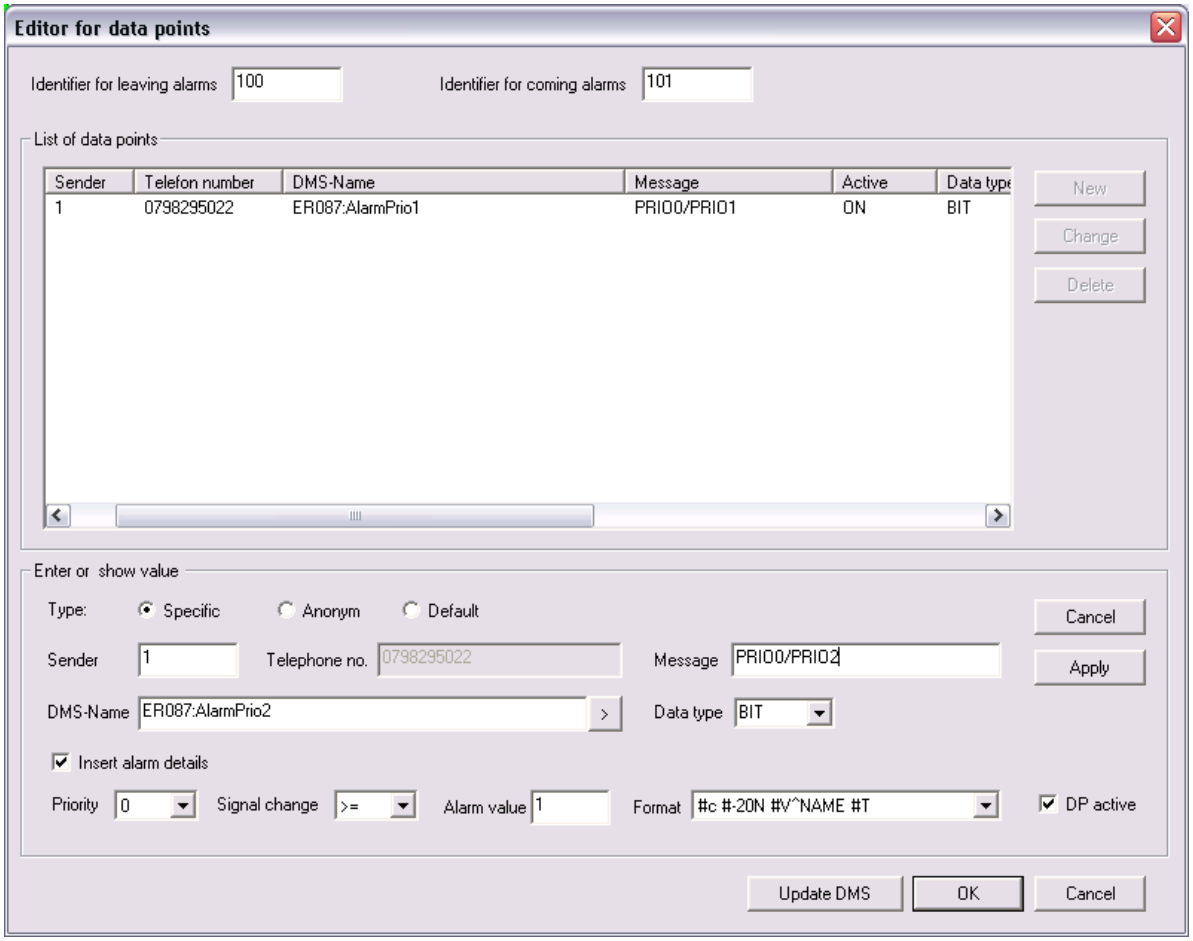


Capital and lower-case letters are differentiated between. It is therefore acceptable to enter "prio1".

- The message that is entered may not already be present in the list based on the same PLC.

As already mentioned, the SMS texts that are sent from the same PLC for different alarms must be unique. Therefore it must be ensured that the same SMS text (e.g. Prio0 for outgoing alarms) is not sent for two different alarms, because pSMS will not be able to uniquely assign a data point to the message in this case.

For example, the entry of "PRIO0/PRIO2", as shown here:

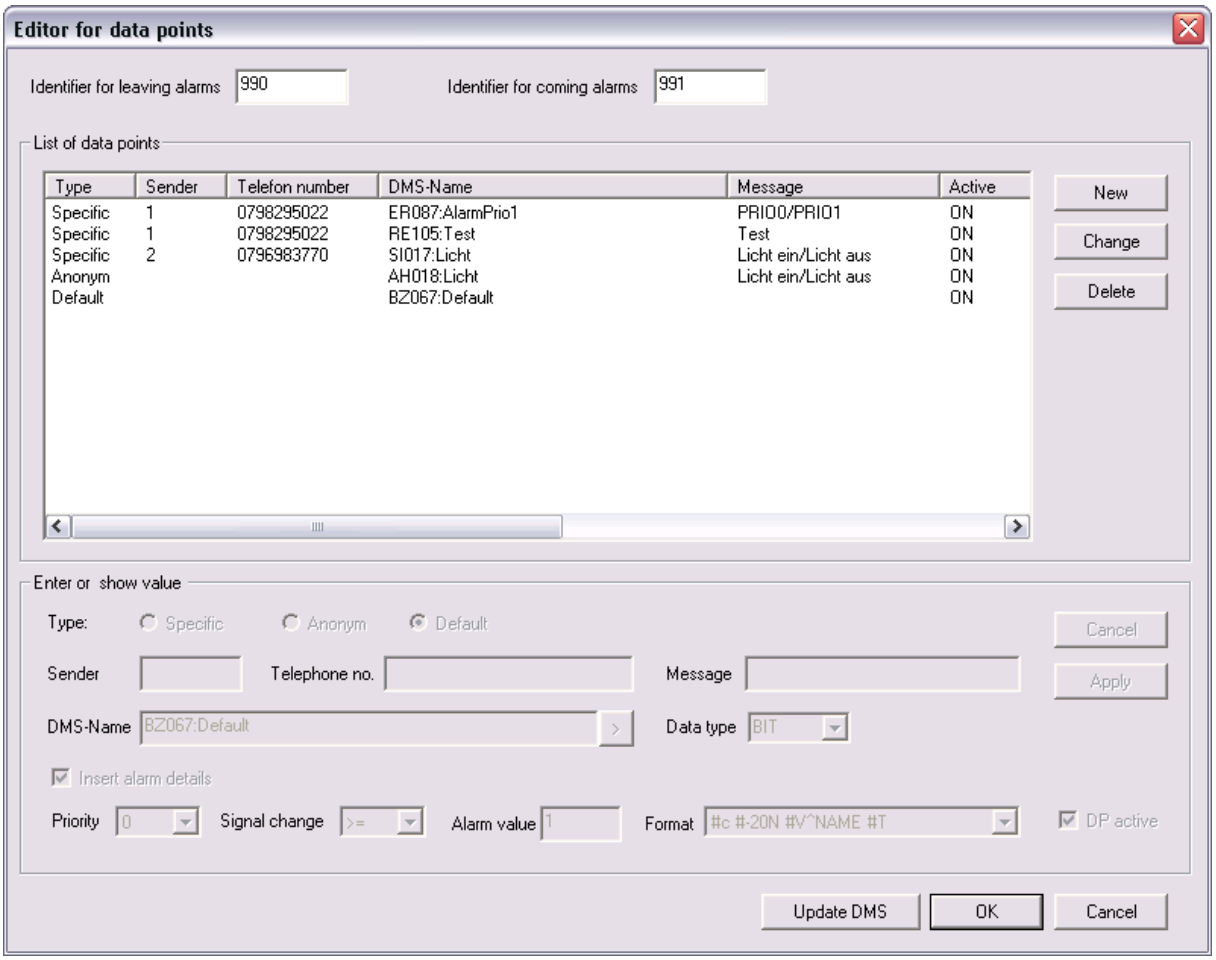


leads to the following error:

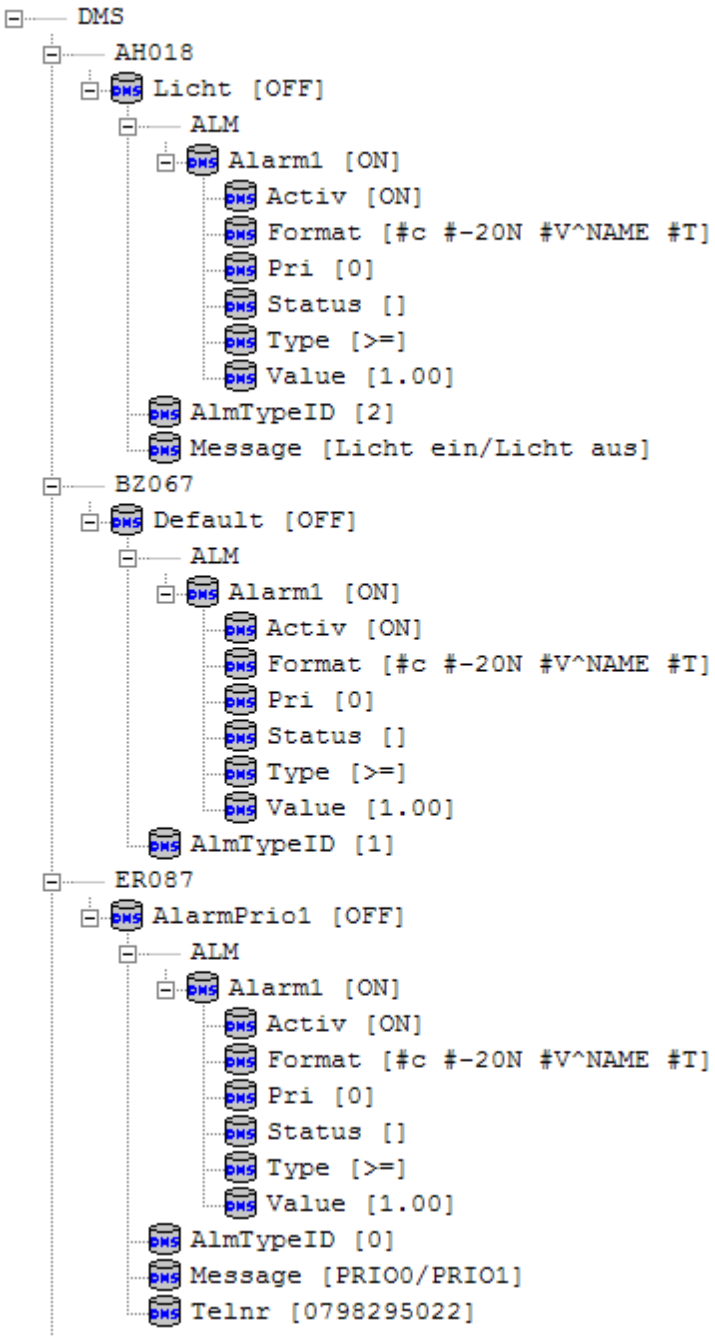
If this message were be permitted, pSMS would not be able to decide whether the value of the data point "ER087:AlarmPrio1" or the data point "ER087:AlarmPrio2" should be changed on receipt of an SMS with the text "PRIO0".

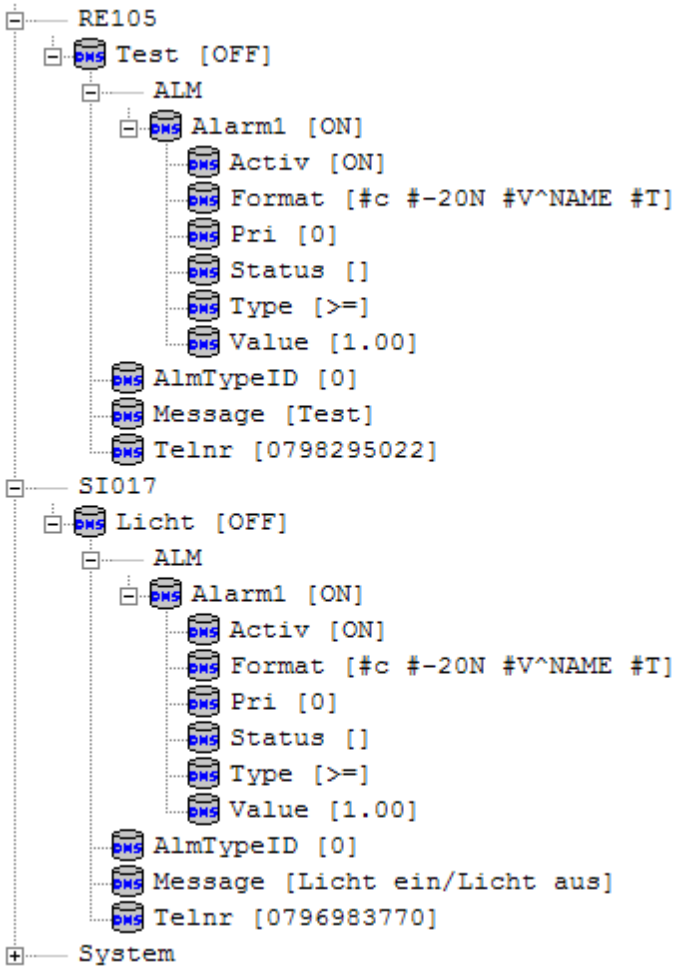
8.12.5 Example

The following list was created (Signal transmission >=, Limit value = 1). This example is drawn on in the chapters Deleting alarms and Evaluating an SMS.



When clicking on the **Update DMS** button, the following DMS tree is generated:





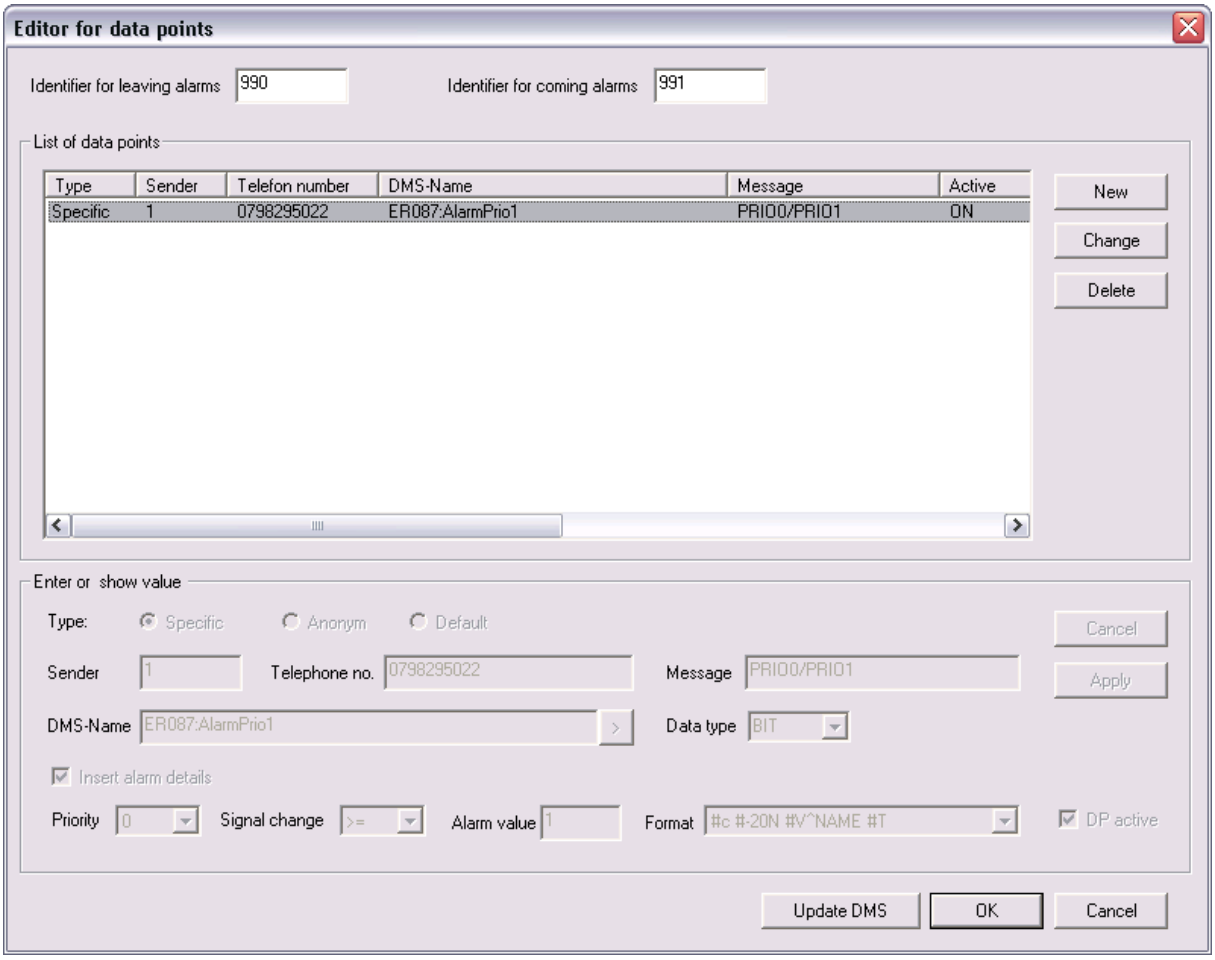
The alarm data is saved in the current project directory (subfolder \cfg) in the file pSMS.cfg (section [PLC1], [PLC2], [Anonymous], [Default]):

```
[Settings]
Port=COM1
Timer=1
Modem=Proton GTM
AlmOutg=990
AlmIncom=991
MngWatchDogActive=0
Modem Comm OK=
[PLC1]
TelNo=0798295022
DMSName1=ER087:AlarmPrio1
Message1=PRIO0/PRIO1
Active1=1
Prio1=0
Type1=>=
Value1=1.00
Format1=#c #-20N #V^NAME #T
DMSName2=RE105:Test
Message2=Test
Active2=1
Prio2=0
```

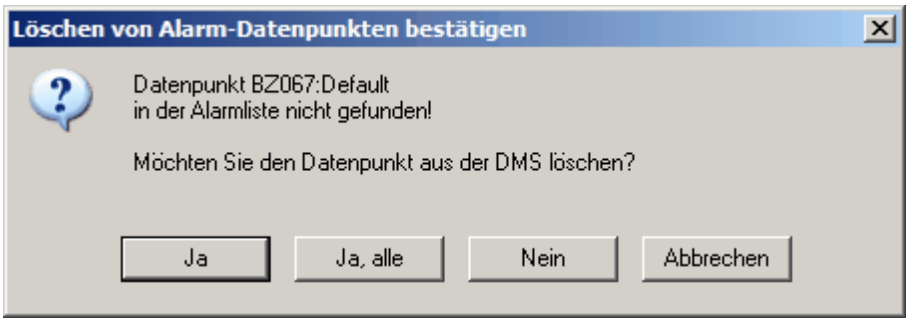
```
Type2=>=
Value2=1.00
Format2=#c #-20N #V^NAME #T
[PLC2]
TelNo=0796983770
DMSName1=SI017:Light
Message1=Light on/Light off
Active1=1
Prio1=0
Type1=>=
Value1=1.00
Format1=#c #-20N #V^NAME #T
[Anonymous]
DMSName1=AH018:Light
Message1=Light on/Light off
Active1=1
Prio1=0
Type1=>=
Value1=1.00
Format1=#c #-20N #V^NAME #T
[Default]
DMSName=BZ067:Default
Active=1
Prio=0
Type=>=
Value=1.00
Format=#c #-20N #V^NAME #T
```

8.12.6 Deleting alarms

Three alarms were deleted from the alarm list of the preceding example by clicking on the line in question in the alarm list with the left mouse button and then pressing the **Delete** button:



When clicking on the **Update DMS** button, a check is now performed to see whether alarm data points are present in the DMS, which are not entered in the list.If this is the case, it is indicated in a dialog:



It is then possible to delete it from the DMS:

<Yes>
Deletes the displayed data point from the DMS.

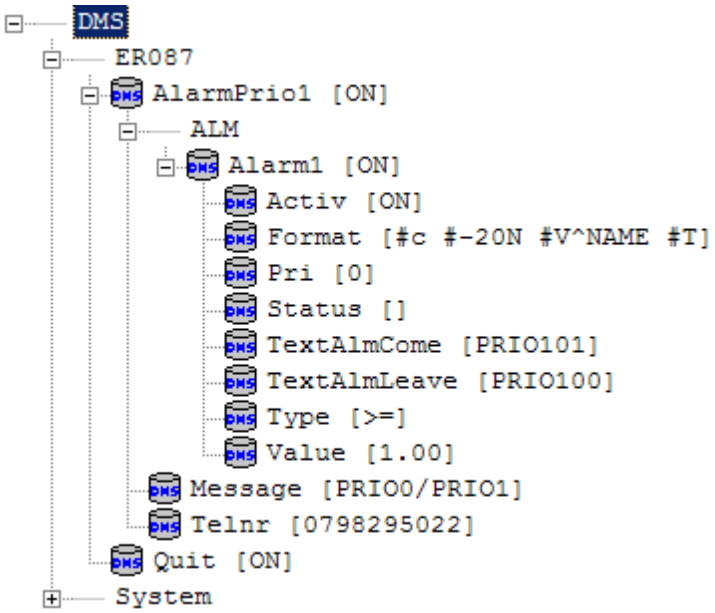
<Yes to All>
Deletes all data points that are not found in the alarm list from the DMS without having to confirm each time.

<No>
The displayed data point is not deleted from the DMS.

<Cancel>
The deletion of the data point is cancelled.

The deleted data points are displayed in the Debug window of pSMS.

Assuming you have clicked **Yes to All**, the view after deleting the DMS tree structure should look as follows:



The file pSMS in the current project directory (subfolder \cfg) is modified correspondingly:

```
[Settings]
Port=COM1
Timer=1
Modem=Proton GTM
AlmOutg=100
AlmIncom=101
MngWatchDogActive=0
[PLC1]
TelNo=0798295022
DMSName1=ER087:AlarmPrio1
Message1=PRIO0/PRIO1
Active1=1
Prio1=0
Type1=>=
Value1=1.00
Format1=#c #-20N #V^NAME #T
```

8.12.7 Evaluating an SMS

The modem connected to the serial interface receives the SMSes sent from the PLC. They are read and evaluated by pSMS.

pSMS currently supports 4 modems:

- **TCS MICRO**

  If this modem is used, pSMS queries the modem at regular intervals as to whether an SMS has been received. If a message has been received, it is read by pSMS and the evaluated. An SMS requested from this modem appears as follows:

  Hello -> -> SMS IN Date: 2008/09/25,09:49:28 From: 0798295022 Msg: PRIO0----------

- **Proton GTM**

  This modem sends the string #SMS to the serial port on receipt of an SMS. This is continuously monitored by pSMS. On receipt of #SMS, pSMS retrieves the message and then evaluates it. An SMS requested from this modem appears as follows:

  +CMGR: REC UNREAD,0798295022,, 08/09/25,14:01:05+08 PRIO0 OK

- **PhoenixContact GSM/GPRS**

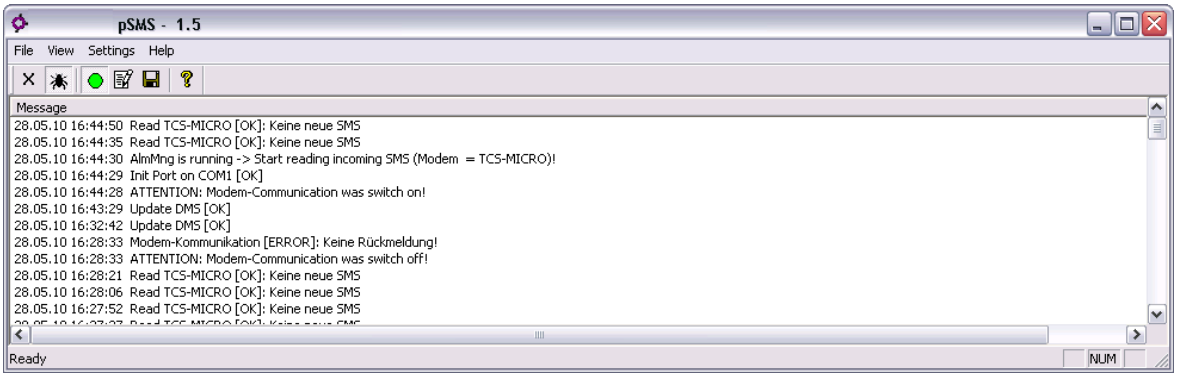
  On receipt of an SMS, this modem automatically forwards it to the serial port. Since pSMS continuously monitors the serial port, the SMS is immediately retrieved after its receipt by pSMS and then evaluated. An SMS forwarded from this modem appears as follows:

  +CMGR: REC UNREAD,0798295022,, 08/09/25,16:12:25+08 PRIO0 OK (the same as the Proton GTM modem)

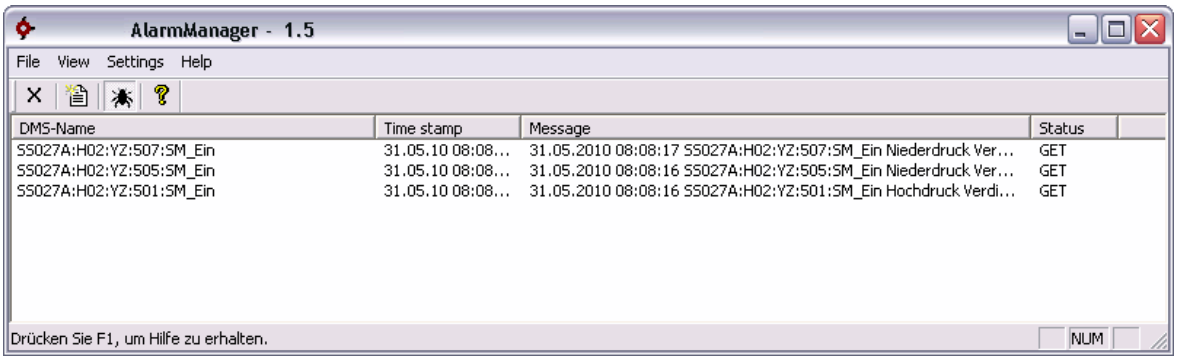
In all three cases the SMSes include the telephone number of the sender (0798395062) and the sent SMS text (PRIO0). Both pieces of information are used for assigning the alarm to a data point. The evaluation of a received SMS will be explained using 7 examples. The DMS configuration from the chapter Defining alarms is taken as a basis.

1. An SMS with the text "PRIO0" is sent by the PLC with the telephone number 0798295022.

On the basis of the telephone number and the SMS text, pSMS now knows the value to set for the data point "ER087:AlarmPrio1" (TelNo = 0798295022, Message = PRIO0/PRIO1). Since the sent SMS text is before the forward slash, it is an outgoing alarm. The value of the data point "ER087:AlarmPrio1", therefore, is set to 0 by pSMS:



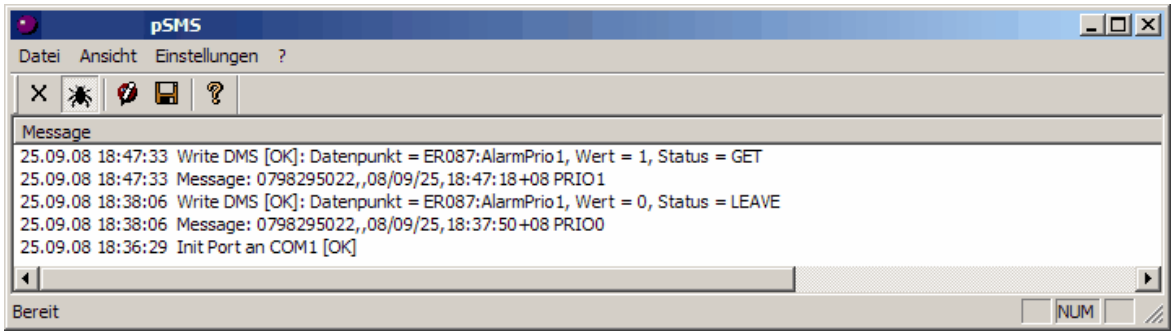
Since the data point is registered with the Alarm Manager, it registers the value change (outgoing alarm, status = OUTG):



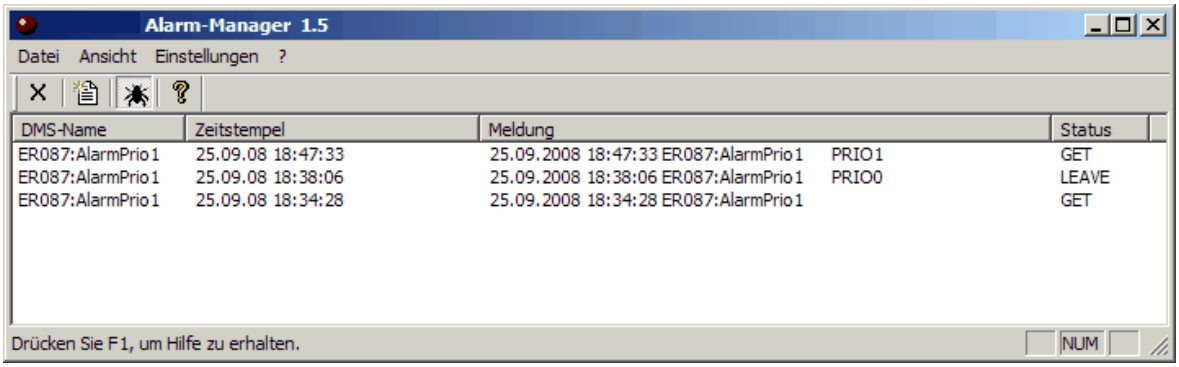
The alarm received by the modem is therefore forwarded to the Alarm Manager through the DMS.

2. An SMS with the text "PRIO1" is sent by the PLC with the telephone number 0798295022.

   Since PRIO1 is after the forward slash in this case (Message = PRIO0/PRIO1), it is an incoming alarm. Therefore the data point "ER087:AlarmPrio1" is now set to 1.

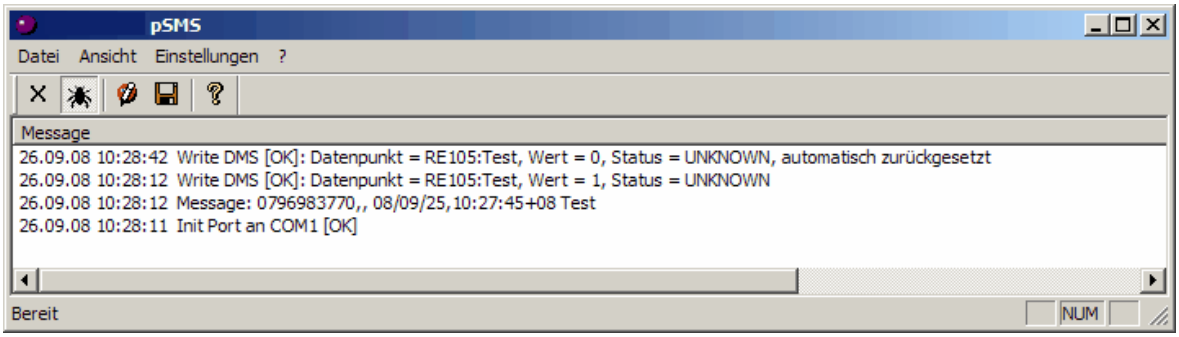


and the received alarm is displayed in Alarm Manager as incoming (Status = INCOM):

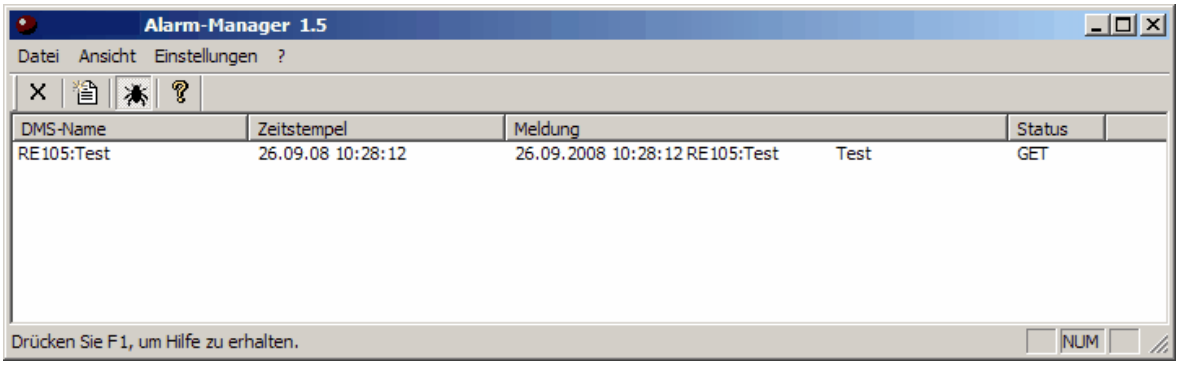


3. An SMS with the text "Test" is sent by the PLC with the telephone number 0798295022.

   When searching the DMS, pSMS finds that the telephone number and SMS text belong to the data point RE105:Test (TelNo = 0796983770, Message = Test). Since no assignment for an outgoing or incoming alarm can be made in this case, as indicated in the output window by the status UNKNOWN, the data point RE105:Test is set to 1. The value is reset 30 seconds later.

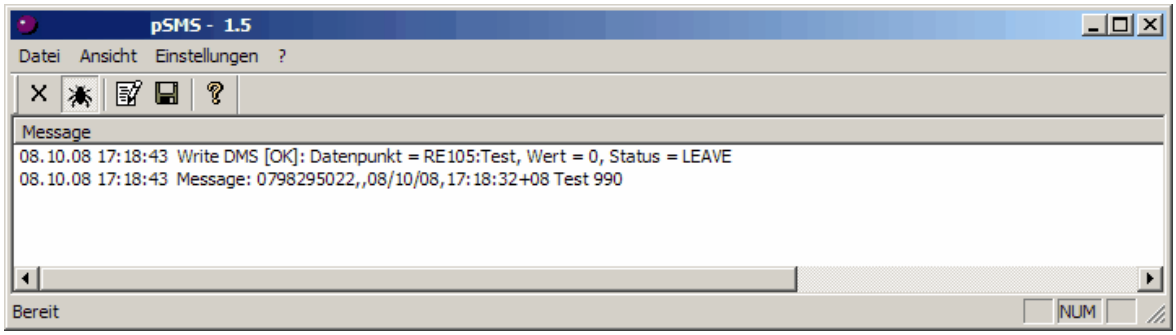


   Now the initial setting of the data point to 1 is registered by the Alarm Manager.

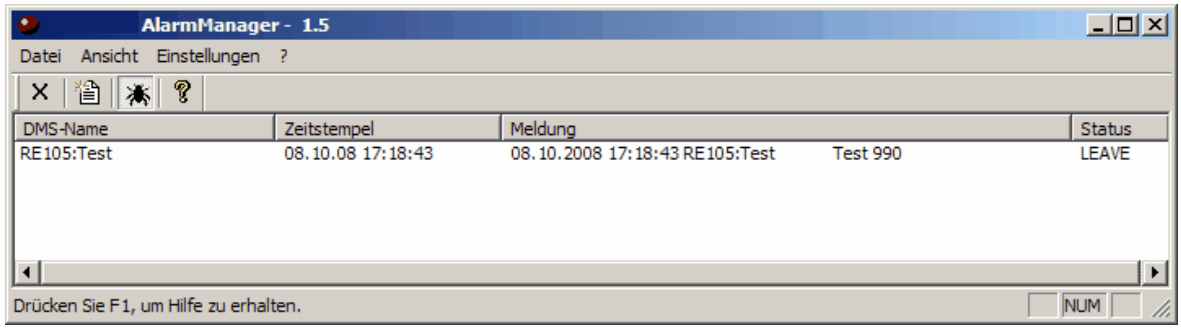


4. An SMS with the text "Test 990" is sent by the PLC with the telephone number 0798295022.

   On the basis of the telephone number and the SMS text, pSMS now knows that the data point in question is "RE105:Test" (TelNo = 0798295022, Message = Test). Since the SMS text includes the identifier for outgoing alarms (990), the value of the data point "RE105:Test" is set to 0 by pSMS:

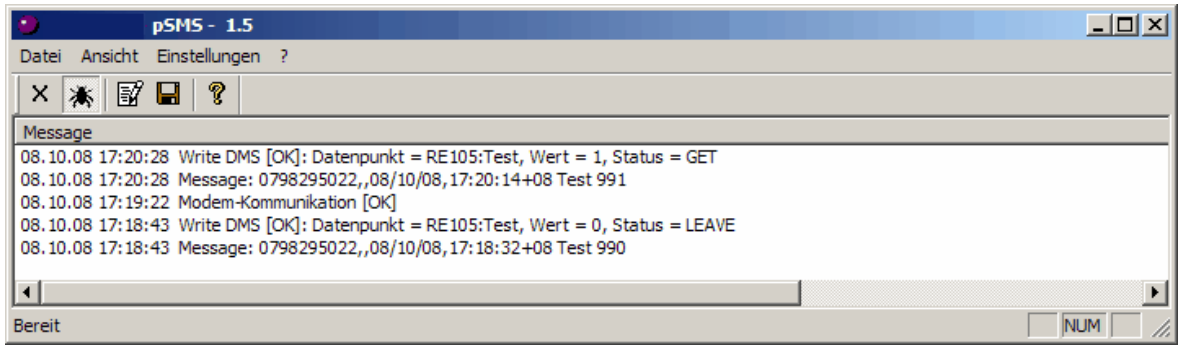


The alarm (Status = OUTG, outgoing) is displayed almost simultaneously by the Alarm Manager:

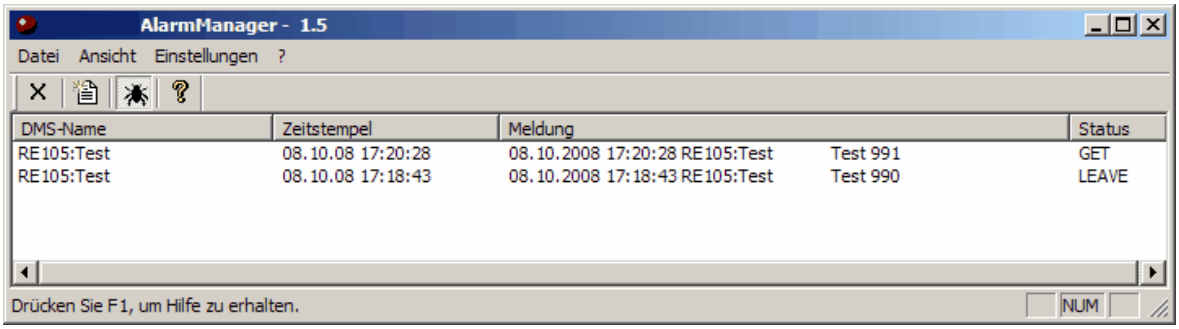


5. An SMS with the text "Test991" is sent by the PLC with the telephone number 0798295022.

The identifier 991 shows that it is an incoming alarm. Therefore the data point "RE105:Test" (TelNo = 0798295022, Message = Test) is set to 1 by pSMS:

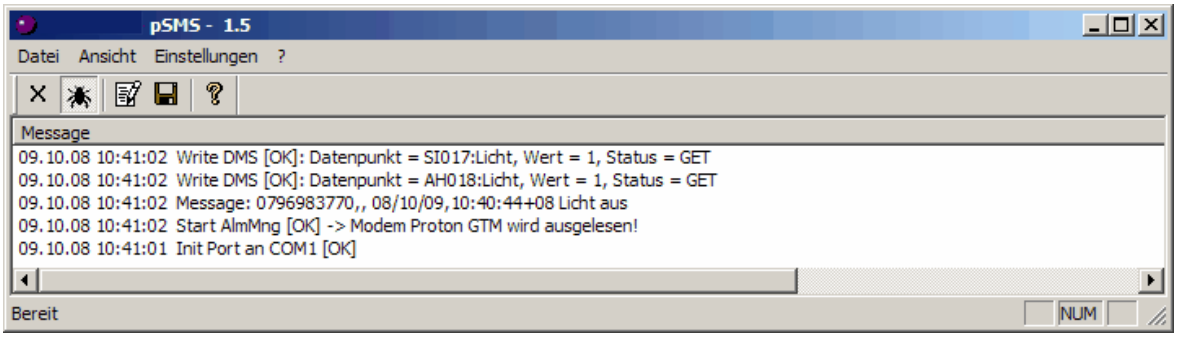


The alarm is consequently logged by the Alarm Manager as incoming (Status=INCOM):

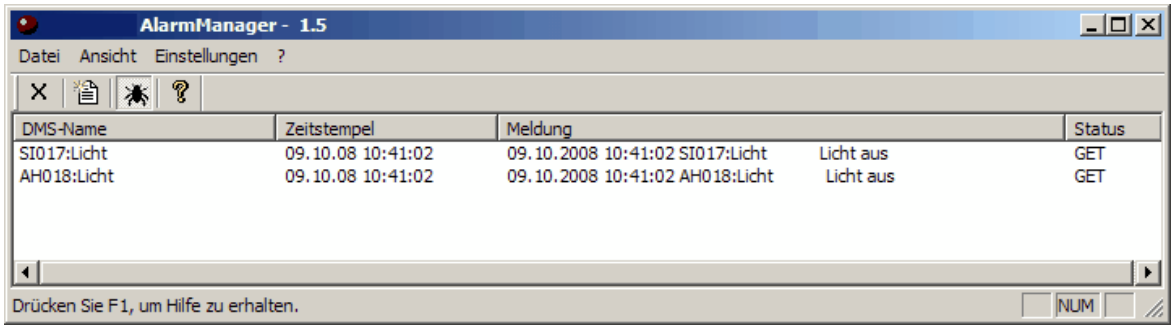


2. An SMS with the text "Light off" is sent by the PLC with the telephone number 07969833770.

   Since the sent SMS text matches both the messages of the data point SI017:Light (specific alarm, message = Light on/Light off) and the message of the data point AH018:Light (anonymous alarm, message = Light on/Light off), both data points are set to 1 (incoming alarm, because the text "Light off" is after the forward slash):

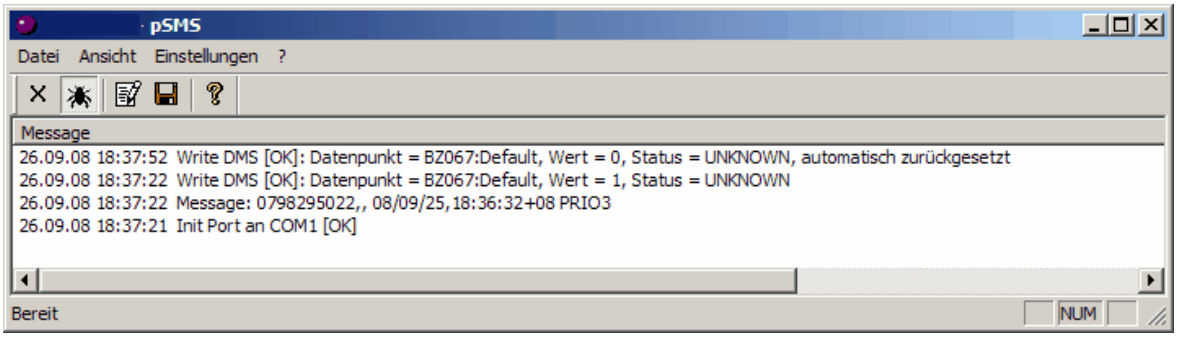


and the received alarm is displayed in Alarm Manager as incoming (Status = INCOM) for both data points:

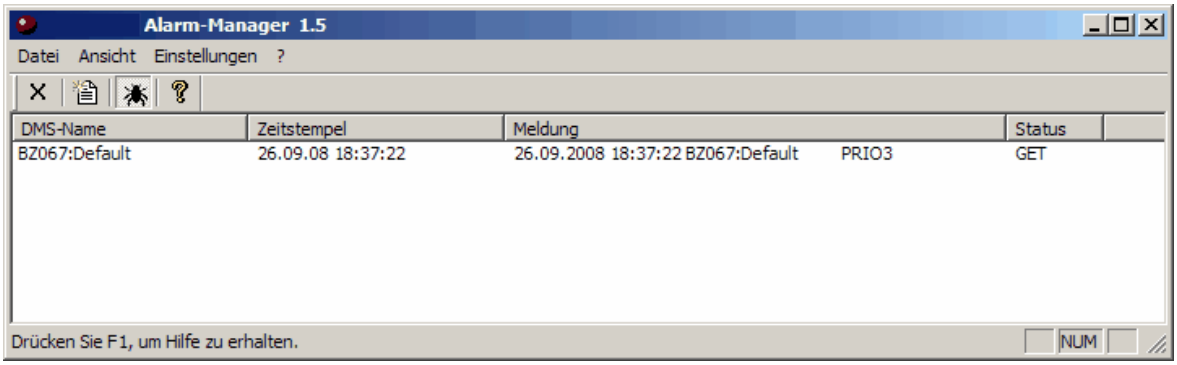


7. An SMS with the text "PRIO3" is sent by the PLC with the telephone number 0798295022.

   In this case the SMS text does not match the messages for the PLC. Therefore the default data point "BZ67:Default" is set to the value 1 and then reset 30 seconds later.



In this case the initial setting of the data point to 1 is also registered by the Alarm Manager.



8.12.8 Modem monitoring

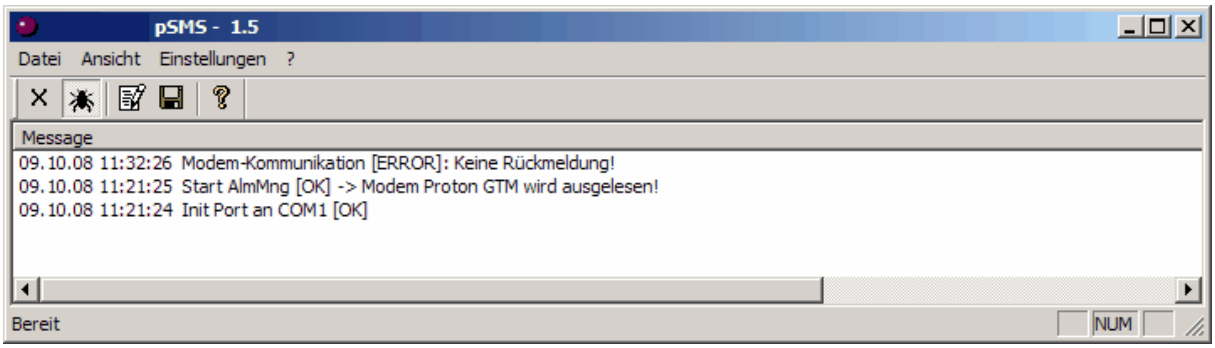
With the modems **Proton GTM** and **PhoenixContact GSM/GPRS**, pSMS checks every 10 minutes whether the modem is receiving.

The **Proton GTM Modem** sends a sign of life every 10 minutes. If this is absent, there is a fault.

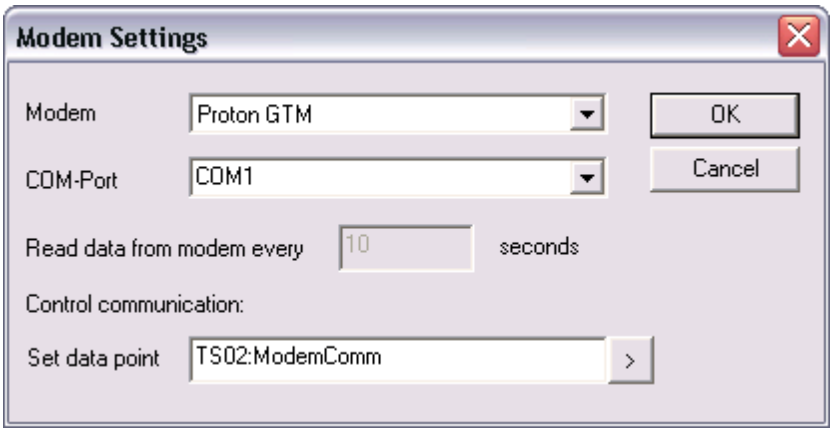
With **PhoenixContact GSM/GPRS** the network status is queried on startup of pSMS and every 10 minutes thereafter. If the modem does not answer or an error code is returned, there

is a communications fault.

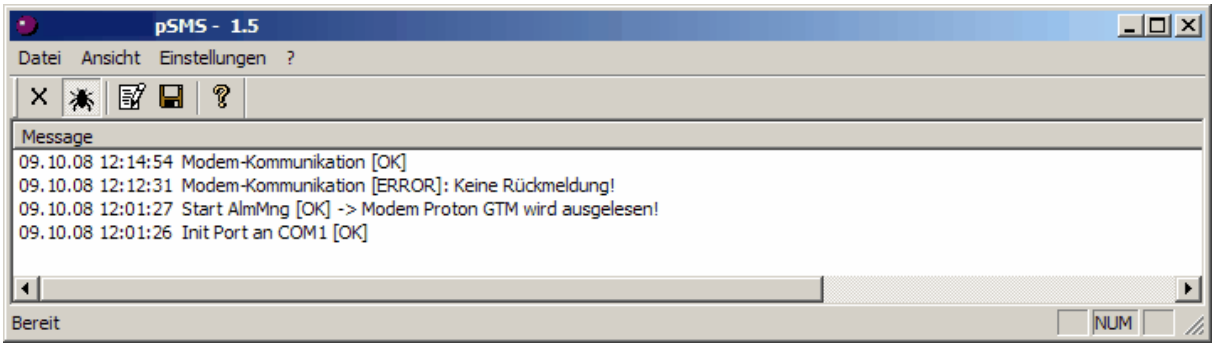
In case of a fault, a message is output in the Debug window:



and the data point (in this case TS02:ModemComm) entered in the Modem Configuration dialog (see the chapter Modem Settings) is set to 0.



If the fault is remedied, the data point is reset to 1 and a corresponding message is output in the Debug window:



Error messages for the Proton GTM Modem:

- **Modem communication [ERROR]: No response**
  The sign of life is absent.
  Cause: Modem disconnected from the power or RS-232 cable not connected.

- **Modem communication [ERROR]: ERROR +CPIN**
  The modem reports that the SIM card has not been inserted.

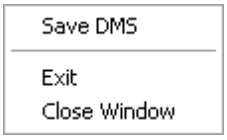
Error messages for the PhoenixContact GSM/GPRS Modem:

- **Modem communication [ERROR]: No response**
  The modem does not respond to a network status query.
  Cause: Modem disconnected from the power or RS-232 cable not connected.

- **Modem communication [ERROR]: Network status = 2**
  The modem is not logged on. It is searching for the network.
  Cause: Antenna not connected

- **Modem communication [ERROR]: Network status = 3**
  The modem is not logged on, because it cannot log on to the network.
  Cause: Antenna not connected

8.12.9 File menu



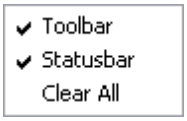
Save DMS or the **icon** 
Saves the DMS.

Exit
Exits the application.

Close window or **the icon** 
This menu item closes the control window of **pSMS**. This does not exit the program; it is only minimised to the taskbar (violet dot).

8.12.10 View menu

In the "**View**" menu, the **toolbar** and/or **status bar** can be switched on or off.



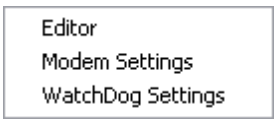
Toolbar
Switches the toolbar on or off.

Status bar
Switches the status bar on or off.

Delete All
This menu item deletes all messages in the Debug window. This has no influence on the functionality of **pSMS**.

8.12.11 Settings menu

In the Settings menu



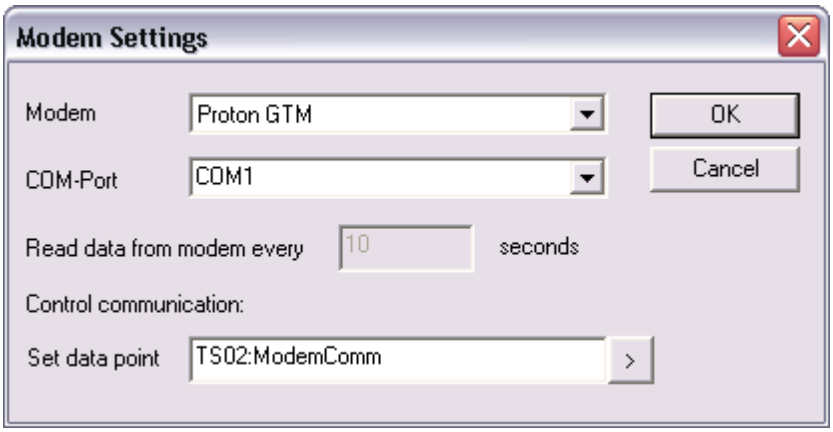
Editor for Alarms or the **icon**
Calls up the dialog for entering the alarm data (for more information see the chapter Defining alarms).

Modem Settings
Important modem settings can be made here.

WatchDog Settings
Here you can define whether pSMS or the DMS should be monitored.

8.12.11.1 Modem Settings

The following dialog window is opened by clicking on the menu item **Settings > Modem Settings**:



Modem
The connected modem must be selected from a list. The following modems are currently supported:
- TCS MICRO

- Proton GTM
- PhoenixContact GSM/GPRS

COM Port
Here the COM port to which the modem is connected must be selected from a list containing all available serial interfaces.

Read Modem
Is only enabled if the Proton GTM mode is selected.pSMS must actively retrieve the SMSes from the modem with this model. The time intervals for this to take place can be indicated here.

On Communications Error
The data point that should be set to 0 in case of a communications error can be indicated here (see the chapter Modem monitoring). Only the modems Proton GTM and PhoenixContact GSM/GPRS are monitored, not MICRO TCS. Therefore when this modem is selected this field is locked.

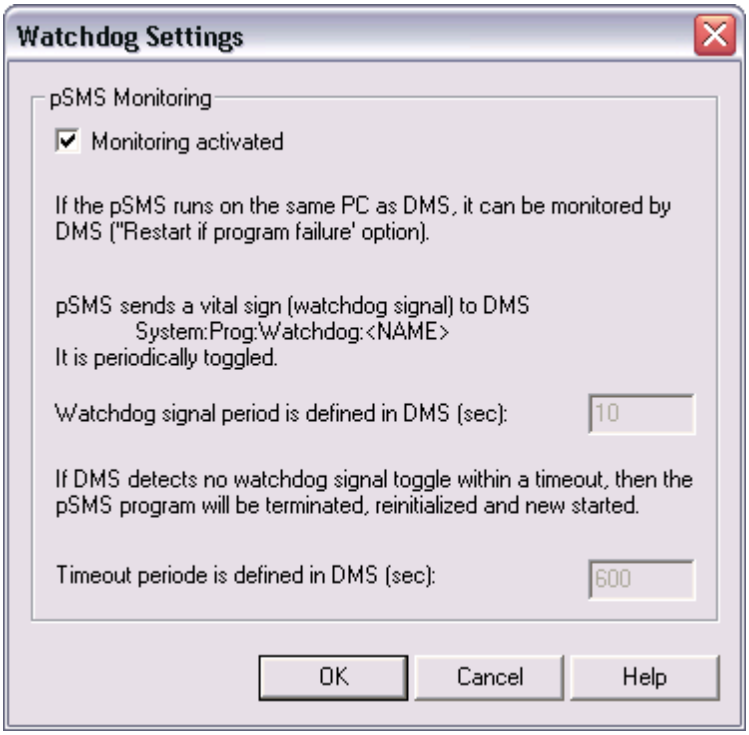
<OK>
Closes the dialog and writes the values to the file pSMS.cfg (section [Settings]):

```
[Settings]
Port=COM1
Timer=1
Modem=Proton GTM
Modem Comm OK=TS02:ModemComm
```

<Cancel>
Closes the dialog without adopting the settings.

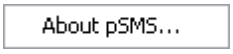
8.12.11.2 WatchDog Settings

Here it can be indicated whether **pSMS** should be monitored by the Data Management System (DMS). If the **Monitoring Activated** checkmark is set, **pSMS** and the **DMS** exchange a live bit.



8.12.12 "?" or Help menu

The "**?**" menu contains current information for the **pSMS** version.



The version must always be given in case of any support queries!

Should support queries become necessary, the internet forum is preferential to contact via telephone or e-mail. This offers two advantages: First, help is available more quickly; and secondly, other users also benefit from the entries.

8.13 SyncDMS

8.13.1 Introduction

SynchDMS is a program which enables the comparison of data points from and to different DMSs.
It can also be used to create a csv file with the requested values.

The data points can be freely configured, and different functions can be performed with the data points collected:

- *add*
- *min*

- *max*
- *set*

The user interface is available in the current language of the operating system, and the following languages are supported here:

- German
- English

8.13.2 Fault analysis

Meaning of the tray icon:

All DMS connections OK

Fault
- DMS not connected (also appears briefly at start until all DMSs are connected)
- Data point read from a DMS not available

Stopped (from tray menu - "running")

Any faults can have different causes:

- General fault of the connection to a DMS
- Incorrect configuration (e.g. incorrect connection parameter or attempted reading of an unavailable data point)
  For example, "registration errors" in the status window are displayed if a data point that is unavailable should be read
- Invalid licensing
  "Licence invalid" appears in the status window if the licence for SyncDMS resp. SyncDMS is not activated with file output.
  The corresponding licence must be set on every connected DMS on which data is written!
  SyncDMS can be licensed from the following DMS versions:
  - Version 1.6.8.102
  - Version 1.7.8.20

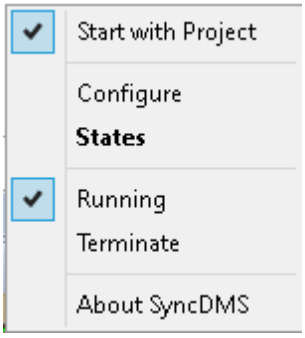
  Without a valid licence:
  - no data is written in DMS (however for test purposes the data monitor displays this)
  - no data is written in a file (however for test purposes the data monitor displays this)

The details can be viewed from the log.

The log file located is under C:\Visi.Plus\proj\PROJECTNAME\log, resp. can be viewed via menu item "Status" -> Log tab.

8.13.3 Operation

There are different menu entries available in the tray icon:

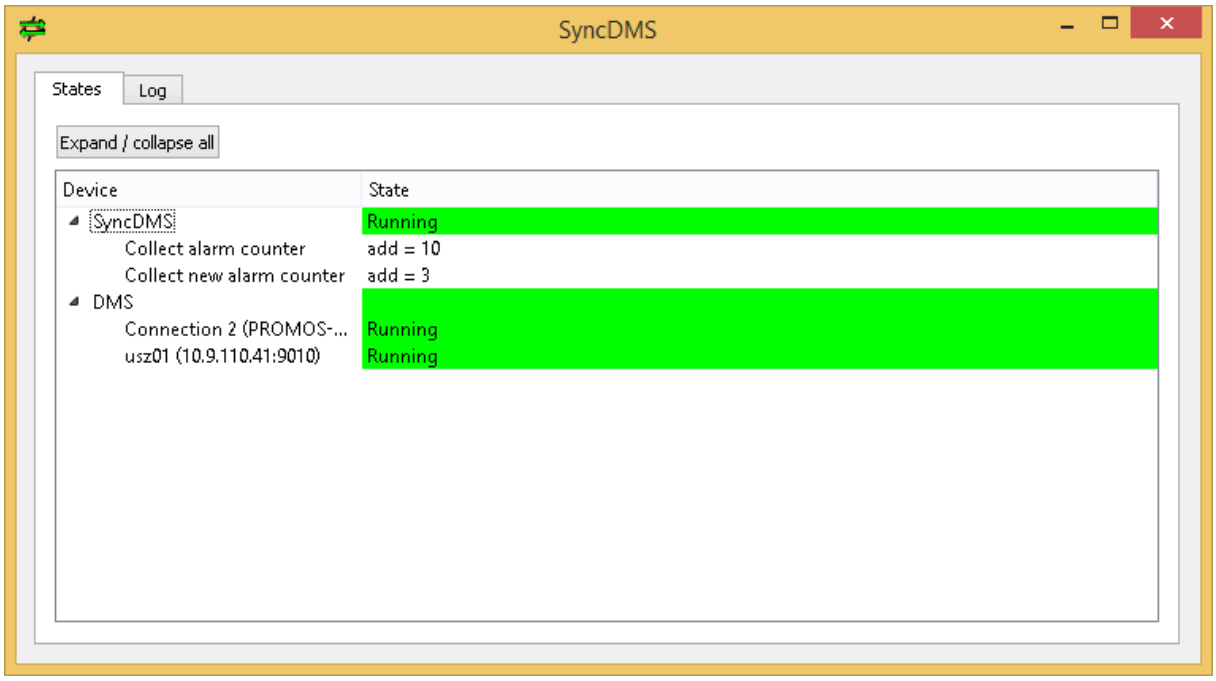


Double-click on the tray icon to display the current status.

- **"Start with Visi.Plus"**
  The entry can be influenced in the `"Start.cfg"` of the current project (check mark = entry is provided -> SyncDMS is automatically started with Visi.Plus).

- **"Configuration"**
  Call up the configurations window (see configuration).

- **"Status"**
  The current status of the connections and the currently calculated values of the set rules are displayed:

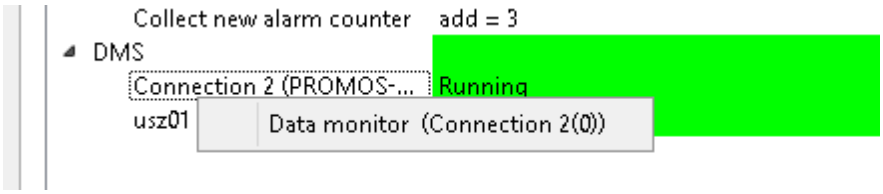


If a connection does not feature the "Running" status, a corresponding tray icon is displayed

as a warning: 
The window is continually updated.

The current log file can be viewed under the "Log" tab.

Monitor windows for monitoring can be opened by right-clicking:

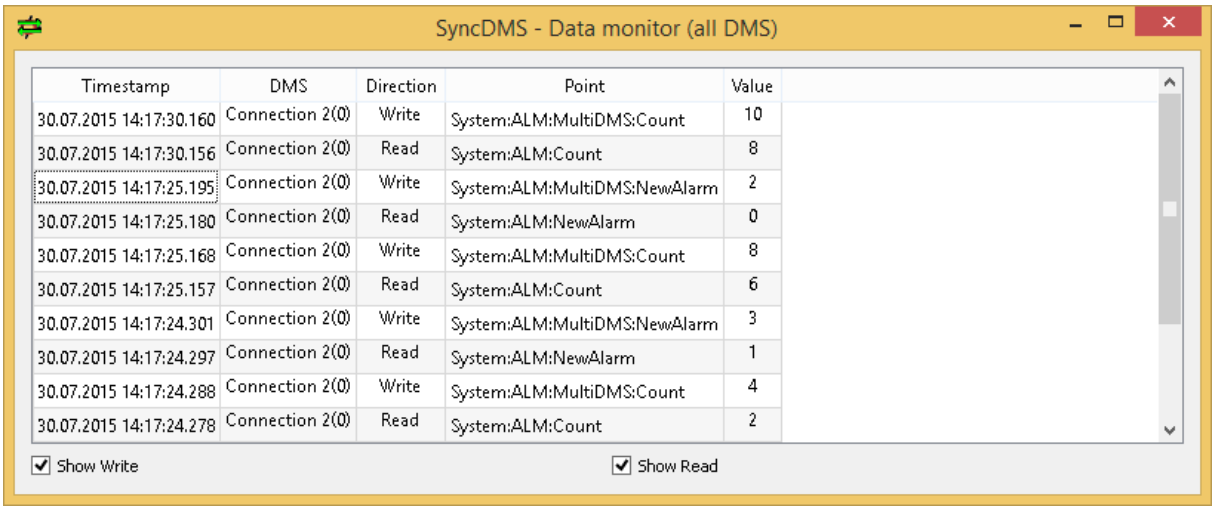


- On the main path "DMS": Monitor with all messages of all DMS

- On a path under DMS: Monitor with all messages of the relevant DMS

- On "SyncDMS": Monitor for the file outputs

Example for data monitor:



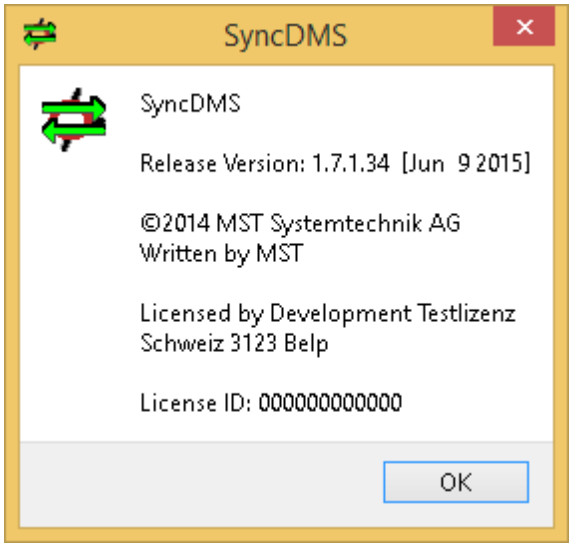
The last 200 entries are displayed for each window.

- **"Running"**
  The execution of the program can be influenced with this setting (check mark = values are processed).

  If processing is stopped, a corresponding tray icon appears:  .

- **"End"**
  With this, SyncDMS is ended.

- **"About SyncDMS"**
  Info screen about SyncDMS:



8.13.4 Special data points

SyncDMS places the following special data points at its local DMS:

- "System:Driver:SyncDMS:SyncDMS_Paused"

Bit ON / OFF, shows whether SyncDMS is running.

- "System:Driver:SyncDMS:SyncDMS_Error"

Bit ON / OFF, shows whether all connections are OK.

- "System:Driver:SyncDMS:DMSOkCount"

Number of DMS connections that are OK.

- "System:Driver:SyncDMS:DMSErrCount"

Number of DMS connections that are faulty.

- "System:Driver:SyncDMS:LastError"

Str, last error that occurred.

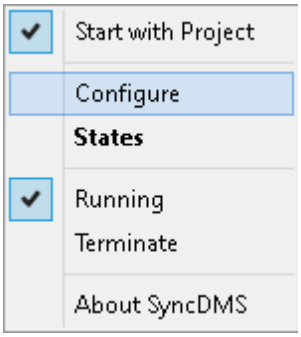
- "System:Driver:SyncDMS:Connections:xxx"

whereby an entry (xxx = connection identification) for each connection displays whether the corresponding connection is OK (Bit ON / OFF)

=> These special data points are only entered 30 seconds after the start of SyncDMS so that connections to individual DMSs that are "still" missing are not incorrectly interpreted as errors.

8.13.5 Configuration

The configuration window can be displayed via the menu of the tray icon (right-click):



There are 3 tab's in the configuration:

- DMS
  Here the DMS connections and the read/write activations of the individual rules for the connections are set.

- Rules
  The different rules are set here.

- File output
  To create csv files

Details about the individual configuration pages can be found in the following chapters.

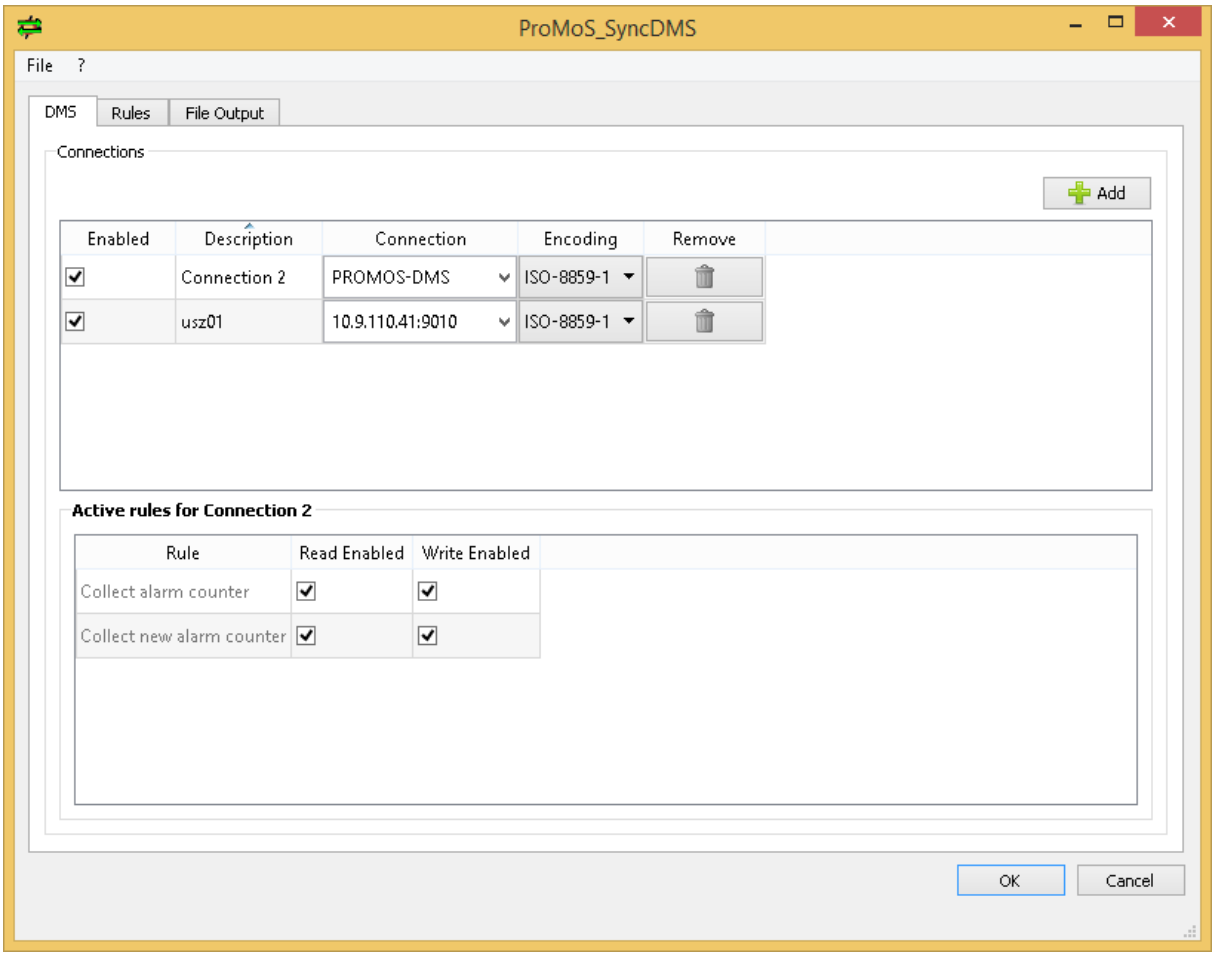
After initial startup, a standard configuration is provided as a template (collection of alarm counters).

The size and position of the configuration window can be changed as required and is maintained even after closing the window.

Note:
The licence information (menu item "?)" - "About") is called up from the currently selected connection.

8.13.5.1 DMS

(Select an item on the screen to be directed to the respective description)



Connections

- **add**
  Insert a new entry.

- **Activated**
  Activates a connection.

- **Description**
  Freely available text.

- **Connection**
  The following can be specified as a connection:
  - For pipe connections:
    "." or "local" or "PROMOS-DMS"
  - For Tcp connections:
    "ip:port" (e.g. "192.168.1.1:9010") or "name:port" (e.g. "WS096:9010")

- **Coding**
  Setting of the character set used by DMS (standard "ISO-8859-1").

- **Delete**
  Deletes the current entry

Active rules
Here individual rules for the ***connection selected above*** can be activated for the read or write procedure.

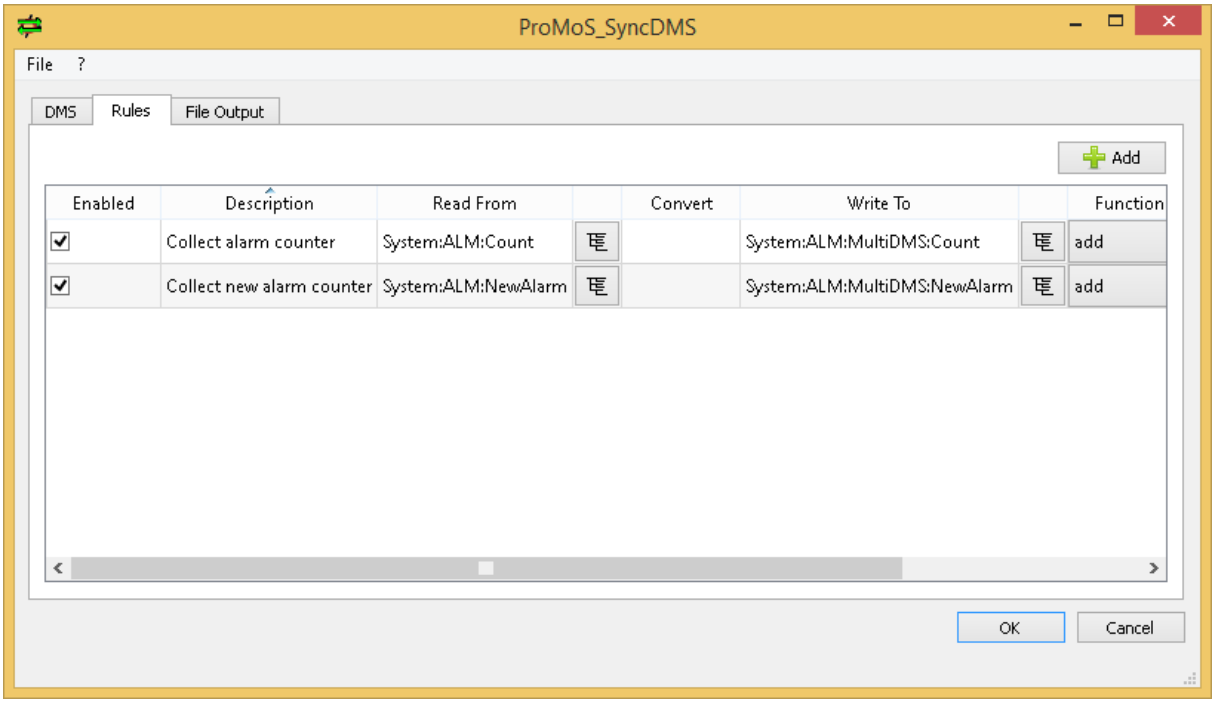
- **Rule**
  Here all recorded rules (see next chapter) are listed.

- **Read activated**
  Read activation of the corresponding rule of this connection.

- **Write activated**
  Write activation of the corresponding rule at this connection.

8.13.5.2 Rules

(Select an item on the screen to be directed to the respective description)



- **Add**
  Adds a new rule.

- **Activated**
  Activates the corresponding rule.

- **Description**

Freely available text.

- **Reading of**
  Data point in DMS that should be read.
  A data point can be selected with the switch on the right side of the input field.

- **Conversion**
  In this section, the read value can be converted. If this section remains empty, the read value is used without conversion.
  The name of the value to be calculated is "`Val`".
  All common operands pursuant to Ecma-262 can be used.
  Mathematical functions such as "`Math.round(Val)`" or "`Math.pow(Val ,2)`" pursuant to Ecma-262 are also available.
  Example: The read value should be multiplied by a factor of 22.5:
  "`Val * 22.5`"

- **Write after**
  Data point in DMS that should be written.
  A data point can be selected with the switch on the right side of the input field (if this already exists).
  A data point that is not available is automatically created by SyncDMS during operation if it does not exist.

- **Function**
  - add
  Add up (e.g. alarm counter)

  - min
  Determination of the smallest value

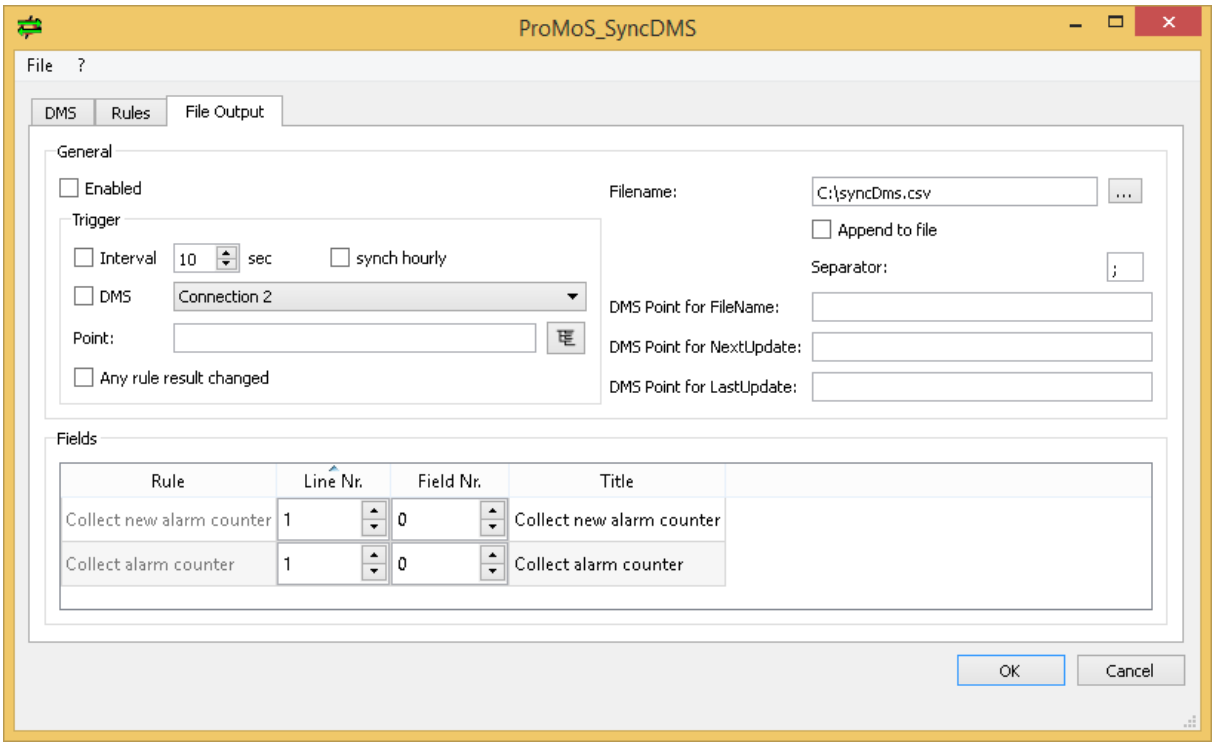
  - max
  Determination of the largest value

  - set
  Transfer (of the last updated value)
  e.g. for copying data points

- **Delete**
  Deletes the relevant rule.

8.13.5.3 File output

(Select an item on the screen to be directed to the respective description)



- **Activated**
  Activates the output.

- **Trigger - Interval**
  With this setting, the file is written every x seconds (in this example 10).
  Any maximum/minimum values are recalculated according to the output.
  With the option "Synch with hour", the time of triggering is synchronised with hours (e.g. 900sec (15min) -> for example, results in 10:00 / 10:15 / 10:30 / 10:45 / 11:00...)

- **Trigger - DMS**
  If this setting is active, the file is written when the specified DMS data points (of the specified DMS) are changed.
  Any maximum/minimum values are recalculated according to the output.

- **Trigger - Rule result changed**
  The file is written when a result from a rule is changed..

- **File name**
  The file to be output.
  Date and time specifications can be entered in the file name, the file name is interpreted as such when inverted commas are included in it.
  Fixed parts must be enclosed in inverted commas, all other parts are interpreted as follows:

Expression	Output

d	the day as number without a leading zero (1 to 31)
dd	the day as number with a leading zero (01 to 31)
ddd	the abbreviated localized day name (e.g. 'Mon' to 'Sun').
dddd	the long localized day name
M	the month as number without a leading zero (1-12)
MM	the month as number with a leading zero (01-12)
MMM	the abbreviated localized month name (e.g. 'Jan' to 'Dec')
MMMM	the long localized month name (e.g. 'January' to 'December')
yy	the year as two digit number (00-99)
yyyy	the year as four digit number
h	the hour without a leading zero (0 to 23 or 1 to 12 if AM/PM display)
hh	the hour with a leading zero (00 to 23 or 01 to 12 if AM/PM display)
m	the minute without a leading zero (0 to 59)
mm	the minute with a leading zero (00 to 59)
s	the second without a leading zero (0 to 59)
ss	the second with a leading zero (00 to 59)
z	the milliseconds without leading zeroes (0 to 999)
zzz	the milliseconds with leading zeroes (000 to 999)
AP	use AM/PM display. AP will be replaced by either "AM" or "PM".
ap	use am/pm display. ap will be replaced by either "am" or "pm".

Examples:

```
C:\FixerName.csv           -> C:\FixerName.csv
'C:\File'yyyyMMdd'.csv'    -> C:\File20130222.csv   (on 22/2/2013)
```

- "Continually add to file" -> thus the new values of an existing file are added.

- As a "separator", the character sequence to separate the fields can be specified. If a tabulator should be used, "`[TAB]`" (as text, 5 characters) must be specified.

The composed title line (configuration "Fields" - "Title") is always (and only once) entered as the first line in the file.

- **DMS point for file name**
  Here it can be specified where the file name should be saved in the DMS.
  A change to this data point in the DMS is considered by the running SyncDMS, i.e. the new file name is then used.
  Can remain empty (no monitoring/transmission).

- **DMS point for counter of next update**
  If the trigger interval is active, the remaining time until the next write procedure is continually transferred at this data point (in seconds).
  Can remain empty (no transmission).

- **DMS point for date/time of last update**
  For a write procedure, the current date/time ("DD.MM.YYYY HH:MM:SS") is transferred as a character string at this data point.
  Can remain empty (no transmission).

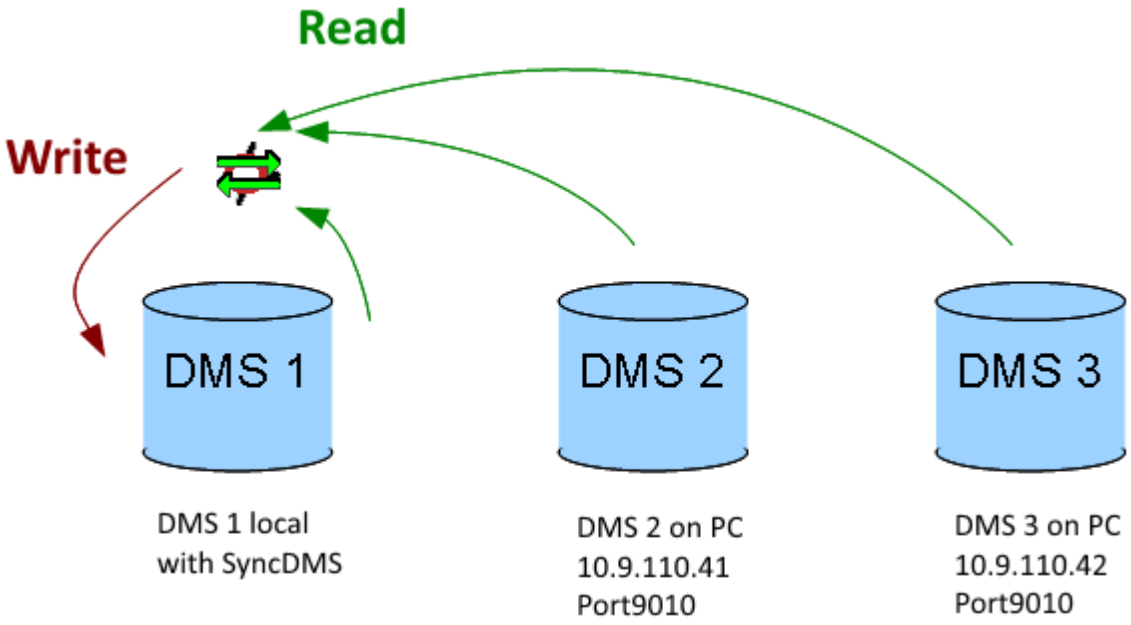
- **Fields**
  A field number and the corresponding title (1st line in the CSV file) can also be specified for

an active rule.
Field no. 0 means no output.
Several lines can be written with the line number (line 0 means field is used for each line).
The title of the first line is used as a title.
If no titles are specified, there is also no output of the title in the file.

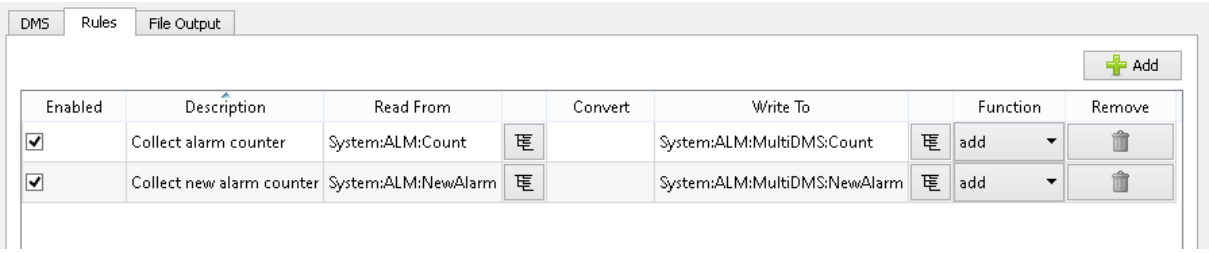
Note: The data points to be transferred ("DMS point for xxx") are sent to all DMSs, when a rule "Write activated" has been actively set for a DMS connection.

8.13.5.4 Examples

8.13.5.4.1 Collection of alarm counters at a DMS_2



To collect alarm counters, the following rules must be created:

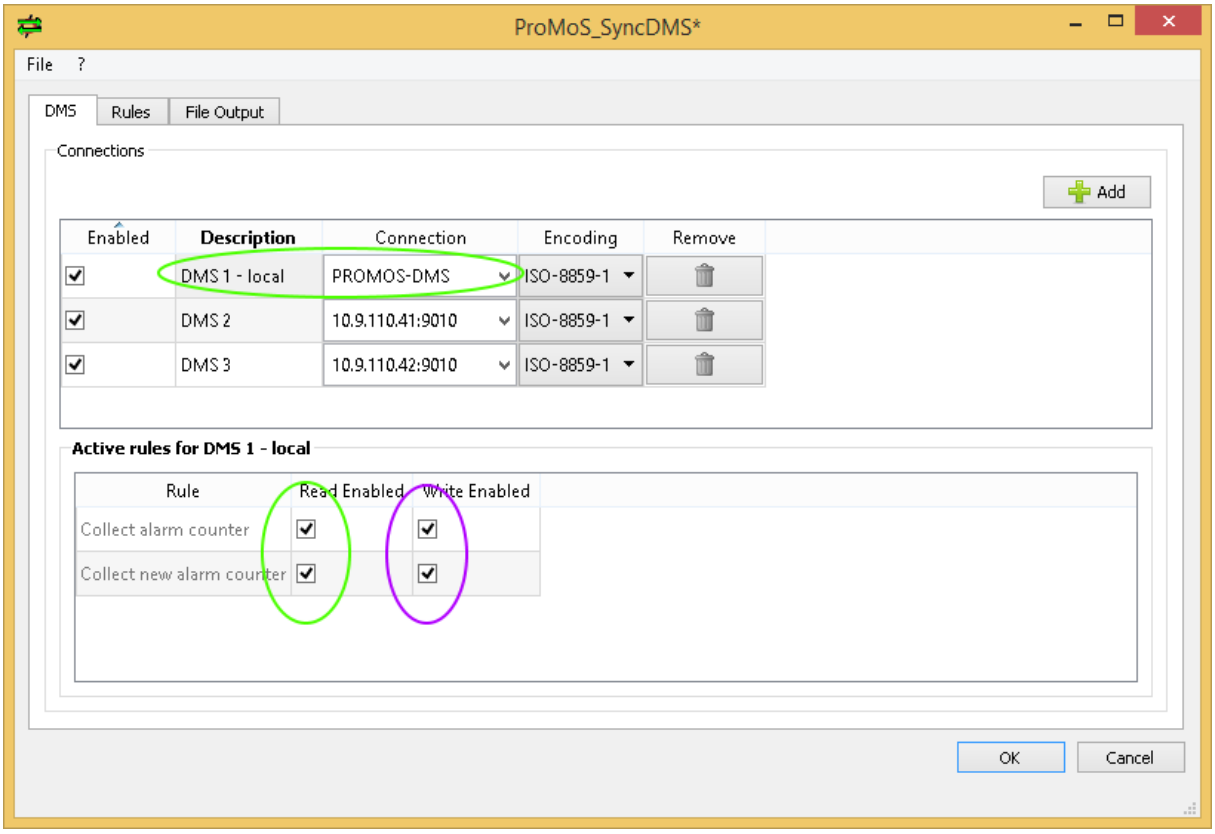


The values are then read and counted.
The result is written in `"System:ALM:MultiDMS:Count"` (resp. `"...:NewAlarm"`).

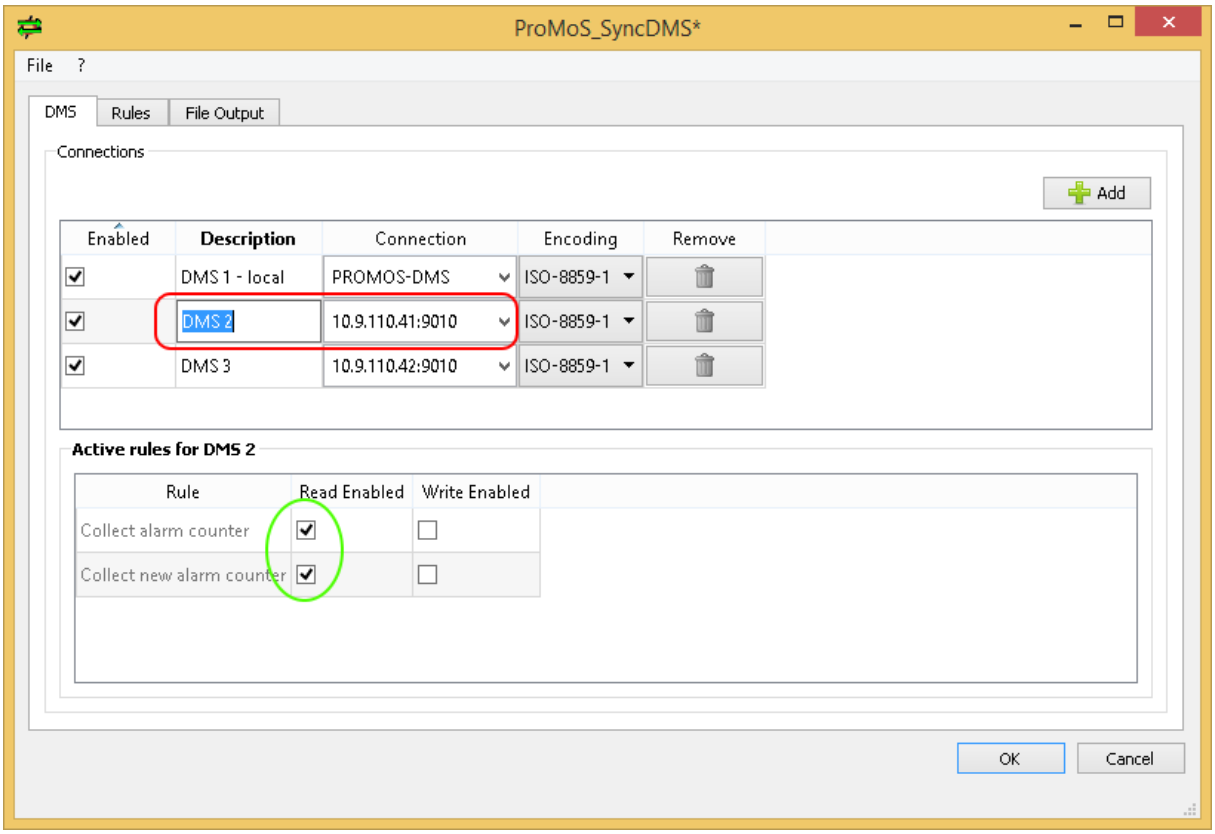
Note: Using the button,  a data point from the DMS can be selected.

- The values are read from all connections (read active)

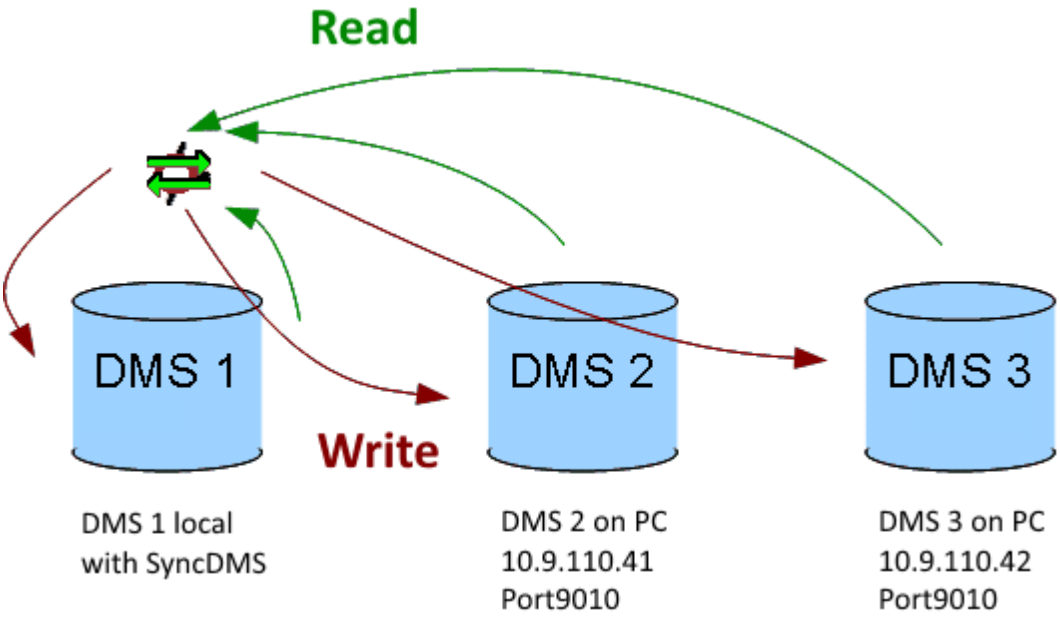
- The result is written at DMS 1 (write active)



- For all other connections, no values are written (read only)

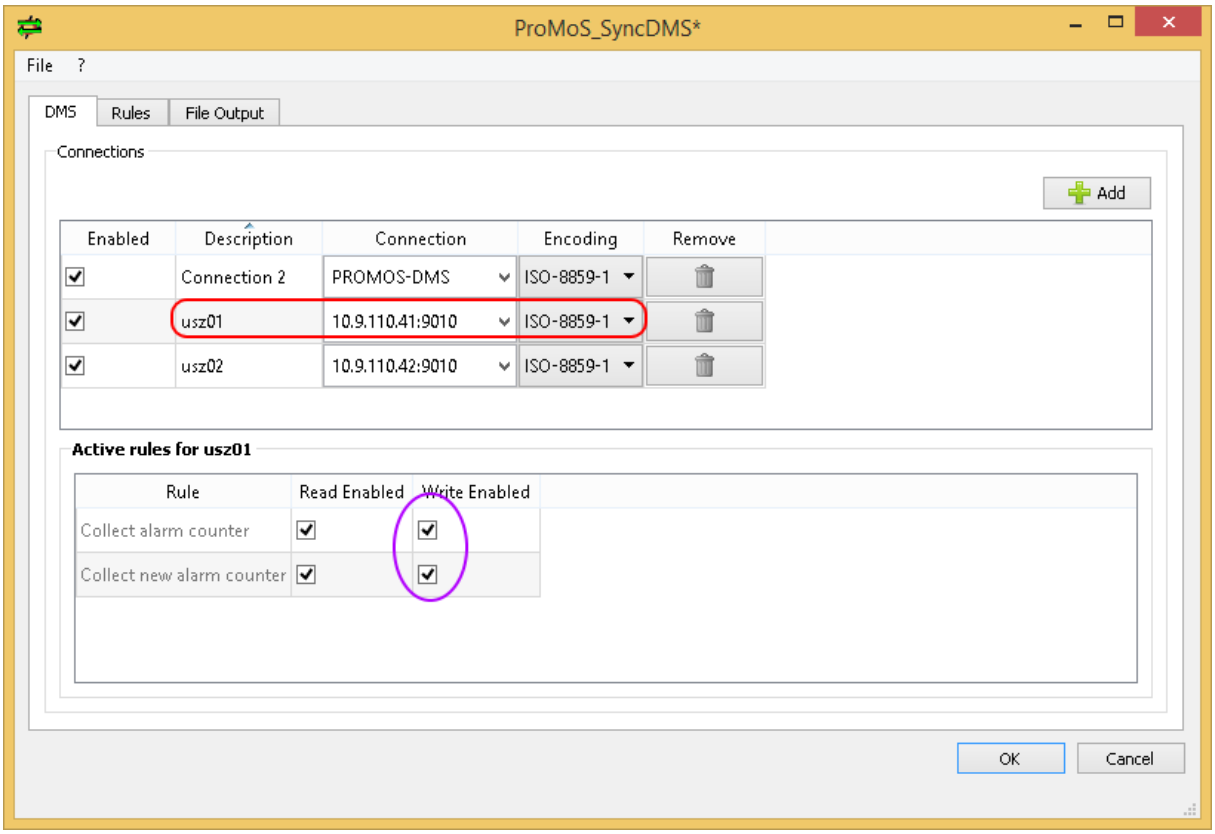


8.13.5.4.2 Collection of alarm counters at several DMSs (1 SynchDMS)_2

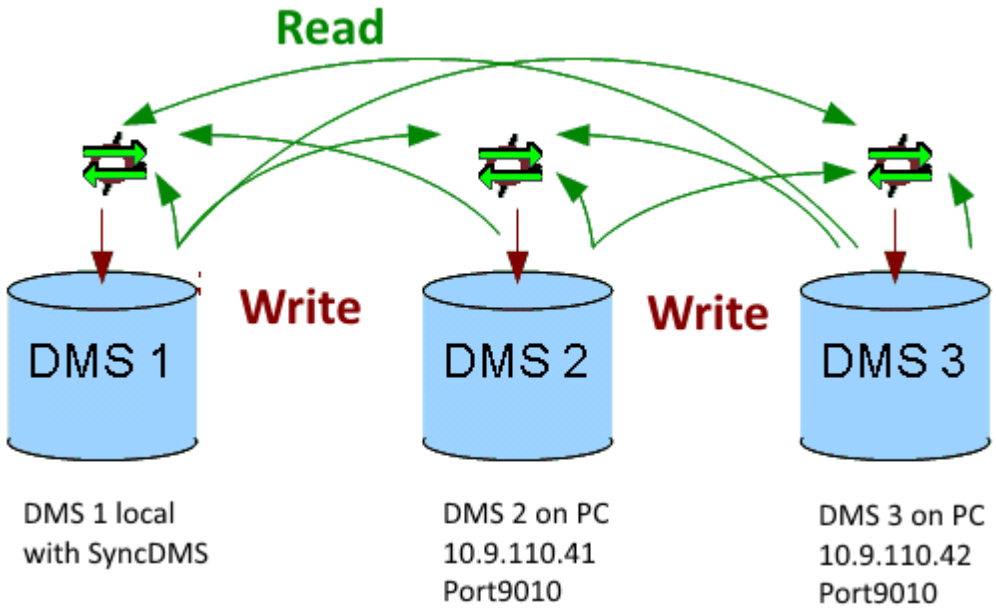


The same rules apply as for example 1.

Unlike example 1, the results are now written on all connections:



8.13.5.4.3 Collection of alarm counters at several DMSs (3 SynchDMS)_2



A SyncDMS is running at all stations.
Only its own DMS is described.

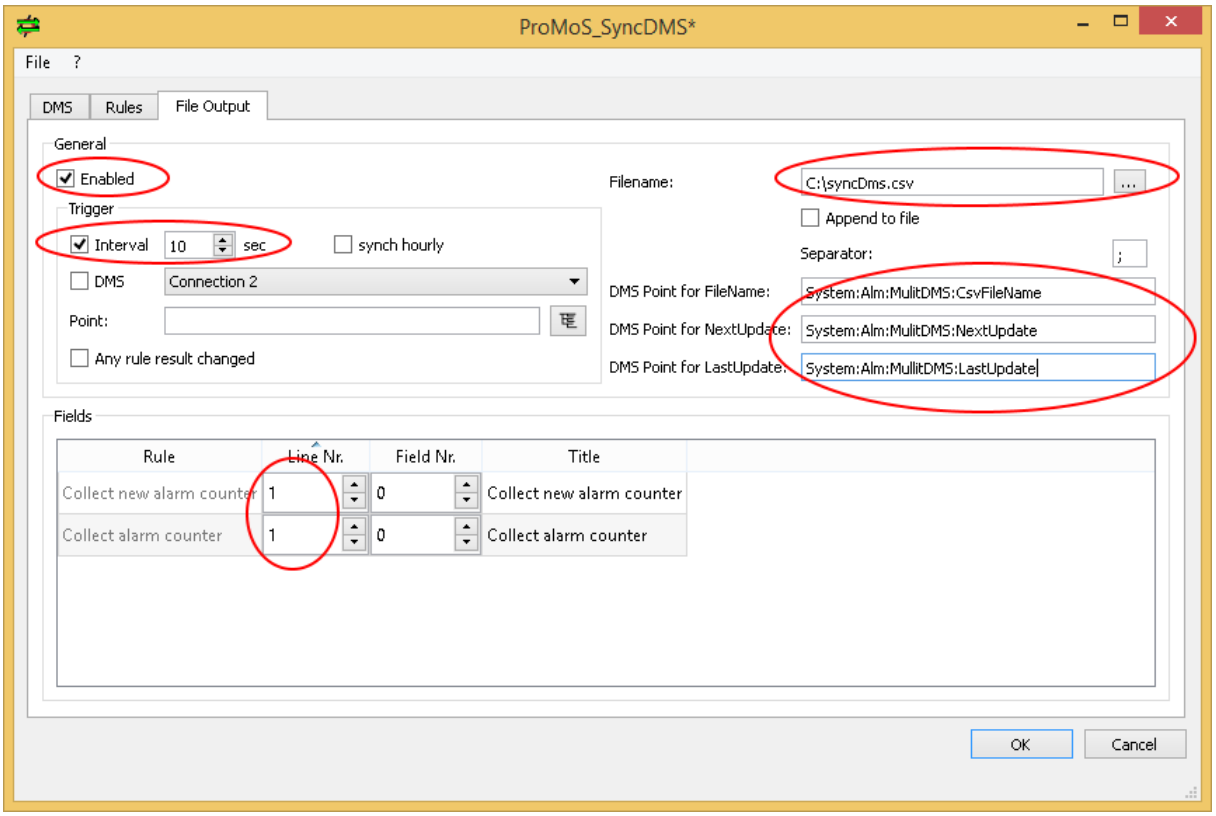
The same rules apply as for example 1 and 2.

Only writing on its own connection is activated (as in example 1).

8.13.5.4.4 Write a CSV file_2

The current values (from example 3) should be written in a CSV file every 10 seconds.

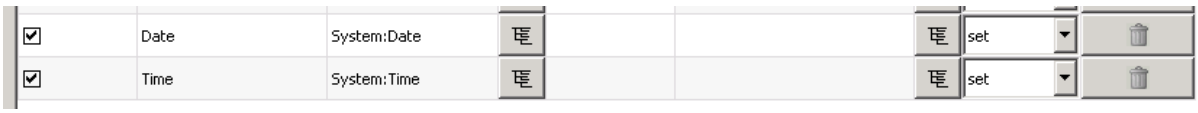
To do this, the following settings must be made in the "file output" tab, for example:



The file "C:\AlarmCounter.csv" is written every 10 seconds and contains the following data, for example:

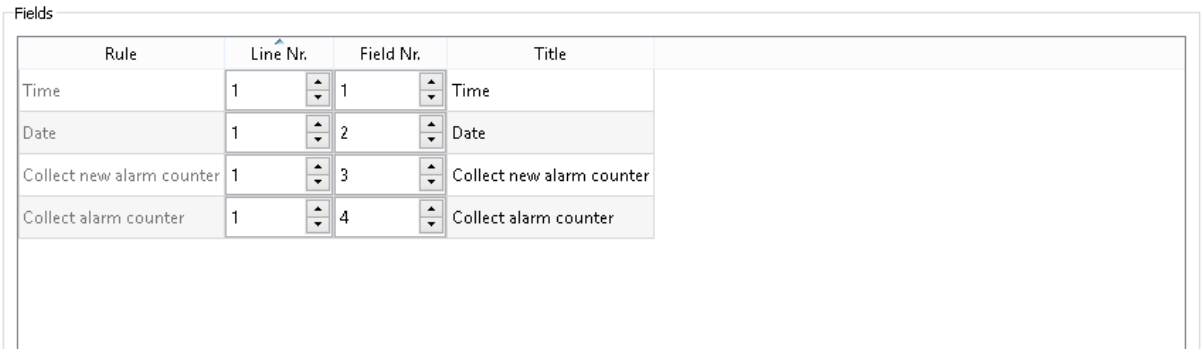
```
Collect alarm counter;Collect new alarm counter
1;6
```

If a time stamp should also be saved in the file, this can be achieved with 2 new rules:



=> read only from only 1 DMS activated, no write activated!

The field no. must then be adjusted accordingly:



The content of the file then appears as follows, for example:

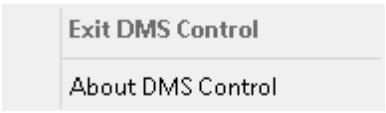
```
Date;Time;Collect alarm counter;Collect new alarm counter
11.01.13;14:09:59;1;6
```

8.14 System-Startprgramm

With this application the Visi.Plus Project is starting as configured in the ProjectCfg. Depend on the setting the promos.exe will be stopped after the start process or check afterward's the DMS function. For this there will be the following TrayIcon:

In case DMS isn't answering (internal Watchdog for one minute) the project will be automatically restarted. All other module (except pWA, PCDDriver who are connecting automatically new if the DMS is running again) will be terminated first. All other modules are controlled by the DMS itself.

The Application doesn't has any menu for the settings. All are adjust in the ProjectCfg. With a right mouse click on the TrayIcon the DMS controlling can be stopped.



The promos.exe application must be execute after restart the computer.

Saia Visi.Plus

Communications drivers

Chapter 9

9 Communications drivers

9.1 Summary

The data flow between Visi.Plus and the connected PLC is regulated by the Vis.Plus communications drivers. Visi.Plus has the following communications drivers:

- **SDriver**
  The standard driver can communicate through TCP/IP, SBUS, PC104, PGU, SBus modem and SOCKET with the PCD controllers.

- **OPCDriver**
  Visi.Plus offers an OPC driver which enables all data to be read directly into the DMS from an OPC server.

- **ESPADriver**

  The ESPA driver can read all data from corresponding systems (telephone/central fire alarm system, staff paging systems) right into the DMS using ESPA protocol 4.4.4.

- **SNMPDriver**

  The SNMP Driver (Simple Network Management Protocol) can monitor all software and hardware components ("managed objects") on the local PC or on the network (e.g. router, server, switches, printer, computer, etc.).

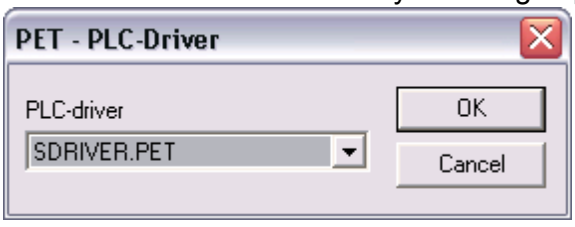
- **TAPIDriver**

  The TAPI Driver (Telephone Application Interface) accepts telephone calls and, after successful verification of the caller (code input check), can acknowledge all alarms that were sent via MalmMng (SMS, e-mail, pager, ESPA, Tel.Voice).

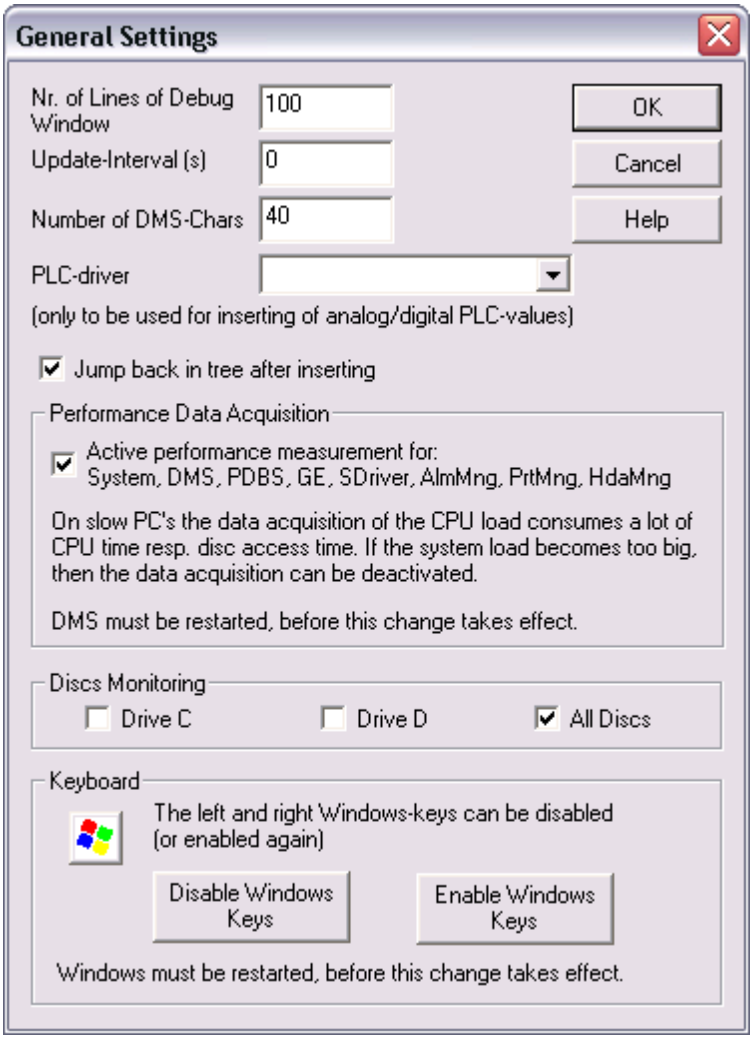
9.2 Selection of the PLC driver

The installed PLC drivers can be adjusted in the **DMS** and **PET** modules.

In the **PET** this is carried out by selecting "**Options > PLC Drivers**".



In the **DMS** this takes place by selecting" **Settings > General Settings > PLC Drivers**".



*If **all data has been entered in the PET**, selecting a **PLC driver** is not necessary. The DMS only requires this setting if it must be defined directly in the DMS communications data.*

9.3 The BACnet driver (BacDriver.exe)

9.3.1 Introduction

BacDriver is based on the BACstac software from Cimetrics, Inc.

The program is used to forward BACnet data to another system: BacDriver receives (via COV/COVP) or reads/polls data from the BACnet network and automatically transfers it to the VisiPlus control system (DMS).

After startup, BacDriver searches for (remote) devices and their objects by sending a BACnet Who Is request.

The configuration takes place in a text file. Any amount of data can be exported.

The data readout occurs:
1. periodically, according to the selected configuration, once a day, hourly or at any interval
2. by subscribing to the BACnet services - COV Notification (**C**hange **o**f **V**alue); with this service, all value changes in a property of a data point object are independently reported to BacDriver.

BACnet = Building Automation and Control networks

COV (COVP) = Change of Value (Change of Value of Property)

9.3.1.1 Pending issues

GUI

- User GUI: Configuration, view, etc.

BACnet

- Scan option: automatic creation of BMO templates of all BACnet objects from scan data (ensures correct property names, etc.)
- Delete device DPs in DMS (incl. all objects in DMS and internal DB)

Commandable objects: Manual control, replacement value, default value (relinquish default)

- Write replacement value/default value with any priority.

9.3.1.2 Supported BACstac version

Current **BacDriver version** is **1.7.1.293**.

This BacDriver 1.7.0.55 actively supports **BACstac 6.2.f**, but also runs under **BACstac 6.4**.

The current BACstac version, which is supported by the current BacDriver, is output in BacDriver.Log and BacDriverScan.log after the BacDriver start.

```
01.02.13 15:25:21
01.02.13 15:25:21 Supported BACstac Version
01.02.13 15:25:22
================================================================================
01.02.13 15:25:23  BACstac Version Major = [6]   BACstac Version Minor = [2])
01.02.13 15:25:23
```

BACnet Protocol Version and Revision Numbers

BACstac 6.2.f supports 135-2008: BACnet-2008, Ver. 1, Rev. 7, Addendum 2008w

In preparation winter 2017:
BACstac 6.7, which supports 135-2008: BACnet-2010, Ver. 1, Rev. 12

9.3.2 QuickStart

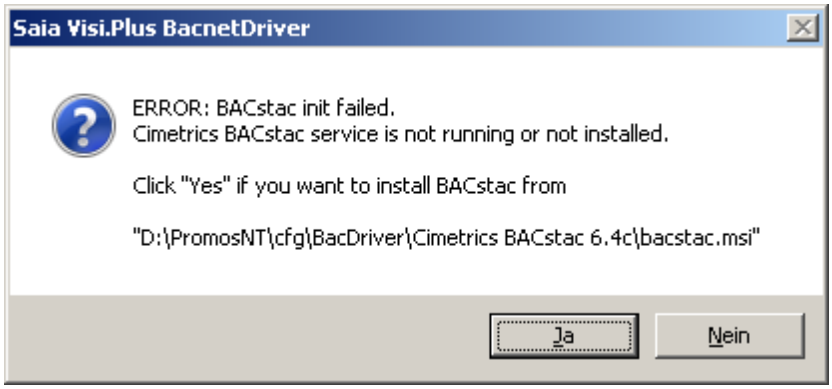
9.3.2.1 First steps: first run

Missing BACstac service

For correct execution of BacDriver there must be BACstac service from Cimetrics (BACnet Protocol Stack, see www.cimetrics.com) installed on the PC.
If BacDriver runs on a PC without installed BACstac or without running BACstac service, then following error message appeares in the message window:

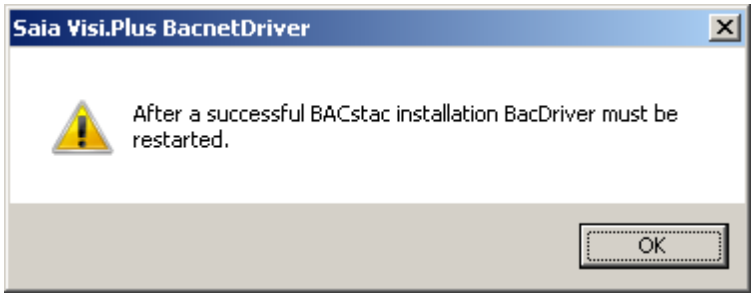
```
   14.06.16 11:12:02 ERROR: BACstac init failed: Cimetrics BACstac service is not
running or not installed: status=[2]=[Service not found])
```

At the same time the following message box is displayed:



BACstac installation is described in the next chapter Installation of BACstac.

Without running BACstac service BacDriver cannot connect to BACnet controllers. After successful installation of BACstac BacDriver must be restarted:



New project

If BacDriver starts in a new project and there is in the project's \drv folder no BacDriver's configuration file (BacDriver.cfg), then all missing files which are needed for BacDriver's correct execution are copied from the setup's \cfg\BacDriver location into the project:

- Visi.Plus\cfg\BacDriver**BMO.dms** is copied into Visi.Plus\proj\<project>\drv and at the same time merged into running DMS
- Visi.Plus\cfg\BacDriver**BacDriver.cfg** is copied into Visi.Plus\proj\<project>\drv
- Visi.Plus\cfg\BacDriver**DeviceList.csv** is copied into Visi.Plus\proj\<project>\drv
- and all other configuration templates from Visi.Plus\cfg\BacDriver are copied into Visi.Plus\proj\<project>\drv

At the same time following modifications are made:

- in DMS is "**System:UnitsFile:{CFGDIR}\BacDriver\BACnet_Units.txt**" created
- in BacDriver.cfg in section [Settings] AddUnknownBACProperties=2, which causes unknown BACnet properties to be inserted into corresponding BMO.dms BACnet base object templates

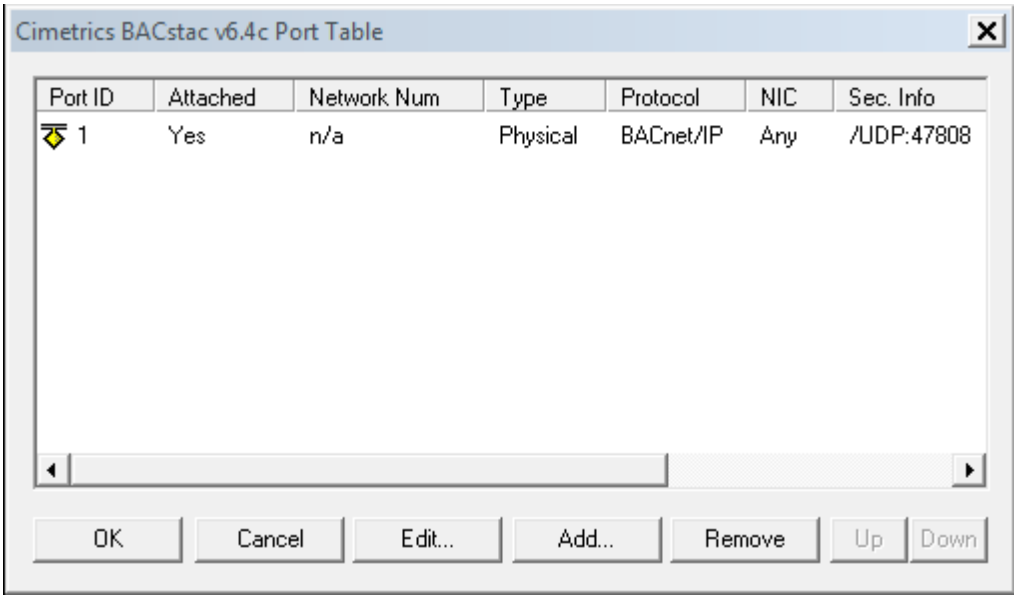
Usually after these modifications are made, BacDriver is capable to connect with all BACnet devices found in local network and depict them in DMS in a structured way. For this functionality default settings from BacDriver.cfg are used.

9.3.2.2 Installation of BACstac

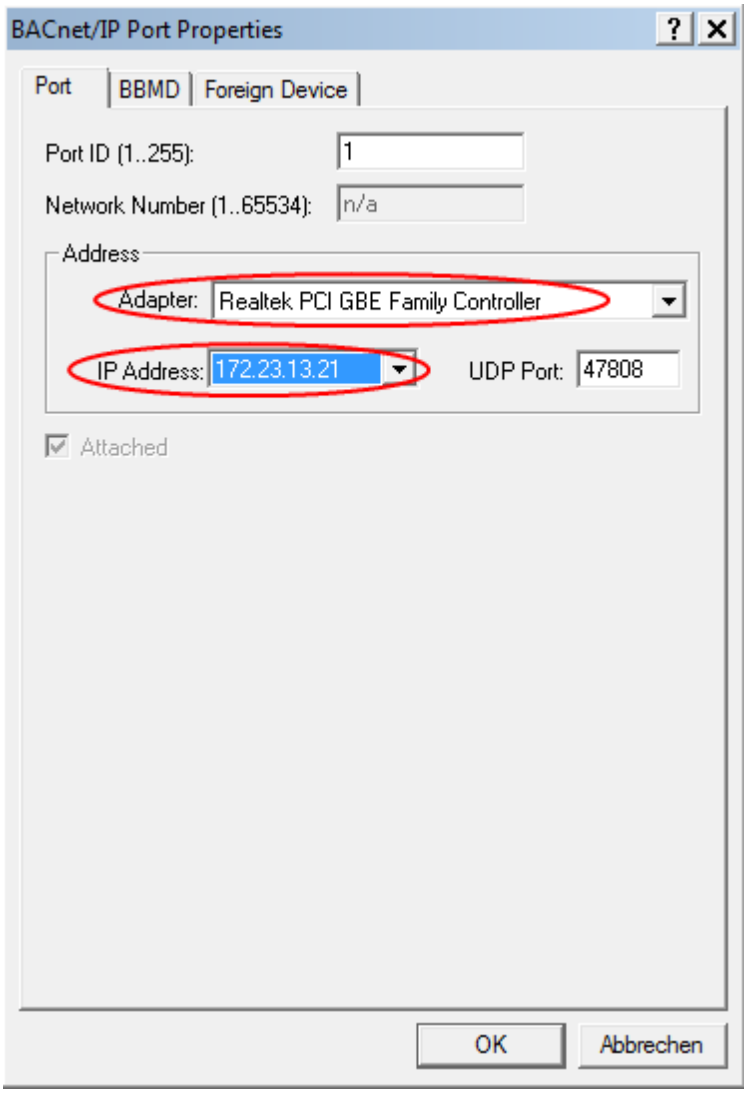
Cimetrix BACstac setup is located in the installation folder of **Visi.Plus** in the folder **cfg \BacDriver**.

Current version is **BACstac 6.4c**. It is installed by executing the Windows Installer **bacstac.msi**.
As standard BACstac is installed in the folder **C:\Program Files (x86)\Cimetrics\BACstac v6.4c**.

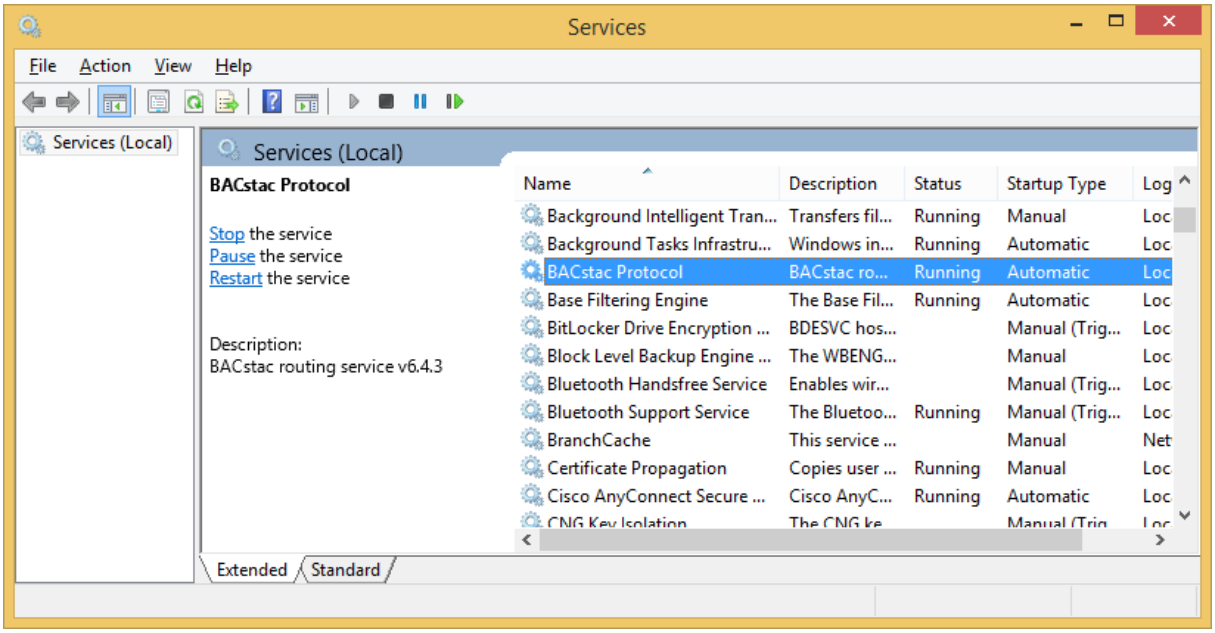
A BACstac configuration with **baccfg.bat** is also started here:



As standard it is sufficient to set the network card adapter and the IP address:

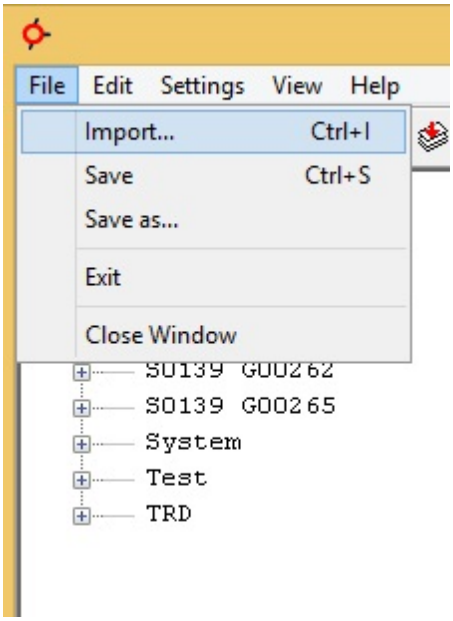


By pressing the <OK> button, the BACstac service should start automatically. This can be checked under *Run Windows* > and **services.msc**:



9.3.2.3 Import PG5 BACnet objects (DDC Suite)

If the BACnet objects are generated with the PG5 program, it will need the DMS for the import. For this purpose, the project must be started and a user must be logged on the control system as administrator.



In DMS under **File -> Open**, the subtree created can then be imported.

In BACDriver it must also be ensured that the driver does not scan the network and newly create the objects, but reads the information of the BACnet objects from the DMS. For this, in the BACDriver.cfg file under the rubric **[Settings]**, the setting must be **ScanDMS=1**.
The configuration file is saved in the directory *<Projektpfad>\<Projektname>\cfg*.

9.3.2.4 Internal test tool: BACnet Browser window

As soon as there is a connectivity to BACnet, the internal build in BACnet Browser might be very useful.

The BACnet Browser Window will be activated via Menu, View, BACnet Browser.

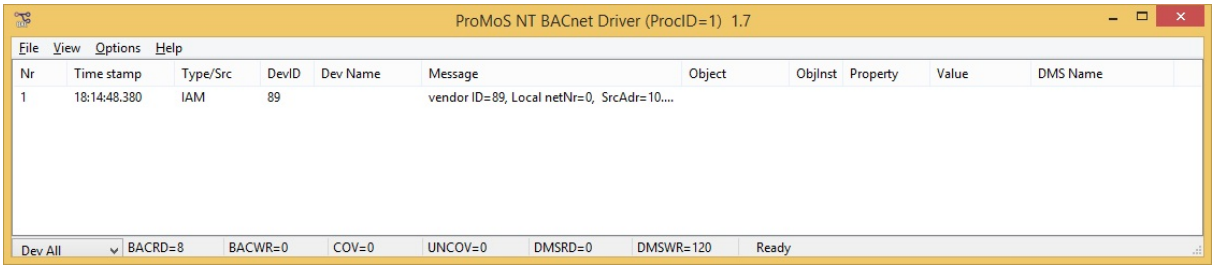
- BACnet Browser is a tool for analyzing and diagnosing BACnet networks.
- BACnet Browser is an easy way to discover BACnet devices, objects and their properties and display their connection to DMS.

It provides devices, objects and properties in a structured view, reflecting the actual BacDriver configuration settings.
In particular, the device range selection and the DMS name labeling definition will be considered.

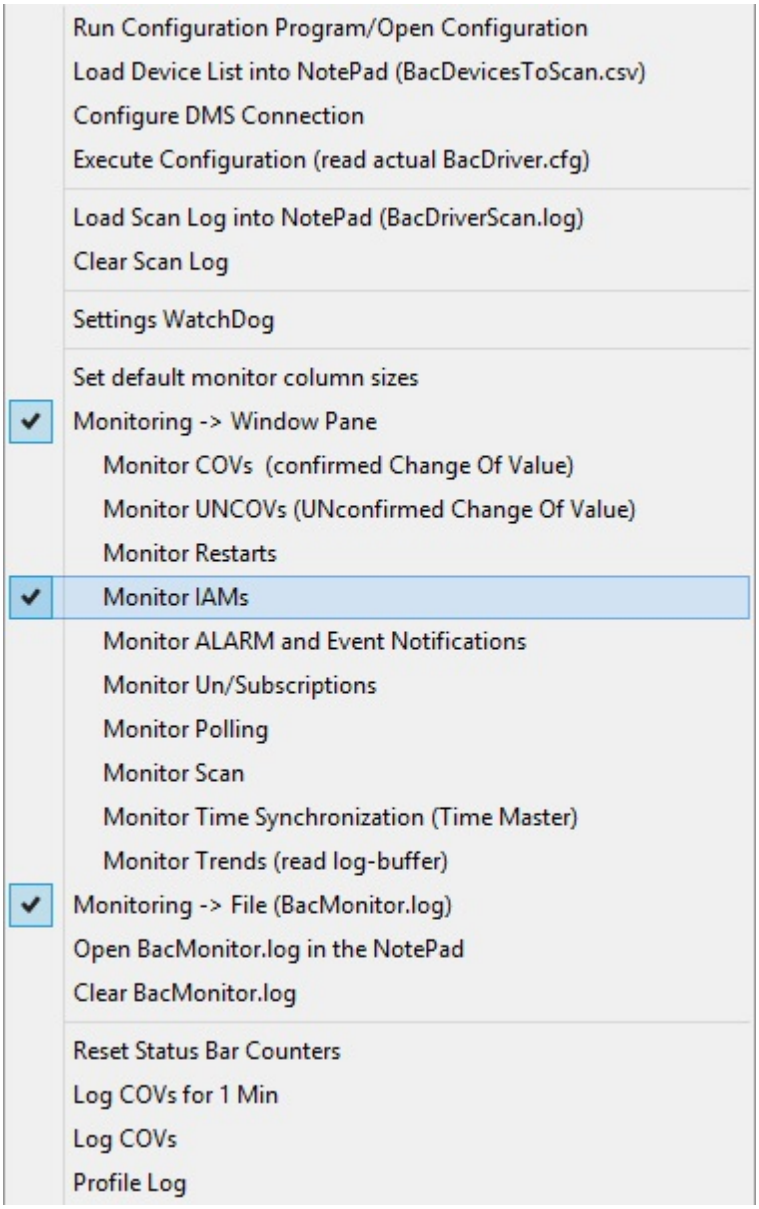
For more detailed description see chapter BACnet Browser Window.

9.3.2.5 Connectivity control BacDriver <-> BACnet devices

As standard after each start, BacDriver shows all accessible devices in the monitor window that can be integrated for the corresponding settings in the DMS Visi.Pluscontrol system:



Required setting for the above view in the Settings menu (monitoring and monitor IAMs must be checked):



If no I_AM broadcasts are received/displayed

- Test accessibility of the individual devices with ping
- Check BACstac service in Windows (has it started)
- Check BACstac settings: Network cards adapter and PC IP address (see Chapter Installation of BACstac)
- Check firewall: if the PC has a local firewall (e.g. in the Windows security centre), the BACstac port used here must be activated as otherwise communication (COV, I-Am and other broadcasts) is prevented.

All I Am broadcasts are always displayed, even from those devices that are outside of the set device instance number range.

9.3.2.6 Define device range

As standard a device ID range is defined which only covers devices that should be incorporated in DMS.

If additional devices should be explicitly excluded from communication with Visi.Plus, they must be realised with a device list.

The configurations file BacDriver.cfg (from the folder <proj>\drv) is automatically loaded in Notepad if the following command is executed in BacDriver:
Menu, settings, start configuration.

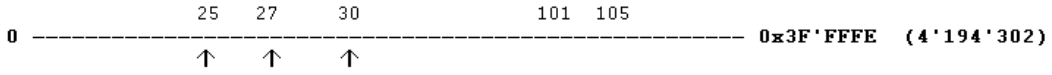
Device range

The device range is defined in BacDriver.cfg in **[Settings]** (see chapter Device instance numbers range):

```
; Default =-1 = no restricted WhoIs range is used, all IDs from 0 to 0x3FFFFE are recorded.
WhoIsDeviceInstNrMin=-1
WhoIsDeviceInstNrMax=-1
```

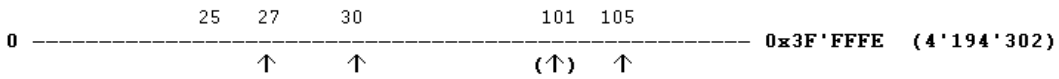
Examples for BACnet device ID range selection:

1. Devices 25..30 are incorporated, all others ignored:



WhoIsDeviceInstNoMin=25
WhoIsDeviceInstNoMax=30

2. Devices 27..105 are incorporated; the device list option must be used so that device 101 can be excluded:



WhoIsDeviceInstNoMin=27
WhoIsDeviceInstNoMax=105

The device lists file (from the folder <proj>\drv) is automatically loaded in Notepad if the following command is executed in BacDriver:
Menu > Settings > Open > Device lists file.

Device list

The device list is defined in BacDriver.cfg in **[Settings]** (see chapter Selective device scan):

```
; Option for selective Device-Scan: if BacDevicesToScanFile=file name and if this file
```

```
; exists in the <proj>\drv folder, then only those devices are scanned, which
; are defined and activated in the file (activation option in the 3rd column=1).
BacDevicesToScanFile=devicelist.csv
```

The different columns are separated by a semicolon.

1st column: Comment or IP
2nd column device ID
3rd column: Activation flag. 1= activated; 0= deactivated (default)
4th column: Comment/ description.

Example of a device list, for which for the above case:

```
; the following devices are selected/activated  ----> 3rd field = '1'
27; 27; 1;device 27 ventilation
172.23.13.30; 30; 1;Device 30 heating
--; 105; 1; Device 105 Motor

; next device is deactivated  ----> 3rd field = '0'  or optionally the device line does not hav
;present at all...
-; 101; 0; test device..
```

9.3.2.7 Control of all communicating devices

A combo box is integrated in the status display at the lower left which includes all connected devices. See also chapter Status bar display.

9.3.2.8 Check for update

A new setup not only brings new functions but also new parameters for the configuration as well as, in certain circumstances, new/expanded BACnet object templates.

New functionality has been realised in BacDriver.exe and does not require any further notice. However, new functions must often be initially activated and/or configured with new parameters.
New or modified BACnet objects can also be located in BMO.dms (as elementary templates).

- The new settings/parameters are located in the BacDriver.cfg configuration file, which is available as a template in two language versions:
  English version <Installations Pfad>\cfg\BacDriver\BacDriver.cfg.En
  German version <Installations Pfad>\cfg\BacDriver\BacDriver.cfg.Ge

- The BACnet objects are in <Installations Pfad>\cfg\BacDriver\BMO.dms

See also chapter Automatic update control

Update function

1. the current BacDriver.cfg version in the project is compared with the one which was copied from the last setup in the installation folder ..\cfg\BacDriver\,
   with the corresponding cfg template BacDriver.cfg.En/Ge (naturally which language version the customer installed in ..\proj\drv\BacDriver.cfg is considered here).

2. If the version number of the cfg file in the installation folder is higher than that of the customer, the following actions are performed:
   - backups of the customer versions of BacDriver.cfg and BMO.dms are made (with date/ time stamp)
   - BMO.dms from the installation folder is saved in the customer -<proj>\cfg\, read in DMS and saved in DMS.
   - The corresponding language version of the BacDriver.cfg file template is copied to the customer <proj>\drv\ and compared with the original customer version, whereby all previous customer settings are adopted.
   - All backup/copy/paste actions are logged in the message window as well as in BacDriver.log in the \log folder.

Manual update: Check for update via menu command

In the BacDriver menu, the command file, "Check for update", can be executed at any time.

Steps 1 and 2 from "Automatic Update" are performed exactly as described above with the difference that before an update is started, the user is informed via a request window and a confirmation is required (OK, cancel).

If no update is pending, a corresponding message window also appears.

Manual update

The operations described above can also be performed manually:

1. Copy the English BacDriver.cfg.En (or the German BacDriver.cfg.Ge) configuration template from the installation folder ..\cfg\BacDriver\
   to the project <proj>\<Projekt-Ordner>\drv and rename it BacDriver.cfg. (Any existing cfg file should be backed up)

2. Start DMS, make a backup copy of BMO.dms, copy BMO.dms from Installation folder .. \cfg\BacDriver\ in <proj>\<Projekt-Ordner>\cfg
   and read BMO.dms into DMS: Menu File, Open... DMS should then be backed up: Menu, File, Save.

9.3.2.9 Objects

To be able to work with BACnet and Visi.Plus, template objects are required. The HLK or DDC Suite template objects are usually composed of several BACnet objects. This gives rise to so-called Super VLOs. More information about Super VLO can be found in the manual of Visi.Plus under the chapter PET > template objects. Information about BACnet objects is in the manual under the rubric BACnet driver> BACnet object templates.

9.3.3 Restrictions, BacDriver limitations

There are some restrictions for the number of communicating data points:

- The number of BACnet devices (controllers, devices) may not be more than 500.
- A maximum of 32 BACnet object types can be defined (there are currently 28 types).
- A maximum of 4000 object instances are possible for each BACnet device.
- A maximum of 64 properties can communicate for each BACnet object.
- The DMS limit must be considered. See chapter Performance characteristics and technical data

Demo license
The number of objects is limited to max. 10.

Engineering / Development license
The number of objects is limited to max. 100.

9.3.4 The BacDriver user interface

9.3.4.1 Status bar display

The status bar contains a combo box with current devices as well as the following text fields that display different event counters:

Event counter

- BACRD Number of read BAC objects (generated from poll and scan procedure)
- BACWR Number of BACnet write procedures (generated by GE/WebServer/DMS)
- COV Number of received Change of Values (confirmed)
- UNCOV Number of received UNconfirmed Change of Values
- DMSRD Number of read data point value changes from DMS (generated by GE/WebServer/DMS)
- DMSWR Number of data point change values written in DMS (generated by BACnet COV/Poll/Scan)

The current event counter display always refers to the set device in the combo box; the highest setting"**_Dev All**" means that the results for all devices are displayed added up, otherwise the display (and monitor window settings) applies only for the individually selected device.

Current devices combo box

All current devices with their name (property object name of the device object) are listed alphabetically in the combo box.

- Current devices All detected devices (via IAM broadcast) that are in the

configured Who Is range
(WhoIsDeviceInstNoMin/Max) and not set inactive (in BacDevicesToScanFile)

Standard display is "**_Dev All**", which symbolically stands for all devices, i.e. all devices are selected with this setting.
If an individual device is displayed in the combo box, then this selection also applies for all displayed counter events as well as all settings in the monitor window.

Current filter setting display in the combo box

Special case is "**_Filter**", wherein only those objects are displayed, which fulfill the filter setting.
The filter string (from **BacDriver.cfg** section **[Monitoring] MonitorFilter**) will be displayed in the status line.
Current version allows a filter definition for only one object. Syntax:
MonitorFilter=<DeviceID>:<ObjTyp>,<ObjID>

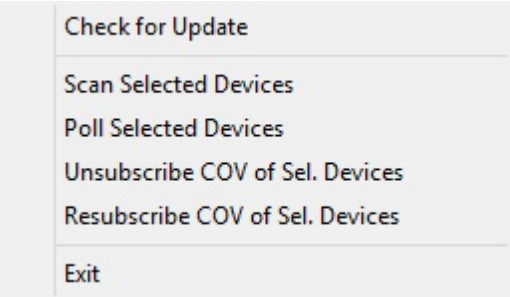
Example: **MonitorFilter=5010:0,4** e.g. device = 5010, object type = 0 = analog input, object instance (=ID) = 4

See chapter Objekt Filter.

9.3.4.2 The BacDriver menus

9.3.4.2.1 "File" menu

The ""File"menu contains the following functions:



Check for update
See also chapter Check for update
1. The current BacDriver.cfg version in the project is compared with the version copied from the last setup in the installation folder ..\cfg\BacDriver\,
   with the corresponding cfg template BacDriver.cfg.En/Ge (the language version that the customer used for installation in ..\proj\drv\BacDriver.cfg is of course taken into account).

2. If the version number of the cfg file in the installation folder is higher than that of the customer, the following actions are performed:
   - backups of the customer versions of BacDriver.cfg and BMO.dms are made (with date/

time stamp)
- BMO.dms from the installation folder is copied in the customer -<proj>\cfg\, read in DMS and saved in DMS.
- The corresponding language version of the BacDriver.cfg file template is copied to the customer <proj>\drv\ and compared with the original customer version, whereby all previous customer settings are adopted.
- All backup/copy/paste actions are logged in the message window and in BacDriver.log in the \log folder.

Scan BACnet

Before each Scan command, the configuration file BacDriver.cfg is newly read and evaluated.
If the option **BacDevicesToScanFile** is defined, this file is also read.

All **active** BACnet devices, which have responded after the BacDriver Start to the **Who Is** request are newly scanned.

An **active** BACnet device means that its instance number
3. is in the defined range of the **Who Is** request in the configuration file (see Chapter Device instance numbers range (WhoIsDeviceInstNoMin/Max)).
4. in the case of an activated **BacDevicesToScanFile** option, the device must be under the listed, active instance numbers. (see chapter Selective device scan)

During the scan, a progress bar is shown that visually reports the progress of the scan process.

Poll
Property values of all objects of all devices are newly read and written in DMS.

Unsubscribe COV
COVs of all objects of all devices are logged off (unsubscribe, unregistered).

Re-subscribe COV
COVs of all objects of all devices are logged off (unsubscribe, unregistered) and reregistered.
The subscription lifetime is obtained from the settings: BacDriver.cfg, [Settings] COVsubscriptionLifetime.
Default 1 is one week = 604800 seconds.
If the following option is activated in the section [Debug] of the BacDriver.cfg configuration file, then COVs are also only explicitly unregistered before the new subscription:

```
; COV is always initially unsubscribed before a new (re)-subscription is executed.
; Default =0 -> only one new subscription request is output
; (with the same existing COV context)
COVUnsubBeforeResub=0
```

Manual time synchronisation

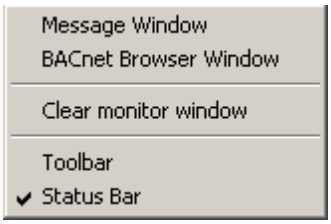
Selected BACnet devices can also be manually synchronised at any time via Menu Command File, "Re-subscribe COV of selected devices".
For the COV resubscription, the time synchronisation services (time synchronisation and UTC time synchronisation) are performed.

Exit

The BacDriver is ended: all registered COVs are cancelled (unsubscribe). No COVs therefore remain active in the devices after BacDriver no longer runs.
The current BacDriver window is closed and the driver is removed from the RAM of the computer. The driver is no longer active.

9.3.4.2.2 "View" menu

The ""**View**" menu contains the following functions:



Message window

Displays the message window. The message window is a non-modal (it can remain open for runtime without affecting BacDriver functionality) message window, mainly for error and system info display, see Chapter The message window)

BACnet browser window

Opens the BACnet browser window. The BACnet browser window is a non-modal browser window (it can remain open for runtime without affecting BacDriver functionality), mainly for diagnostic purposes; it displays in a structured way all BACnet devices which are connected to DMS, theirs objects and properties, see Chapter BACnet Browser Window.

Clear monitor window

The content of the monitor window is deleted.

Symbol bar

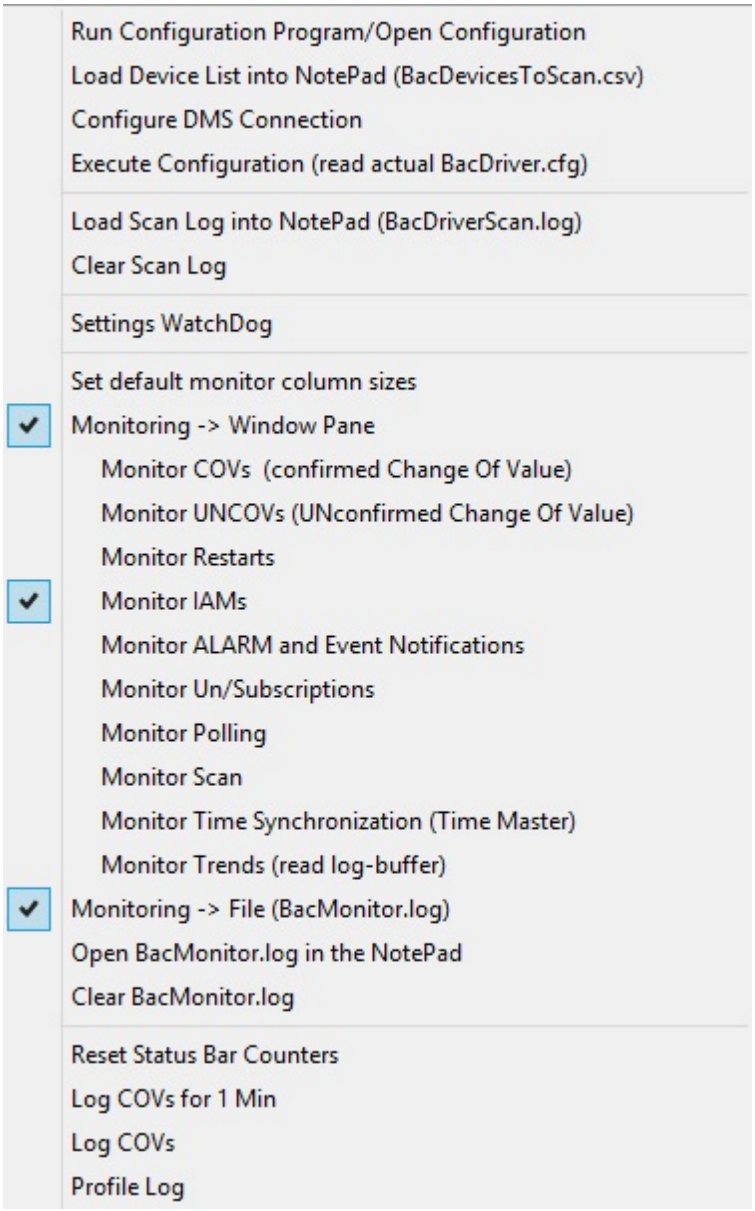
Switches the symbol bar on or off.

Status bar

Switches the status line on or off. The status bar contains a combo box with current devices as well as the following text fields that display different event counters, see Chapter Status bar display

9.3.4.2.3 "Settings" menu

The "**Options**" menu contains the following functions:



Start configuration/Configuration program

An external configuration program is called up.

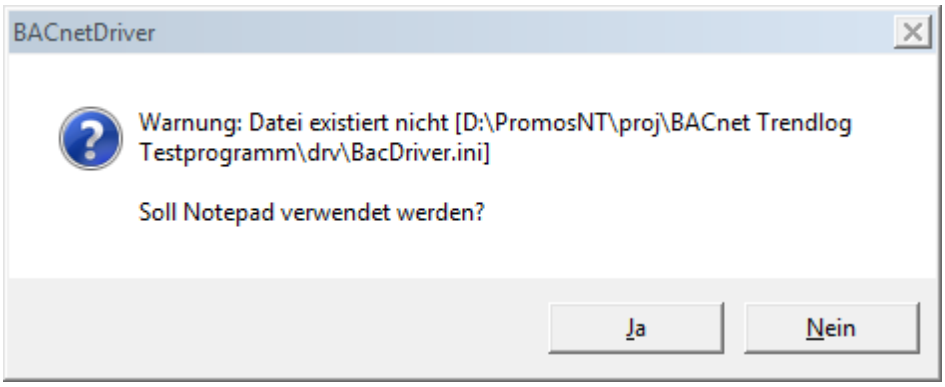
The program name (and possibly path) must be entered in a **BacDriver.ini** file in section [GUI] CfgProgram=<ProgrammName.exe>.
The BacDriver.ini must either be in the execution folder of BacDriver.exe (so for example C:**Visi.Plus**\bin), or in the <proj>\drv folder.

Example of a BacDriver.ini:

```
; List here in section [GUI] BacDriver's configuration program
[GUI]
CfgProgram=notepad.exe
```

If no external GUI configuration program is defined in BacDriver.ini, **Notepad.exe** is called up with the current BacDriver.cfg configuration file by default after a corresponding warning has

been output:



Open device lists file in Notepad (BacDevicesToScan.csv)

The device configuration file is loaded and displayed in Notepad. The name of the file is determined from BacDriver.cfg, from

```
[Settings]
BacDevicesToScanFile=devicelist.csv
```

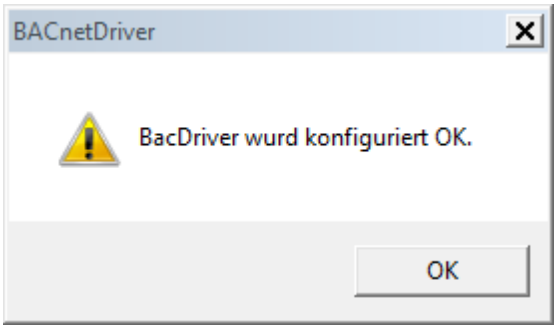
In this example, the device configuration file is called "`devicelist.csv`".

DMS connection

Opens a configuration dialog for the DMS connection settings.

Perform configuration instructions (current BacDriver.cfg)

The BacDriver is newly reconfigured with current values from BacDriver.cfg. Depending on which settings have been changed, devices may have to be re-subscribed. As this can take a long time in certain circumstances (a few minutes), the end of the reconfiguration process is announced with a confirmation message box:



Open scan log file in Notepad (BacDriverScan.log)

The scan log file is loaded and displayed in Notepad. See chapter Scan options, Log of the scanned objects and their properties

Delete scan log file content

The scan log file BacDriverScan.log is deleted.

WatchDog settings
Opens a configuration dialog for the WatchDog settings.

Set standard width for monitor columns
All column sizes are reset to the standard widths and saved. This function is mainly useful when one or more column widths are reduced to zero (hidden).

Monitoring -> Window
All monitor actions can be switched on and off with this function.
This is a checkbox menu command. If checked, the monitor is activated.

Monitor COVs
Monitor events of type COV (value changes)

Monitor Restarts
Monitor events of type UNCOV (unconfirmed broadcasts), which are used for device restart detection.

Monitor IAMs
Monitor events of type IAM messages.

Monitor Un/subscriptions
Monitor all COV registration types: Un- and/or subscription commands which are initiated by BacDriver.

Monitoring -> File (BacMonitor.log)
All monitor entries are written in a BacMonitor.log file, and independent of whether monitoring is generally switched on or off.
This is a checkbox menu command. If checked, the monitor ->file function is activated.

Open BacMonitor.log file in NotePad
The BacMonitor.log file is loaded in Notepad.

Delete BacMonitor.log file content
The BacMonitor.log file is deleted.

Reset status bar counter
All counter displays on the status bar are set to zero.

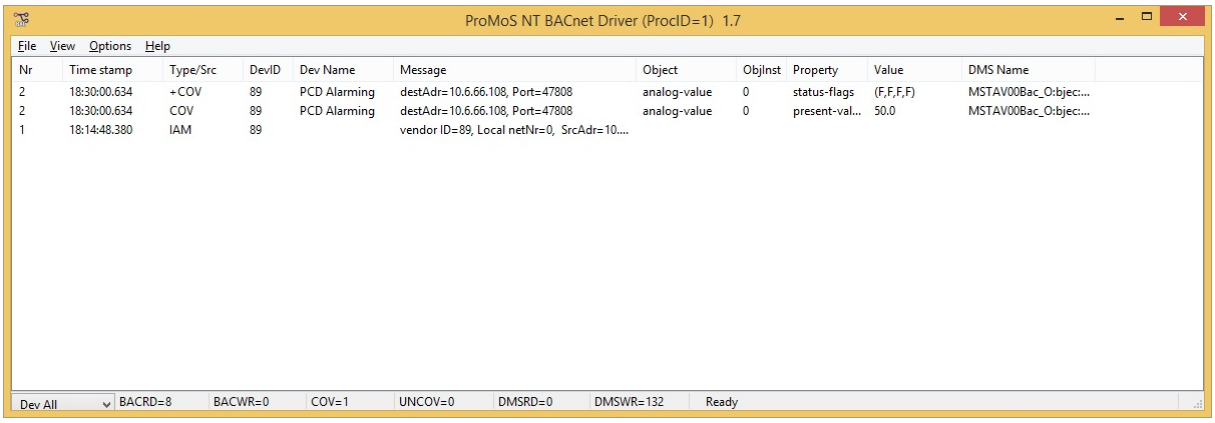
Log COVs for 1 minute
This is a checkbox menu command. If checked, then all COV activities for one minute are entered in **BacDriverTrace.log** or in the separate trace file of the corresponding device (if configured), for example **BacDriver-Dev-25-Trace.log**
After one minute has elapsed, this option is automatically reset.

Log COVs
This is a checkbox menu command. If checked, then all COV activities (unlimited) are entered in **BacDriverTrace.log** or in the separate trace file of the corresponding device (if configured), for example **BacDriver-Dev-25-Trace.log**
As this can generate a considerable number of I/O accesses in certain circumstances and as

a result reduce the performance of the BacDriver, this setting is not advised.

9.3.4.3 The monitor window

Here the current events (alarms, notifications, value changes, etc.) can be represented.



- Monitor function can be activated/deactivated in the Settings menu, Monitoring -> Window.
- The monitor actions only refer to the device set (all or individual) on the status bar.
- The monitor window can be deleted at any time; the counter is then set to zero again.
- The column width can be changed; the sizes are automatically saved after each adjustment.
- Different filters can be set for the individual event types (COV, IAM, restarts, subscriptions).
- All monitor entries can be written in a BacMonitor.log file, independently of whether monitoring is generally switched on or off.
- With the menu command "Open BacMonitor.log file in NotePad", the BacMonitor.log file can be loaded in Notepad.
- With the menu command "Delete BacMonitor.log file content", the file is deleted.

Column description

1. **No** is the sequence numbering of the logged events. It is reset to 1 if the monitor window is deleted.
2. **Time stamp** hh:mm:ss.milli created at the moment when the event has arrived in BacDriver or is output by BacDriver.
3. **Type/Src** specifies the type and the source of the event. If in capital letters, this means that the source is a BACnet event, otherwise a BacDriver action:
   IAM is a broadcast; additional information is displayed in the message column: Manufacturer ID, Remote / Local Network No., SrcAdr / Port, etc.
   COV is a value change broadcast registered (subscribed) by BacDriver.
   +COV indicates that this information is an additional part of the previous line. COVs can also supply property values (status, etc.).
   UNCOV is unconfirmed COV Broadcast, mainly used for the device restart detection.
   +UNCOV indicates that this information is an additional part of the previous line. UNCOVs can also supply property values (status, etc.).
   subCov / unsubCov are general BacDriver's COV un/subscriptions.
   subCovP / unsubCovP are general BacDriver's COVP un/subscriptions von Property priority-array.
   un/subCovTimer : comes from BacDriver's Timer, periodical Re-Subscription,

Period is configured in [Settings] COVresubscrPeriod.
un/subCovRestartUncov comes from detected Restart Notification Broadcast Event. The device will be again re-subscribed.
un/subCovRestartWD comes from BacDriver's Timer, periodical Watchdog Control. The device will be again re-subscribed, after it had been some time unavailable (e.g. LAN failure). The Watchdog period is configured in [Settings] WDPeriod.
un/subCovCmdMenu comes from Re-subscription, which is executed from the command menu.
un/subCovScan comes from Re-subscription, which is executed from the BACscan function, either from BacDriver's start or from the command menu.
unsubCovExitApp comes from Un-subscription, which is executed, when BacDriver exits.

4. **DevID** is the source device ID (instance number).
5. **Dev Name** is the source device name (value of the property object name from device object).
6. **Message** is additional specifications about the event, dep. on the type, for example, Source/Dest. Addr, network number, port, etc.
7. **Object** is the name of the participating object (property object name).
8. **ObjInst** is the instance number (object ID) of the participating object.
9. **Property [BAC typ]** is the name of the participating property and their BACnet typ [BOOLEAN/UNSIGNED/SIGNED/REAL/DOUBLE/ENUM/BIT_STRING/ ANSI_X34_STRING...].
10. **Value** indicates the read/written value of the BACnet property, or of the DMS data point.
11. **DMS Type** indicates the type of the participating data point in DMS: BIT/BYS/BYU/WOS/ WOU/DWS/DWU/FLT/STR.
12. **DMS name** is the name of the associated data point in DMS.

Event filter

The monitoring of each event type can be individually activated/deactivated with the corresponding menu command:

- Monitor **COVs / UNCOVs**
- Monitor **Restarts**
- Monitor **IAMs**
- Monitor **Un/Subscriptions**
- Monitor **Polling**
- Monitor **BAC writes**
- Monitor **Scan**
- Monitor **Time synchronisation**
- Monitor **Trends**

Configuration

The monitor window has a defined maximum number of lines. If it is reached, the oldest lines are deleted (ring buffer).
The standard is 1000 lines, max. 10000, min. 100.
This setting as well as the others are kept in the BacDriver.cfg file, in section [Monitor]:

```
[Monitor]
;=============================================================================================
==
; Following options are for monitor window pane display.
```

```
; Monitor window: if =1 then the data broadcast and notification traffic from PLCs will be displayed
; (default =1)
MonitorActive=1

; Monitor COVs (Change of Value) notifications, default =0
MonitorCOV=0

; Monitor UNCOVs (Change of Value) Notifications, default =0
MonitorUNCOV=0

; Monitor EVTs (Alarm and Event) notifications, default =0
MonitorEVT=0

; Monitor devices restart notifications, which are delivered by an Uncofirmed COV, default =0
MonitorRestart=0

; Monitor devices IAM broadcasts, default =1
MonitorIAM=1

; Monitor objects COV subscriptions, default =0
MonitorSub=0

; Monitor objects polling, default =0
MonitorPoll=0

; Monitor objects scanning, default =0
MonitorScan=0

; Monitor time synchronization (TimeMaster), default =0
MonitorTimeSynch=0

; Monitor activated for DMS writes -> BACnet (incl. following object-poll), default =0
MonitorBacWrt=0
; Monitor objects trending, default =0
MonitorTrends=0

; If =1 then monitoring goes into the BacMonitor.log file, regardless of whether MonitorActive=1 or not.
; (default=0)
MonitorToFile=0

; Nr. of buffer lines in monitor window. Default=1000, max. 10000, min. 100.
MonitorLines=1000
```

9.3.4.3.1 Object Filter

Current BacDriver version offers one filter definition for the monitoring of just one object.
In the global configuration file BacDriver.cfg:

Section [Monitor]

```
; Monitor Filter: if this setting is activated, then only such objects
will be monitored,
; which fulfill the filter option (filter string):
"<devID>:<ObjTyp>,<ObjID>"
; Example: MonitorFilter=5010:5,263
```

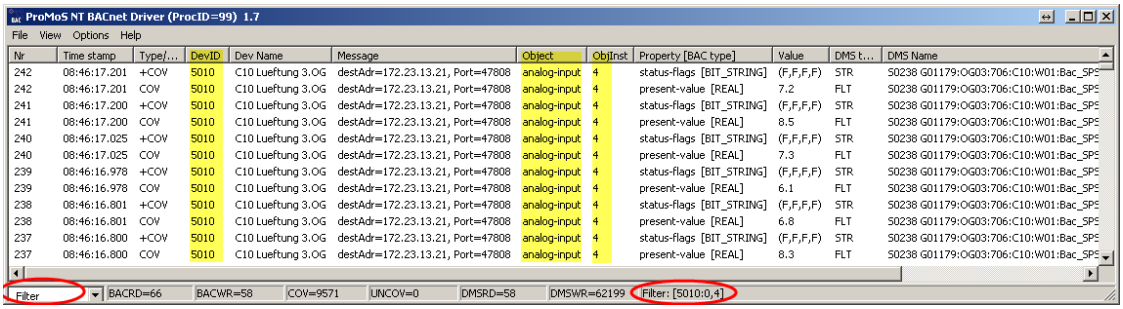
MonitorFilter=<DeviceID>:<ObjTyp>,<ObjID>

Example.: **MonitorFilter=5010:0,4** e.g device = 5010, object type = 0 = analog input, object instance (=ID) = 4
With this filter setting, only the following events appear:

Display: current filter setting in the combo box

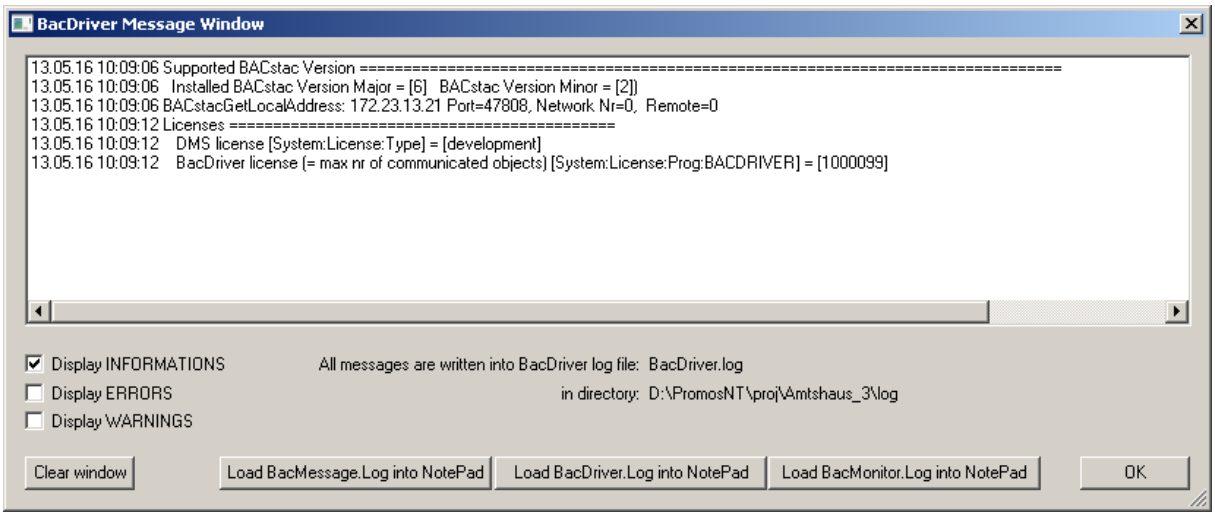
Special case is "**_Filter**", wherein only those objects are displayed, which fulfill the filter setting.
The filter string (from **BacDriver.cfg** section **[Monitoring] MonitorFilter**) will be displayed in the status line.

Example: **MonitorFilter=5010:0,4** e.g. device = 5010, object type = 0 = analog input, object instance (=ID) = 4
With this filter setting, only the following events appear:

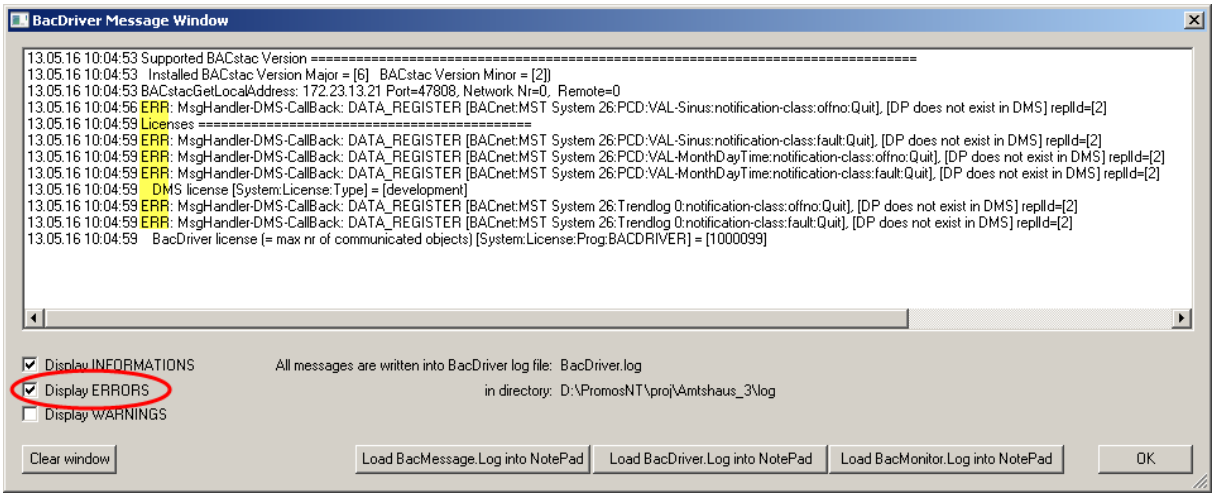


9.3.4.4 The message window

The message window is a non-modal (e.g. it can remain open for runtime without influencing the BacDriver functionality) message window, mainly for error and system info display. It can be displayed anytime via **Menu, View, Message** Window.



If an error is detected during the BacDriver start and the option **Display ERRORS** is selected, then the message window will be displayed **automatically:**



Example of an error display:

- BACstac service is not installed or the version is not compatible
- A connection to DMS cannot be established
- BACnet read/write error
- Non existing data point in DMS
- A DMS name exceeds the maximum length of 80 characters

Example of an info display:

- The number of scanned objects exceeds the licensed number
- A device restart was
- An unknown BACnet property is detected which is not included in the corresponding BMO template

After the BacDriver start, the following system specifications are written in the message window:

- Current licenses
- Detected BACstac version
- Detected BacDriver address, network number, and specification whether remote or local network

Configuration

The Message Window display options will be saved in **BacDriver.cfg** in the **[Monitor]** section:

```
[Monitor]

; For Message Window: if MsgWinShowInfo=1 then information messages will be displayed, othe
; (but they are still written into the BacDriver.log)
MsgWinShowInfo=1
; For Message Window: if MsgWinShowWarn=1 then warning messages will be displayed, otherwis
; (but they are still written into the BacDriver.log)
MsgWinShowWarn=0
; For Message Window: if MsgWinShowErr=1 then error messages will be displayed, otherwise n
; (but they are still written into the BacDriver.log)
MsgWinShowErr=0
```

9.3.5 Configuration

All setting options and other options are stored in a configuration file (ASCII text file).
The file must have the name "**BacDriver.cfg**" and saved in the current project directory under \DRV.

The "**BacDriver.cfg**" configuration file is generally valid for all devices.

It is also possible (and generally also useful) to define a separate configuration for each device.
The device-dependent configuration files must comply with the following naming rules: "**BacDevice-<InstNr>.cfg**", or "BacDevice-100020.cfg", see the next subchapter Device-dependent Configuration.

 Make sure that the configuration file of the text editor is saved in the **ANSI** format and not in the **UTF-8** format!
Otherwise in certain circumstances the options from the [Settings] section cannot be read by BacDriver and BacDriver only accepts the default values throughout.

 Each configuration file has a version key: Section [Settings] CfgVersion=Ge_1.134_2015-May-01. This allows an automatic update control, see Chapter Check for updates

Example of a configuration file:

```
; Do not delete this line! Section [Settings] should not be on the 1st line!
; (Just in case that this file was saved in UTF-8 format...)
;

; ====> Individual (custom) configurations per device <====
; All options, which have the additional comment "(===> This option can be configured per devic
; can be configured in separate device config file "BacDevice-<InstNr>.cfg" (which is device de
; The individual device config files have higher priority than the general BacDriver.cfg config

[Settings]

; Language and version nr of this config file. Needed for automatic check for updates. DO NOT M
CfgVersion=En_1.257_2016-Dec-09
;========================= English version ================================================

; Option for update check of BacDriver.cfg and BMO.dms from last setup,  Default: CheckForUpdat
; If <project>\drv\BacDriver.cfg is of older version than <installation path>\cfg\BacDriver\Bac
; then <project>\drv\BacDriver.cfg will be updated with <installation path>\cfg\BacDriver\BacDr
; and as well <project>\cfg\BMO.dms will be updated with <installation path>\cfg\BacDriver\BMO
; If an update is performed, then both updated files (BacDriver.cfg and BMO.dms)
; will be first backuped (with date/time stamp included in file names).
; If CheckForUpdates=0 (this is default): no update checks will be performed.
; If CheckForUpdates=1: if an update checks is indicated, then a Yes/No MsgBox will be displaye
```

```
;   "Check for update: a new BacDriver configuration / BACnet BMO object template is available"
;    if answer = Yes update will be performed, otherwise not.
; If CheckForUpdates=2: an update check and a possible update will always be performed.
CheckForUpdates=0

; BacDriver's process ID: BacDriver registers with following ID (as well as PC's IP adr) all CO
; subscriptions. This ProcessIDcov must be unique inside a PC system.
ProcessIDcov=1

; If DMSRoot is empty (=default), then all scanned BacObjects from BACnet will be inserted in D
; directly under their name (assembled from the "object-name" or "description" property, depend
; on the option ScanBacToDMSname - see below), otherwise the DMSRoot name will be preceeded.
; ==> This option is only relevant if option ScanDMS=0 e.g. only if BACnet scan is activated <=
DMSRoot=BACnet

; DeviceRoot: gives a posibility to insert all BacObjects of a belonging device in DMS under a
; node, e.g. <DeviceName> or <DeviceName>-<DeviceInstNr> or <anyDeviceNameString>-<DeviceInstNr
; Depending if DMSRoot option is defined (see above), DeviceRoot will succeed DMSRoot,
; e.g. BacObjects are in DMS inserted under: <DMSRoot>:<DeviceRoot>
; If DeviceRoot=1: then DeviceRoot node name = <DeviceName> (= object-name property of the devi
; If DeviceRoot=2: then the device instance nr is appended to the DeviceRoot: <DeviceName>-<Dev
; If DeviceRoot=<anyDeviceNameString>: then DeviceRoot node name is <anyDeviceNameString>-<Devi
; Default = DeviceRoot is empty: e.g. all device BacObjects are NOT grouped under a common Devi
; ==> This option is only relevant if option ScanDMS=0 e.g. only if BACnet scan is activated <=
DeviceRoot=1

; DMS connection: PC name, hostname or TCP/IP adr: e.g. for local DMS = "."
; or remote "172.18.0.217:9010" or "www.host.ch".
; If a DMS port nr will be attached (:9010), then TCP/IP connection will be used, otherwise pip
; Example: DMSConn=10.0.0.142:9010
; Default: local DMS "."
DMSConn=.

; BACSTAC Destination network number: 0=local (default), 0xFFFF=65535=global, else remote.
BacnetDestNETnr=65535

; Optional arguments for the WhoIs request, which limit the range of Device instance numbers
; expected to respond.
; In the defined range must be found also all device instance numbers from the file
; BacDevicesToScanFile, otherwise they cannot be scanned.
; Default =-1 if not used, = no range restriction for the WhoIs request
; The device instance must be:
;   (1) unique within the BACnet internetwork, and
;   (2) an integer number in range of 0 to 4.194.302 = 0..0x3FFFFE
WhoIsDeviceInstNrMin=-1
WhoIsDeviceInstNrMax=-1

; Interval in seconds in which time the WhoIs-BACnet responses are awaited (default=10 secs).
; If WhoIsTimeout=0 then no WhoIs-BACnet request is issued.
WhoIsTimeout=10

; Option for the selective device-Scan: if BacDevicesToScanFile=file name and if this file exis
; in <proj>\drv folder, then only those devices will be scanned, which are defined in the file
; and are as well activated (activation option is in the 3rd column=1).
; Required condition is, that all listed device instance numbers are located within
; the range defined above: WhoIsDeviceInstNrMin/Max
; csf-format of BacDevicesToScanFile: all empty and comment (1st character ';') lines will be
; ignored, 1st line will be skipped (column title),
; all following lines: ID; Device Instance Number; Activations Option; Comment...
;  Default = no selective scan (no file name)
BacDevicesToScanFile=DeviceList.csv

; Scan source. (===> This option can be configured per device <===)
; if ScanDMS=0 (=default), then only BACnet is scanned.
; if ScanDMS=1, then only DMS is scanned.
```

```
; if ScanDMS=2, then first DMS is scanned and subsequently BACnet, while only such objects will
;               inserted into DMS which are not yet contained there.
; ==> If ScanDMS=1 then options DMSRoot, DeviceRoot and [Filter] have no relevancy <==
ScanDMS=0

; If during a BACnet scan an unknown property is detected, i.e. a property which is not contain
; in the according base BMO Bac_<ObjectTyp> template, then there are three possibilities how to
; - AddUnknownBACProperties=0 (=default): property will be ignored, not connected and not inclu
; - AddUnknownBACProperties=1: property will be connected and included into DMS but not into BM
; - AddUnknownBACProperties=2: property will be connected and included into DMS and into BMO
; (===> This option can be configured per device <===)
AddUnknownBACProperties=0

; Property ID which is the source of the string for the DMS object node name (sub-tree).
; (===> This option can be configured per device <===)
; From this string the plant code key is generated (DMS name) and the BMO template VLO name
; (aggregate template) extracted, see [Filter] section below.
; Default is ScanBacToDMSname=77 which is the ID of the "object-name" property.
; Another possibility is ScanBacToDMSname=28 which is the ID of the "desctription" property.
ScanBacToDMSname=77

; Scan period in seconds. (===> This option can be configured per device <===)
; if ScanPeriod=0 (=default), then no scanning occurs.
; if ScanPeriod=86400 then all detected devices, incl. devices which have been selective scanne
; (one after another) will be re-scanned once a day. New objects will be inserted into DMS,
; unused objects, which do no longer exist, will still remain in DMS.
; (future option: they will be then in this case deleted from DMS)
ScanPeriod=0

; Poll period in seconds. (===> This option can be configured per device <===)
; See also option [Trending] PollTrends. If set =1 the also trend-log buffer data will be polle
; During this period, all objects and their properties of all devices will be sequentially poll
; The resulting poll interval is therefore calculated as following: poll period divided by the
; of all objects. The Pollperiode should be chosen so that the poll interval is not less
; than ~ 500-100 ms. If PollPeriod=0, then no polling occurs.
; Default = 1 hour=3600 secs. Shortest subscription period value is 900 = 15 minutes.
PollPeriod=3600

; Poll period in milliseconds: (===> This option can be configured per device <===)
; For polling of visualized (=opened screens in GE/WEB viewer) properties.
; if VisPollPeriod=0 -> no visualized properties will be polled.
; Default = 0 e.g. no visualized properties will be polled.
; Recommended value = 4000 (= 4 second).
VisPollPeriod=4000

; Watchdog period in seconds: poll the device object "program" (all properties contained in
; the template "BMO:Bac_program") and update it in DMS into
; "System:Driver:BacDriver:Device-<xy>:_Program". The template must contain at least the proper
; "program-state". In addition there will be polled also the property "system-status" of the de
; object itself: "System:Driver:BacDriver:Device-<xy>:system-status".
; If all OK: .."system-status" = "operational" and .."_Program:program-state" = "running"
; if WDPeriod=0, then no watchdog polling occurs.
; Default = 120 secs
; Shortest subscription period value is 10 seconds.
WDPeriod=120

; Device restart notification via unconfirmed COV notifications broadcast: if a device restarts
; it broadcasts unconfirmed COV notifications: system-status, time-of-device-restart und last-r
; If this broadcast is detected and the sender device's instance nr. is on the active-devices l
; then all device's COVs will be resubscribed (provided either this option DeviceRestartNotif=1
; the corresponding device cfg line option in the device list in column 13 =1).
; If DeviceRestartNotif=1 then this option is then valid globally for all devices, regardless
; of any individual device cfg settings (in the corresponding device list file or device cfg fi
; If DeviceRestartNotif=-1 or this option is not present (commented out), then individual devic
; will be considered.
```

```
; (general default=0 e.g. option is not activated)
DeviceRestartNotif=1

; Automate device restart notification (via AddListElement): add BacDriver's adr(+netNr) or loc
; broadcast to the restart-notification-recipient property of every device object, but only the
;  a) not already contained and b) if there is no local broadcast already contained in the reci
; DeviceRestartAddNotif=1 : Add local broadcast (networkNr = 0, no adr)
; DeviceRestartAddNotif=2 : Add BacDriver's local adr + network number
; If DeviceRestis respondedartAddNotif=1/2 then this option is then valid globally for all devi
; of any individual device cfg settings (in the corresponding device list file or device cfg fi
; If DeviceRestartAddNotif=-1 or this option is not present (commented out),
; then individual device cfg settings will be considered.
; (general default=0 e.g. option is not activated)
DeviceRestartAddNotif=1

; Use confirmed or unconfirmed notification for COV subscription
; If UseCOVconfirmedNotification=1 then every COV service is required to be responded by an ack
; If UseCOVconfirmedNotification=0 then COV service will be an unconfirmed one, no response is
; (general default=1 e.g. option is activated)
UseCOVconfirmedNotification=1

; COV (Change-Of-Value) resubscription period in seconds.
; If COVresubscrPeriod=0, then no COV will ever be resubscribed.
; Default = 1 hour=3600 secs
; Shortest subscription period value is 900 = 15 minutes.
COVresubscrPeriod=3600

; COV subscription lifetime in seconds. Default 1 week = 604800
; After the expiration is the subscription canceled and an automatic resubscribe procedure is i
; The value COVsubscriptionLifetime=0 indicates an indefinite lifetime,
;     which shall not automatically cancelled after any time (should NOT be used).
; Shortest lifetime value is 3600 = 60 minutes.
COVsubscriptionLifetime=7200

; Option to automatically subscribe COVP of priority-array properties of all objects of all dev
; if available. Default = 0, e.g. priority-array will not be subscribed (if present).
COVPprioArr=1

; BacDriver as BACnet Time Master (this option is linked to COVresubscrPeriod):
; If the option TimeMaster=1 (and COVresubscrPeriod>0), then at each COV-Resubscription period
; of a device object will be its time and date brought to the DMS time.
; To all BACnet devices there are both services supported:
;    TimeMaster=1 Time-Synchronization (local DMS time) = default
;    TimeMaster=2 UTC-Time-Synchronization (Greenwich mean time)
TimeMaster=1

; StartUpDelay in seconds
; After all BacDriver's start sequences have been finished (scan, etc),
; then until this time expires, it will be awaited, before the scan/poll/WatchDog and COVresub
; timer functions will be activated.
StartUpDelay=0

; BacDriver start option: shall a following message box appear before the start or not?
; If =1, then the question will be, whether the first start scan shall be executed or not.
; -> if BacAskIfStartScan=1 -> display Yes/No MsgBox: "Scan: update VLO templates from BMO?"
;    if answer = Yes scan will be performed, otherwise not.
; -> if BacAskIfStartScan=0 -> no MsgBox will be displayed, the network will be scanned
; and accordingly to the options in the [Filter] section, DMS will be updated with BMO templat
; default=0 then no question message box before the start
BacAskIfStartScan=0

; Global Default BACnet write priority.
; This setting is valid for all write operations of all object properties of all devices:
; 1 Manual-Life Safety
; 2 Automatic-Life Safety
```

```
; 3 - 4 freely available
; 5 Critical Equipment Control
; 6 Minimum On/Off  ------> must never be used: is reserved for BACstac
; 7 freely available
; 8 Manual Operator
; 9 - 16 freely available (default =15)
BacWritePrio=15

; Enable individual BACnet write priorities pro object, valid for all write operations.
; (===> This option can be configured per device <===)
; If =1, then all BACnet objects will be extended with additional auxiliary DMS data points:
; "_Prio" type DWS and "_PrioSetNULL" type BIT.
; If the value of the "_Prio" data points >= 1 and <=16, then this priority will be used for al
; write operations of all properties of this BACnet object, instead of the global
; BACnet priority "BacWritePrio" (see above).
; If the trigger DP "_PrioSetNULL" is set = ON, then NULL will be written into priority-array
; at the according priority position due to "_Prio".
; default=0 then no individual write priorities are possible, only the glogal BacWritePrio
UseIndividualObjWritePrio=0

; For nested (grouped)-VLO template objects:                (e.g. "UG01:M01:E1:Bac_BET01")
; BacPushUpBMONAME=1: -> Determine from the embedded BMO-VLO template object the original BMO N
;   (="UG01:M01:E1:Bac_BET01:RM_Ein"), rename ..:NAME -> ..:BacNAME
;   and copy the contained string without the last segment ("RM_Ein") one DMS level higher
; BacPushUpBMONAME=2: -> Determine from the embedded BMO-VLO template object the "description"
;   and copy the contained string one DMS level higher into "NAME"
; BacPushUpBMONAME=3: -> Determine from all embedded BMO-VLO template object "description" prop
;   the "highest common name" and copy it one DMS level higher into "NAME"
;   e.g. "Pumpe 1 Freigabe" and "Pumpe 1 SM" and "Pumpe 1 RM" --> result to be copied is "Pumpe
; BacPushUpBMONAME=3 = default
BacPushUpBMONAME=3

; For nested (grouped)-VLO template objects: option for BacDriver BMO-update in scan procedure
; delete all sub-BMO objects, which are part of a (grouped)-VLO template objects,
; but they are not contained in the actual PLC, e.g. its property "NAME" is empty (unused).
; BacDelUnusedBMONAME=1: all unused sub-BMO objects will be deleted after an object scan
; BacDelUnusedBMONAME=0: (=default) no test if NAME empty, no delete.
BacDelUnusedBMONAME=1

; Option for scan:
; BacForceBMOUpdate=1: all BMO template objects will be updated into DMS, even if they are
; in current DMS already present; in this case their value will be overwritten with the one fro
; BacForceBMOUpdate=0: (=default) only such BMO template objects will be updated, which do not
; in current DMS.
BacForceBMOUpdate=0

; After scan has completed: option to compile PLS-functions in DMS
; compileDMSFunc=1: (=default) all DMS functions (especially new functions which have been adde
; by scan's BMO-update) will be compiled
; compileDMSFunc=0: DMS functions will not be compiled (new functions which have been added
; by scan's BMO-update will remain inactive)
CompileDMSFunc=1

; After scan has completed and possibly PLS-functions has been compiled (if CompileDMSFunc=1):
; Option to execute functions in DMS:
; ExecuteDMSFunc=1: all functions (particular new functions, which Were added by the scan-BMO-u
; will be performed.
; ExecuteDMSFunc=0: (=default) all new added functions from Scan's BMO-update will not be perfo
ExecuteDMSFunc=0

; Scan Logging: if =1 or 2: scan details will be logged into BacDriverScan.log
; If ScanLogActive=1 then all objects of scanned devices will be listed
; If ScanLogActive=2 then all object properties of scanned devices will be listed
; ScanLogActive=1: =Default : all objects of scanned devices will be listed
ScanLogActive=1
```

```
; Watchdog: through monitoring, the DMS can recognise a BacDriver process which is not respondi
; and restart BacDriver.exe as necessary.
; MngWatchDogActive=0: BacDriver will NOT be monitored by DMS. If it stops working (crashes)
; then the BacDriver.exe program will not be restarted.
; MngWatchDogActive=1: (=default) BacDriver will be monitored by DMS. If it stops working (cras
; then the BacDriver.exe program will be restarted.
MngWatchDogActive=1

; Schedule objects: if following option is activated, then "Act" flags will be used for writing
; values into the schedule object. The "Time" value will be written into the schedule object or
; if "Act" =ON, else the "Time" field input will be ignored. If a new "Time" segment has to be
; to the weekly-schedule property, then its "Act" flag must be set =ON.
ScheduleObjUseAct=0



[Filter]

;=============================================================================================
; (===> This whole section can be configured per device <===)
; ==> This section is only relevant if option [Settings]ScanDMS=0 e.g. only if BACnet scan is a

; Mainly used to extract the BMO template VLO name (aggregate template) from the object's BACne
; This name (assembled from the "object-name" or "description" property, depending
; on the option ScanBacToDMSname) is assumed to be structured and correspond to the plant code
; Several positions (BMOLevels) can be specified inside the BACnet name for different templates
; So are several definable BMO possible (BMOStartPos1, BMOEndPos1, BMOLevel1, BMOStartPos2, ..
; and thereby nested as well as parallel integrated BMO structures.
; Following filter methods will be applied to OBJECT_NAME in this order:
; 1) BMOStartPos..BMOEndPos: BMO template name is extracted by position
; 2) BacToDMSKeyMask: fix the DMS levels; insert ':' DMS delimiter into corresponding positions
; 3) BacToDMSErase: single letters/delimiters inside the object's BACnet name can be deleted.


; 1) If a fixed part i.g. a segment of the plant key code (object's NAME property) is intended
; to deliver the BMO template name, then its start and end positions must be defined here:
; position 1 = first letter (not 0). Position is inactiv (vacant) if = 0 or -1 or empty.
; Start and end position letters are part of the resulting BMO template name.
; -> If only one BMO position is defined, then the keywords BMOStart/EndPos/Level
;    may be written without indexes.
; -> If several BMO positions are defined, then the position sequences must be gaples
;    and its index must start with 1.
; To every position also belongs the level specification (= order of the plant key code stage),
; e.g. DMS name level, where the BMO template will be inserted. In other words, the number
; of the segment in the position mask 'BacToDMSKeyMask' - see below.

;-------------------------------------------------------------
; Example A) for plant key code "021701L304ULA_E01PPE001TL01REP"
;            its positions:     "12345678901234567890123456789O"
; In this example there are only 2 template names used (others are kept in 'reserve'),
; starting with BMOStartPos1, BMOEndPos1, BMOLevel1 for the first one
; and ending with BMOStartPos2, BMOEndPos2, BMOLevel2 for the second (and last) template name.
; BMO template name = "L" <- "021701L304ULA_E01PPE001TL01REP"
;BMOStartPos1=7
;BMOEndPos1=7
;BMOLevel1=3
; BMO template name "PPE" <- "021701L304ULA_E01PPE001TL01REP"
;;BMOStartPos2=18
;;BMOEndPos2=20
;;BMOLevel2=8
;
; BMOLevel determines the insert position in the DMS name, see BacToDMSKeyMask:
;      BacToDMSKeyMask     = "....:..:.:...:...:....:...:...:...."
;   -> DMS-Name            = "0217:01:L:304:ULA_:E01:PPE:001:TL01"
```

```
;                                    |              |
;                       3rd stage: BMO-Obj "L"        |
;                                    7th stage: BMO-Obj "PPE" - but will be inserted at 8th level
;
;      see above: BMO objects will be attached in these stages
;
; BMO template name "1L" <- "021701L304ULA_E01PPE001TL01REP"
;BMOStartPos2=6
;BMOEndPos2=7
;BMOLevel2=4
; BMO template name "304" <- "021701L304ULA_E01PPE001TL01REP"
;BMOStartPos3=8
;BMOEndPos3=10
;BMOLevel3=5
; BMO template name "ULA" <- "021701L304ULA_E01PPE001TL01REP"
;BMOStartPos4=11
;BMOEndPos4=13
;BMOLevel4=6
; BMO template name "_E01" <- "021701L304ULA_E01PPE001TL01REP"
;BMOStartPos5=14
;BMOEndPos5=17
;BMOLevel5=7
; BMO template name "001" <- "021701L304ULA_E01PPE001TL01REP"
;BMOStartPos6=21
;BMOEndPos6=23
;BMOLevel6=9
; BMO template name "TL01" <- "021701L304ULA_E01PPE001TL01REP"
;BMOStartPos7=24
;BMOEndPos7=27
;BMOLevel7=10

; 2) Position mask for the BACnet name (plant key code). Each '.' stands for a plant key code l
; A ':' means, that on this positions (between two letters) a DMS ':' delimiter will be inserte
; Example for plant key code "021701L304ULA_E01TRU001TL01"
;      BacToDMSKeyMask =     "....:..:.:...:...:....:...:...:...."
;   -> DMS name        =     "0217:01:L:304:ULA_:E01:TRU:001:TL01"
;BacToDMSKeyMask=....:..:.:...:....:...:...:...:....
BacToDMSKeyMask=

; 3) BACnet name -> conversion -> DMS name: all listed letters of the 'BacToDMSErase' keys will
; be deleted in BACnet string (=Property OBJECT_NAME).
; No spaces, several letters possible.
; Default =empty (no entry)
; Example: "0217:01:L:304:ULA:_E01:TRU:001:TL01" -> "0217:01:L:304:ULA:E01:TRU:001:TL01" (;BacT
BacToDMSErase=

; Write operation: BACnet str-property in DMS -> conversion -> BACnet (write string property to
; All listed letters of the 'DMSToBacStrErase' keys, which are present in the string property,
; will be deleted in the string, before it is written to the BACnet.
; The BACnet string property in DMS remains unchanged.
; Several letters possible. Default = empty = no deletion.
; Example: description property = ""70K1 switch HMI L301 AS X"" -> "70K1 switch HMI L301 AS X"
DMSToBacStrErase="



[BlackListScan]

;=============================================================================================
; (===> This whole section can be configured per device <===)
; Options in this section are only then relevant if scan source is BACnet, e.g. [Settings] Scan
; If there are no keywords defined in this section, then BlackListScan wan't be performed.

; -------> "ObjName<Nr>" <-----------
; Strings defined in folowing "ObjName<Nr>" keynames are compared (regex like) with BACnet prop
```

```
; "object-name" (of the according device) and if they match then the object is not scanned and
; in DMS. The keynames must be numbered, but don't need to be in sequence and there may gaps ex
;    ObjName23=PCD : all objects which name matches exactly "PCD" will be ignored.
;    ObjName24=PCD* : all objects which name begins with "PCD" will be ignored
;    ObjName25=*PCD : all objects which name ends with "PCD" will be ignored
;    ObjName26=*PCD* : all objects which name contains "PCD" will be ignored

; -------> "ObjDesc<Nr>" <-----------
; Strings defined in folowing "ObjDesc<Nr>" keynames are compared (regex like) with BACnet prop
; "description" (of the according device). Same rules as above.
;    ObjDesc1=PCD : all objects which description matches exactly "PCD" will be ignored.
;    ObjDesc2=PCD* : all objects which description begins with "PCD" will be ignored
;    ObjDesc3=*PCD : all objects which description ends with "PCD" will be ignored
;    ObjDesc4=*PCD* : all objects which description contains "PCD" will be ignored

; -------> "ObjType<Nr>" <-----------
; Objects defined in folowing "ObjType<Nr>" keynames are compared with scanned BACnet object
; of the according device and if they match then the object is ignored and not included into DM
; The keynames must be numbered, but don't need to be in sequence and there may gaps exist.
; Keynames "ObjTyp<Nr>" define object(s) by type and instance nr.
;    ObjType1=BO 5 : binary-output instance nr=5 will be ignored
;    ObjType4=AI 9-13 : analog-input instances in range from 9 till 13 will be ignored
;       Objects can be identified by: 1) 2..3-chars abbreviation e.g. "AO" = "analog-output"
;       or written out: "analog-output". Capitalization does not matter.
;       Abbreviation list: "analog-input" oder "ai", "analog-output" "ao", "analog-value" oder
;       "binary-input" oder "bi", "binary-output" oder "bo", "binary-value" oder "bv", "calenda
;       "command" oder "cmd", "event-enrollment" oder "eeo", "file" oder "fi", "group" oder "gr
;       "multi-state-input" oder  "msi", "multi-state-output" oder "mso", "notification-class"
;       "schedule" oder "sch", "averaging" oder "ave", "multi-state-value" oder "msv", "trend-l
;       "life-safety-point" oder "lsp", "life-safety-zone" oder "lsz", "accumulator" oder "ac",
;       "pulse-converter" oder "pc", "event-log" oder "el", "trend-log-multiple" oder "tlm",
;       "load-control" oder "lc", "structured-view" oder "sv", "access-door" oder "ad"


[WhiteListScan]

;=============================================================================================
; (===> This whole section can be configured per device <===)
; Options in this section are only then relevant if scan source is BACnet, e.g. [Settings] Scan
; If there are no keywords defined in this section, then BlackListScan wan't be performed.

; -------> "ObjName<Nr>" <-----------
; Strings defined in folowing "ObjName<Nr>" keynames are compared (regex like) with BACnet obje
; "object-name" (of the according device) and only if they match then the object will be scanne
; included into DMS, while all other objects will be ignored (no scanned, not included into DMS
; The keynames must be numbered, but don't need to be in sequence and there may gaps exist.
;    ObjName23=PCD : all objects which name matches exactly "PCD" will be scanned and included
;    ObjName24=PCD* : all objects which name begins with "PCD" will be scanned and included int
;    ObjName25=*PCD : all objects which name ends with "PCD" will be scanned and included into
;    ObjName26=*PCD* : all objects which name contains "PCD" will be scanned and included into

; -------> "ObjDesc<Nr>" <-----------
; Strings defined in folowing "ObjDesc<Nr>" keynames are compared (regex like) with BACnet obje
; "description" (of the according device). Same rules as above.
; The keynames must be numbered, but don't need to be in sequence and there may gaps exist.
;    ObjDesc1=PCD : all objects which description matches exactly "PCD" will be scanned and inc
;    ObjDesc2=PCD* : all objects which description begins with "PCD" will be scanned and includ
;    ObjDesc3=*PCD : all objects which description ends with "PCD" will be scanned and included
;    ObjDesc4=*PCD* : all objects which description contains "PCD" will be scanned and included

; -------> "ObjType<Nr>" <-----------
; Objects defined in folowing "ObjType<Nr>" keynames are compared with scanned BACnet objects
; of the according device and only if they match then the object will be scanned and included i
; while all other objects will be ignored (no scanned, not included into DMS).
; The keynames must be numbered, but don't need to be in sequence and there may gaps exist.
```

```
;    ObjType1=BO 5 : binary-output instance nr 5 will be scanned
;    ObjType4=AI 9-13 : analog-input instance nr from 9 to 13 will be scanned
;       Objects can be identified by: 1) 2..3-chars abbreviation e.g. "AO" = "analog-output"
;       or written out: "analog-output". Capitalization does not matter.
;       Abbreviation list see above in [BlackListScan]



[Alarming]

;=============================================================================================
; Following options are for alarm and event notification handling.
; (===> Some options of this section can be configured per device <===)

; Alarm and event notification activation: if =1 then the alarm and event notification from BAC
; will be enabled (default =1)
; This option must be activated for trending (if UseTrending=1) and alarming (if UseAlarming=1)
AlarmNotification=1

; (===> This option can be configured per device <===)
; BACnet alarming activation: if =1 (and AlarmNotification=1) then the alarming via
; notification-class:fault:ALM:... and notification-class:offno:ALM:... will be enabled (defaul
; =0: - run time: ignore NC notification
;     - scan BAC: no BMO import of NC..ALM DP's   (scan DMS: no action)
; =1: - run time: use NC notification             ----> this is default
;     - scan BAC: no action (complete BMO import) (scan DMS: no action)
Use_NC_Alarming=1

; (===> This option can be configured per device <===)
; 'Classic' alarming activation: if =1 then the alarming via
; present-value:ALM:... will be enabled (default =0) (this option is independent from AlarmNoti
; =0: - scan BAC: no BMO import of present-value:ALM DP's  (scan DMS: no action) -> this is de
;     - run time: no action
; =1: - scan BAC: no action (complete BMO import)          (scan DMS: no action)
;     - run time: no action
Use_PV_Alarming=0


; Alarm and event notification message-text transfer: if =1 then the alarm and event notificati
; message-text pass-through from BACnet into DMS/AlmMng will be enabled (default =0)
AlarmTextTransfer=0

; -----> CAUTION! Next option is not yet operational. Reserved for future solution
; Activation of alarm and event acknowledgement with two Quit flags (QuitGet and QuitLve):
; if =0 then the alarm and event acknowledgement is handled with one Quit flag (default=0)
; if =1 then the alarm and event acknowledgement is handled with two Quit flags
; -------> this feature (acknowledgement handled with two Quit flags) is NOT yet ready to use,
;          AlarmViewer is not yet prepared to display these two Quit's.
; default =0 (-------> option =1 is reserved for future solution)
QuitGetAndQuitLveFlags=0

; Alarm priority mapping type: defines the BACnet to Visi.Plus priority mapping (default =2)
; 1 Pass-through: the BACnet priority values will be written directly into the DMS data-point
; 2 Visi.Plus-light: the BACnet priority values will be converted to 6 predefined priority clas
AlarmPrioMapType=2

; Alarm priority class: when Visi.Plus-light mapping is enabled the 6 priority classes has to b
; defined by the following mapping schema: Visi.Plus priority class = BACnet priority value ran
; Pay attention that the priorities 5 and 6 are interchanged in comparison to BACnet definition
; (default AlarmPrio1=0-31) Life safety / Safety
; (default AlarmPrio2=32-63) Property safety / Security
; (default AlarmPrio3=64-95) Supervisory / Technical
; (default AlarmPrio4=96-127) Trouble / Fault
; (default AlarmPrio5=192-255) Miscellaneous lower priority / Status
; (default AlarmPrio6=128-191) Miscellaneous higher priority / Maintenance
```

```
AlarmPrio1=0-31
AlarmPrio2=32-63
AlarmPrio3=64-95
AlarmPrio4=96-127
AlarmPrio5=192-255
AlarmPrio6=128-191



[Trending]

;=============================================================================================
; Following options are for trend-objects and log-buffer alarm/event notification handling.
; -------> For Trending also following option must be activated: [Alarming] AlarmNotification=1

; If UseTrending=0 then all kind of trend operations will be disabled. Default=1.
UseTrending=1

; Poll trend-log buffer data (see [Settings] PollPeriod)
; If PollTrends=1 then trend-log buffer will be polled within the period defined in [Settings]
; Default =0 = deactivate trend polling. Trends will be then read in only if manual trigger "_P
;            in DMS trend-log will be set _Pull=ON or on log-buffer alarm/event notification.
PollTrends=1

; If ReadTrendsAtStartup=1 then at the BacDriver's startup, just before COVs are subscribed,
; log-buffer of all trend-log objects are read in, starting with the last date/time stamp recor
; Default =0, no trend data from log-buffers will be read in at startup.
ReadTrendsAtStartup=0

; If referenced trended object property (referenced in trend-log in log-device-object-property)
; DMS also another TRD data point and object defined for trending via HDAMng:
; -> then if the option Set_HDA_TRD_Off=1 then BacDriver deactivates HDAMng TRD object in DMS
;    by resetting TRD:CHANGES, TRD:DIFF and TRD:TIME to OFF.
; -> If Set_HDA_TRD_Off=0, then no modifications are made in object's trended property in DMS i
; ==> This option is only relevant if option UseTrending=1 e.g. only if trend operations are en
; Default =1 = deactivate trending via HDAMng.
Set_HDA_TRD_Off=1


[Server]
;=============================================================================================
; In the moment for future development reserved.
; BacDriver Server process ID. This ProcessID must be unique inside a PC system.
ProcessID=98

; Server functionality activation
Active=0

[AdvancedClient]
;=============================================================================================
; In the moment for future development reserved.

; Advanced Client functionality activation
Active=0


[Monitor]

;=============================================================================================
; Following options are for monitor window pane display.

; Monitor window: if =1 then the data broadcast and notification traffic from PLCs will be disp
; (default =1)
MonitorActive=1
```

```
; Monitor Filter: if this setting is activated, then only such objects will be monitored,
; which fulfill the filter option (filter string): "<devID>:<ObjTyp>,<ObjID>"
; Example: MonitorFilter=5010:5,263
MonitorFilter=

; Monitor COVs (Change of Value) notifications, default =0
MonitorCOV=0

; Monitor UNCOVs (Change of Value) Notifications, default =0
MonitorUNCOV=0

; Monitor EVTs (Alarm and Event) notifications, default =0
MonitorEVT=0

; Monitor devices restart notifications, which are delivered by an Uncofirmed COV, default =0
MonitorRestart=0

; Monitor devices IAM broadcasts, default =1
MonitorIAM=1

; Monitor objects COV subscriptions, default =0
MonitorSub=0

; Monitor objects polling, default =0
MonitorPoll=0

; Monitor objects scanning, default =0
MonitorScan=0

; Monitor time synchronization (TimeMaster), default =0
MonitorTimeSynch=0

; Monitor activated for DMS writes -> BACnet (incl. following object-poll), default =0
MonitorBacWrt=0
; Monitor objects trending, default =0
MonitorTrends=0

; If =1 then monitoring goes into the BacMonitor.log file, regardless of whether MonitorActive=
; (default=0)
MonitorToFile=0

; Nr. of buffer lines in monitor window. Default=1000, max. 10000, min. 100.
MonitorLines=1000

; For Message Window: if MsgWinShowInfo=1 then information messages will be displayed, otherwis
; (but they are still written into the BacDriver.log)
MsgWinShowInfo=1
; For Message Window: if MsgWinShowWarn=1 then warning messages will be displayed, otherwise no
; (but they are still written into the BacDriver.log)
MsgWinShowWarn=0
; For Message Window: if MsgWinShowErr=1 then error messages will be displayed, otherwise not
; (but they are still written into the BacDriver.log)
MsgWinShowErr=1


[Debug]

;=============================================================================================
;Following options are only for debugging purposes. DO NOT CHANGE them to other than the defaul

; Option for BACnet objects readout method, Default: BacObjectReadoutOnce=1
; if BacObjectReadoutOnce=1 then read all object properties in a single command
; -> fast method, but not neccessarily compatible with all BACnet devices
; if BacObjectReadoutOnce=0 then read each object property in a single command
```

```
; -> slow method, but compatible with all BACnet devices
; CAUTION: if BacObjectReadoutOnce=0 then Black- und WhiteListScan functions will be deactivate
BacObjectReadoutOnce=1

; General switch for COV subscription: if COVactive=0 -> no COV subscription at all, no COV not
; Default=1
; To switch COVs off might be useful to reduce the network load, if at the same time the option
; VisPollPeriod is activated (e.g. VisPollPeriod=1000, e.g. all visualized properties are polle
; each 1 second) and alarming is not dependent on incoming COVs.
COVactive=1

; General switch for unconfirmed COV subscription (functionality for restart notification):
; if UnconfCOVactive=0 -> no unconfirmed COV subscription at all, no unconfirmed COV notificati
; Default=1
; To switch unconfirmed COVs off might be only useful for testing or debugging purposes.
UnconfCOVactive=1

; COV is always first unsubscribed before a new (Re)-Subscription will be performed.
; Default =0 -> only new subscription request with the same existing COV context is made.
COVUnsubBeforeResub=0

; If BacDriver exits, then it removes its local address/netNr from all device-objects,
; restart-notification-recipients lists, if DeviceRestartRemoveAdrNotif=1 (=default)
; For debug purposes this function can be switched off by DeviceRestartRemoveAdrNotif=0
DeviceRestartRemoveAdrNotif=1

; All BACnet CharacterStrings are enclosed in double quotes (e.g. "On_Low").
; Before sending the strings to DMS, BacDriver filters these quotes away. If DMS writes the mod
; strings back, then the leading and trailing quotes are restored again in BACnet automatically
; For debug purposes this function can be switched off by DeleteStringQuotes=0, default = 1.
DeleteStringQuotes=1

; Just for degugging: trace logging: if =1 (default = 0) then activate detail
; logs into BacDriverTrace.log
TraceActive=0

; Just for degugging: profiling logging: if =1 (default = 0) then activate profiling
; logs into BacDriverProfile.log
ProfileActive=0

; Just for degugging: Sleep time in milli seconds: if >0 then after each scanned and registered
; there will be a Sleep-time inserted. (Default = 0)
ScanRegSleep=0

; Just for degugging: master timer thread period for poll/visual poll/watchdog/resubscription/s
; For debug or load/stress tests can be switched off or extended.
; Default =1000 ms = 1 second
BacTimer=1000

; Just for degugging: additional time stamp check for received properties from BACnet
; If ChkPollCOVTimeStamp=0 (=default) then no check is performed
; If =1 then time stamps are compared and if received property is older
; than the DMS one, then no the property will be not written into DMS
ChkPollCOVTimeStamp=0

; Just for degugging: priority setting for controlled competing between
; the BacTimer (polling/watchdog/resubscription) and the COV thread.
; If BacCOVPrio=0 (=default) then controlled competing is not activated (both threads have same
; BacCOVPrio=1..20 (max value), 20 is highest prio for the COV thread, 1 is lowest
; useful for heavy load stress tests for massive (high) COV frequency
BacCOVPrio=0

; For debugging/testing: if ShowTestMenu=1 then test menu entries in menu File will be visible
; These menus are for tesing of trending, file objects, etc.  (Default = 0)
ShowTestMenu=0
```

```
; For schedule object: if BacScheduleTypeToDMS=1 then the data types of the switched objects wi
; used/enforced in DMS. Under circumstances, the BMO template data types may be overwritten in
; If BacScheduleTypeToDMS=0 (=default), then the included BMO template data type is decisive an
; all BACnet data types will be converted accordingly.
; (default = 0)
BacScheduleTypeToDMS=0
```

9.3.5.1 Device-dependent configuration

Each device can have its own configuration file, which must meet the following naming rule:

"**BacDevice-<InstNr>.cfg**", e.g. "BacDevice-100020.cfg"

All options that are listed in this individual configuration files have higher priority than those of the general BacDriver.cfg.

The following options can currently be configured depending on the device:

```
[Settings]
ScanDMS
ScanPeriod
PollPeriod
VisPollPeriod
```

as well as the entire section

```
[Filter]
```

Example of a device-dependent configuration file (this template is in the setup installation directory ..\cfg\BacDriver):

```
; Do not delete this line! Section [Settings] should not be on the 1st line!
; (Just in case that this file was saved in UTF-8 format...)
;

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; This is a cfg template for an individual device/controller configuration.
;; All settings are listed here, which are currently possible to configurate.
;;
;; If you want to use it for a device, you must rename/copy this template file and replace
;; the <INSTNR> term by the according device instance number, e.g. "BacDevice-100020.cfg"
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

[Settings]

; Language and version nr of this config file. Needed for automatic check for updates. DO
NOT MODIFY!
CfgVersion=En_1.189_2015-Nov-29
;======================== English version
====================================================
```

```
; Scan source
; if ScanDMS=0 (=default), then only BACnet is scanned.
; if ScanDMS=1, then only DMS is scanned.
; if ScanDMS=2, then first DMS is scanned and subsequently BACnet, while only such
objects will be
;             inserted into DMS which are not yet contained there.
; ==> If ScanDMS=1 then options DMSRoot, DeviceRoot and [Filter] have no relevancy <==
ScanDMS=0

; Property ID which is the source of the string for the DMS object node name (sub-tree).
; From this string the plant code key is generated (DMS name) and the BMO template VLO
name
; (aggregate template) extracted, see [Filter] section below.
; Default is ScanBacToDMSname=77 which is the ID of the "object-name" property.
; Another possibility is ScanBacToDMSname=28 which is the ID of the "desctription" property.
ScanBacToDMSname=77

; Scan period in seconds
; if ScanPeriod=0 (=default), then no scanning occurs.
; if ScanPeriod=86400 then all detected devices, incl. devices which have been selective
scanned
; (one after another) will be re-scanned once a day. New objects will be inserted into DMS,
; unused objects, which do no longer exist, will still remain in DMS.
; (future option: they will be then in this case deleted from DMS)
ScanPeriod=0

; Poll period in seconds. During this period, all objects and their properties of all devices
; will be sequentially polled. The resulting poll interval is therefore calculated as following:
; poll period divided by the number of all objects. The Pollperiode should be chosen so that
; the poll interval is not less than ~ 500-100 ms.
; if PollPeriod=0, then no polling occurs.
; Default = 1 hour=3600 secs
; Shortest subscription period value is 900 = 15 minutes.
PollPeriod=0

; Poll period in milliseconds for polling of visualized (=opened screens in GE/WEB viewer)
properties.
; if VisPollPeriod=0 -> no visualized properties will be polled.
; Default = 0 e.g. no visualized properties will be polled.
; Recommended value = 4000 (= 4 second).
VisPollPeriod=0



[Filter]

;================================================================
================================
; ==> This section is only relevant if option [Settings]ScanDMS=0 e.g. only if BACnet scan is
activated <==

; Mainly used to extract the BMO template VLO name (aggregate template) from the object's
BACnet name.
; This name (= object-name property) is assumed to be structured and correspond to the
```

```
plant code key.
; Several positions (BMOLevels) can be specified inside the BACnet name for different
templates.
; So are several definable BMO possible (BMOStartPos1, BMOEndPos1, BMOLevel1,
BMOStartPos2, ...)
; and thereby nested as well as parallel integrated BMO structures.
; Following filter methods will be applied to OBJECT_NAME in this order:
; 1) BMOStartPos..BMOEndPos: BMO template name is extracted by position
; 2) BacToDMSKeyMask: fix the DMS levels; insert ':' DMS delimiter into corresponding
positions.
; 3) BacToDMSErase: single letters/delimiters inside the object's BACnet name can be
deleted.


; 1) If a fixed part i.g. a segment of the plant key code (object's NAME property) is intended
; to deliver the BMO template name, then its start and end positions must be defined here:
; position 1 = first letter (not 0). Position is inactiv (vacant) if = 0 or -1 or empty.
; Start and end position letters are part of the resulting BMO template name.
; -> If only one BMO position is defined, then the keywords BMOStart/EndPos/Level
;    may be written without indexes.
; -> If several BMO positions are defined, then the position sequences must be gaples
;    and its index must start with 1.
; To every position also belongs the level specification (= order of the plant key code stage),
; e.g. DMS name level, where the BMO template will be inserted. In other words, the number
; of the segment in the position mask 'BacToDMSKeyMask' - see below.

;---------------------------------------------------------------
; Example A) for plant key code "021701L304ULA_E01PPE001TL01REP"
;          its positions:     "12345678901234567890123456789O"
; In this example there are only 2 template names used (others are kept in 'reserve'),
; starting with BMOStartPos1, BMOEndPos1, BMOLevel1 for the first one
; and ending with BMOStartPos2, BMOEndPos2, BMOLevel2 for the second (and last)
template name.
; BMO template name = "L" <- "021701L304ULA_E01PPE001TL01REP"
BMOStartPos=7
BMOEndPos=7
BMOLevel=3
; BMO template name "PPE" <- "021701L304ULA_E01PPE001TL01REP"
BMOStartPos2=18
BMOEndPos2=20
BMOLevel2=8
;
; BMOLevel determines the insert position in the DMS name, see BacToDMSKeyMask:
;     BacToDMSKeyMask     = "....:.:...:...:...:...:...."
;   -> DMS-Name           = "0217:01:L:304:ULA_:E01:PPE:001:TL01"
;                                 |           |
;                   3rd stage: BMO-Obj "L"      |
;                                 7th stage: BMO-Obj "PPE" - but will be inserted at 8th level...
;
;     see above: BMO objects will be attached in these stages
;
; BMO template name "1L" <- "021701L304ULA_E01PPE001TL01REP"
;BMOStartPos2=6
;BMOEndPos2=7
```

```
;BMOLevel2=4
; BMO template name "304" <- "021701L304ULA_E01PPE001TL01REP"
;BMOStartPos3=8
;BMOEndPos3=10
;BMOLevel3=5
; BMO template name "ULA" <- "021701L304ULA_E01PPE001TL01REP"
;BMOStartPos4=11
;BMOEndPos4=13
;BMOLevel4=6
; BMO template name "_E01" <- "021701L304ULA_E01PPE001TL01REP"
;BMOStartPos5=14
;BMOEndPos5=17
;BMOLevel5=7
; BMO template name "001" <- "021701L304ULA_E01PPE001TL01REP"
;BMOStartPos6=21
;BMOEndPos6=23
;BMOLevel6=9
; BMO template name "TL01" <- "021701L304ULA_E01PPE001TL01REP"
;BMOStartPos7=24
;BMOEndPos7=27
;BMOLevel7=10



; 2) Position mask for the BACnet name (plant key code). Each '.' stands for a plant key code
letter.
; A ':' means, that on this positions (between two letters) a DMS ':' delimiter will be inserted.
; Example for plant key code "021701L304ULA_E01TRU001TL01"
;      BacToDMSKeyMask =     "....:.:.:...:....:.:...."
;   -> DMS name        =     "0217:01:L:304:ULA_:E01:TRU:001:TL01"
BacToDMSKeyMask=....:.:.:...:....:.:...:....

; 3) BACnet name -> conversion -> DMS name: all listed letters of the 'BacToDMSErase'
keys will
; be deleted in BACnet string (=Property OBJECT_NAME).
; No spaces, several letters possible.
; Default =empty (no entry)
; Example: "0217:01:L:304:ULA:_E01:TRU:001:TL01" ->
"0217:01:L:304:ULA:E01:TRU:001:TL01" (;BacToDMSErase=_)
BacToDMSErase=



; Write operation: BACnet str-property in DMS -> conversion -> BACnet (write string property
to BACnet):
; All listed letters of the 'DMSToBacStrErase' keys, which are present in the string property,
; will be deleted in the string, before it is written to the BACnet.
; The BACnet string property in DMS remains unchanged.
; Several letters possible. Default = empty = no deletion.
; Example: description property = ""70K1 switch HMI L301 AS X"" -> "70K1 switch HMI L301
AS X"
DMSToBacStrErase="
```

```
[BlacklistScan]

;=================================================================
================================
; (===> This whole section can be configured per device <===)
; Options in this section are only then relevant if scan source is BACnet, e.g. [Settings]
ScanDMS=0/2
; If there are no keywords defined in this section, then BlackListScan wan't be performed.

; -------> "ObjName<Nr>" <-----------
; Strings defined in folowing "ObjName<Nr>" keynames are compared (regex like) with
BACnet property
; "object-name" (of the according device) and if they match then the object is not scanned and
not included
; in DMS. The keynames must be numbered, but don't need to be in sequence and there may
gaps exist.
;    ObjName23=PCD : all objects which name matches exactly "PCD" will be ignored.
;    ObjName24=PCD* : all objects which name begins with "PCD" will be ignored
;    ObjName25=*PCD : all objects which name ends with "PCD" will be ignored
;    ObjName26=*PCD* : all objects which name contains "PCD" will be ignored

; -------> "ObjDesc<Nr>" <-----------
; Strings defined in folowing "ObjDesc<Nr>" keynames are compared (regex like) with
BACnet property
; "description" (of the according device). Same rules as above.
;    ObjDesc1=PCD : all objects which description matches exactly "PCD" will be ignored.
;    ObjDesc2=PCD* : all objects which description begins with "PCD" will be ignored
;    ObjDesc3=*PCD : all objects which description ends with "PCD" will be ignored
;    ObjDesc4=*PCD* : all objects which description contains "PCD" will be ignored

; -------> "ObjType<Nr>" <-----------
; Objects defined in folowing "ObjType<Nr>" keynames are compared with scanned BACnet
object
; of the according device and if they match then the object is ignored and not included into
DMS.
; The keynames must be numbered, but don't need to be in sequence and there may gaps
exist.
; Keynames "ObjTyp<Nr>" define object(s) by type and instance nr.
;    ObjType1=BO 5 : binary-output instance nr=5 will be ignored
;    ObjType4=AI 9-13 : analog-input instances in range from 9 till 13 will be ignored
;       Objects can be identified by: 1) 2..3-chars abbreviation e.g. "AO" = "analog-output"
;       or written out: "analog-output". Capitalization does not matter.
;       Abbreviation list: "analog-input" oder "ai", "analog-output" "ao", "analog-value" oder "av"
;       "binary-input" oder "bi", "binary-output" oder "bo", "binary-value" oder "bv", "calendar" oder
"ca",
;       "command" oder "cmd", "event-enrollment" oder "eeo", "file" oder "fi", "group" oder "gr",
"loop" oder "lo",
;       "multi-state-input" oder    "msi", "multi-state-output" oder "mso", "notification-class" oder
"nc",
;       "schedule" oder "sch", "averaging" oder "ave", "multi-state-value" oder "msv", "trend-log"
oder "tl",
;       "life-safety-point" oder "lsp", "life-safety-zone" oder "lsz", "accumulator" oder "ac",
```

```
;       "pulse-converter" oder "pc", "event-log" oder "el", "trend-log-multiple" oder "tlm",
;       "load-control" oder "lc", "structured-view" oder "sv", "access-door" oder "ad"



[WhiteListScan]

;=================================================================
================================
; (===> This whole section can be configured per device <===)
; Options in this section are only then relevant if scan source is BACnet, e.g. [Settings]
ScanDMS=0/2
; If there are no keywords defined in this section, then BlackListScan wan't be performed.

; -------> "ObjName<Nr>" <-----------
; Strings defined in folowing "ObjName<Nr>" keynames are compared (regex like) with
BACnet object property
; "object-name" (of the according device) and only if they match then the object will be
scanned and
; included into DMS, while all other objects will be ignored (no scanned, not included into
DMS).
; The keynames must be numbered, but don't need to be in sequence and there may gaps
exist.
;    ObjName23=PCD : all objects which name matches exactly "PCD" will be scanned and
included into DMS.
;    ObjName24=PCD* : all objects which name begins with "PCD" will be scanned and
included into DMS.
;    ObjName25=*PCD : all objects which name ends with "PCD" will be scanned and
included into DMS.
;    ObjName26=*PCD* : all objects which name contains "PCD" will be scanned and
included into DMS.

; -------> "ObjDesc<Nr>" <-----------
; Strings defined in folowing "ObjDesc<Nr>" keynames are compared (regex like) with
BACnet object property
; "description" (of the according device). Same rules as above.
; The keynames must be numbered, but don't need to be in sequence and there may gaps
exist.
;    ObjDesc1=PCD : all objects which description matches exactly "PCD" will be scanned
and included into DMS.
;    ObjDesc2=PCD* : all objects which description begins with "PCD" will be scanned and
included into DMS.
;    ObjDesc3=*PCD : all objects which description ends with "PCD" will be scanned and
included into DMS.
;    ObjDesc4=*PCD* : all objects which description contains "PCD" will be scanned and
included into DMS.

; -------> "ObjType<Nr>" <-----------
; Objects defined in folowing "ObjType<Nr>" keynames are compared with scanned BACnet
objects
; of the according device and only if they match then the object will be scanned and included
into DMS,
; while all other objects will be ignored (no scanned, not included into DMS).
```

; The keynames must be numbered, but don't need to be in sequence and there may gaps exist.
; ObjType1=BO 5 : binary-output instance nr 5 will be scanned
; ObjType4=AI 9-13 : analog-input instance nr from 9 to 13 will be scanned
; Objects can be identified by: 1) 2..3-chars abbreviation e.g. "AO" = "analog-output"
; or written out: "analog-output". Capitalization does not matter.
; Abbreviation list see above in [BlackListScan]

9.3.5.2 Firewall setting

If the PC has a local firewall (e.g. in the Windows security center), the BACstac port used must be activated as otherwise
communication (COV, I-Am and other broadcasts) are prevented.

9.3.5.3 Call up of an external GUI configuration program

It is possible to incorporate an external customer-specific GUI configuration program in BacDriver.
The call-up takes place from the Menu, Options, Configuration (or Settings).

The program name (and possibly path) must be entered in a **BacDriver.ini** file in section [GUI] CfgProgram=<ProgrammName.exe>.
The BacDriver.ini must either be in the execution folder of BacDriver.exe (so for example C:**Visi.Plus**\bin), or in the <proj>\drv folder.

Example of a BacDriver.ini:

```
; List here in section [GUI] BacDriver's configuration program
[GUI]
CfgProgram=notepad.exe
```

If no external GUI configuration program is defined in BacDriver.ini, Notepad.exe is called up with the current BacDriver.cfg configuration file by default.

9.3.5.4 Automatic update control

Section **[Settings]**

The automatic update control ("Check for Updates" function) can be configured in BacDriver.cfg in the section [Settings] as follows:

CheckForUpdates=0 (= default): **no** update check is performed.

CheckForUpdates=1 if an update is pending, then a Yes/No MsgBox **request** appears, whether this update should be performed or not.
CheckForUpdates=2 an update check and possible update are always performed **automatically**.

A “Check for Updates” function can be also be performed as a menu command at any time: Menu, file, “Check for Updates” (resp. "Check for update")

See also Chapter Check for update

9.3.5.5 DMSRoot

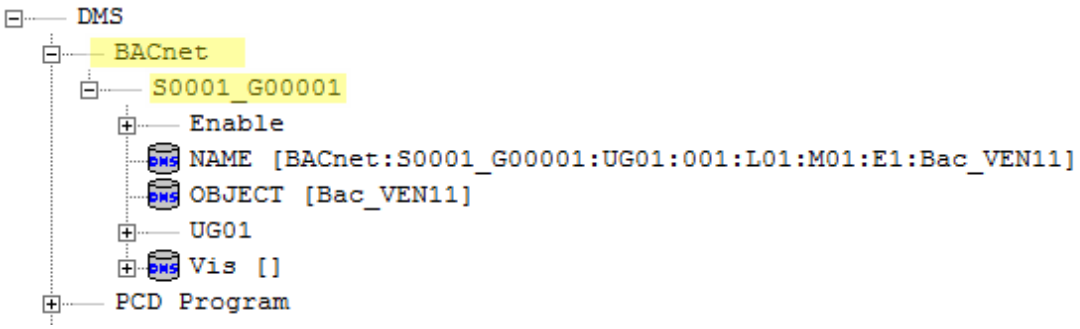
Section **[Settings]**

If **DMSRoot** remains empty, the BACnet objects directly under its name (= object-name or description property) are inserted into DMS.
Otherwise the **DMSRoot** name is put in front.

Example: for BACnet object name = S0001_G00001:UG01:001:L01:M01:E1:Bac_VEN11

DMSRoot=BACnet

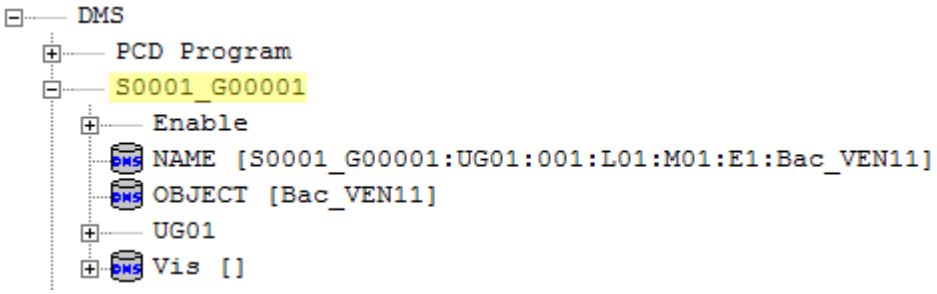
Resulting DMS structure:



DMSRoot=

i.e. DMSRoot remains empty, undefined

Resulting DMS structure:



9.3.5.6 DMS connection (DMSConn)

Section **[Settings]**

BacDriver can be operated in Remote mode, i.e. with a remote connection to the DMS server.
If no connection can be made to a local DMS after the BacDriver start, an attempt is made to establish a connection to the remote DMS.

DMSConn is a string that defines the DMS connection:

- Connection log: TCP/IP or Windows pipe connection:
  If a colon with a four digit DMS port number follows the DMS server name (without a space in between), then a TCP/IP connection is established, otherwise always a pipe connection.
- Connection type: local or remote; defined via the DMS server name; if the string only consists of ".", then BacDriver is connected to a DMS that runs on the same (local) computer otherwise the DMS server is automatically searched on the network (exception. "localhost" equivalent to "127.0.0.1" also identifies the local PC).

Examples:

Local connection via pipes:

DMSConn=.

Local connection via:

DMSConn=127.0.0.1:9010

Remote connection via pipes:

DMSConn=WS097

Remote connection via TCP/IP:

DMSConn=172.18.0.217:9010

This setting can also be made via Menu, Options, Configure DMS Connection. A connection dialog is displayed, the entered connection configuration tested and if OK, this is entered in `<proj>\drv\BacDriver.cfg [Settings] "DMSConn"` .

9.3.5.7 BACnet network options

Section **[Settings]**

9.3.5.7.1 Destination network number (BacnetDestNETno)

Section **[Settings]**

Define BACnet address (destination network number):

BacnetDestNETno=0

BacnetDestNETno=65535

BacnetDestNETno=<Remote-Adress>

These values have the following meaning:

1. **0**=local (default): the address is installed so that BACnet messages sent with this address are transferred on the local BACnet network.
2. **65535**=global (hex 0xFFFF): the address is installed so that BACnet messages sent with this address are transferred on all connected BACnet networks
3. **<Remote-Adress>**: the address is installed so that BACnet messages sent with this address are transferred onto the remote BACnet network.

 Before each scan, the destination network number is newly read by BacDriver.cfg, i.e. it can be modified for runtime.
In this way, the BACnet devices can be scanned individually (or as a group) from different networks.

9.3.5.7.2 Device instance numbers range (WhoIsDeviceInstNoMin/Max)

Section **[Settings]**

Here optional arguments for the **WhoIs** request are set that restrict the range of the devices that are expected to respond.

The device instance should:
1. occur once within the BACnet internetwork, and
2. be an integer in the range of 0 to 4.194.302 = 0..0x3FFFFE

WhoIsDeviceInstNoMin=100000

WhoIsDeviceInstNoMax=105000

Default = -1: then **WhoIsDeviceInstNoMin**=0 and **WhoIsDeviceInstNoMax**=4194302 (=0x3FFFFE)

All explicitly listed device instance numbers must be in the defined range from the BacDevicesToScanFile otherwise they cannot be scanned. See: Selective device scan.

9.3.5.7.3 Selective device scan

Section **[Settings]**

Here an optional name of a csv file is defined, which contains a list of device instance numbers.
If this file is provided in the <proj>\drv folder, then only the listed devices are scanned.

BacDevicesToScanFile=IMOzh.csv

A required condition is that all listed instance numbers of the devices are within the defined range WhoIsDeviceInstNoMin/Max,
see chapter Device instance numbers range

csv format of BacDevicesToScanFile is the following:

- empty lines and comment lines can be used in the entire document (first character ';' is used); these are ignored.
- 1st line is skipped (columns title)
- all of the following lines must feature the following format: ID or IP-Adr of the device; device instance number; activation option; comment...

Example:

```
# Title line: example template: one line = one device
VERSION_OF_LAYOUT;1.43;

# columns
# 1     ;    2   ;   3  ;     4    ;       5       ;     6       ;       7         ;
    8      ;     9        ;      10      ;     11   ;   12    ;  13
# ID/IP ; Device ;Device; Comment; Resubscription; Poll        ; Scan Source     ;
Scan Source ; WatchDog     ; Scan         ; TraceLog ; ScanLog ; Restart
# No/Adr; inst.No; active;          ;  Period (secs);Period (secs);Inp/Outp=EDE-File;
  Inp=DMS   ; Period (secs); Period (secs); active    ; active   ; Notification
# 1     ;    2   ;   3  ;     4    ;       5       ;     6       ;       7         ;
    8      ;     9        ;      10      ;     11   ;   12    ;  13

;1;1001;1;MY first device to scan;1200;0;0;0;0;0;1;0
10.6.200.3;222;1;MY 2nd device to scan;1200;
```

9.3.5.7.4 BACnet WhoIs Timeout

Section **[Settings]**

Time period in seconds in which the I Am-BACnet responses await Who Is requests.
The period duration should be selected according to the local BACnet network architecture.

WhoIsTimeout=10

If an "I Am" response only arrives from a device after this timeout (for example due to a router delay, etc.), then this device is no longer recorded as an accessible device by BacDriver and is no longer considered for scanning/communication.

9.3.5.7.5 Write priority (BacWritePrio)

Section **[Settings]**

Key word **BacWritePrio**:

```
; BACnet priority for write
; 1 Manual life safety (safety - manual)
; 2 Automatic life safety (safety - automatic)
; 3 - 4 Freely available
; 5 Critical equipment control (critical application)
; 6 Minimum On/Off (on/off) ------> may never be used: reserved for BACstac
; 7 Freely available
; 8 Manual operator (manual)
; 9 - 16 Freely available (15 is default)
BacWritePrio=15
```

Default setting is write priority = 15.

The write priority **BacWritePrio** can be changed for runtime at any time, i.e. adjusted in **BacDriver.cfg**.
When the cfg file is saved, the new write priority becomes active.

The write priority **6** may never be used: it is reserved for BACstac.

Key word **UseIndividualObjWritePrio**:

```
; Enable individual BACnet write priorities pro object, valid for all write oper
; (===> This option can be configured per device <===)
; If =1, then all BACnet objects will be extended with additional auxiliary DMS
; "_Prio" type DWS and "_PrioSetNULL" type BIT.
; If the value of the "_Prio" data points >= 1 and <=16, then this priority will
; write operations of all properties of this BACnet object, instead of the globa
; BACnet priority "BacWritePrio" (see above).
; If the trigger DP "_PrioSetNULL" is set = ON, then NULL will be written into p
; at the according priority position due to "_Prio".
; default=0 then no individual write priorities are possible, only the glogal Ba
UseIndividualObjWritePrio=0
```

9.3.5.7.6 Debug options

Section **[Debug]**

Key word **BacObjectReadoutOnce**:

```
; Option for BACnet objects read method:
; if BacObjectReadoutOnce=1 then all object properties are read out in a single
; -> fast method, but not always compatible with all BACnet devices
; if BacObjectReadoutOnce=0 then every object property is read out in a command
; -> slow method, but compatible with all BACnet devices
; CAUTION: if BacObjectReadoutOnce=0 then Black- und WhiteListScan functions wil
BacObjectReadoutOnce=1
```

This option does not influence the scan/poll functionality, but the Black/WhiteListScan will be disabled if `BacObjectReadoutOnce=0`.

In normal mode, never use `BacObjectReadoutOnce=0`.

9.3.5.8 Assembly of the DMS name

The BACnet object name, under which all the object's properties are inserted into DMS, is by default assembled from the property "**object-name**".

But this is an option, which is relevant only for the case, when there no BACnet object structure, e.g. tree, exists yet in DMS.
In this case the DMS tree for the BACnet objects is composed at run time when the BACnet is scanned by BacDriver for the first time, see next chapter Scan options.

In the section **[Settings]** there is an option for for the determination of the object property, from which string value the DMS name is assembled:

ScanBacToDMSname=77

The number is a property ID which is the source of the string for the DMS object node name (sub-tree).
From this string the plant code key is generated (DMS name) and the BMO template VLO name
(aggregate template) extracted, see [Filter] section below.
Default is **ScanBacToDMSname=77** which is the ID of the "**object-name**" property.
Another possibility is **ScanBacToDMSname=28** which is the ID of the "**desctription**" property.

9.3.5.9 Scan options

Section **[Settings]**

ScanDMS=0

Scan source. (===> This option can be configured per device <===)
If **ScanDMS=0** (=default), then only BACnet is scanned.

If **ScanDMS=1**, then only DMS is scanned.

If ScanDMS=1 then options **DMSRoot**, **DeviceRoot** and **[Filter]** have no relevancy.

If **ScanDMS=2**, then first DMS is scanned and subsequently BACnet, while only such objects will be inserted into DMS which are not yet contained there.

The option ScanDMS=2 is particularly useful to use, when new items have been inserted in to a device and you want to implement them in DMS. There are 2 ways how the device might be implemented in the DMS:

- The object name corresponds to the DMS structure, e.g. object name =
  "S0006 G00387UG01701H08C03Bac_CLK01Time"

is converted 1:1 into DMS name:
  "S0006 G00387:UG01:701:H08:C03:Bac_CLK01:Time"
The DMS structure is defined in section [Filter], by BacToDMSKeyMask and BMOStart / EndPos and BMOLevel.

- The object name does not correspond to the DMS structure, e.g. object name =
  "S0006 G00387UG01701H08C03Bac_CLK01Time"

is implemented in DMS as
  "UG01:H08:3:Bac_CLK01Time"
Depending on which name structure is used, adjustments in the DMS structure or DMS naming may need to be made subsequently.
You can use the DMS rename function for it, even at run time, it is not necessary to exit BacDriver and restart it again.

The scan, or re-scan function can be performed in BacDriver also several times (and at runtime): menu File, Scan Selected Devices.
Either all devices are scanned → combo box on the status bar = "Dev All" or in the combo box only one device may be selected.
Currently it is not possible to make a selection of only a few devices.

If you want to implement only newly introduced objects from the device into DMS, then a usefull method may be to use BlackListScan and/or WhiteListScan filters, s. chapter "Filter for scanned objects: Black and White Lists".

ScanPeriod=3600

Scan period in seconds: if **ScanPeriod=0**, then no scan (**This is the default setting**).
Otherwise the network is requested according to the set BACnet address in **BacnetDestNETnr**, after the BACnet devices with **WhoIs**.
The range of the device instance numbers is taken from the **WhoIsDeviceInstNrMin/ Max** setting.
The selective scan option **BacDevicesToScanFile** is also considered.

Before each new scan, the configuration file "**BacDriver.cfg**" is newly read and BacDriver is configured accordingly.

Before each new scan, all COV subscriptions are cancelled (unregistered) and

reregistered after the scan, whereby the set **COVsubscriptionLifetime** is considered (seeCOV options).


It is not recommended to enable this option! By sending WhoIs broadcasts the network can be stressed, sometimes significantly.

BacAskIfStartScan=0

Option for BacDriver start: should the following (see below) request MsgBox appear before the start or not?
If =1, then it is asked whether the first start scan should be performed or not, i.e.
-> if **BacAskIfStartScan=1** -> display "Yes/No" MsgBox: "Scan: Update VLO from BMO?" for answer "Yes", scan otherwise not.
-> if **BacAskIfStartScan=0** -> no request appears, the network is scanned and DMS is updated with VLOs from BMO corresponding with the filter options.

9.3.5.9.1 BMO templates option "BacPushUpBMONAME"

Section **[Settings]**

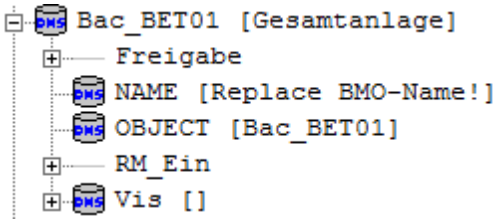
BacPushUpBMONAME=1

(default setting = 3, see below)

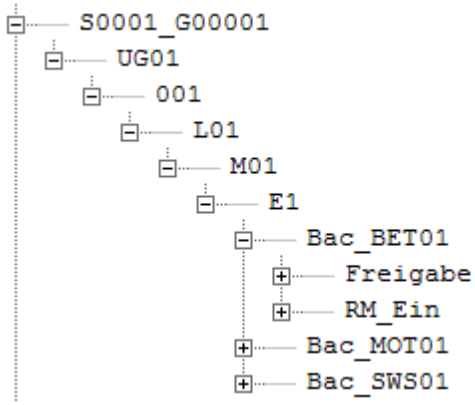
Option for nested/grouped (super) VLO objects.
After (during the scan procedure) a BMO-VLO object (for example "...:M01:E1:Bac_BET01") was embedded in DMS, its "**NAME**" is determined (for example "...:M01:E1:Bac_BET01:RM_On") and is copied a DMS level higher without the last link ("RM_On").

Example:

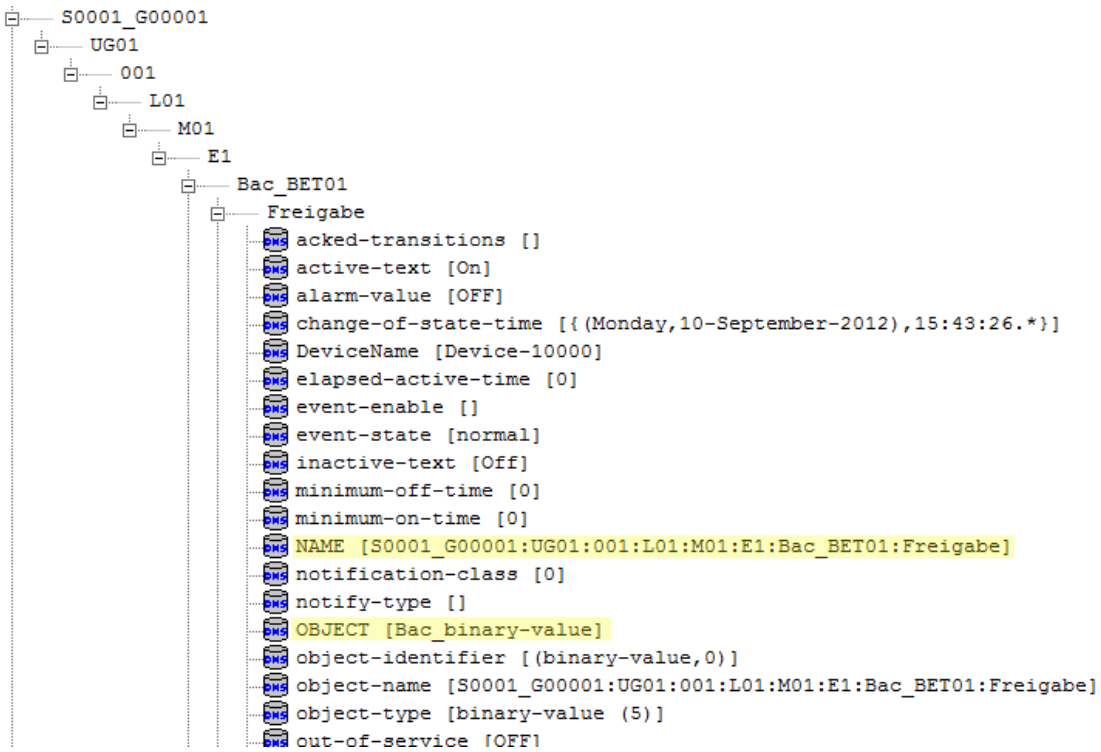
BMO object of the Bac_BET01



Inserted Bac_BET01 after scanning the object "S0001_G00001:UG01:001:L01:M01:E1:Bac_BET01"

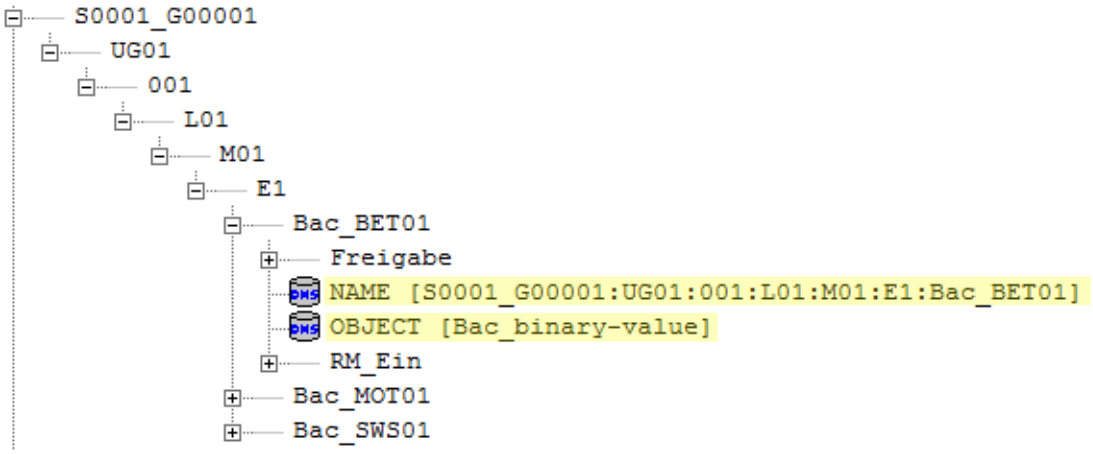


Open subtree of Bac_BET01



The data points "NAME" and "OBJECT" may not appear in this subtree, but must be moved to the subtree under Bac_BET01.
Before the DP "NAME" is shortened by the last name link: new value = "S0001_G00001:UG01:001:L01:M01:E1:Bac_BET01"

Subtree of Bac_BET01 after moving the "NAME" and "OBJECT"



The only object is now "**S0001_G00001:UG01:001:L01:M01:E1:Bac_BET01**", because there is only one DP "NAME".

BacPushUpBMONAME=2
(default setting = 3, see below)

Option for nested/grouped (super) VLO objects:
- Get the "**description**" from the last embedded BMO-VLO object and copy the content string one DMS level higher into "**NAME**".

BacPushUpBMONAME=3
(this is the default setting)

Option for nested/grouped (super) VLO objects:
- Get from all embedded BMO-VLO "**description**" properties the "**largest common name**" and copy it one DMS level higher into "**NAME**".
  Example: "*Pump 1 Release*" and "*Pump 1 SM*" and "*Pump 1 RM*" --> result to be copied is "*Pump 1*"

9.3.5.9.2 BMO templates option "BacDelUnusedBMONAME"

Section **[Settings]**

BacDelUnusedBMONAME=1

Option for nested (super) VLO objects.
After (during the scan procedure) a BMO-VLO object (for example "...:M01:E1:Bac_BET01") has been embedded in the DMS, all sub-BMO objects are deleted which are part of a (super) VLO object, but are not included in the current PLC, i.e. their "NAME" property is empty.

BacDelUnusedBMONAME=0

This is the default setting.
No test whether NAME empty, no delete.

9.3.5.9.3 BMO templates option "BacForceBMOUpdate"

Section **[Settings]**

BacForceBMOUpdate=1

All BMO templates are updated in the DMS, even if they are already provided in the DMS; in this case, their value is overwritten with the one from BMO.

BacForceBMOUpdate=0

This is the default setting.
Only those types of BMO templates are updated that are not yet available in DMS.

9.3.5.9.4 DMS control functions compilation and execution

Section **[Settings]**

After each scan, PLS functions are compiled, if

CompileDMSFunc=1

This is the default setting.
With this PLS functions newly inserted in DMS are then activated from the BMO update.
The PLS compilation is not performed if

CompileDMSFunc=0

All newly inserted PLS functions from the BMO update remain inactive.


This option setting is suitable for debug and test purposes.

After each scan, PLS functions are executed, if

ExecuteDMSFunc=1

With this, all value changes (depending on newly scanned data) are also reported by DMS

to all registered clients (for example AlmMng, AlmView).

ExecuteDMSFunc=0

This is the default setting.
No PLS functions are executed so that also no value changes (depending on newly scanned data) are reported by DMS to registered clients.

9.3.5.9.5 Log of the scanned objects and their properties

Section **[Settings]**

ScanLogActive=1

Scan Logging: if =1 or 2: scan details will be logged into **BacDriverScan.log**
If ScanLogActive=1 then all objects of scanned devices will be listed
If ScanLogActive=2 then all object properties of scanned devices will be listed
ScanLogActive=1: =Default : all objects of scanned devices will be listed

9.3.5.10 Time synchronisation of the BACnet devices (BacDriver as Time Master)

Section **[Settings]**

TimeMaster=1

If the option is **TimeMaster=1** (and **COVresubscrPeriod**>0), its time and date for each COV resubscription period of a device object is brought to the DMS time, i.e. the properties local date and local time are described with DMS time values.

 The option **TimeMaster** is linked with the option **COVresubscrPeriod** , i.e. the time synchronisation only takes place when the re-subscription period is defined.

Both services are supported for all BACnet devices:

- Time synchronisation (local DMS time)
- UTC Time Synchronisation (GMT)

Manual time synchronisation
Selected BACnet devices can also be manually synchronised at any time via Menu Command File, "Re-subscribe COV of selected devices".
For the COV resubscription, the time synchronisation services (time synchronisation and UTC Time Synchronisation) are performed.

9.3.5.11 COV (Change Of Value) options

Section **[Settings]**

COVresubscrPeriod=3600
This is the default setting.

Re-subscription period in seconds: if **COVresubscrPeriod=0**, then COV subscriptions are no longer renewed.
Smallest subscription period is 900 = 15 minutes.

COVsubscriptionLifetime=604800
This is the default setting.

COV subscription lifetime in seconds. Default 1 week = 604800 seconds.
After its expiration, the COV subscription is cancelled and a new Re-subscribe procedure is automatically started.
Smallest lifetime is 900 = 15 minutes.

The value **COVsubscriptionLifetime**=0 means an endless lifetime, which is never automatically cancelled (should NOT be used)

COVPprioArr=0
This is the default setting.

Option for automatic subscription of COVP of priority array properties of all objects of all devices, if available.
Default = 0, i.e. priority array is not subscribed (if available).

In Section **[Debug]** the COV subscription can generally be deactivated:

COVactive=0
Default setting is COVactive=1.

If **COVactive**=0 -> then no COV subscription is executed at all and, as a result, no COV notifications are received.
It may be useful to deactivate the COVs to reduce the network load when the option **VisPollPeriod** is simultaneously activated (for example VisPollPeriod=1000, i.e. all visualised properties are polled every second) and the alarm is not dependent on incoming COVs.

9.3.5.12 Poll options

Section **[Settings]**

9.3.5.12.1 Poll of all active devices

Section **[Settings]**

PollPeriod=3600
This is the default setting.

Poll period in seconds. During this period all objects and their properties of all active devices are polled in sequence.
If **PollPeriod**=0 -> then no poll.
Default = 3600, i.e. all objects are polled in one hour.

The resulting poll interval is calculated as follows: Poll period divided by the number of all objects.
The poll period should be selected so that the poll interval is not smaller than ~ 500-100 ms.

Ex.: for 40 devices each with ~400 objects, a poll period of ~4000 seconds is optimum; 4 objects are then polled per second

 The polled object with all its properties is read with read access, if **[Debug]** BacObjectReadoutOnce=1.

9.3.5.12.2 Poll of all visualised properties (in GE/browser webserver)

Section **[Settings]**

VisPollPeriod=0
This is the default setting.

Poll period in milliseconds for polling of visualised properties.
If **VisPollPeriod**=0 -> no visualised properties are polled.
Default =0, i.e. no visualised properties are polled.
Recommended value = 1000 (= 1 second). Minimum value = 100 (milliseconds).

Automatic polling of visualised data points in GE/browser client viewer

After a new screen has been displayed, all corresponding BACnet DMS data points are automatically updated and regularly polled until the screen is closed again.

 If the **VisPollPeriod** option is activated, the COV subscription can generally be deactivated in the section **[Debug]** .
See chapter **Debug and logging options**

9.3.5.13 Weekly-schedule options

Section **[Settings]**

ScheduleObjUseAct=1

This is the default setting.

If =1, then BacDriver automatically generates the following help data points for the **weekly-schedule** property (of the schedule object): "**Sav**" and "**Act**":

"**Act**" is attached to every "Time<No>". If it is = OFF, this time DP only keeps the time values (for future use), but it is deactivated, i.e. if BacDriver composes the weekly schedule string, these time DPs are ignored.

"**Sav**" is attached to every "Val<No>". It retains a copy of "Val" (it also has the same data type). Only "Sav" DPs should be visualised in the schedule BMO template instead of the actual "Val" values. The reason for this is because it is how the switched value remains saved and it is then also displayed when this point in time is deactivated. The value is adopted again in the current switch DP "Val" when reactivated at a later date.

If =0, then the help data points: "**Sav**" and "**Act**" are not generated, not communicated and are ignored if they already exist in the DMS.

Section **[Debug]**

BacScheduleTypeToDMS=0

This is the default setting.

If =1, then BacDriver automatically converts the DMS data type of the switched object in **weekly-schedule** property (of the schedule object): "**Val<nr>**" to reflect the same BACnet data type of the switched object, e.g. if BACnet data type is boolean then the DMS data type of "**Val<nr>**" will be BIT.

 This setting is generally not recommended! The dynamic change of the data types in DMS may cause trouble in the visualization (which operates with fixed data types).

If BacScheduleTypeToDMS=0 (=default), then the included BMO template data type is decisive and all BACnet data types will be converted accordingly,
e.g. if the BACnet data type is enumerated and the DMS data type of "**Val<nr>**" is BIT, then the switched value is converted from enumerated to BIT and vice versa.

9.3.5.14 Filter options

Section **[Filter]**

Based on the selected filter rules in Section [Filter]:

1. a unique DMS name is created from the BACnet object name
2. a BMO template name is converted from the object name. During the first scan (after

the BacDriver start), this template is inserted in the corresponding position in DMS.

The options in this section are mainly used to extract the BMO template name (aggregate VLO) from the BACnet name of the respective object (=OBJECT_NAME property). This name represents the address key=system identification AKS.

Using its AKS name = content of the property "object name" (string type), the BACnet objects are inserted accordingly into the DMS structure.

Several positions (BMOLevels) can be specified within the AKS name for different VLO's. Several BMO templates can therefore be defined and integrated and therefore also nested and entire BMO structures, see Options BMOStartPos1, BMOEndPos1, BMOLevel1, BMOStartPos2,...

The following filter methods are used on the OBJECT_NAME property in the sequence:
1) **BMOStartPos..BMOEndPos, BMOLevel**: BMO template name is extracted according to the absolute position specifications
2) **BacToDMSKeyMask**: individual positions can be replaced by ':'
3) **BacToDMSErase**: individual letters can be deleted

9.3.5.14.1 BMO template name extraction: Positions, levels, masks

Section **[Filter]**

If a fixed part (=part of the AKS address key) of the OBJECT_NAME property should be used for the BMO template name conversion, its **Start** and **End position** must be entered here:
1. Position 1 = first letter (not 0). Position is inactive (unoccupied) if = 0 or -1 or empty.
2. Start and end position letters are a part of the resulting BMO name.
3. Position indices must start with 1 and be numbered consecutively without gaps 1,2,3,...
4. The level specification = the order of the AKS level in the DMS name, belongs to each position, where the BMO_Template Object is added,
   i.e. the segment number in the position mask 'BacToDMSKeyMask' - see below.

Example:

The address key is "021701**L**304ULA_E01**TRU**001TL01REP"
The corresponding positions are " 1 2 3"
"12345678901234567890123456789O"

In this example only two BMO names are used: "**L**" on position from 7 to (and with) 7
"**TRU**" from 18 to 20

The address key is divided into 9 levels, which corresponds to a DMS tree depth of 9 levels.
The levels are defined in a string mask: **BacToDMSKeyMask**

```
BacToDMSKeyMask                          =
"....:..:...:...:....:...:...:...."
; this results in the following DMS name=
"0217:01:L:304:ULA_:E01:TRU:001:TL01"
```

The levels (=DMS levels) are separated from each other with the DMS separator character "**:**". The levels are numbered in sequence, starting with 1.
Therefore the 3rd. level = L and the 7th = TRU
If the corresponding VLO's should be inserted at the same level in DMS (which is logical), their level must also have corresponding numbers.
The 2 BMO template objects are defined as follows:

```
; BMOName = "L" from "021701L304ULA_E01TRU001TL01REP"
BMOStartPos1=7
BMOEndPos1=7
BMOLevel1=3

; BMOName = "TRU" "- "021701L304ULA_E01TRU001TL01REP
BMOStartPos2=18
BMOEndPos2=20
BMOLevel2=8
```

9.3.5.14.2 BACnet name conversion: BacToDMSDelimiter and BacToDMSErase

Section **[Filter]**

In the AKS address key (= OBJECT_NAME property), individual letters can also be replaced by DMS separators ":".
Ex.:

```
; if BACnet-Name = "MST.REG#001/R0004" DMS = "MST:REG:001:R0004" can be converted f
BacToDMSDelimiter=./#
```

The rule is:
- One or more letters can be specified.

- No spaces
- Default: =empty (no entry)

A conversion rule can then still be used: **BacToDMSErase**
All listed letters are deleted in the AKS address key (=Property OBJECT_NAME).
Ex.: the letter "_" is deleted in the AKS name:

```
; Ex.: Name = "0217:01:L:304:ULA:_E01:TRU:001:TL01" -> "0217:01:L:304:ULA:E01:TRU:0
BacToDMSErase=_
```

The rule is:
- One or more letters can be specified.
- No spaces
- Default: =empty (no entry)

9.3.5.14.3 Write operation conversion: DMSToBacStrErase

Section **[Filter]**

During a write operation, a BACnet-String-Property from DMS is written in BACnet. A conversion can still be defined before the write operation:
all listed letters of the **'DMSToBacStrErase'** key, which are available in the string property, are deleted after the property variable transfer before the write procedure in the string.

The BACnet-String-Property in the DMS remains unchanged. Several letters can be defined.
Ex.:

```
; description property = ""70K1 system switch HMI X""
; is written in BACnet as "70K1 system switch HMI X" (all quotation marks are delet
DMSToBacStrErase="
```

9.3.5.14.4 Filter for scanned objects: Black and White Lists

Section **[**BlackListScan**]** and Section **[**WhiteListScan**]**

In these sections, additional conditions for objects can be defined so, that if such objects meets the conditions, then:
1. they **CANNOT** be inserted into DMS when scanning BACnet (**BlackListScan**)
2. **ONLY** such objects will be inserted into DMS when scanning BACnet (**WhiteListScan**)

Configuration options: global or device specific
Both sections can be defined globally in **BacDriver.cfg** for all devices, or device specific (just for single device) in the respective **BacDevice-<ID>.cfg**.

 If a device specific Black- and/or WhiteListScan is defined, then a possible global Black/ WhiteListScan will be ignored during the BACnet scan.

Filter methods

Two different filtering methods can be applied and/or combined:

1. Object name: this property string will compared according to regex rules with all filter definitions
2. Object description: this property string will compared according to regex rules with all filter definitions
3. Object type and its instance number or instance number range.

All filtering methods can be combined, e.g. if the WhiteListScan object range is defined, then only objects from this range will be scanned and inserted into DMS.
If additionally even BlackListScan object range is defined, then such objects will be eliminated from the resulting WhiteListScan object range.

Example:

In WhiteListScan is a (floor) range defined **"*UG01*"** and in BlackListScan is a (room) range defined **"*701*"**,
e.g. all objects of the floor UG01 should be inserted, but without the room 701.
During the scan successively objects with following names are detected:
"C02:UG01:702:H08:Bac_SPS01" → will be inserted: fulfills WhiteListScan and does not violate BlackListScan
"C02:UG02:702:H08:Bac_SPS02" → will not be inserted: violates WhiteListScan, BlackListScan is not relevant anymore
"C02:UG01:701:H08:Bac_DIG01" → will not be inserted: fulfills WhiteListScan, but conflicts with BlackListScan (BlackListScan is fulfilled)

Filter by object name

Keyword is/are:

ObjName<Nr>=<FilterString>

If there are no keywords defined, then no Black/WhiteListScan's are accomplished.

WhiteListScan: the objects will be scanned and inserted into DMS only if they match, otherwise they will be ignored.
BlackListScan: if the objects do match, then they will not be scanned and not inserted into DMS.

- Strings defined in the following keywords, will be compared with the BACnet objekt property "object-name" (of the belonging device) according regex rules.
- The keywords must be numbered. It does not matter whether they occur in order or there are gaps in numbering.

Example for WhiteListScan:

ObjName23=PCD all objects which name matches **exactly** "PCD" will be scanned and included into DMS
ObjName24=PCD* all objects which name **starts with** "PCD", will be scanned and included into DMS

ObjName35=*PCD all objects which name **ends with** "PCD", will be scanned and included into DMS
ObjName37=*PCD* all objects which name **contains** "PCD", will be scanned and included into DMS

Filter by object description

Keyword is/are:

ObjDesc<Nr>=<FilterString>

If there are no keywords defined, then no Black/WhiteListScan's are accomplished.

WhiteListScan: the objects will be scanned and inserted into DMS only if they match, otherwise they will be ignored.
BlackListScan: if the objects do match, then they will not be scanned and not inserted into DMS.

- Strings defined in the following keywords, will be compared with the BACnet objekt property "object-name" (of the belonging device) according regex rules.
- The keywords must be numbered. It does not matter whether they occur in order or there are gaps in numbering.

Example for WhiteListScan:

ObjDesc3=PCD all objects which description matches **exactly** "PCD" will be scanned and included into DMS
ObjDesc4=PCD* all objects which description **starts with** "PCD", will be scanned and build into DMS
ObjDesc5=*PCD all objects which description **ends with** "PCD", will be scanned and build into DMS
ObjDesc7=*PCD* all objects which description **contains** "PCD", will be scanned and build into DMS

Filter by object type and instance number

Keyword is/are:

ObjType<Nr>=<Object identifier> <Instance number>

or for instance number ranges

ObjType<Nr>=<Object identifier> <Instance number From> - <Instance number To>

If there are no keywords defined, then no Black/WhiteListScan's are accomplished.

WhiteListScan: the objects will be scanned and inserted into DMS only if they match, otherwise they will be ignored.
BlackListScan: if the objects do match, then they will not be scanned and not inserted into DMS.

Example for WhiteListScan:

ObjType1=binary-output 5 binary-output object of the instance number 5 will be scanned and build into DMS
ObjType4=analog-input 9-13 all analog-input objects of the instance numbers from 9 till 13 (incl.) be scanned and build into DMS

Table of the object type identifiers

Use of capital and/or small initial letters does not matter

<Object-identifier> can be written as:

"analog-input"	or	ai
"analog-output"	or	ao
"analog-value"	or	av
"binary-input"	or	bi
"binary-output"	or	bo
"binary-value"	or	bv
"calendar"	or	ca
"command"	or	cmd
"device"	or	dv
"event-enrollment"	or	eeo
"file"	or	fi
"group"	or	gr
"loop"	or	lo
"multi-state-input"	or	msi
"multi-state-output"	or	mso
"notification-class"	or	nc
"program"	or	pr
"schedule"	or	sch
"averaging"	or	ave
"multi-state-value"	or	msv
"trend-log"	or	tl
"life-safety-point"	or	lsp
"life-safety-zone"	or	lsz
"accumulator"	or	ac
"pulse-converter"	or	pc
"event-log"	or	el
"trend-log-multiple"	or	tlm
"load-control"	or	lc
"structured-view"	or	sv
"access-door"	or	ad

9.3.5.15 Debug and logging options

BacDriver writes 3 different logs:

- BacDriver.log: Standard log for error/event logging
- BacDriverScan.log: only scan events are logged here. Structured list of all BACnet devices and their objects and properties, instance numbers, data types, names and DMS names generated from them.
  If the `[Settings] TraceActive=1` is set in BacDriver.cfg, the COV events are also logged here.
- BacDriverTrace.log:for detailed debug messages. Only active when the option `[Settings] TraceActive=1` is set in BacDriver.cfg.

- BacDriverProfile.log: for debug profile messages. Only active when the option `[Settings] ProfileActive=1` is set in BacDriver.cfg. Here the runtimes of the respective functions (for example `Look_For_Devices`) are entered.


The `TraceActive` option should only be activated briefly with control as it can cause a significantly high CPU load and large log files.

Section [Monitor]

MonitorActive=0
Monitor window: if MonitorActive=1 , then the data traffic is displayed.

MonitorFilter=<DeviceID>:<ObjTyp>,<ObjID>

Example.: **MonitorFilter=5010:0,4** e.g. device = 5010, object type = 0 = analog input, object instance (=ID) = 4
see chapter Object Filter.

All other Monitor options see chapter The monitor window.

Section [Debug]

Options in this section are only intended (and useful) for debugging.

TraceActive=0
Trace Logging: if TraceActive=1, then detail logs are logged in **BacDriverTrace.log**.

ProfileActive=0
Profiling Logging: if ProfileActive=1, then profile loggings are logged in **BacDriverProfile.log**.

ScanRegSleep=0
Sleep time in milliseconds: if >0 then this time is awaited after each scanned object registered in DMS. (Default = 0).

COVUnsubBeforeResub=0
COV is always initially unsubscribed before a new (Re)-subscription is executed.

COVactive=0
Here the COV subscription is generally deactivated.
Default setting is COVactive=1.
If COVactive=0 -> then no COV subscription is executed at all and as a result no COV notifications are received.
It can be useful to deactivate the COVs to reduce the network load when the option **VisPollPeriod** is simultaneously activated (for example VisPollPeriod=1000, i.e. all visualised properties are polled every second) and the alarm is not dependent on

incoming COVs.

 Condition for the above option: the alarm should not be dependent on incoming COVs (otherwise they are also deactivated).

See Chapter **Visualisation in GE and Browser**

UnconfCOVactive=1

General switch for unconfirmed COV subscription; this functionality is necessary for restart notifica
If UnconfCOVactive=0 -> no unconfirmed COV subscription, no unconfirmed COV notifications at a
Default 1, deactivating unconfirmed COVs can only be useful for test and debug purposes.

DeviceRestartRemoveAdrNotif=1

If the BacDriver ends, it removes its local address/network no. from all device objects, restart-notifi
if DeviceRestartRemoveAdrNotif=1 (=default).
This functionality can be switched off for test and debug purposes if DeviceRestartRemoveAdrNotif

DeleteStringQuotes=1

All BACnet CharacterStrings are set in double quotation marks (for example, "On_Low"). These are
If DMS writes the modified strings back in BACnet, the quotation marks (in BACnet) are restored, if
This functionality can be switched off for test and debug purposes if DeleteStringQuotes=0

BacObjectReadoutOnce=1

Option for BACnet objects read method, default: BacObjectReadoutOnce=1.
If BacObjectReadoutOnce=1 then all object properties are read out in a single command
-> fast method, but not always compatible with all BACnet devices
If BacObjectReadoutOnce=0 then every object property is read out in a command
-> slow method, but compatible with all BACnet devices

 CAUTION: if BacObjectReadoutOnce=0 then Black- und WhiteListScan functions will b

 This option does not influence the scan/poll functionality.
In normal mode, never use `BacObjectReadoutOnce=0`.

BacTimer=1000

Master Timer Thread period for poll/visual poll/watchdog/resubscription/scan.
Can be deactivated or extended for debug or Load/Stress Tests.
Default =1000 msec

ChkPollCOVTimeStamp=0

additional time stamp check for received properties from BACnet
; If ChkPollCOVTimeStamp=0 (=default) then no check is performed
; If =1 then time stamps are compared and if received property is older
; than the DMS one, then no the property will be not written into DMS

BacCOVPrio=0

priority setting for controlled competing between

; the BacTimer (polling/watchdog/resubscription) and the COV thread.
; If BacCOVPrio=0 (=default) then controlled competing is not activated (both threads have same priority)
; BacCOVPrio=1..20 (max value), 20 is highest priority for the COV thread, 1 is lowest
; useful for heavy load stress tests for massive (high) COV frequency

9.3.5.16 Watchdog monitoring of the active controllers (BACnet devices)

Section **[Settings]**

WDPeriod=<Anzahl Sekunden> = time period in which the state of all devices is checked.
If WDPeriod = 0, the watchdog monitoring for all devices remains inactive (switched off).

WDPeriod=120

During this time period, the program objects of all devices are read and their values (properties) written in DMS,
i.e. all properties which are included in the "BMO:Bac_program" template.
In particular, the property "**program state**" for device monitoring and monitoring can be used (visualised).

The device objects are written in DMS under the following data point:

System:Driver:BacDriver:Device-<ID> e.g.:
System:Driver:BacDriver:Device-20

The program object of the respective device is listed in DMS as an independent branch:

System:Driver:BacDriver:Device-<ID>_Program

The device program status during the specified period in the data point
- **System:Driver:BacDriver:Device-<ID>:_Program:program state**

is therefore always updated. The numeric value of the **program state** data point as well as DWS is simultaneously written into
- **System:Driver:BacDriver:Device-<ID>:_Program:program state:_pr_state**.

In addition to the project object, the property "**system status**" of the device object itself is polled: "**System:Driver:BacDriver:Device-<ID>:system-status**".

If everything is OK: .."**system-status**" = "**operational**" and .."_Program:**program-state**" = "**running**"

Device monitoring

The following DPs can be used for monitoring:
- "**System:Driver:BacDriver:Device-<ID>:_Program:program-state**" of type STR

and
- "**System:Driver:BacDriver:Device-<ID>:_Program:program-state:_pr_state**" of type DWS (this DP is automatically created after the BacDriver scan).

In addition a comment string is appended to **System:Driver:BacDriver:Device-<ID>:_Program:program-state:_pr_state:Comment** of type STR,
which reflects the status (= Meaning) as a string.

These DPs can have the following values - BACnet program state codes: (_pr_state = program state):

DP = "**.....:_Program:program-state:_pr_state**" (DWS)	**Status**	**Meaning**
-1	Critical	not connected
0	Warning	idle
1	Warning	loading
2	OK	Normal operation / running
3	Warning	waiting
4	Critical	halted
5	Critical	unloading
10	OK	operational
11	OK	operational-read-only
12	OK	download-required
13	OK	download-in-progress
14	Critical	non-operational
15	OK	backup-in-progress

 For each transition from an **'inaccessible'** state to -> **'accessible'** (device offline -> online), all COV(P)s are re-subscribed and all objects polled.
This procedure is identical to the one performed for a restart notification.

9.3.5.17 Restart notification (event-controlled start notification)

The restart notification is based on unconfirmed COV notifications (broadcasts), the following information is generally sent:
system status, time of device restart and last restart reason.

These notifications are sent for a cold/warm start if the device was programmed accordingly. A device sends a restart notification to all recipients/networks which are entered in the device object in the property **restart notification recipient**.
This could be a broadcast (normally in the local network) or an address.

In the chapter Automatic entry of BacDriver in restart recipients list, it is described how BacDriver can make an entry to all configured devices as a recipient of the restart message.

If this broadcast is detected and the sender device instance no. is on the list of active devices, then the following actions are performed for this device:
1. all COV (and if configured COVP) subscriptions are renewed
2. then **all objects (with all properties) are polled** and written in DMS.

9.3.5.17.1 Configuration

Restart notification is controlled via a global configuration (activated/deactivated). Default: Restart notification is not activated.

This method can be configured/activated in BacDriver.cfg and/or in the respective device list (option BacDevicesToScanFile) in different ways:
- globally for all devices in BacDriver.cfg in section [Settings] DeviceRestartNotif=1/0/-1
- individually for each device in the device list on the corresponding device configuration line in the associated 13th option column 1/0

Global configuration, one setting valid for all devices

Section **[Settings]**

DeviceRestartNotif=1/0/-1

This option is global (applies to all devices).
If **DeviceRestartNotif=1**, Restart-Notification for all devices is then activated, regardless of the individual device configuration.
If **DeviceRestartNotif=0**, restart notification for all devices is deactivated, regardless of the individual device configuration.
If **DeviceRestartNotif=-1** then this global setting is deactivated; in this case the individual configuration for each device is valid (see next section).

Individual configurations, each device has its own setting

In the section **[Settings]** a name of a csv file must be defined that includes a list of device instance numbers.
If this file is provided in the <proj>\drv folder, then the listed devices and their individual settings are read.

BacDevicesToScanFile=devicelist_XY.csv


A required condition is that all listed instance numbers of the devices are within the defined range WhoIsDeviceInstNrMin/Max,
see Chapter Device instance numbers range.

csv format of BacDevicesToScanFile is the following:
- empty lines and comment lines can be used in the entire document (first character ';') is used; these are ignored.

- 1st line is skipped (columns title)
- all of the following lines must feature the following format: ID or IP addr. of the device; device instance number; activation option; comment...

The option for the restart notification of the respective device is in the **corresponding device configuration line in the 13th column:**
=1 → Restart notification for this device is activated but only when the global option in section [Settings] DeviceRestartNotif=-1
=0 → Restart notification for this device is deactivated but only when the global option in DeviceRestartNotif=-1

Example:

```
# Title line: example template: one line = one device
VERSION_OF_LAYOUT;123;

# columns
# 1  ;    2   ;   3  ;   4   ;       5    ;     6      ;      7        ;     8    ;
   9       ;     10     ;   11   ;  12   ;    13   ;   14
# ID ; Device ;Device;Comment; Resubscr.  ; Poll       ; Scan Source   ; Scan Src ;
WatchDog   ; Scan       ;TraceLog;ScanLog;Restart  ;Restart ;
# No.; inst.No; active;       ;Period (sec);Period (sec);In/Out=EDE-File;  Inp=DMS
;Period (sec);Period (sec); active ;active ;Notif.act;AddNotif;
# 1  ;    2   ;   3  ;   4   ;      5     ;     6      ;      7        ;    8     ;
    9      ;     10     ;    11  ;   12  ;   13    ;   14

;1;1001;1;MY first device to scan;1200;0;0;0;0;0;1;0
10.6.200.3;222;1;MY 2nd device to scan;1200;
```

9.3.5.17.2 Restart counter

The restarts are counted for monitoring purposes as well as monitoring.

The counters are available for each device in "**System:Driver:BacDriver:Device-<InstNr>:_RestartCnt**".

- After the BacDriver start, all device counters "_RestartCnt" are initialised to 0.
- Set to 1 after subsequent successful connection.
- Each detected (and successful) restart (detected via broadcast or also via watchdog) increments this counter.
- If BacDriver is ended, all "_RestartCnt" are set to -1.

9.3.5.17.3 Automatic entry of BacDriver in restart recipients list (AddListElement)

After startup, the BacDriver reads the property restart notification recipients from all devices and their device object.
The recipient data is compared and if the BacDriver identifies either a local broadcast (network number =0) or its own local address (and the same network number) under the recipients, no action is performed.

Automatic AddListElement action is controlled via a global configuration (activated/ deactivated). Default: AddListElement action is not activated.

This method can be configured/activated in BacDriver.cfg and/or in the respective device list (option BacDevicesToScanFile) in different ways:
- globally for all devices in BacDriver.cfg in section [Settings] DeviceRestartAddNotif=2/1/0/-1
- individually for each device in the device list on the corresponding device configuration line in the associated 14th option column: 2/1/0

Global configuration, one setting valid for all devices

The following options can be configured: in BacDriver.cfg section [Settings] DeviceRestartAddNotif:

Section **[Settings]**

DeviceRestartAddNotif=1/2/0/-1

This option is global (applies to all devices).
If **DeviceRestartAddNotif=1**, then a local broadcast for the recipient list is added to the restart-notification-recipient (of the device object) (AddListElement), regardless of the individual device configuration.
If **DeviceRestartAddNotif=2**, then BacDriver's local address and network number for the recipient list added to restart notification recipient (of the device object) (AddListElement), regardless of the individual device configuration.
If **DeviceRestartAddNotif=0**, then no AddListElement action is performed, i.e. no restart notification is received for this device, regardless of the individual device configuration.
If **DeviceRestartNotif=-1** then this global setting is deactivated; in this case the individual configuration for each device is valid (see next section).

Individual configurations, each device has its own setting

In the section **[Settings]** a name of a csv file must be defined that includes a list of device instance numbers.
If this file is provided in the <proj>\drv folder, then the listed devices and their individual settings are read.

BacDevicesToScanFile=devicelist_XY.csv


A required condition is that all listed instance numbers of the devices are within the defined range WhoIsDeviceInstNoMin/Max,
see Chapter Device instance numbers range

csv format of BacDevicesToScanFile is the following:
- empty lines and comment lines can be used in the entire document (first character ';' is used); these are ignored.
- 1st line is skipped (columns title)
- all of the following lines must feature the following format: ID or IP addr. of the device;

device instance number; activation option; comment...

The option for the AddListElement action of the respective device is in the **corresponding device configuration line in the 14th column:**
=2 → AddListElement action or this device is activated but only when the global option in section [Settings] DeviceRestartNotif=-1
BacDriver's local address and network number for the recipient list is added to restart notification recipient (of the device object).
=1 → AddListElement action for this device is activated but only when the global option in section [Settings] DeviceRestartNotif=-1
A local broadcast for the recipient list is added to restart-notification-recipient (of the device object).
=0 → AddListElement action for this device is deactivated but only when the global option DeviceRestartAddNotif=-1
No action is performed.

Example:

```
# Title line: example template: one line = one device
VERSION_OF_LAYOUT;123;

# columns
# 1  ;    2   ;   3  ;   4   ;       5    ;     6      ;      7        ;     8    ;
   9       ;     10     ;   11   ;  12   ;    13   ;   14
# ID ; Device ;Device;Comment; Resubscr.  ; Poll       ; Scan Source   ; Scan Src ;
WatchDog   ; Scan       ;TraceLog;ScanLog;Restart  ;Restart ;
# Nr.; inst.Nr; aktiv;       ;Period (sec);Period (sec);In/Out=EDE-File;  Inp=DMS
;Period (sec);Period (sec); active ;active ;Notif.act;AddNotif;
# 1  ;    2   ;   3  ;   4   ;       5    ;     6      ;      7        ;     8    ;
    9      ;     10     ;    11  ;   12  ;   13    ;   14

;1;1001;1;MY first device to scan;1200;0;0;0;0;0;1;0
10.6.200.3;222;1;MY 2nd device to scan;1200;
```

9.3.5.18 Watchdog monitoring of the BacDriver

BacDriver toggles in the time (configured in DMS, see “System:Prog:WatchDog:Period”) its value (“System:Prog:WatchDog:BACDRIVER”, DWS) from 0 to 1 resp. from 1 to 0.
The flag is set to -1 when BacDriver ends. “System:Prog:WatchDog:BACDRIVER”

After starting the BacDriver,
the “_UP” Flag of BacDriver is written (ON / OFF): “System:Prog:BACDRIVER_UP” = 1 (only for local connection).
The flag is set to OFF when BacDriver ends.

9.3.6 BacDriver's system data points in DMS

After startup, the BacDriver carries the following information in DMS under **System:Driver:BacDriver:**

Property	DMS data point	DMS type	Ex.
To get signalling for Alarm Mng: isset/reset if Bad Driver fi	_ALMChange	BIT	OFF

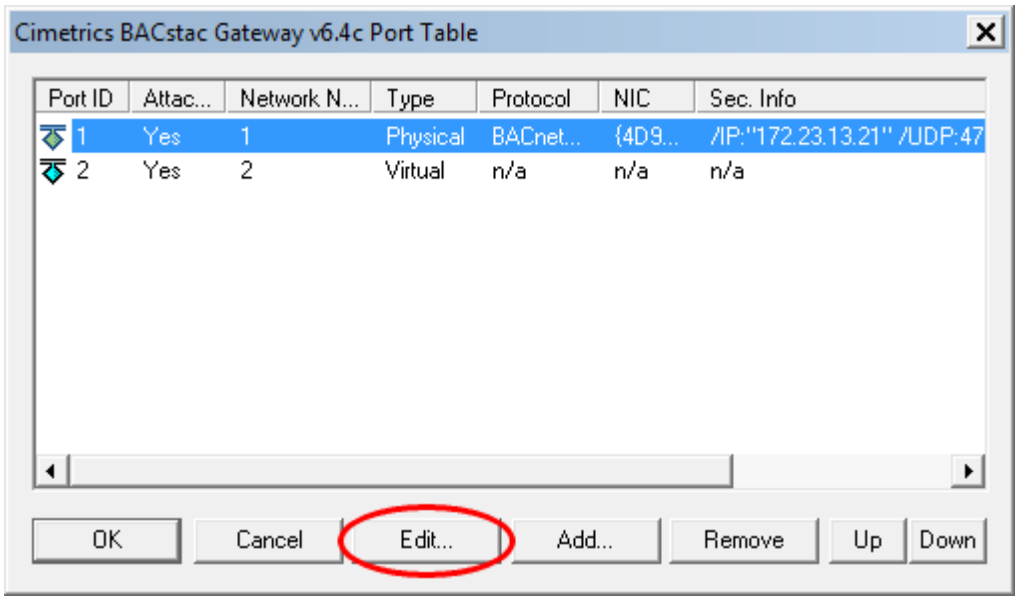
Property	DMS data point	DMS type	Ex.
Property is shared BACnet scan/resscan			
Toggle signal for AlrMng:	_CfgChange	BIT	OFF

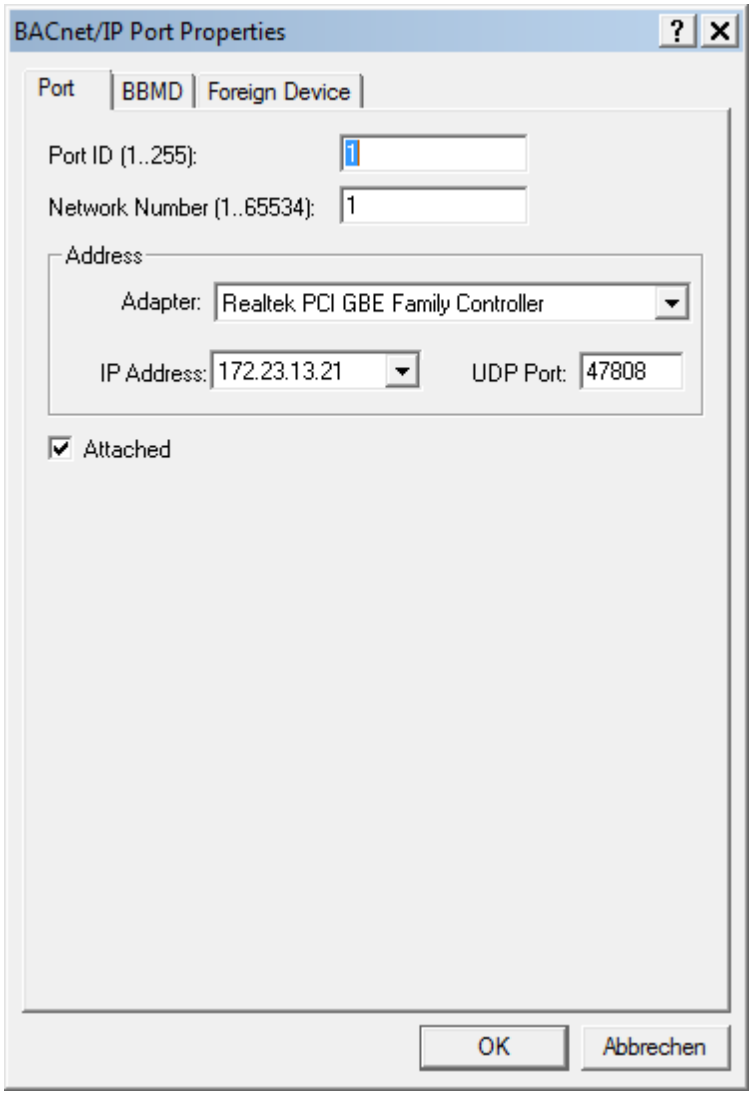
Property	DMS data point	DMS type	Ex.
Property set/reset if Bad Driver configuration.			

Property	DMS data point	DMS type	Ex.
Local address	_LocalAdr	STR	"172.23.13.21"
Local network number	_NetNr	DWS	0
BACnet Station Port	_Port	DWS	47808
BacnetDr	_ProcID	DWS	Value from [Settings] ProcessIDcov

Property	DMS data point	DMS type	Ex.
Driver's specific Device ID:			

- Local address, network number and Port are defined in Cimetrics BACstac settings; it is the BacDriver's IAM address (see also Installation of BACstac):

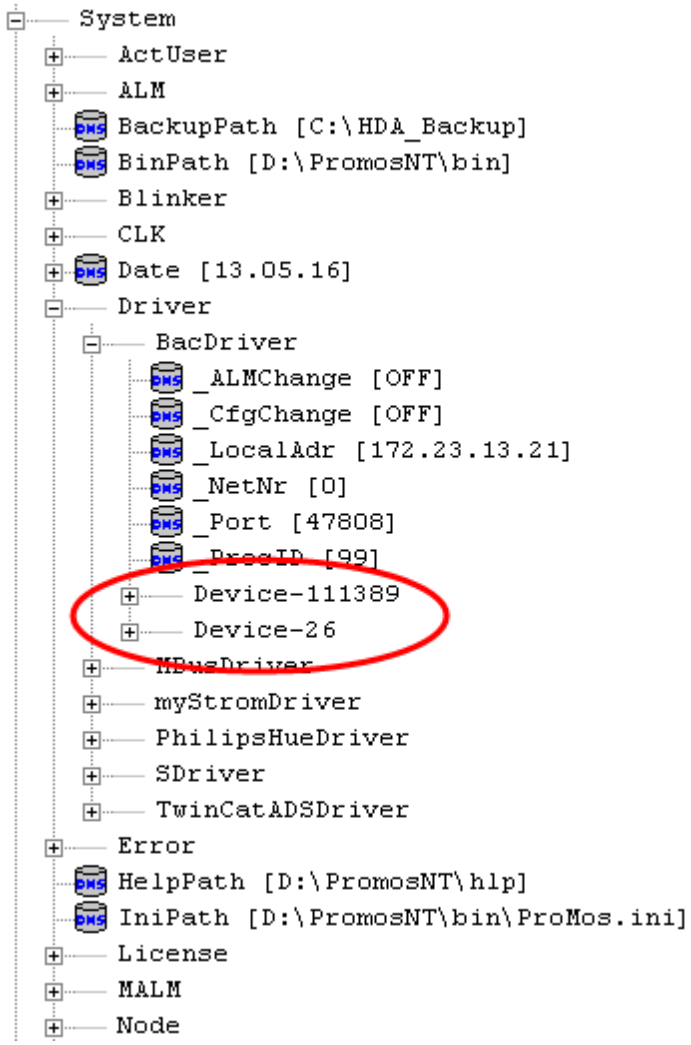




- BacDriver's process ID is defined in the global BacDriver.cfg configuration in [Settings] **ProcessIDcov**:

```
; BacDriver's process ID: BacDriver registers with following ID (as well
as PC's IP adr) all COV(P)
; subscriptions. This ProcessIDcov must be unique inside a PC system.
ProcessIDcov=99
```

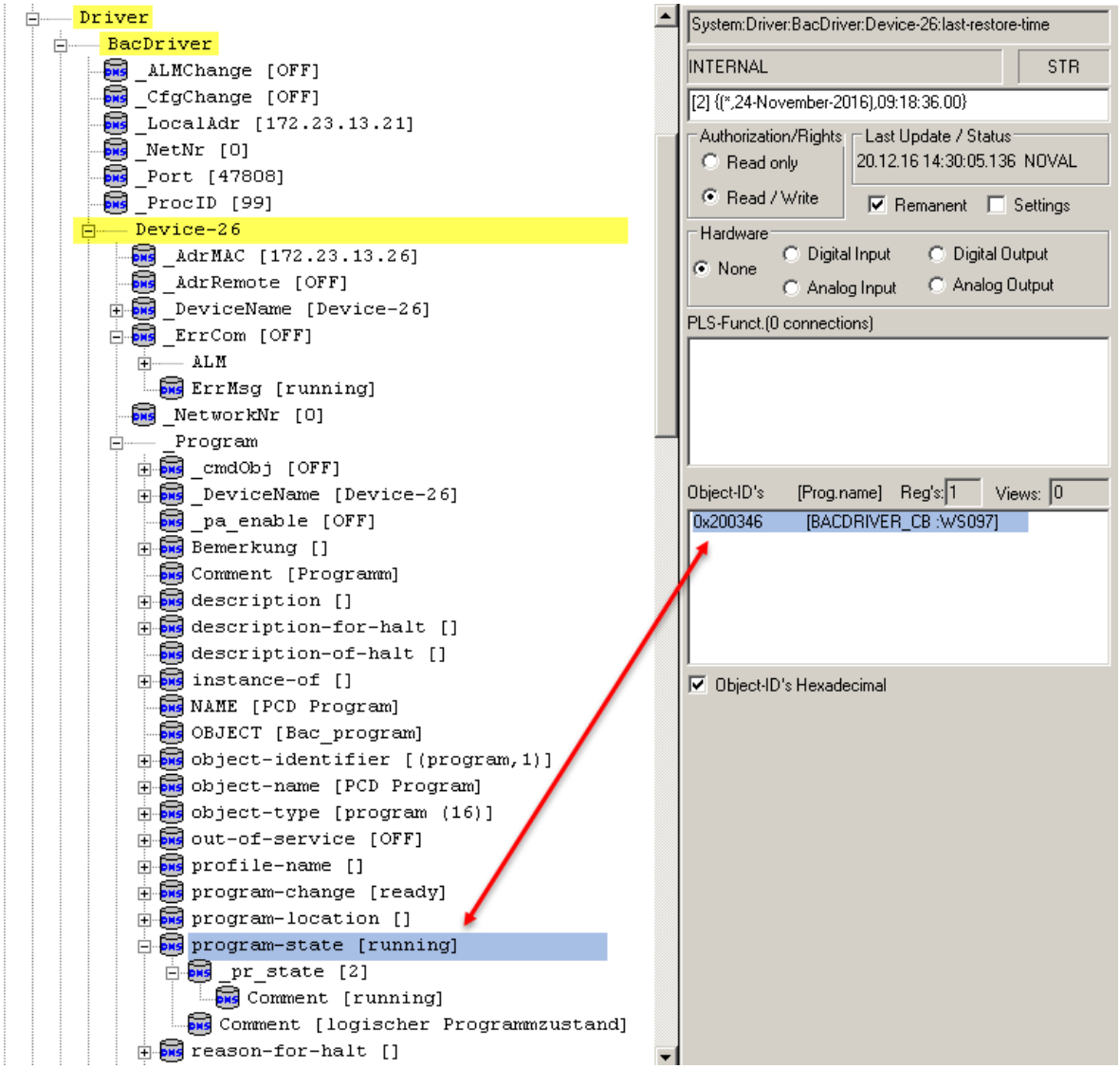
After all devices has been scanned, then all connected devices are as well listed in **System:Driver:BacDriver**



If a device is connected and online, all data points in **BacDriver:Device- <ID>** should be completed and registered by **BACDRIVER_CB**,
see How to check if an object is connected.

Before this check, DMS tree view should be absolutely updated s. How to check if device is online and reachable.

- The data point **System:Driver:BacDriver:Device-17:_Program:program-state** should exist and have the value "**running**",
  s. Watchdog monitoring of the active controllers (BACnet devices).



9.3.7 Summary of DMS help variables in BACnet objects

After startup, the BacDriver carries the following information in DMS under each BACnet object node:

Description, purpose	DMS data point	DMS type	Registered by BacDriver	Automatic creation: if **yes** ➔ does not need to be included in the **BMO:Bac_<object name>**
device name e.g. "Device-47810"	**_DeviceName**	STR	no	yes
flag DP: set if this object has **priority-array** property, i.e. it is **commandable**	**_cmdObj**	BIT	no	no (must be contained in basic BMO:Bac...)
flag DP: set =ON if the object (one among of grouped VLO set)	**_Enabled**	BIT	no	yes

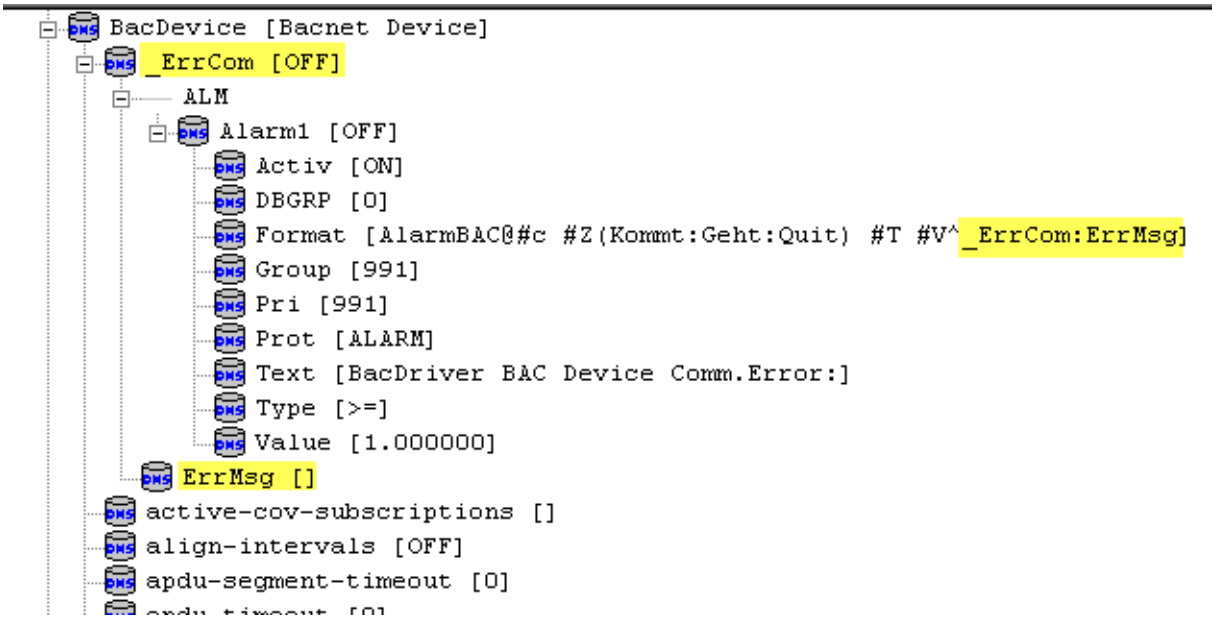
Description, purpose	DMS data point	DMS type	Registered by BacDriver	Automatic creation: if **yes** ➔ does not need to be included in the **BMO:Bac_<object name>**
exists. Not always are all VLO object present. With _Enabled then the visibility can be controlled				
only for **commandable** objects: for manual control: trigger DP: replacement value, default value (Relinquish Default)	**_pa_enable**	BIT	yes	no (must be contained in basic BMO:Bac...)
only for **commandable** obj: copy/ container of **present-value**	**_pa_value**	*)	yes	no (must be contained in basic BMO:Bac...)
individual BAC object's write priority	**_Prio**	DWS	no	yes (will be aut.created in base BMO: Bac..)
trigger DP: reset the corresponding token of priority-array (due to object's write priority) to NULL	**_PrioSetNULL**	BIT	yes	yes (will be aut.created in base BMO:_Bac..)
trigger DP: if 0 ➔ 1 then pull (read) BACnet object into DMS	**_Pull**	BIT	yes	yes (will be aut.created in base BMO: Bac..)
only for the "schedule" object, property "weekly-schedule": trigger DP: if 0 ➔ 1 then the complete **schedule** object is written into BACnet and re-read again	**_Push** **)	BIT	yes	yes
only for the "**trend-log**" object: flag DP: will be set in the trend-log if new buffer data were read from trend	**_TRDready** ***)	BIT	no	yes
only for the "**trend-log**" object: trigger DP: if 0 ➔ 1 then BacDriver polls the **trend-log's linear buffer** (i.e. only if its related property **stop-when-full** = ON). Subsequently the buffer will be reset and the trend-log restarted.	**_TRDpullFix** ***)	BIT	yes	no (must be contained in basic BMO:Bac...)
for any BACnet object:				

Description, purpose	DMS data point	DMS type	Registered by BacDriver	Automatic creation: if **yes** ➔ does not need to be included in the **BMO:Bac_<object name>**
flag DP: set = ON if **this object** is referenced by a trend-log, i.e. its data are logged. Otherwise reset = OFF.	**_EnabledTL**	BIT	no	no

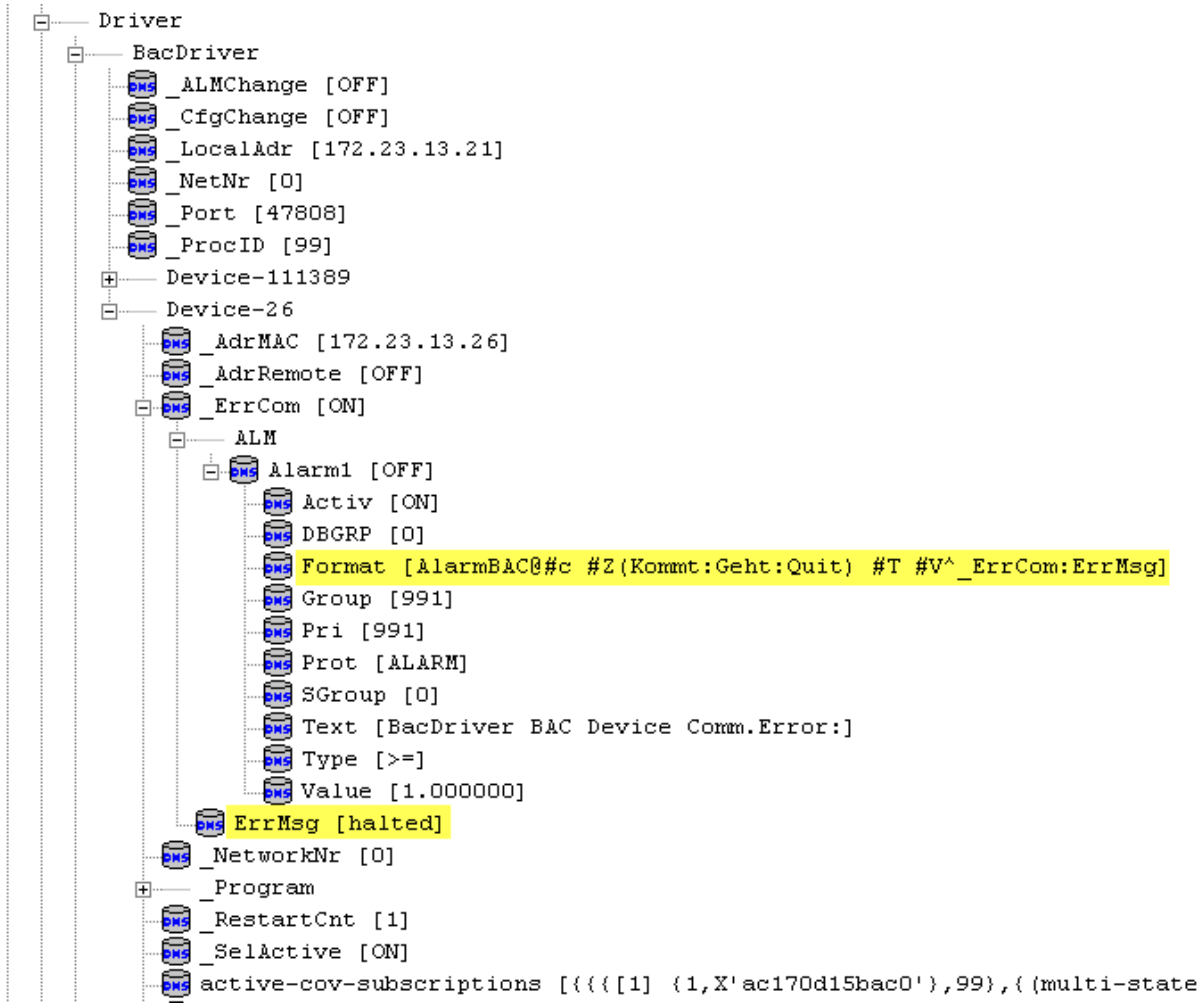
*) must be of the same type as **present-value**
) only for the "schedule**" object, property "weekly-schedule", e.g. for instance "...Schedule Binary 0:weekly-schedule:_Push"
***) only for the "**trend-log**" object

9.3.8 Monitoring of the BACnet devices (via alarm data points)

Each device can be monitored by its own separate alarm.
For this purpose an **_ErrCom** data point is included in the BMO:BacDevice with a ALM alarm add-on:



At the BACnet scan-time BacDriver inserts this structure into the according **"System:Driver:BacDriver:Device-<ID>"**



If the device gets unreachable or there is a read or write error, BacDriver writes the error message and code into

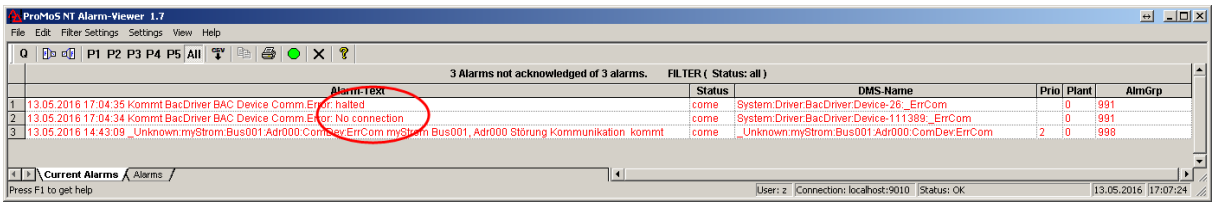
- System:Driver:BacDriver:Device-<ID>:_Program:program-state:_pr_state

and

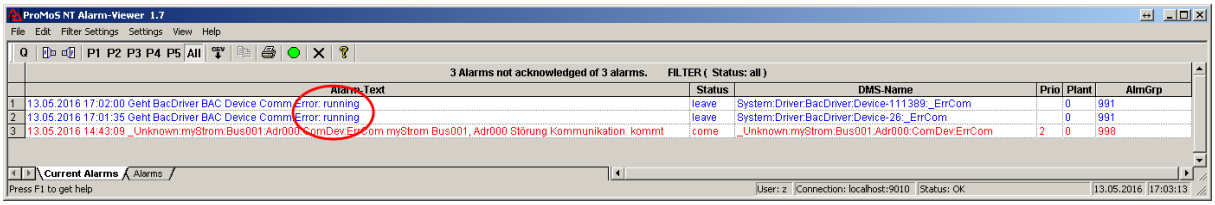
- System:Driver:BacDriver:Device-<ID>:_Program:program-state

and a copy of the message string into **System:Driver:BacDriver:Device-<ID>:_ErrCom:ErrMsg.**

The alarm DP **System:Driver:BacDriver:Device-<ID>:_ErrCom** will be set due to the state of the error code; if it is <> 0 then an alarm is invoked and the error message is transferred into the alarm text:



As soon as the device is communicating again and the error code = 0, the alarm(s) is gone:



Synchronization with AlmMng

Every time when BacDriver scans BACnet and the ALM structure of the devices might have change, a synch-signal
System:Driver:BacDriver:_ALMChange is triggered -> 0 -> 1 -> 0.
If AlmMng is running, it registers this DP and after this change is detected, then AlmMng updates all ALM's in DMS.

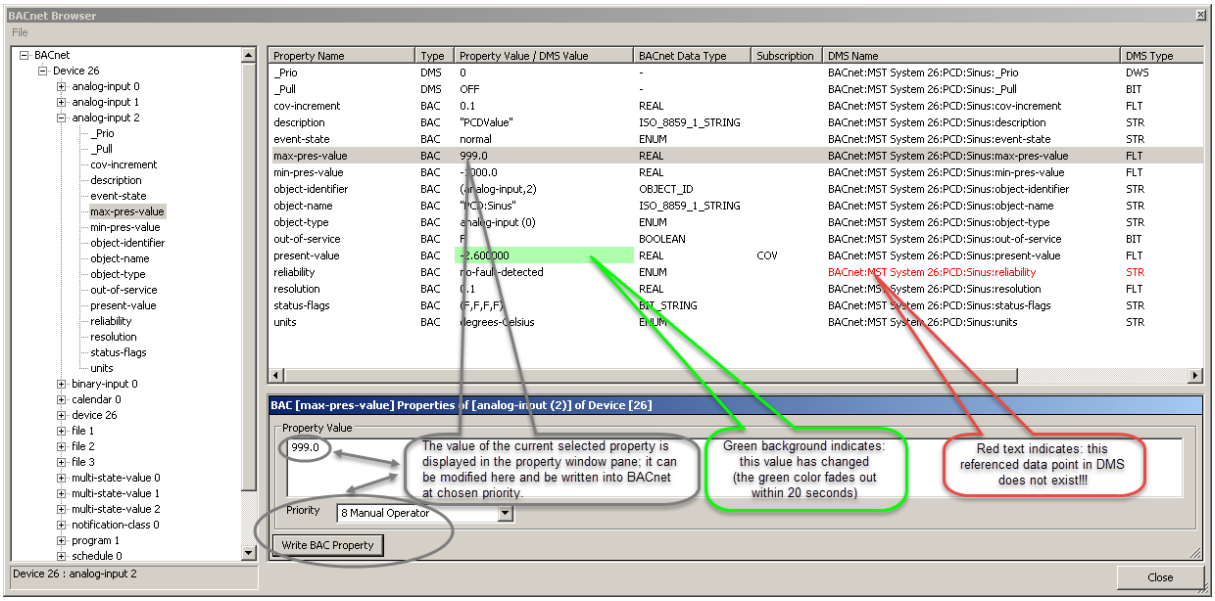
9.3.9 BACnet Browser Window

The BACnet Browser Window will be activated via Menu, View, BACnet Browser.

- BACnet Browser is a tool for analyzing and diagnosing BACnet networks.
- BACnet Browser is an easy way to discover BACnet devices, objects and their properties and display their connection to DMS.

It provides devices, objects and properties in a structured view, reflecting the actual BacDriver configuration settings.
In particular, the device range selection and the DMS name labeling definition will be considered.

This tool is under development, features are still not finished.



Automatic Refresh

The display is automatically refreshed every 2 seconds.

Manual Refresh

You can force the display refresh manually:

- via menu, File, Refresh
- double click on device, object or property

Write BAC Property

The actual selected property is written indirectly to the BACnet:
if clicked, first the current value from the Property Value window is written into the according DMS data point; and then DMS subsequently writes the value into the connected BACnet device.

9.3.10 Troubleshooting, error handling

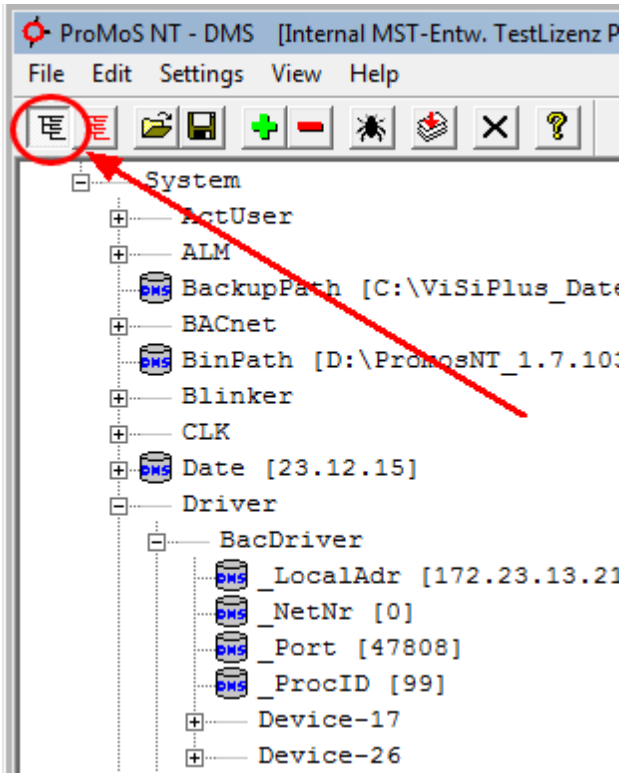
For more complex troubleshooting / diagnosis, we also recommend the use of the following Tools, which are independent from Bacdriver.exe Program:

- BACnet OPC-Server from Cimetrics (demo version available)
- BACnet Explorer von Cimetrics (demo version available)
- BACeye (www.baceye.com)

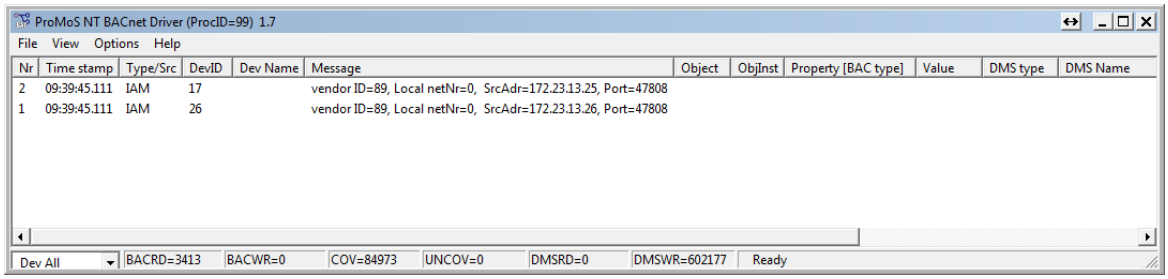
9.3.10.1 How to check if device is online and reachable

After all devices has been scanned, then all connected devices are listed in **System:Driver:BacDriver** (see BacDrivers system data points in DMS).

Before you start investigating DMS data points, you should refresh and update the DMS tree view; just press the "Object View" tool bar button and expand the System view after:



1. If you ping the device, then **ping** command must be successful
2. If you start BacDriver with the activated menu option "**Monitor IAMs**", then the BacDriver's monitor window there must show the corresponding **IAM** notification:

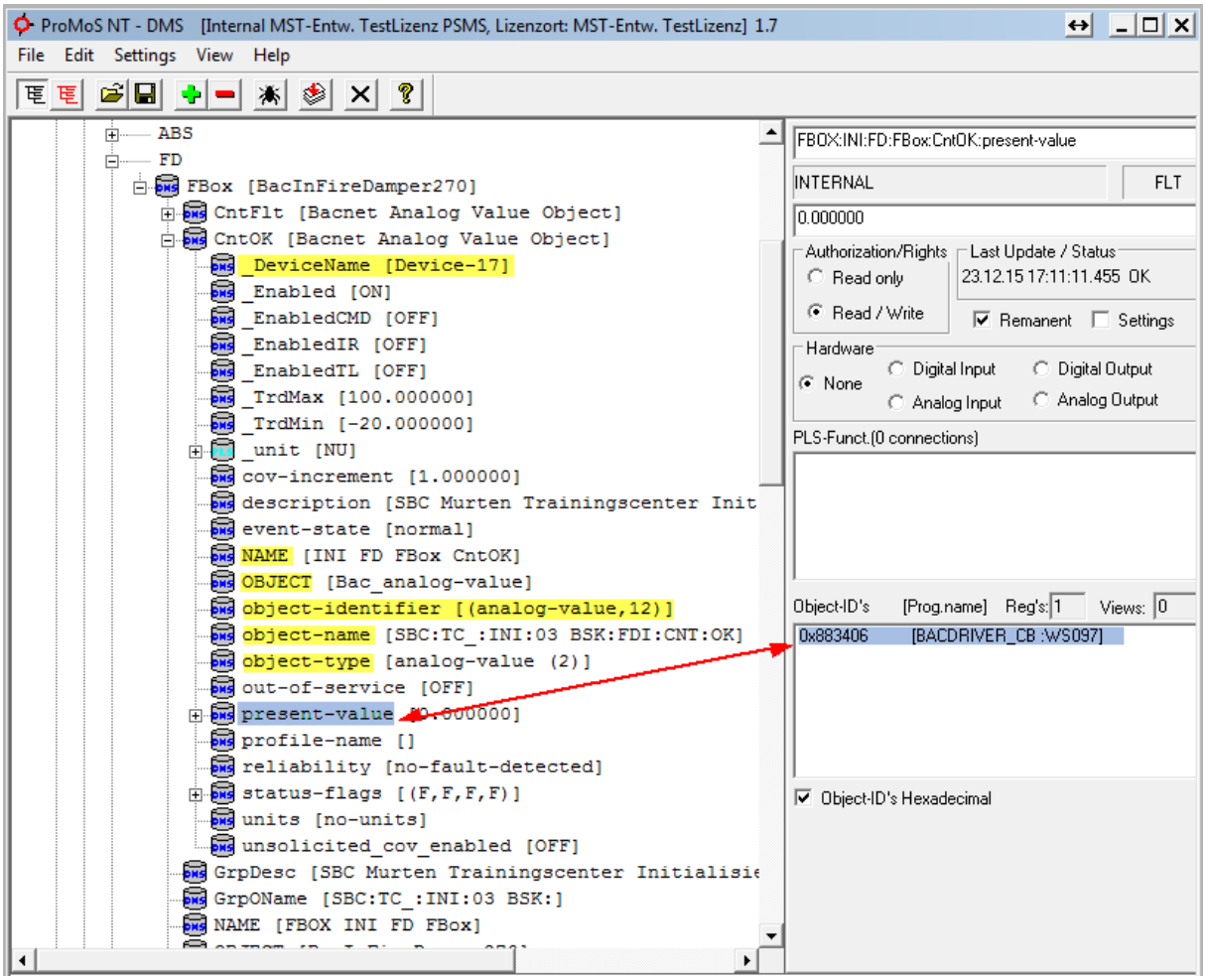


In Case of no success do following checks:

3. In the global configuration (BacDriver.cfg) the corresponding device ID must match the WhoIs request range in [Settings] **WhoIsDeviceInstNrMin** / **WhoIsDeviceInstNrMax**.
4. If there is an selective device list defined in BacDriver.cfg in [Settings] **BacDevicesToScanFile** defined, check if in the list the corresponding device is defined and activated.

9.3.10.2 How to check if an object is connected

For a well detected and connected BACnet object following DMS data points are mandatory, see example below, marked yellow:



- **_DeviceName**: keeps the reference to the belonging device, has the form **Device-<ID>**
- **NAME** names the belonging BMO template
- **OBJECT** identifies the belonging BMO template
- **object-identifier** identifies the belonging BACnet object, has the form (**<object-type-name>,<object-ID>**)
- **object-name** name of the BACnet object
- **object-type (<object-type>)** determines the BACnet object type, has the form **<object-type-name>**

If the object is connected, then its properties, e.g. present-value, must be registered by the BacDriver client. The registration is displayed in the right property pane window. At least one of the displayed registrations must be named "**BACDRIVER_CB**"

Basically all standard BACnet properties, e.g. data points with standard BACnet property name, should be registered by **BACDRIVER_CB**.
If this is not the case, then
- this property is only in the BMO BACnet template present, but not in the device object.

9.3.10.3 Most frequent errors

1. DMS name is > 80 characters
Entry in BacDriver's message window, as well as in BacDriver.log:

```
ERROR: DMS name lenght > 80 (=84): [BACnet:DDC 2.7 FBox:SBC:TC_:BAC::05
S:che:dul:e B::SC REF O:BJ:status-flags:in-alarm])
```

This error will not happen anymore in Visi.Plus Version 2.0, because the 80 character limit for maximal length of the DMS names becomes no longer necessary
.

2. A BACnet objekt with the identical ID is mapped twice in DMS
In DMS occures the property "**object-identifier**" with the identical STR value twice (or even more times), e.g.:

"FBOX:AL:ALA1:FBox:Fault:object-identifier" = "(binary-input,0)"
and
"FBOX:INI:FD:FBox:State:object-identifier" = "(binary-input,0)"

Entry in BacDriver's message window, as well as in BacDriver.log and also in BacScan.log:

```
DescObjectToBAC: object already exists in DMS: ObjType=[binary-input]
ObjID=[0] devID=[17] devName=[DDC 2.7 FBox]
DMSName=[FBOX:AL:ALA1:FBox:Fault] ObjName=[SBC:TC_:SM_:02 Sm1:ALA]
                                  existing object in DMS: ObjType=[binary-
input] ObjID=[0] devID=[17] devName=[DDC 2.7 FBox]
DMSName=[FBOX:INI:FD:FBox:State]
```

3. The number of scanned objects exceeds the licensed number
If supernumerary BACnet objects are detected, e.g. there are more objects as the licensed number, then the objects will be included in DMS, but not connected.
Eintrag in BacDriver Messagefenster, sowie in BacDriver.log:

```
ERROR obj beyond licenseLimit: <..> BACnet > DMS: deviceID.InstNr=<..>
objectID.InstNr=<..> objectType=<..>
```

BacDriver's license informations will be written into the following both DPs in DMS:

- System:Driver:BacDriver:LicInfo:Count
- System:Driver:BacDriver:LicInfo:Limit

In addition after the BacDriver start the license informations will be also written into the BacDriverScan.log.

9.3.11 Scan of the BACnet network

After startup, BacDriver searches for (remote) devices and their objects by sending a BACnet Who Is request.

This is the only time when new BACnet devices can be discovered. If new devices are connected to the network, BacDriver must be restarted so that they can communicate.

The references for all accessible BACnet devices remain saved over the entire lifetime of BacDriver and are used to periodically read all available data, to enter in the VisiPlus system (DMS) and, if necessary, to also record new objects (i.e. their data and structure) in DMS.

The data readout (polling of BACnet object properties) occurs:
1. periodically, according to the selected configuration, not at all, once a day, hourly or at any interval

   and / or

2. by subscribing to the BACnet services - COV Notification (**C**hange **o**f **V**alue); with this service, all value changes in a property of a data point object are independently reported to BacDriver.


It is not recommended to periodically scan the BACnet network in which the corresponding configuration option **[Settings] ScanPeriod** is defined. This produces a heavy load on the network.

9.3.11.1 Control of the BACnet properties

After the BacDriver start, all BACnet properties are controlled that are used in the BACnet templates in BMO.
if a property is not supported in the current BACstac (see Supported BACstac version), this is logged in BacDriverScan.log as follows (ex.):

```
       01.02.13 14:40:21 Report unknown BAC properties (in curr. BACstac version) used
in BMO
       01.02.13 14:40:21
=====================================================================
       01.02.13 14:42:21  Unknown: BAC Property [pcd-input-reference] ObjType=[analog-
input] for [BMO:Bac_analog-input]
```

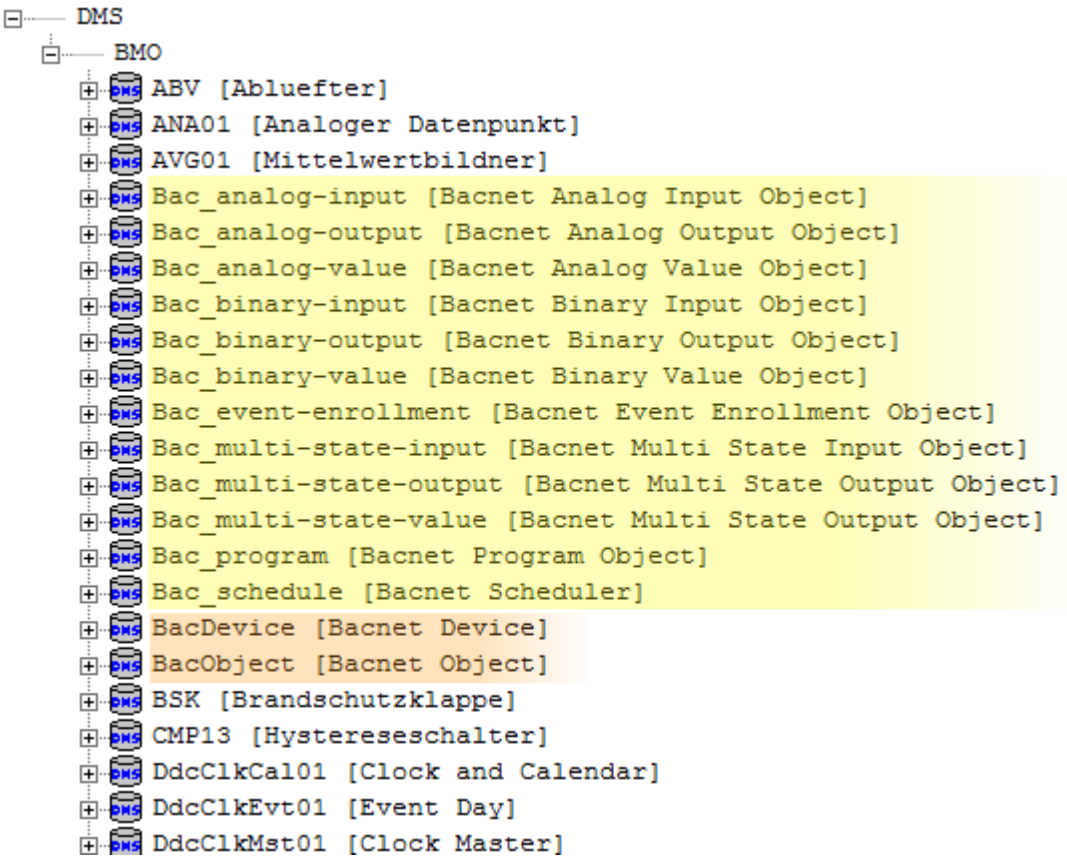
9.3.12 BACnet objects templates (BMO)

For each scanned BACnet object, BacDriver searches for its template in DMS-BMO. To be able to identify the objects, they must have a unique name, for example
"**Bac_binary output**".

If an object is not found in BMO, a general object template is used: "**BacObject**".

There is also a special object template for the BACnet device: "**BacDevice**".

Example of some BACnet objects:



9.3.12.1 Name conventions for BACnet ObjectNames

All BACnet object names must comply exactly with the name convention, must therefore not be renamed or otherwise modified (otherwise they are not identified by the BacDriver). All names are case-sensitive, a distinction is therefore made between lower and upper case.

There are two objects with fixed names:

BACnet object identifier	BMO template name
Device	**"BacDevice"**
general BACnet object	**"BacObject"**

All remaining BACnet objects have a prefix "**Bac_**".

Note **_Underscore**, followed by standard object names (with **'-'** as delimiter):

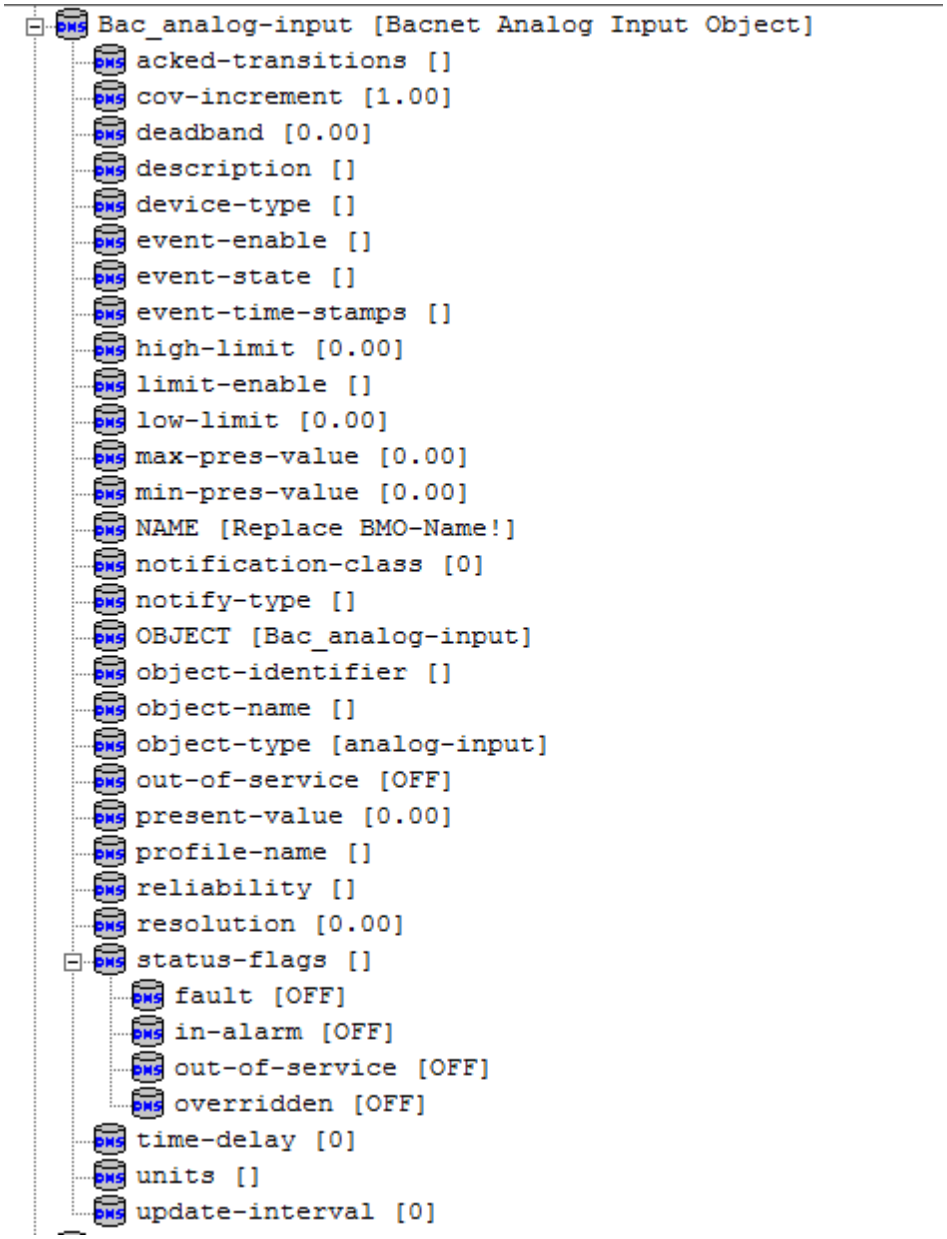
BACnet object identifier	BMO template name
access door	**"Bac_access-door"**

BACnet object identifier	BMO template name
Accumulator	**"Bac_accumulator"**
analog input	**"Bac_analog input"**
analog output	**"Bac_analog output"**
analog value	
Averaging	
binary input	
binary output	

9.3.12.2 Name conventions for BACnet PropertyNames

- All property names are written in lower case with '-' as a separator (if available).
- If a property name starts with _' letters, this means that it is not a standard BACnet property name but a local BacDriver help variable, for example **_pa_value**, **_pa_enable**, etc.

Example template of the BACnet object **Bac_analog input**:

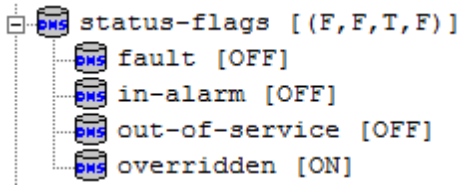


The property names are written in lower case with '-' as a delimiter.

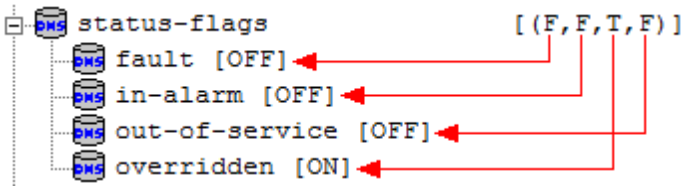
9.3.12.3 Property status flags

This property is of type STR and has the typical form = "(F,F,F,F)".

To also have direct access to the individual flags, they are mapped as independent data points (type BIT):



After the ReadProperty procedure, the string array is parsed and the values written in the respective flag DPs.
The status flags are assigned to the following status string components:



The conversion of the status string to the individual BIT flags is only performed when they exist as DPs in the corresponding BMO template.

9.3.12.4 Scheduler

The BACnet object Bac_CLK01 is used as an input interface for a BACnet schedule object. This involves a weekly timer with 2 switch-on and switch-off times each day. A binary signal is output from the object to switch a system part on or off.

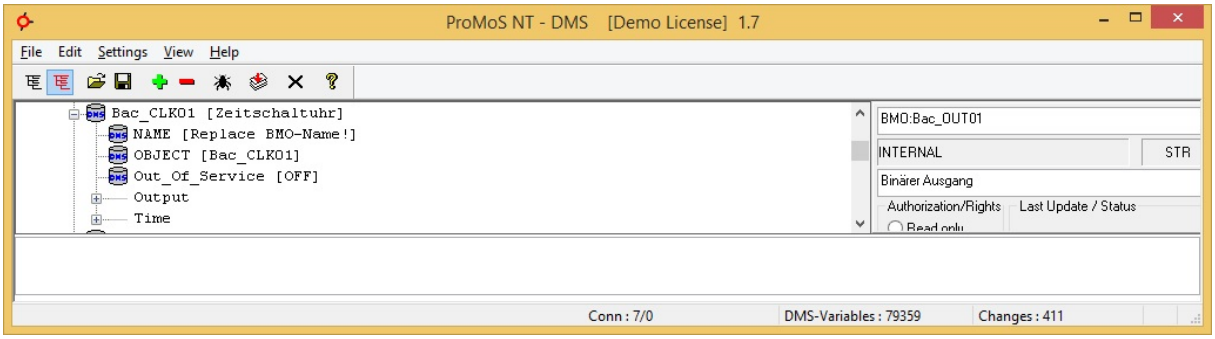
Variable list:
The BAC_CLK01 object is structured as follows:

Description	System part	GA function		BACnet object type	Attribute VLO	Notes
BAC_CLK01	Weekly switching schedule	Switch command each day Time switch output	1 2	Schedule binary Binary value	Time Output	4 switching times per day

1 Schedule binary object to record 4 switching times per day.

2 Binary value object. This is used to visualise the switching output (present value) of the schedule object. This step is necessary as the present value of the schedule object is not COV and thus the state changes are not transferred. This is achieved by using a binary value.

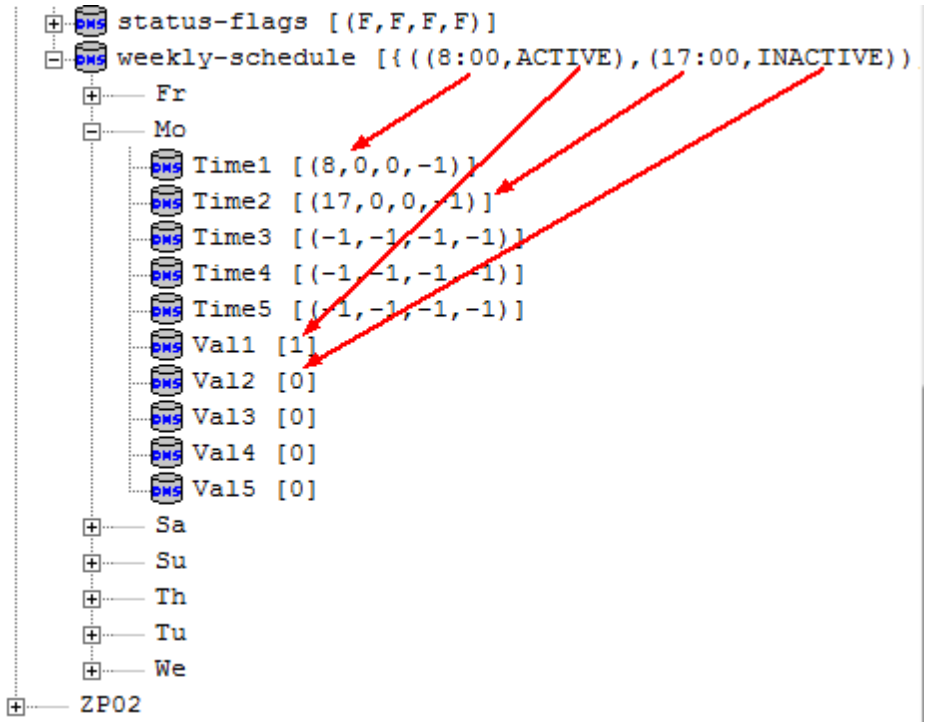
The object template in the DMS is as follows:



The property **weekly schedule** (of the schedule Object) is of string type and has the form of a nested array.
The individual array segments and links represent all weekdays and day events. The number of day events is not limited.

Ex.: {((8:00,ACTIVE),(17:00,INACTIVE)),
((8:00,ACTIVE)),
((8:00,ACTIVE),(17:00,INACTIVE)),
((8:00,ACTIVE),(17:00,INACTIVE),(19:00,ACTIVE),(23:30,INACTIVE)),
((8:00,ACTIVE),(17:00,INACTIVE)),
((00:00,INACTIVE)),
((10:00,ACTIVE),(17:00,INACTIVE))}

This string is converted when read from BACnet into BacDriver to a corresponding DMS subtree structure (only Monday (=Mo) DP is extracted here):



The individual data points are created during the run time in DMS if they do not yet exist (copied from BMO template).

The **weekly schedule** subtree is registered for visualisation and representation of the corresponding schedule tools in GE.
All subtree data points are registered by BacDriver so that each value change (time, value) in BacDriver is recorded in the equivalent DMS callback.
In this event all values of the subtree are read by BacDriver and the **weekly schedule** property newly compiled.

The subtree data points are registered via a separate DMS callback connection and organised in a map structure:

In: DMS name (time or value event)

Out: DMS registration ID of the **weekly schedule** parent DP

With the parent DP Reg. ID as the input value, you receive the corresponding BAC specifications (addr., type and InstanceNo.) from the BAC objects DMS database with which the **weekly schedule** property is written in BACnet. Immediately after (always still within the weekly schedule callback event), the property is read again; the property is thus validated and the event written again in the DMS subtree.

Scheduler type

Default type for DMS-BIT is now BACnet ENUM (previously BOOLEAN).

The following solution is not yet realised:
Normally the BACnet type should be determined from the property 'list of object property references':
for example = "{{(binary-value,0),present-value}}", then it applies for binary value type ENUM (regardless of which DMS type is defined for Val1/2/3..).

To be able to cover all possibilities in the future (even when for example 'list of object property references' is empty), an additional DP of type STR is introduced in BMO:
"Bac_schedule:weekly-schedule:_bacTyp" = "ENUM".
If defined, it determines the BACnet schedule event type. The following values are programmed:
"ENUM" -> bacType = DATA_TYPE_ENUM; "BOOLEAN" -> bacType = DATA_TYPE_BOOLEAN; "UNSIGNED" -> bacType = DATA_TYPE_UNSIGNED;
"SIGNED" -> bacType = DATA_TYPE_SIGNED; "REAL" -> bacType = DATA_TYPE_REAL; "DOUBLE" -> bacType = DATA_TYPE_DOUBLE;

9.3.12.4.1 Function description

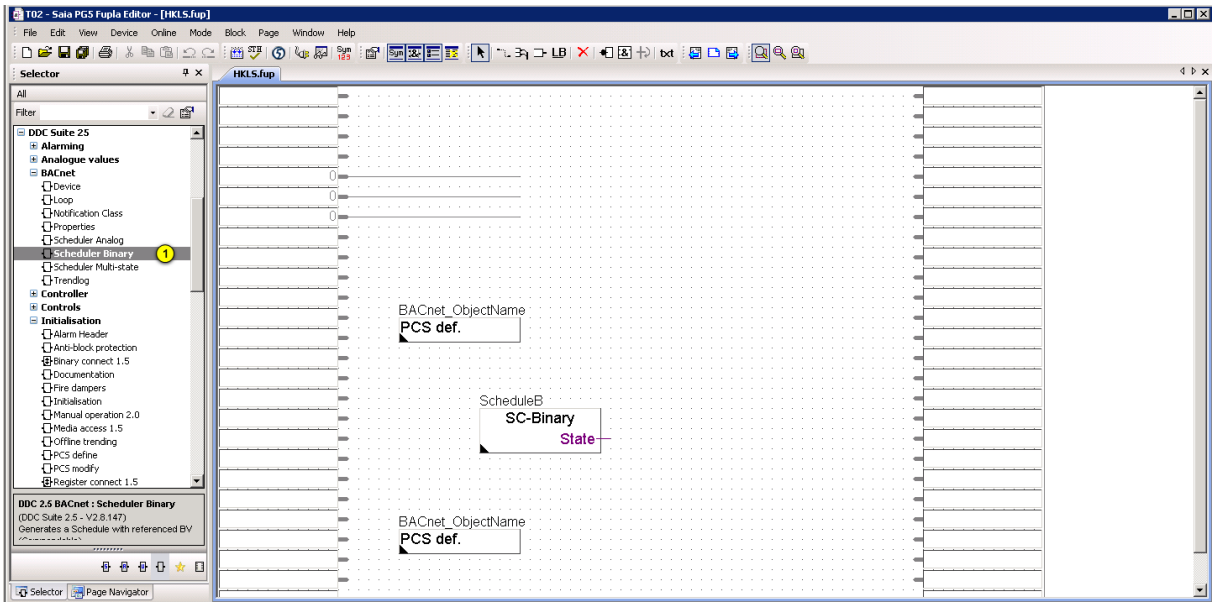
9.3.12.4.1.1 Implement in PG5

This subchapter explains how the BACnet scheduler is inserted in the PG5 and identified and visualised in Visi.Plus during a subsequent BACnet scan.

2 approaches are explained here. How the scheduler object is created directly in the BNT file or how it is created automatically from a Fupla.

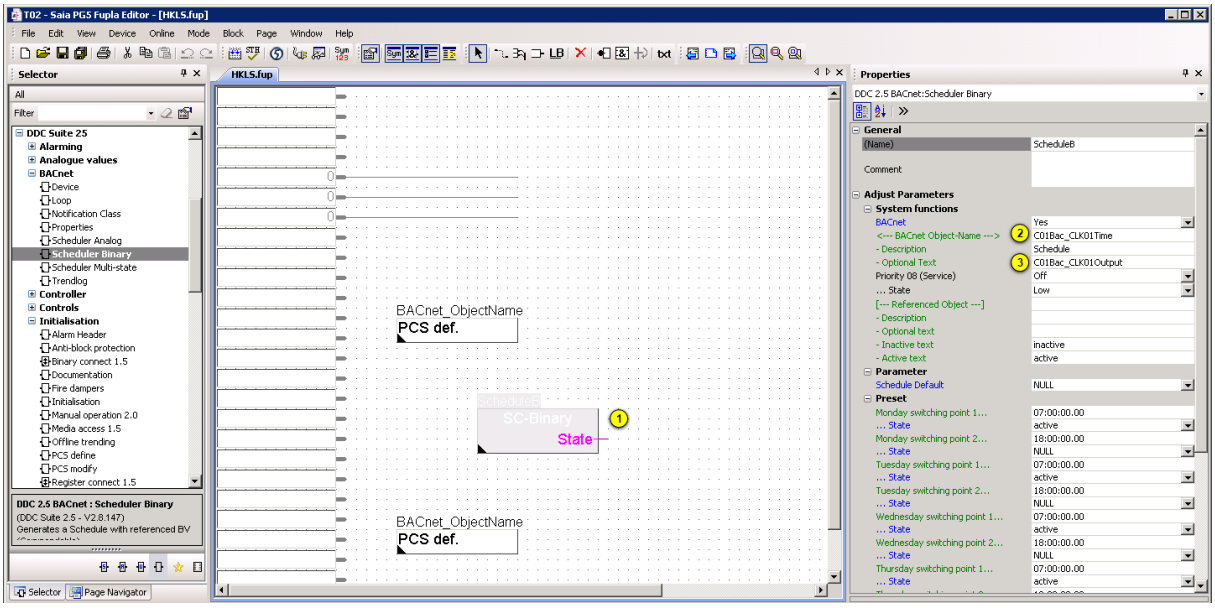
Scheduler object generated from a Fupla:

A Fupla must also be inserted with a timer in the PG5:



The timer object can be found in the DDC Suite 25 library under BACnet. This can be inserted by drag & drop.

The settings can then be made by double-clicking on the scheduler object:



(2) Name of the time object. The ending must match the template object in the DMS.

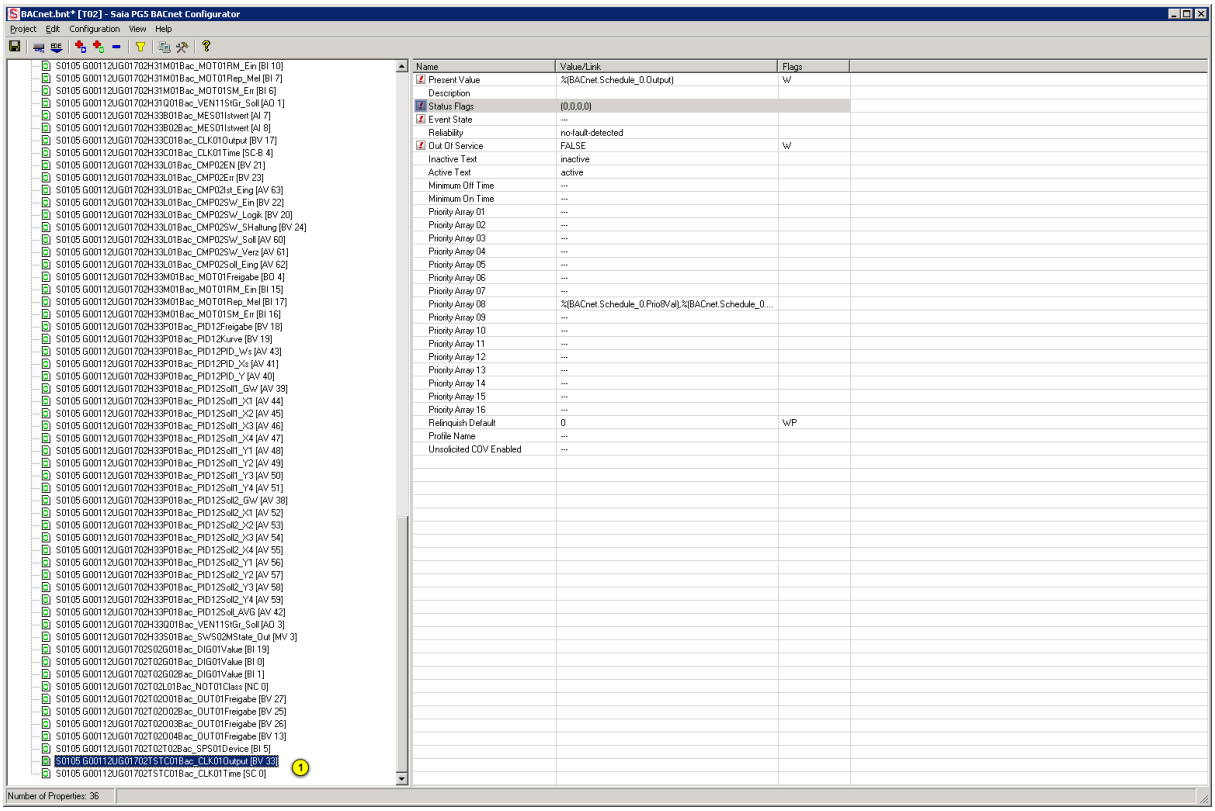
(3) Name of the output object. The ending must match the template object.

The BACnet name is generated from the BACnet_ObjectName and the schedule binary object. Here it is important that the name matches the filter options! Otherwise the BacDriver

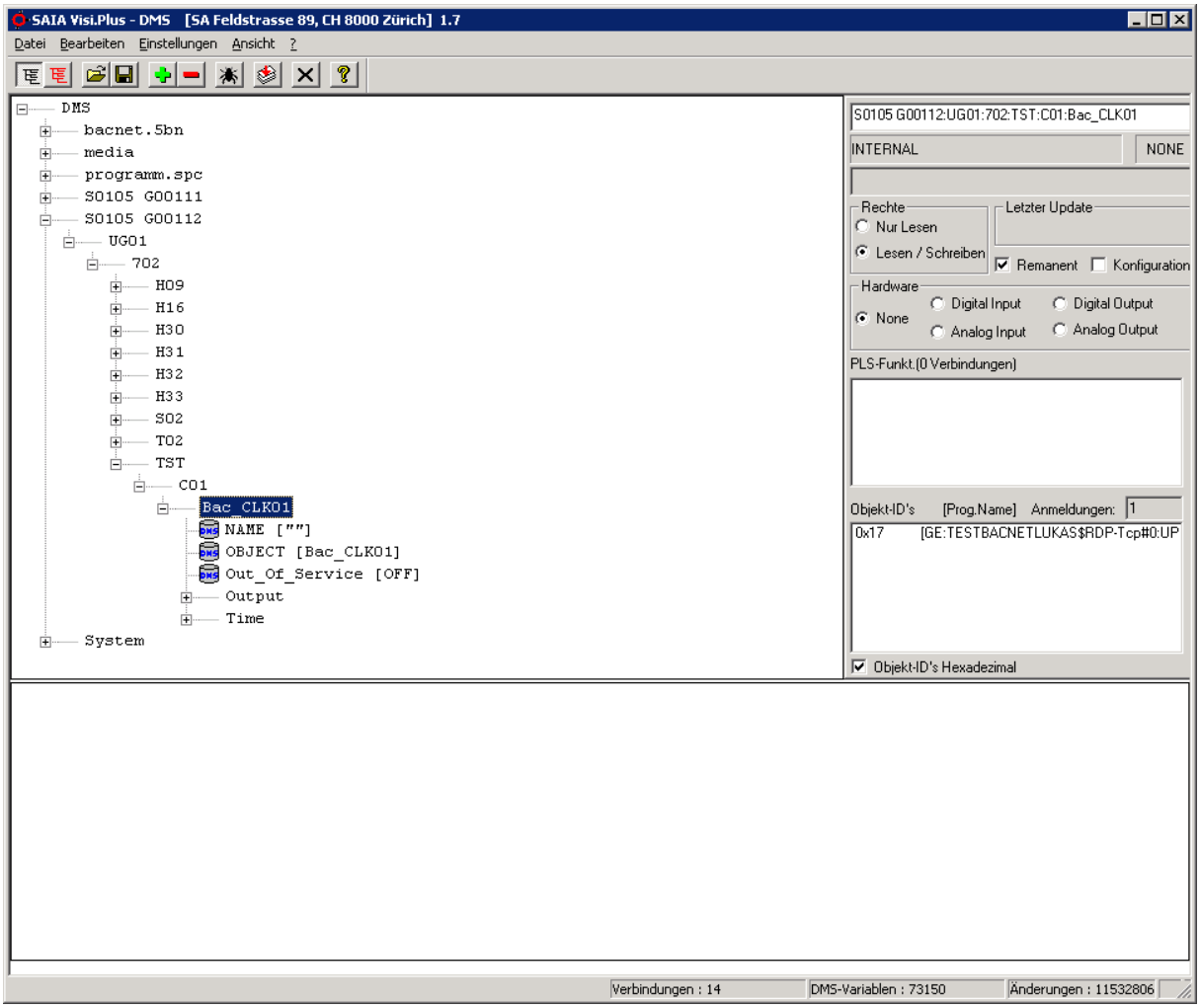
will not find the template object when scanning and cannot assign it! See Chapter Filter Option

For the PG5 to also correctly write the BACnet file, control of the BAC_DDC_BACnetSCB250 file is recommended. This file is saved in the SBC *SBC*\<*PG5 Version*>*Projects*\<*ProjectName*>*T02* or Visi.Plus project folder C:\Visi.Plus\proj \<ProjectName>\pcd\T02. Both object names must match the items (2) and (3).

The Fuplas have been correctly created and the BACnet.bnt file generated. The scheduler will automatically appear in the file as follows:

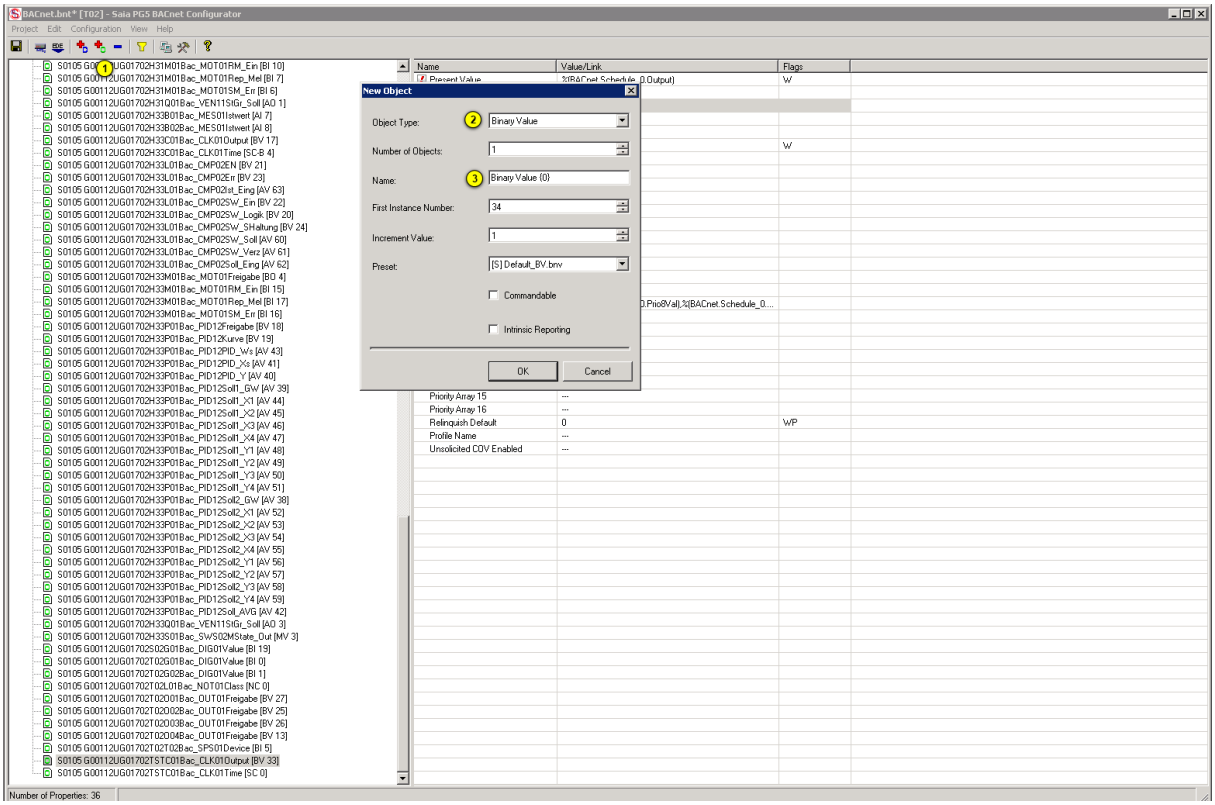


After downloading to the controller and scanning the BacDriver, the object is automatically inserted in the DMS:



Create scheduler object in the bnt file.

Unlike the first variant, the BACnet object is not created as Fupla, but directly in the bnt file. A symbol can then be referenced to the present value.

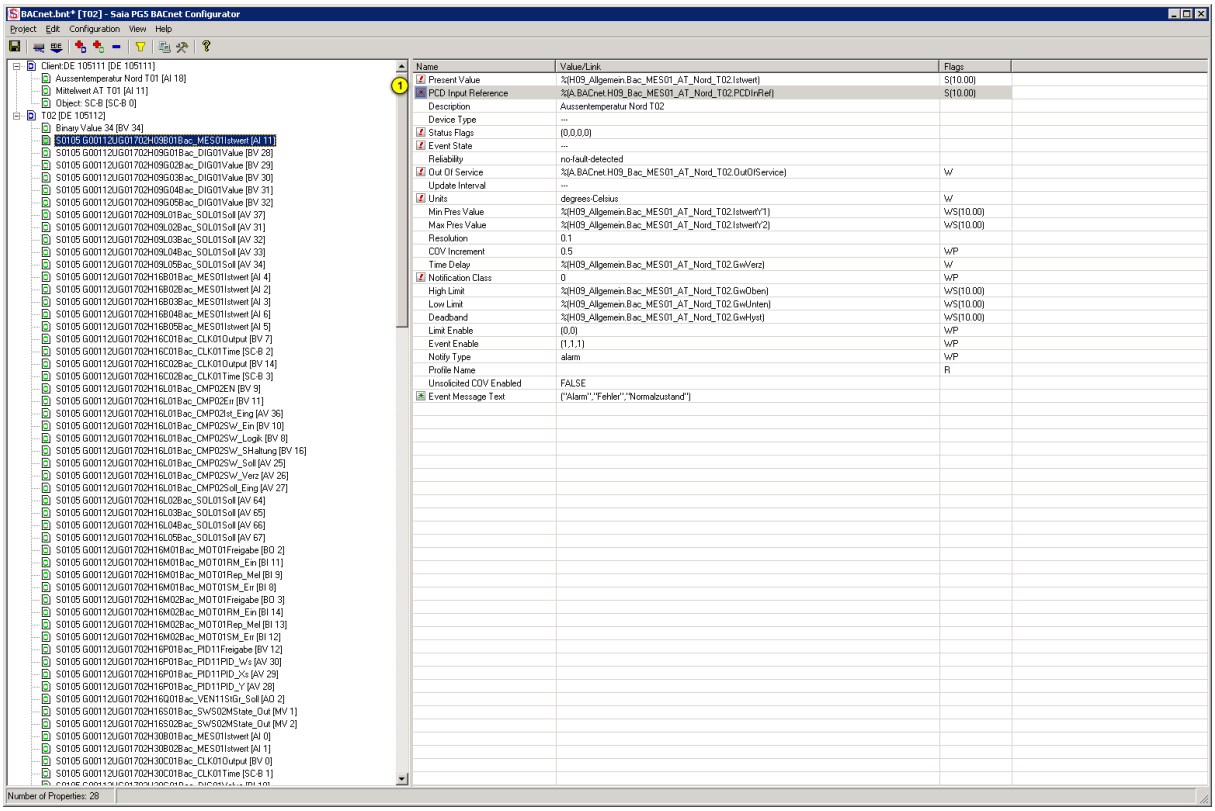


1 The symbol  enables a new BACnet object to be inserted!

2 Select the object binary value here. The 2nd object of the binary scheduler is then still required.

3 The BACnet name must still be specified under Name as defined under the filter option to enable the DMS to recognise the BACnet object.

The symbol reference is generated by double-clicking on **present value, PCD input reference** or the corresponding **property**. For more information refer to the PG5 manual or the sbc-Support page.



9.3.13 Connection of BMO objects

BMO = template object, contained and defined in DMS-BMO.

See also chapter Configuration, Filter Options

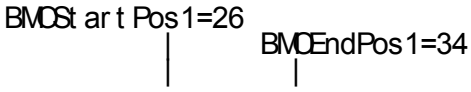
9.3.13.1 Specification of VLO name from BMO from the 'object name' property

Depending on the option **ScanBacToDMSname=77** the according BACnet property might also be "description" instead of "object-name".
See chapter Configuration, Assembly of the DMS name.

Depending on the absolute position specifications (**BMOStartPos1** and **BMOEndPos1**), the object name string is used to extract the VLO name. And from the original string value before it is split up into DMS names (see next chapter).

Warning:
- Position 1 = first letter (not 0).
- Start and end position letters belong to the resulting BMO name.

BMOStartPos1=26
BMOEndPos1=34



```
"S0006 G00387UG01701H08C03Bac_CLK01Time"
 12345678901234567890123456789012345678901234567890
                1         2         3         4
```

Result: extracted VLO name = “Bac_CLK01”

The key names **BMOStartPos1**, **BMOEndPos1** are in BacDriver.cfg in section [**Filter**].

9.3.13.2 Generation of DMS names from a BACnet property

See the above chapter Assembly of the DMS name.

The assembly string (usually ‘object name’ by default, it might be also e.g. 'property') is split up into DMS names using a mask.

The position mask is defined in the configuration key [**Filter**] **BacToDMSKeyMask**.

This is used to generate the address key from the string of the assembly (‘object name’/'description') property (i.e. DMS name).

The **BacToDMSKeyMask** position mask is coded as follows:
- A ‘**.**’ denotes an address key letter.
- A ‘**:**’ denotes that, at this position (between two letters), a DMS ‘**:**’

```
BacToDMSKeyMask = ".....:...:...:...:...:.........:........"
        Resultat: "S0006 G00387:UG01:701:H08:C03:Bac_CLK01:Time"
```

9.3.13.3 Specify the insertion position of the VLOs determined in DMS

The **BMOLevel1** insertion position is in the BacDriver.cfg in section [**Filter**].

From the DMS name specified, it is assumed:

```
                    "S0006 G00387:UG01:701:H08:C03:Bac_CLK01:Time"
                      |          |                        |
                      1. Stufe   2. Stufe ....            |
Für BMOLevel 1=6 gilt =6. Stufe (Level): BMO-Obj = "Bac_CLK01"
```

Result: in DMS under the corresponding node **Bac_CLK01**

9.3.13.4 Grouped VLOs

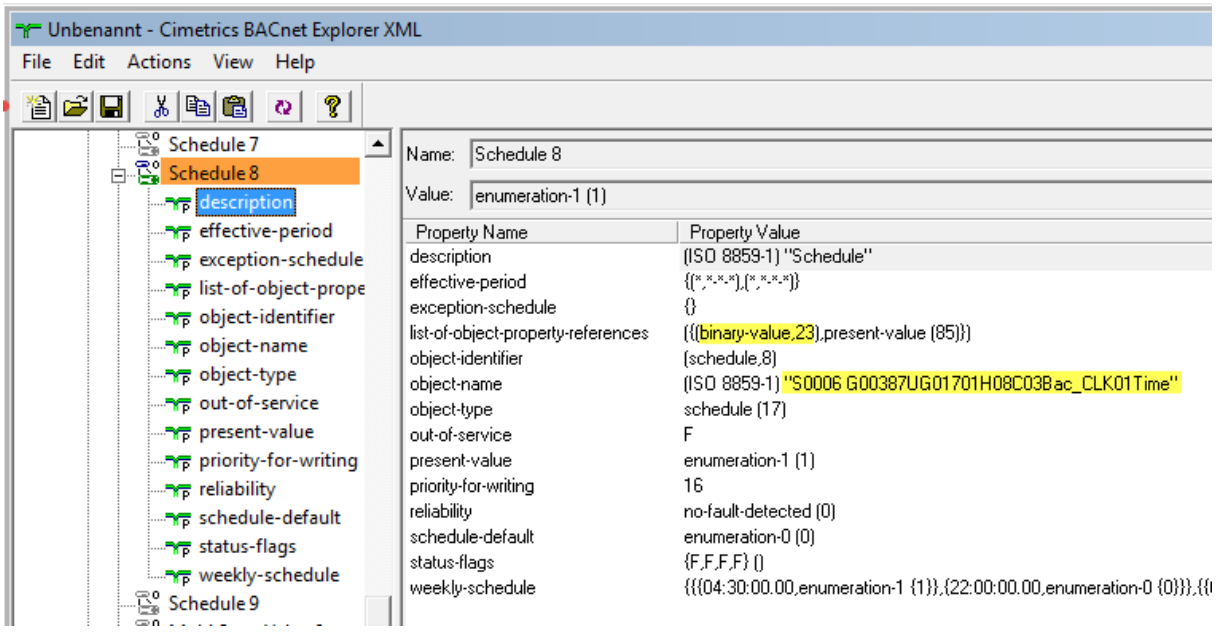
The functionality of the groups VLOs is explained here using an example of a timer_VLO.

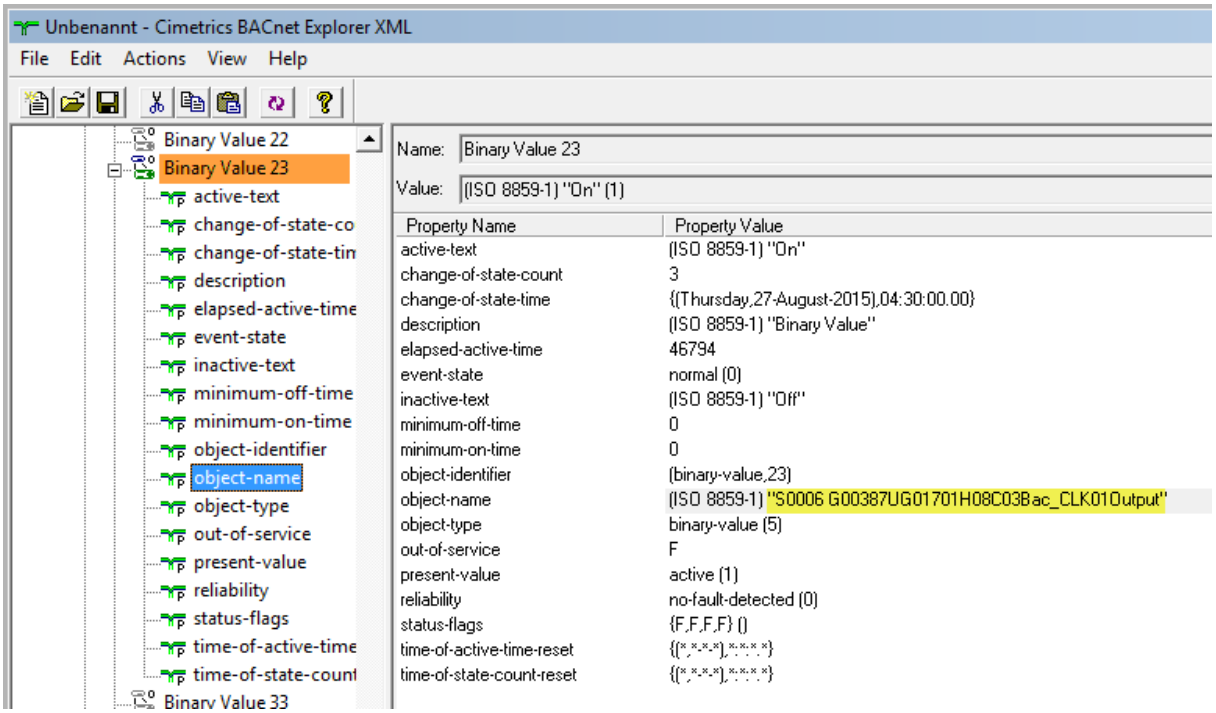
A timer (=scheduler) is made up of two objects:
1. **schedule** object
2. a commandable object that is switched by the schedule, e.g. **binary value**

For both objects to be combined in a groups VLO, their **object name** properties must be consistent, e.g.
- schedule : object name = “S0006 G00387UG01701H08C03**Bac_CLK01**Time”
- binary value : object name = “S0006 G00387UG01701H08C03**Bac_CLK01**Output”

The filter options in BacDriver.cfg [Filter] section (see previous chapters) must be set so that the VLO name can be extracted from the object name string (**Bac_CLK01**).

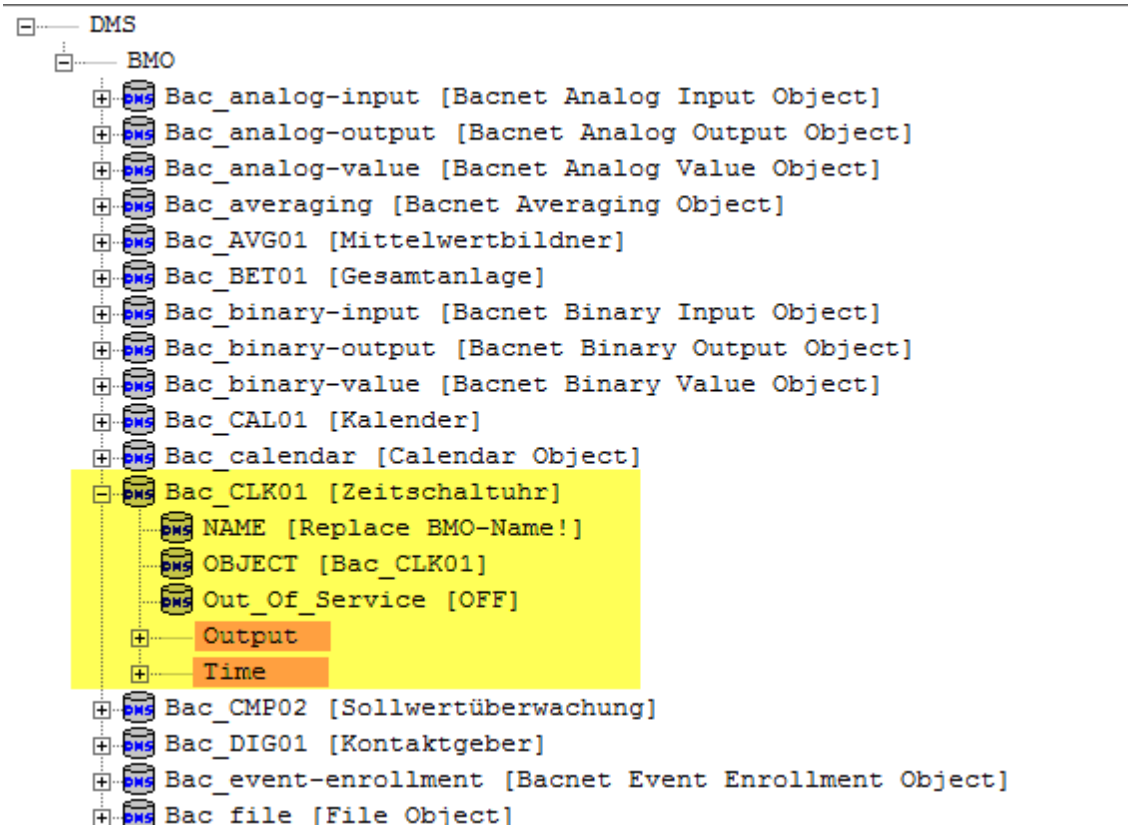
Both objects are presented in BACnet as follows:





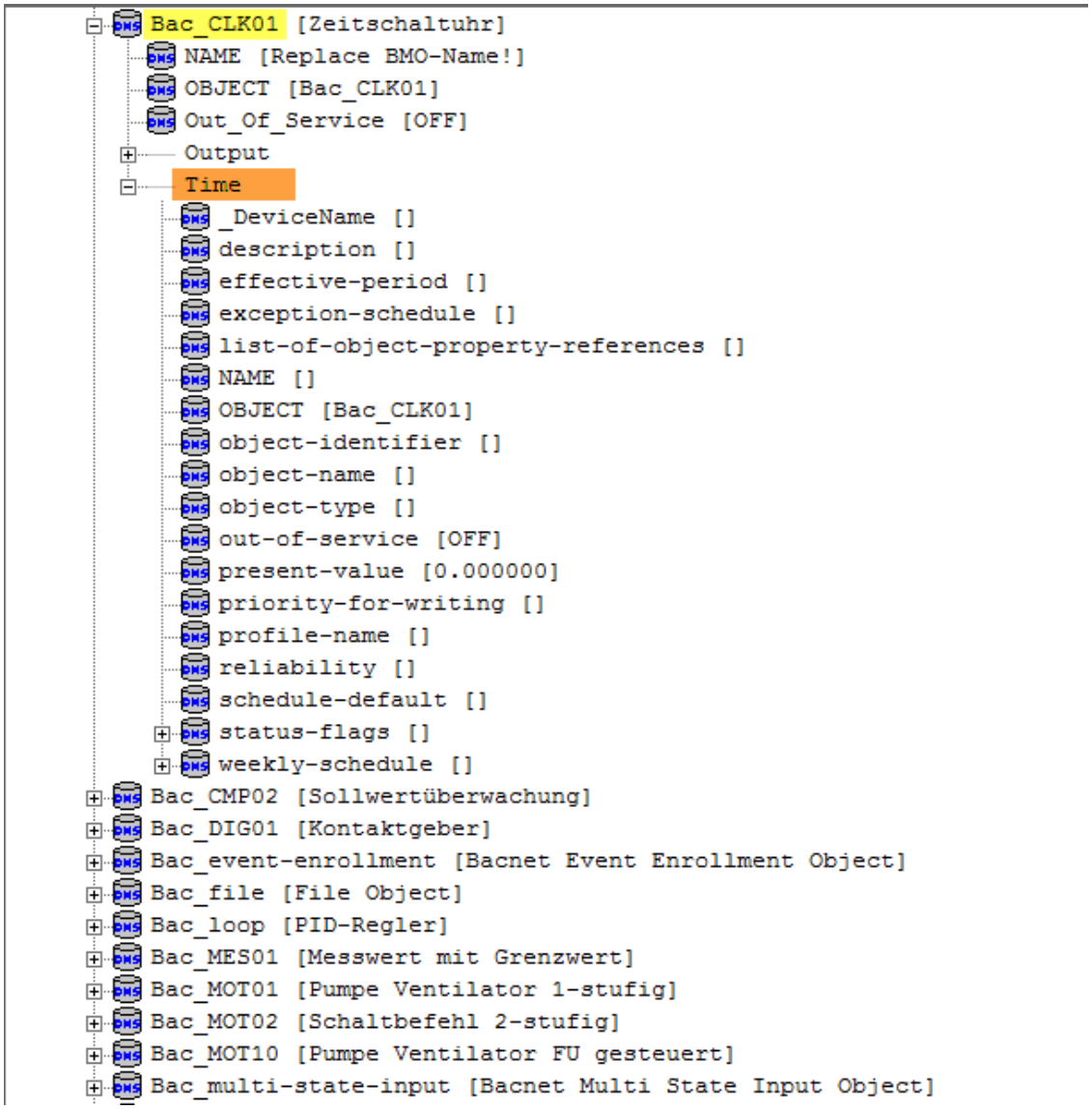
In BMO, the corresponding groups VLO Bac_CLK01 must be present

This comprises two VLOs: **Output** and **Time**



VLO Output

VLO Time



Result in DMS: the composed object:

9.3.13.4.1 Unused objects contained in a grouped VLO

A grouped VLO may contain objects, which are not allways present in the actual controller (PLC).

The visualization of such (unused) objects may be supported by two configuration methods: all unused objects are detected after the BacDriver's startup and subsequently following actions can be taken

a. unused objects are deleted from DMS: configuration option [Settings] BacDelUnusedBMONAME,
see chapter BMO templates option "BacDelUnusedBMONAME"
b. unused objects are indicated by a help variable **_Enabled** by which the visibility can be controlled

9.3.13.5 Error handling - not associable objects

If the source of the string for the DMS object node name (sub-tree) is empty or non-existent, then subsequently no DMS object name can be assembled and the BACnet object can't be inserted in DMS correctly.

Such object, if present, will be inserted under a collective node "**_BAC_UNKNOWN:_OBJ_NAME_MISSING:....**"

9.3.14 Commandable objects: Manual control, replacement value, default value (Relinquish Default)

The manual control, writing of replacement value and restoration of the default value is controlled via 2 help properties.

Each commandable object must have 2 additional properties (mandatory) (these are to be entered in the corresponding BMO object):

_pa_enable (BIT) and **_pa_value** (must be of the same type as present value).

Furthermore it is recommended that another property (not mandatory) be entered in the corresponding BMO object:

_cmdObj (BIT). If an object is recognised as commandable (the property **priority array** is provided and not empty), **_cmdObj** = TRUE (=1) is set by the BacDriver. With the aid of **_cmdObj**, such hidden/inaccessible parameters can be programmed in a general object screen template which are only characteristic for commandable objects

Visualisation

If you want to describe in a commandable object the property **present value** with the set GLT (here: BacDriver) priority, you should not do this directly but via the visualisation of the help property **_pa_value**.

You should not visualise **present value** directly but **_pa_value**.

The difference whether the replacement value or default value is written is oriented towards the value of the help property **_pa_enable:**

- If **_pa_enable** is set, then _pa_value (for example 77.0) is written in present value with the GLT priority (BacWritePrio=8 from BacDriver.cfg) in Bacnet/Bacstac.
  The updated present value (via COV) as well as the priority array (via COVP) then come from Bacstac, for example:
  {NULL,NULL,NULL,NULL,NULL,NULL,NULL,77.0,NULL,NULL,NULL,NULL,NULL,NULL,0.0,NULL}
- If **_pa_enable** is reset (deactivated - user action via GE visualisation), BacDriver writes (automatically) ZERO in present value with the GLT-Prio (pa_value remains unchanged).
  As a result, priority array = {NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,0.0,NULL}and present value = 0.0 comes from Bacstac.

BacDriver internally recognises a commandable object by the existence of the property **priority array** with a defined string content.

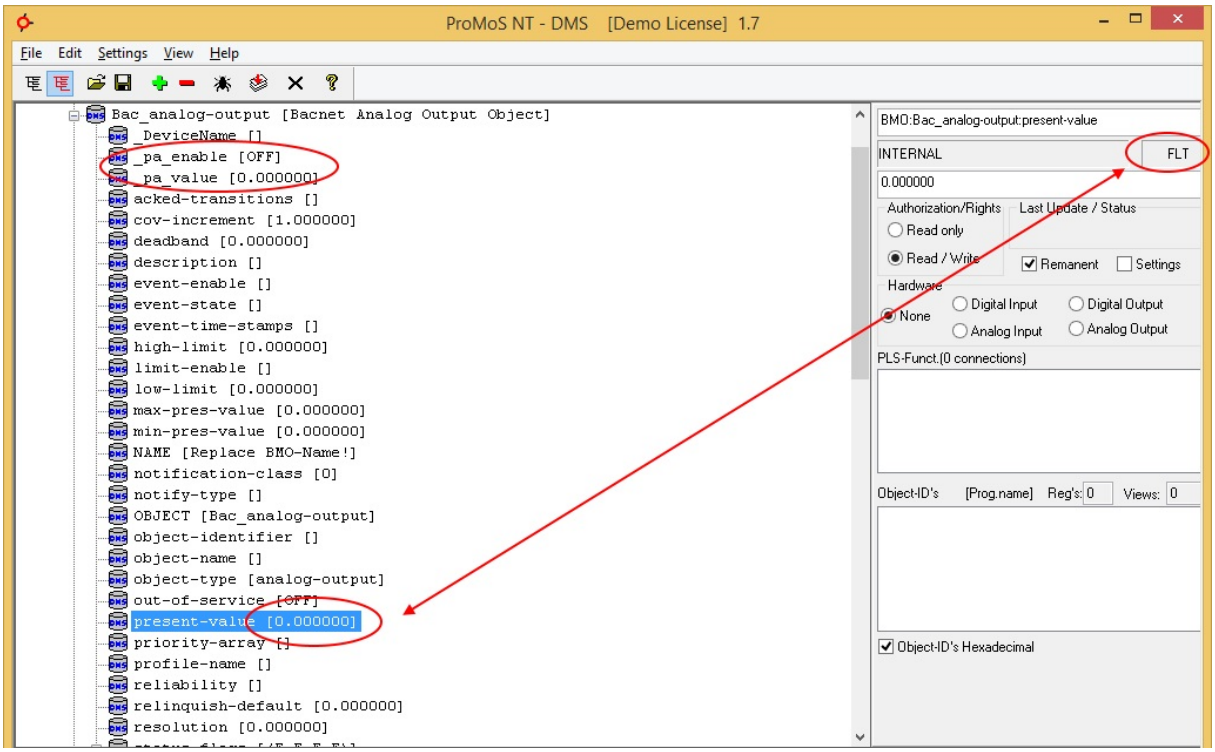
If you want to visualise the **priority array** , in BacDriver.cfg the corresponding option **COVPprioArr=1** must be set!

9.3.14.1 Manual control, replacement value, default value (Relinquish Default)

- A requirement for the correct function of the control of the commandable objects is the activation of the option COVPprioArr=1 in the section Section [Settings] in BacDriver.cfg, see Chapter "COV (Change of Value) options"

- The manual control, writing of replacement value and restoration of the default value is controlled via 2 help properties. Each commandable object must have 2 additional properties (mandatory) (these are to be entered in the corresponding BMO object):

_pa_enable Type BIT
_pa_value should be of the same type as present value (if this is not the case, the type is automatically adjusted to the present value)



- Furthermore it is recommended another property (not mandatory) be entered in the corresponding BMO object:

_cmdObj Type BIT. If an object is recognised as commandable (the property

priority array is provided and not empty), **_cmdObj** = TRUE (=1) is set by the BacDriver. Using **_cmdObj**, such hidden/inaccessible parameters can be programmed in a general object screen template which are only characteristic for commandable objects

9.3.14.2 Visualisation

If you want to describe in a commandable object the property **present value** with the set GLT (here: BacDriver) priority, you should not do this directly but via the visualisation of the help property **_pa_value**.

You should not visualise **present value** directly but **_pa_value**.

The difference, whether the replacement value or default value is written, is oriented towards the value of the help property **_pa_enable:**

- If **_pa_enable** is set, then _pa_value (for example 77.0) is written in present value with the GLT priority (BacWritePrio=8 from BacDriver.cfg) in Bacnet/Bacstac. BACnet then notifies the updated present value (via COV) as well as the priority array (via COVP) for example: {NULL,NULL,NULL,NULL,NULL,NULL,NULL,77.0,NULL,NULL,NULL,NULL,NULL,NULL,0.0,NULL}

- If **_pa_enable** is reset (deactivated> user action via GE visualisation), present value is copied to pa_value and then ZERO is written in present value with the GLT priority. BACnet notifies priority array = {NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,0.0,NULL}and present value = 0.0

BacDriver recognises a commandable object by the existence of the property **priority array**, with a defined string content and in this case sets an object variable (help property) **_cmdObj** to TRUE (=1), provided that this property exists in the corresponding BMO object template.
With the aid of **_cmdObj**, the visibility of the additional controls for commandable objects can be controlled in an object template.

If you want to visualise the **priority array** , in BacDriver.cfg the corresponding option **COVPprioArr=1** must be set!

9.3.14.3 Functionality when writing with the same priority for commandable objects

The following section describes the behaviour, regarding properties of Visi.Plus/BacDriver (GLT) and other clients, for example, control panels in BACnet, when several or all clients write with the same priority.

- If a client writes with the same GLT_Priority, its values are immediately accepted by BacDriver, i.e. present value and priority array are automatically updated in DMS,

whereby the current present value is also copied in _pa_value. _pa_enable is also set, i.e. due to this action, the manual control in GLT is also activated (this behaviour could be made configurable in the future if need be).

- If changes are made to _pa_value (or present-value) in GLT, all clients accept these values automatically, even if GLT relinquishes control (relinquish the priority), _pa_enable resets and writes ZERO in the present-value.
- For the BacDriver start, the client values are accepted (present value, priority array).

This functionality is only ensured when in BacDriver.cfg the corresponding option **COVPprioArr=1** is set!

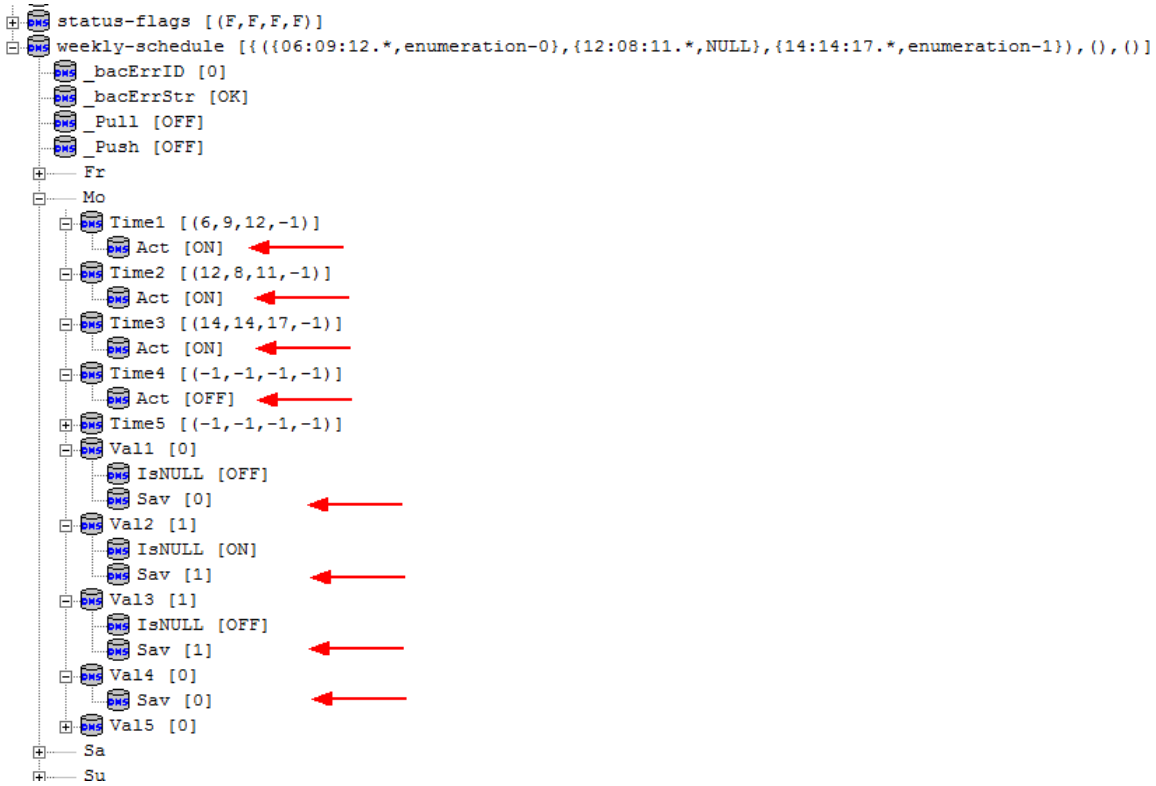
9.3.15 Schedule object

9.3.15.1 Property weekly-schedule

The **weekly-schedule** property is of type string and is in the form of a nested array.
The individual array elements and members represent all weekdays and day events. The number of day events is not limited.

Example: {({06:09:12.*,enumeration-0},{12:08:11.*,NULL},{14:14:17.*,enumeration-1}),(),(),(),(),(),()}

This string is converted into corresponding DMS subtree structure when reading from BACnet in BacDriver:



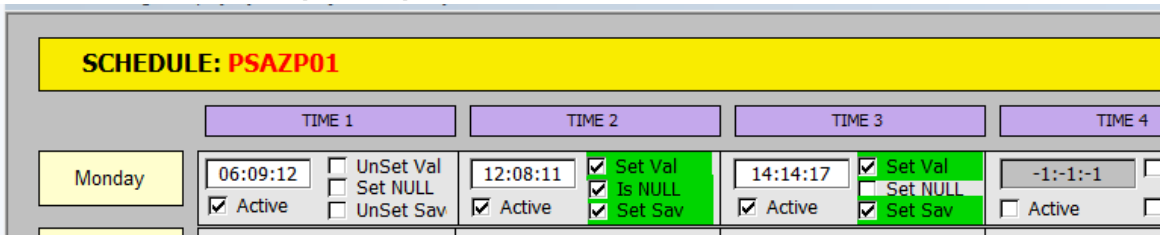
The following visualisation extentions are configurable (s. Weekly-schedule Optionen):

BacDriver automatically generates the following aid datapoints: "**Sav**" and "**Act**":

"**Act**" is appended to each "Time<Nr>". If it is = OFF, then this time-DP keeps only the time values (for future use), but it is deactivated, e.g. if BacDriver composes the weekly-schedule string, then this time DPs are ignored.

"**Sav**" is appended to each "Val<Nr>". It keeps a copy value of "Val" (it has also the same data type). In a schedule visualization or in a schedule BMO template only "Sav" DPs should be visualized, instead of the real "Val" values. The reason is that this switched value remains stored in this way and will also be displayed, even if this time is deactivated. In the following reactivation of the time the value will be transferred to the current switch-DP "Val".

Visualization of the top example:



9.3.15.2 Data type of the switched object

The connected (=switched) object property is referenced in schedule property **"list-of-object-property-reference"**
e.g. "{{(binary-value,0),present-value}}".

BacDriver detects the data type and stores it in the local data point "**_schedTyp**", which is appended to the schedule object.
This is especially useful when the weekly-schedule string is not written (delivered) by GE, but by an external client,
for example by a Web schedule application e.g. "Choose date Time". In such case, the Web client should first read from DMS the corresponding data type of the switched object and then compile the weekly-schedule string (with the correct data type), and then write it back into DMS.
DMS will thereon inform BacDriver, which in turn analyzes the string (dissected), composes the schedule object accordingly and write it back into BACnet.

If the **"list-of-object-property-reference"** property is empty, i.e. the output is switched directly by the schedule object itself, then BacDriver takes the data type of the present-value (of the scheduler).

_schedTyp can have the following values: "ENUM", "REAL", "DOUBLE", "SIGNED", "UNSIGNED", "BOOLEAN".

Depending on _schedTyp value following addendum data points are then created: **_SC_mul**, **_SC_ana** and **_SC_bin**, all of them of type BIT.
- if _schedTyp = "SIGNED" or "UNSIGNED" then "**_SC_mul**" = ON,
- if _schedTyp = "REAL" or "DOUBLE" then "**_SC_ana**" = ON
- if _schedTyp = "BOOLEAN" or "ENUM" then "**_SC_bin**" = ON.

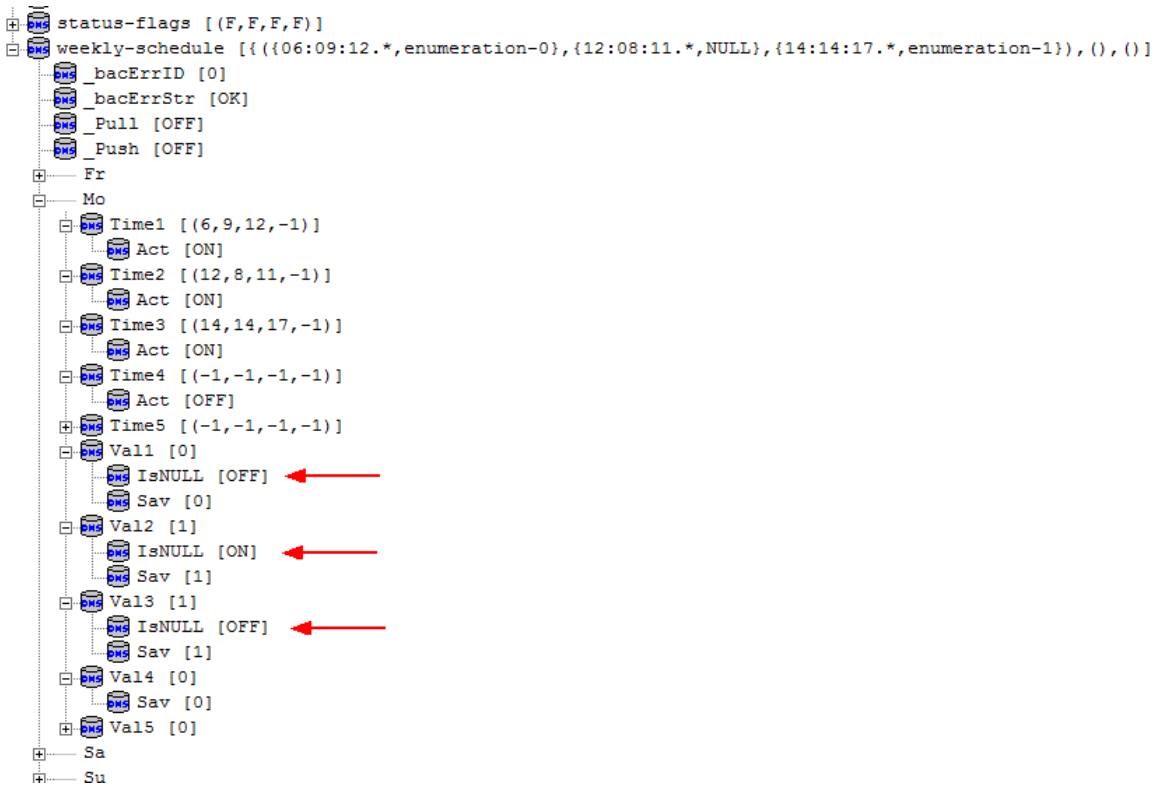
9.3.15.3 Setting NULL of the switched object

BacDriver creates in DMS under the property DP "**weekly-schedule:<day>:Val<Nr>**" automatically following auxiliary data points:

"**IsNULL**" type BIT: this DP is set, if the switching DP is set =**NULL**.

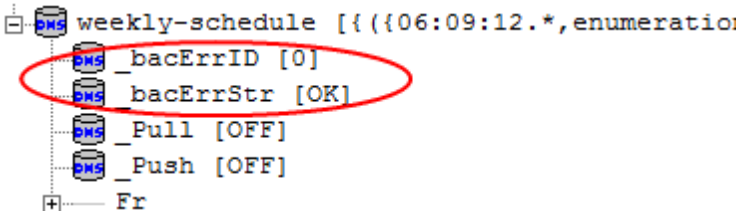
Accordingly "IsNULL" should be used in the visualization, in order to
1) display the "NULL" state of the switching point
2) by setting/reseting of "IsNULL" the switching point can be set to "NULL", respectivelly be reset back to a value

Example.: {({06:09:12.*,enumeration-0},{12:08:11.*,NULL},{14:14:17.*,enumeration-1}),(),(),(),(),()}



9.3.15.4 Error Handling

BacDriver creates in DMS under the weekly-schedule property DP automatically following auxiliary data points:



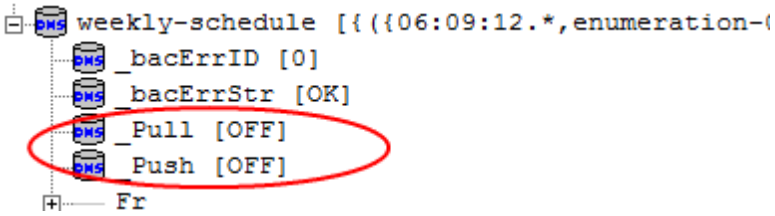
"**_bacErrID**" type DWS, e.g.: 53

"**_bacErrStr**" type STR, e.g.: "Value out of range"

These data points contain the final result of a schedule write operation,
for example _bacErrID = 53 _bacErrStr = "Value out of range" if hour value > 24.

9.3.15.5 Visualisation

BacDriver creates in DMS under the weekly-schedule property DP automatically following auxiliary data points:



"**_Pull**" trigger DP: if set ON, then BacDriver polls the complete schedule-object into DMS. This is useful when it is combined in GE with the "Schedule-image-loading" button, so that all schedule values are automatically updated and polled before the image is opened:
BACnet -> DMS

"**_Refr**" trigger DP: if set ON, then BacDriver writes current schedule values into BACnet and polls them immediatelly back:
DMS -> BACnet -> DMS
This can serve as a validation of all current values in schedule GE.

9.3.15.6 Integration of an external timer application

- The time format string is communicated via the DMS data point property "weekly-schedule"
- Via this DP the schedule object can be completely controlled in an external application, e.g. web based scheduler
- To make this possible, the time format string has a standard format, which is independent from the data type of the switched/connected object.

On the respective device the property **weekly-schedule** of the schedule object can have different formats, depending on the data type of the connected object, on the PCD type (manufacturer), engineering tool, etc.

Example:
- {({04:35:00.00,active},{22:00:00.00,inactive}),(),(),(),(),(),()}
- {({04:35:00.00,TRUE},{22:00:00.00,FALSE}),(),(),(),(),(),()}
- {({04:35:00.00,1},{22:00:00.00,0}),(),(),(),(),(),()}

- {({04:35:00.00,enumeration-1},{22:00:00.00,enumeration-0}),(),(),(),(),(),()}
- {({04:35:00.00,1.0 },{22:00:00.00,0.0 }),(),(),(),(),(),()}

Thus, if the time format can be clearly validated in an external schedule application, there must be uniform format:

- all type information from the time format string will be eliminated from the BacDriver (This has the consequence that from the time format string itself no information / conclusions about the type of switched / connected object are possible)
- there are only integers, real numbers (decimal sign) and / or indication "NULL"
- if a switching value must be initialized with default value (Relinquish Default), then a "NULL" string instead of the value must be written into the time format string

Examples of different file formats of the DMS DP "weekly-schedule"

Note that also instead of each value there can stand a "NULL" indication:

Multistate-value, data type indication in DP "**_schedTyp**"="UNSIGNED"
weekly-schedule="{({08:00:00.00,1},{18:00:00.00,3},{22:00:00.00,2}),
({08:00:00.00,1},{18:00:00.00,3},{22:00:00.00,2}),
({08:00:00.00,1},{18:00:00.00,3},{22:00:00.00,2}),
({08:00:00.00,1},{18:00:00.00,3},{22:00:00.00,2}),
({08:00:00.00,1},{18:00:00.00,3},{22:00:00.00,2}),
({08:00:00.00,1},{18:00:00.00,3},{22:00:00.00,NULL}),
({08:00:00.00,1},{18:00:00.00,3},{22:00:00.00,NULL})}"

Analog-value, data type indication in DP "**_schedTyp**"="REAL"
weekly-schedule="{({08:00:00.00,1.5},{18:00:00.00,15.755},{22:00:00.00,13467.1}),
({08:00:00.00,1.5},{18:00:00.00,15.755},{22:00:00.00,13467.1}),
({08:00:00.00,1.5},{18:00:00.00,15.755},{22:00:00.00,13467.1}),
({08:00:00.00,1.5},{18:00:00.00,15.755},{22:00:00.00,13467.1}),
({08:00:00.00,1.5},{18:00:00.00,15.755},{22:00:00.00,13467.1}),
({08:00:00.00,1.5},{18:00:00.00,15.755},{21:33:33.00,13467.1},{23:44:44.00,NULL}),
({08:00:00.00,1.5},{18:00:00.00,15.755},{22:00:00.00,13467.1})}"

Binary-value, data type indication in DP "**_schedTyp**"="ENUM"
weekly-schedule="{({04:30:00.00,1},{22:00:00.00,0}),
({04:30:00.00,1},{22:00:00.00,0}),
({04:30:00.00,1},{22:00:00.00,0}),
({04:30:00.00,1},{22:00:00.00,0}),
({04:30:00.00,1},{22:00:00.00,0}),
({04:30:00.00,1},{22:00:00.00,0})}"

Operation

- After each change of value of weekly-schedule string, the scheduler will be completely read back from BACnet:
  External web app user input →DMS →BacDriver →**BACnet Schedule validation** →BacDriver →DMS →External web app: **validation of the user's input**

Error handling

- If OK:
  the modified time format string is in weekly-schedule with new values, in DP type DWS "**_bacErrID**" = 0, DP type STR "**_bacErrStr**" = "OK"
- If error:
  weekly-schedule contains again the original content of the time format string (before the modification took place),
  the error code and error message are in "_bacErrID" and "_bacErrStr" , e.g.: "**_bacErrID**" = 53, DP type STR "**_bacErrStr**" = "Value out of range"

9.3.16 Calendar object

Geben Sie hier den Text ein.

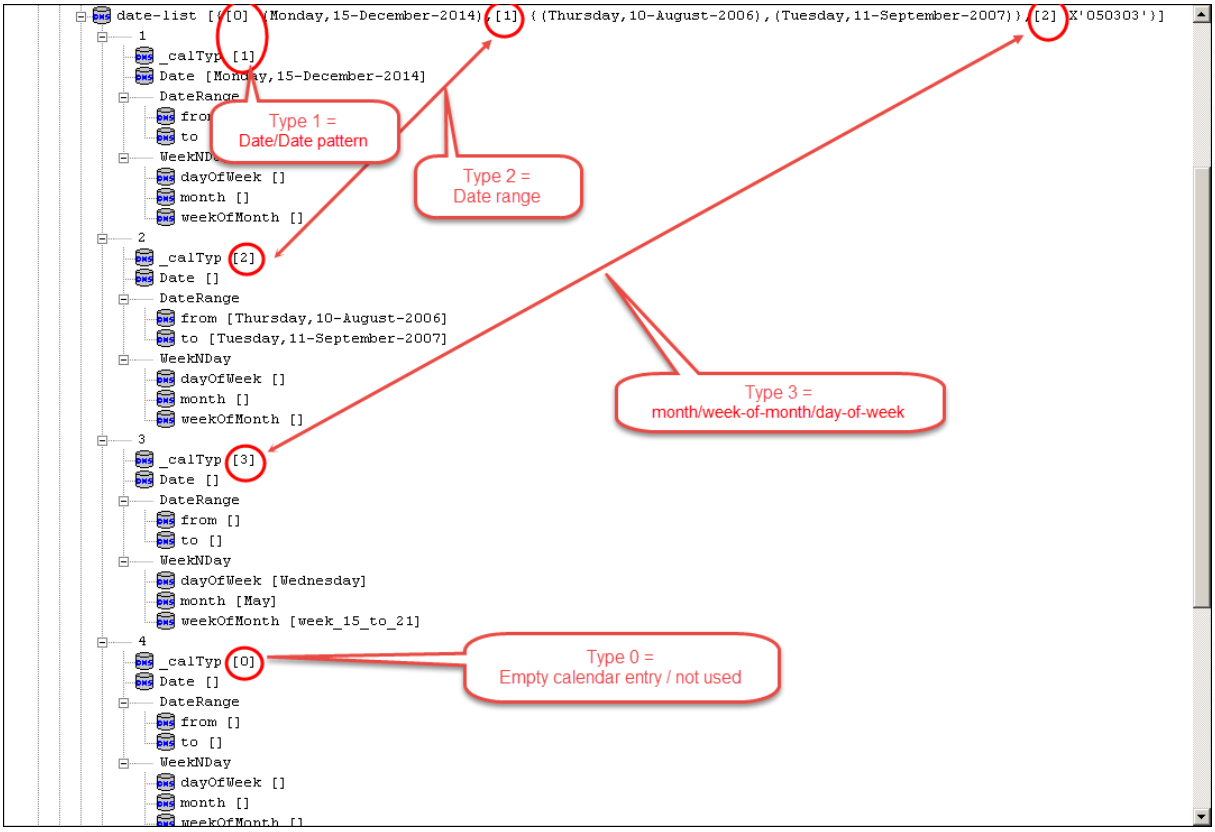
9.3.16.1 Property date-list

The **date-list** property is of type string and is in the form of a nested array: a BACnetLIST of BACnetCalendarEntry, each of which is either a specific date or date pattern (Date), range of dates (BACnetDateRange), or month/week-of-month/day-of-week specification (BACnetWeekNDay).

If the current date matches the calendar entry criteria, the present value of the Calendar object is TRUE.

Example: {[0] (Monday,15-December-2014),[1] {(Thursday,10-August-2006), (Tuesday,11-September-2007)},[2] X'050302'}

This string is converted into corresponding DMS subtree structure when reading from BACnet in BacDriver:



9.3.17 File object

Base BMO object for the BACnet file object should contain following two help properties:

_Pull Type BIT: at each transition 0 → 1 will be the associated file read out of the device, saved on the harddisk and subsequently **_Pull** will be reset to zero again.

_Path Type STR: save destination for the data file of the file object. If **_Path** is empty, then the file will be saved in <proj>\pcd location.

The **_Path** string can contain following macros:

"{INSTDIR}"	e.g. "C:\Visi.Plus"
"{PROJDIR}"	e.g. "C:\Visi.Plus\proj"
"{CFGDIR}"	e.g. "C:\Visi.Plus\cfg"
"{BINDIR}"	e.g. "C:\Visi.Plus\bin"
"{ACTPROJDIR}"	e.g. "C:\Visi.Plus\proj\<ProjectName>"

These are the same macros as in the DMS control function EQF (=EQU from file).

At present only file read operation is supported.
Read file works either with file-stream access or with file-record access in both ways.

9.3.18 Alarming

9.3.18.1 Introduction

This chapter describes specifications of the development of BacDriver for BACnet Alarming.

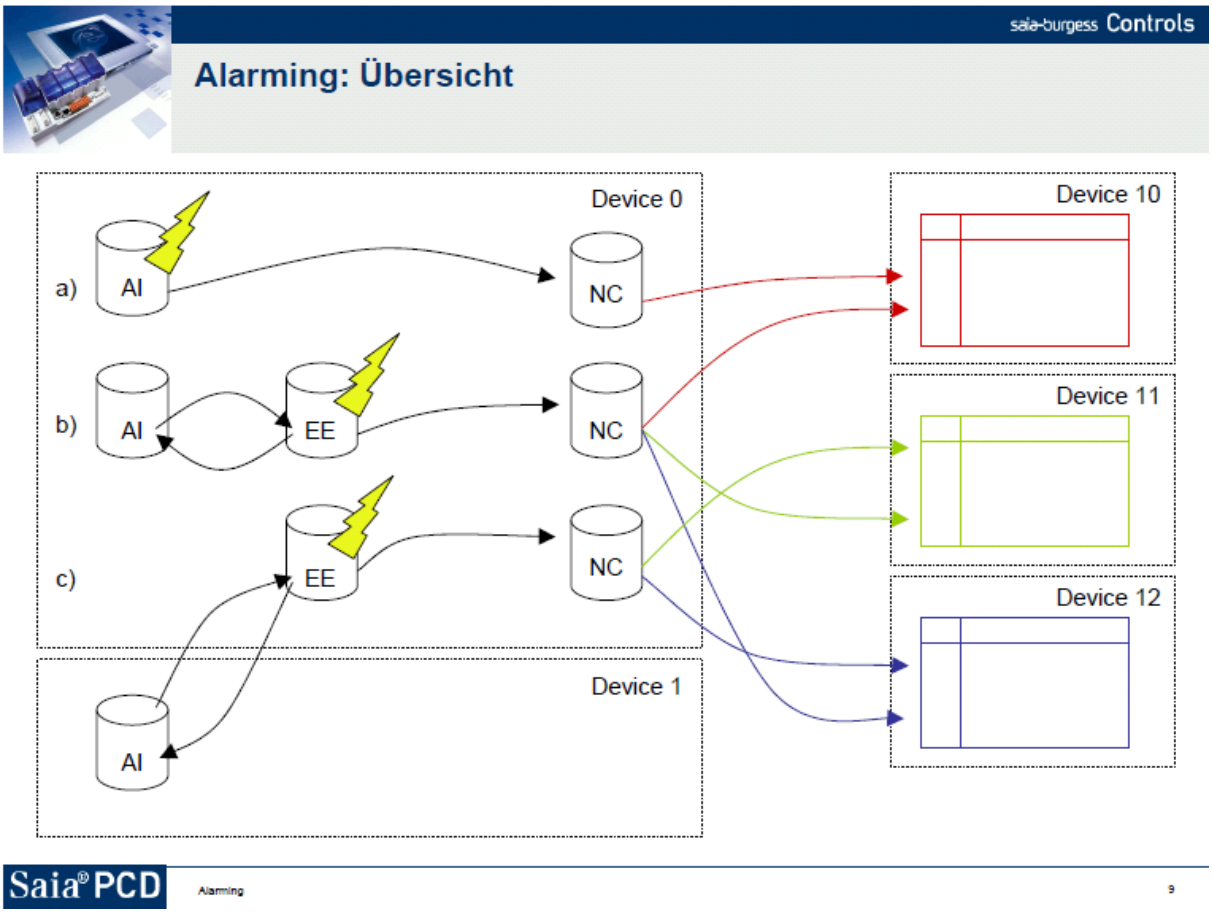
BACnet Alarming is a matter of the event, respectively alarm notification by means of BACnet notification classes.
What it's all about, is described in chapter Specification Alarming.

The chapter Specification BacDriver describes the extensions, which are necessary in BacDriver, to implement the alarming.

The adjustments for the BACnet alarming in the BMO objects are summarized in the chapter Specification BMO Objects.

9.3.18.2 Specification Alarming

BACnet Alarming is a matter of the event, respectively alarm notification by means of BACnet notification classes.
The notification class (NC) takes the task of a dispatcher, which only forwards the events / alarms to the registered recipients.
The alarm handling is processed further in the corresponding BACnet objects AI, BI etc. The device manufacturers may also their BACnet have implemented in in different forms.



BACnet Alarming von Saia-Burgess Control

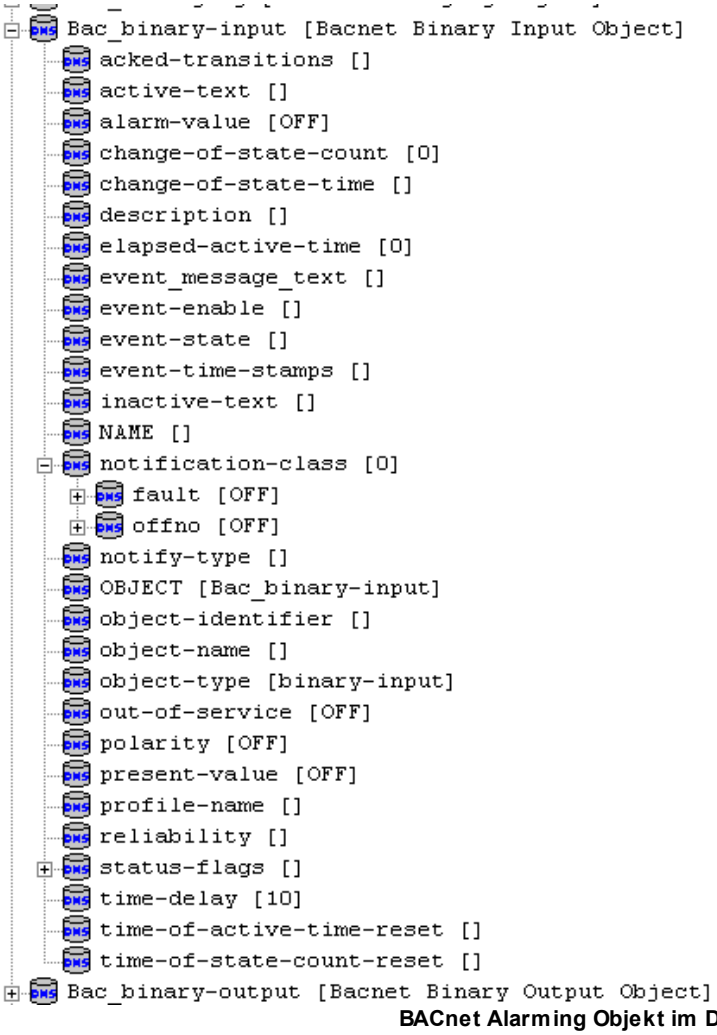
Saia-Burgess Control uses notification classes and confirmed event notifications for the event/alarm notification. The acknowledge of the event/alarm notification also has to take place without timestamp.

9.3.18.3 Specification BacDriver

For the implementation of the BACnet alarming is the existing BacDriver expanded. The Cimetrics BACstac Library provides the already available callback methods for event / alarm notifications. The BacDriver sets the hooks to the callback methods for the Confirmed or Unconfirmed event / alarm notification.
Upon the occurrence of events / alarms the BacDriver will be automatically called. The BacDriver is extended by an event queue and an event thread to process the events / alarms.

Data management

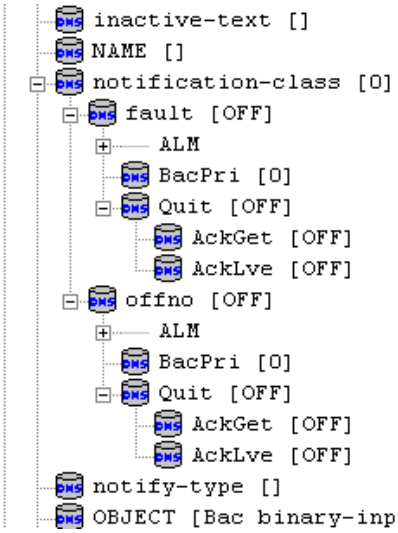
The data management in DMS for the BACnet Alarming in the example Binary Input:



BACnet Alarming Objekt im DMS

For the BACnet alarming the two data points **Offnormal** and **Fault** are in DMS used.

- aks:bac-binary-input:notification-class:offno
- aks:bac-binary-input:notification-class:fault



BACnet Alarming Fault/Offnormal Datenpunkte

Initialization

During the scan of the BACnet objects the BacDriver registers itself for DMS callbacks for the both BACnet quit flags to provide an acknowledgment of the events / alarms.
Per BACnet objects are these the following two flags:

- aks:bac-binary-input:notification-class:offno:Quit
- aks:bac-binary-input:notification-class:fault:Quit

After the scan of the BACnet objects the BacDriver registeres with its address (IP / Port / Process ID) as a recipient in the recipient list of all existing notification class (NC) objects in BACnet Stack of the particular device. After BacDriver's address is entered in the recipient list then all events / alarms are going to be initialized.
Pending or not fully acknowledged events / alarms from the BACnet device will be queried (method GetEventInformation).

Notification

If an event / alarm is notified and the status transition is toOffnormal, toLowLimit toHighLimit, etc., then in DMS the bit aks:bac-binary-input:notification-class:offno will be set to [ON].
When state transition is toNormal then in DMS the bit aks:bac-binary-input:notification-class:offno will be reset to [OFF].
If an event / alarm is notified and the status transition is toFault then in DMS the bit aks:bac-binary-input:notification-class:fault will be set to [ON].
If there is an another state transition then the bit aks:bac-binary-input:notification-class:fault will be again reset to [OFF].
Both offno bit as well as fault bit connect to the existing Visi.Plus alarm mechanism.

AckRequired and Acknowledgement (Quit)

Whether an acknowledgment of an event / alarm is required, is specified by BACnet. The default is read in at an event / alarm occurrence and saved in DMS in the four Ack flags:

- aks:bac-binary-input:notification-class:offno:Quit:AckGet
- aks:bac-binary-input:notification-class:offno:Quit:AckLve
- aks:bac-binary-input:notification-class:fault:Quit:AckGet
- aks:bac-binary-input:notification-class:fault:Quit:AckLve

In order to the predetermined combination of Ack and Quit flags process correctly, then the Visi.Plus alarm handling must be modified.
The acknowledge against the BACnet device is without timestamp.

Priorities

The implementation of a BACnet priority value in a Visi.Plus priority follows by means of a simple, configurable priority mappings.
The mapping provides defaults approximately as follows:

- BACnet Priority Visi.Plus

- 0..31 Life safety messages Prio 1
- 32..63 Safety messages Prio 2
- 64..95 Technical alarms Prio 3
- 96..127 Fault messages Prio 4
- 192..255 Operating messages Prio 5

- 128..191 Maintenance messages Prio 6

Configuration

See also chapter Configuration.

For alarming the BacDriver configuration is expanded by the following items:

[Alarming]
AlarmNotification=0/1

AlarmTextTransfer=0/1

AlarmPrioMapType=2
AlarmPrio1=0-31
AlarmPrio2=32-63
AlarmPrio3=64-95
AlarmPrio4=96-127
AlarmPrio5=192-255
AlarmPrio6=128-191

[Monitor]
MonitorEVT=0/1

9.3.18.4 Specification BMO Objects

In the BMO object templates the new alarming structure is introduced.
The previous alarm definition under the flag "present-value" is omitted.

The following BMO objects support the new alarm handling:

- Analog Input
- Analog Output
- Analog Value

- Binary Input
- Binary Output
- Binary Value

- Multi State Input
- Multi State Output
- Multi State Value

- Event Enrollment
- Loop

9.3.18.5 Configuration

For alarming the BacDriver configuration is expanded by the following items:

The following options are only in the global configuration BacDriver.cfg possible, e.g. these settings are **valid globally for all devices**.

Sektion **[Alarming]**

```
; Alarm and event notification activation: if =1 then the alarm and event
notification from BACnet
; will be enabled (default =1)
; This option must be also activated for trending (see Trending
configuration if [Trending] UseTrending=1)
```
AlarmNotification=0/1

```
; Alarm and event notification message-text transfer: if =1 then the alarm
and event notification
; message-text pass-through from BACnet into DMS/AlmMng will be enabled
(default =0)
```
AlarmTextTransfer=0/1

```
; Alarm priority mapping type: defines the BACnet to Visi.Plus priority
mapping (default =2)
; 1 Pass-through: the BACnet priority values will be written directly into
the DMS data-point
; 2 Visi.Plus-light: the BACnet priority values will be converted to 6
predefined priority classes
```
AlarmPrioMapType=2

```
; Alarm priority class: when Visi.Plus-light mapping is enabled the 6
priority classes has to be
; defined by the following mapping schema: Visi.Plus priority class =
BACnet priority value range
; Pay attention that the priorities 5 and 6 are interchanged in comparison
to BACnet definition
; (default AlarmPrio1=0-31) Life safety / Safety
; (default AlarmPrio2=32-63) Property safety / Security
; (default AlarmPrio3=64-95) Supervisory / Technical
; (default AlarmPrio4=96-127) Trouble / Fault
; (default AlarmPrio5=192-255) Miscellaneous lower priority / Status
; (default AlarmPrio6=128-191) Miscellaneous higher priority / Maintenance
```
AlarmPrio1=0-31
AlarmPrio2=32-63
AlarmPrio3=64-95
AlarmPrio4=96-127
AlarmPrio5=192-255

AlarmPrio6=128-191

Sektion **[Monitor]**

```
; Monitor EVTs (Alarm and Event) notifications, default =0
```
MonitorEVT=0/1

9.3.18.5.1 Alarming types activation

In addition to the BACnet alarm via
notification-class:fault:ALM:... und **notification-class:offno:ALM**:...
there is also the "classic" Visi.Plus alarming via
present-value:ALM:...

In the section **[Alarming]** there are two options for them:

These options can be **configured per device in the device dependent configuration** file BacDevice-<INSTNR>.cfg.

```
; (===> This option can be configured per device <===)
; BACnet alarming activation: if =1 (and AlarmNotification=1) then the
alarming via
; notification-class:fault:ALM:... and notification-class:offno:ALM:...
will be enabled (default =1)
; =0: - run time: ignore NC notification
;     - scan BAC: no BMO import of NC..ALM DP's   (scan DMS: no action)
; =1: - run time: use NC notification             ----> this is default
;     - scan BAC: no action (complete BMO import) (scan DMS: no action)
Use_NC_Alarming=1

; (===> This option can be configured per device <===)
; 'Classic' alarming activation: if =1 then the alarming via
; present-value:ALM:... will be enabled (default =0) (this option is
independent from AlarmNotification)
; =0: - scan BAC: no BMO import of present-value:ALM DP's  (scan DMS: no
action) -> this is default
;     - run time: no action
; =1: - scan BAC: no action (complete BMO import)          (scan DMS: no
action)
;     - run time: no action
Use_PV_Alarming=0
```

9.3.19 Trending (Historic Data Logging)

9.3.19.1 Introduction

BACnet trending, or trend-logging, is performed by two objects:

- Trend Log object

and

- Trend Log Multiple object.

Both objects collect time-stamped data, either periodically or from change-of-value notifications (COV's).

- the Trend Log object collects data only from one property of one object, in one device, somewhere in the system.
- the Trend Log Multiple object can collect data from multiple properties, from multiple objects and from multiple devices in the system.

 The actual BacDriver version (1.7.1.256) supports in the moment only the Trend Log object.

The logged data can be referenced by two ways:
- either via the logged property data point of the logged object
- or also directly in the trend log object itself. In this case the reference DP name is dependent of the logged data type:
  BIT ➔ **_logged-variableBIT** (that is file name is e.g. "System26_PCD_VAL-Sinus__logged-variableBIT.hdbx")
  DWU ➔ **_logged-variableDWU** (that is file name is e.g. "System26_PCD_VAL-Sinus__logged-variableDWU.hdbx")
  FLT ➔ **_logged-variableFLT** (that is file name is e.g. "System26_PCD_VAL-Sinus__logged-variableFLT.hdbx")

9.3.19.2 Configuration of the BacDriver

For trending the BacDriver configuration is expanded by additional items in the section [Trending].

The following options are only in the global configuration BacDriver.cfg possible, e.g. these settings are **valid globally for all devices**.

 The described functionality is the same for both types of trend-log buffer: either ring buffer (stop-when-full = F) or linear (fixed) buffer (stop-when-full = T).

Sektion **[Trending]**

```
;=============================================================================
=======================
; Following options are for trend-objects and log-buffer alarm/event
notification handling.
; -------> For Trending also following option must be activated: [Alarming]
AlarmNotification=1
; (===> All options of this section can be configured per device <===)
```

```
; (===> This option can be configured per device <===)
; If UseTrending=0 then all kind of trend operations will be disabled.
Default=1.
```
UseTrending=1

```
; (===> This option can be configured per device <===)
; Poll trend-log buffer data (see [Settings] PollPeriod)
; If PollTrends=1 then trend-log buffer will be polled within the period
defined in [Settings] PollPeriod.
; Default =0 = deactivate trend polling. Trends will be then read in only
if manual trigger "_Pull"
;            in DMS trend-log will be set _Pull=ON or on log-buffer alarm/
event notification.
```
PollTrends=1

```
; (===> This option can be configured per device <===)
; If ReadTrendsAtStartup=1 then at the BacDriver's startup, just before
COVs are subscribed,
; log-buffer of all trend-log objects are read in, starting with the last
date/time stamp recorded in PDBS.
; Default =0, no trend data from log-buffers will be read in at startup.
```
ReadTrendsAtStartup=0

```
; (===> This option can be configured per device <===)
; If referenced trended object property (referenced in trend-log in log-
device-object-property) has in
; DMS also another TRD data point and object defined for trending via
HDAMng:
; -> then if the option Set_HDA_TRD_Off=1 then BacDriver deactivates HDAMng
TRD object in DMS
;    by resetting TRD:CHANGES, TRD:DIFF and TRD:TIME to OFF.
; -> If Set_HDA_TRD_Off=0, then no modifications are made in object's
trended property in DMS in TRD.
; ==> This option is only relevant if option UseTrending=1 e.g. only if
trend operations are enabled. <==
; Default =1 = deactivate trending via HDAMng.
```
Set_HDA_TRD_Off=1

```
; (===> This option can be configured per device <===)
; Trend recording data file destination - the following options are
available:
; The trend data is stored as a file either in the logged object or in the
trend-log object itself or in both
; If UseTrendRefFile=0 -> trend data file = logged obj file "<logged-obj
DMS-Name>:<logged-property>.hdb"
;                    e.g. "PCD:Sinus:present-value" -> "PCD_Sinus_present-
value.hdb"
; If UseTrendRefFile=1 -> trend data file = trend-log file "<trend-log DMS-
Name>:<_logged-variableTYPE>.hdb"
;          e.g. FLT-type: "PCD:VAL-Sinus:_logged-variableFlt" -> "PCD_VAL-
Sinus__logged-variableFlt.hdb"
;          e.g. BIT-type: "PCD:VAL-Sinus:_logged-variableBit" -> "PCD_VAL-
Sinus__logged-variableBit.hdb"
;          e.g. DWU-type: "PCD:VAL-Sinus:_logged-variableDwu" -> "PCD_VAL-
Sinus__logged-variableDwu.hdb"
; If UseTrendRefFile=2 -> both trend data files will be recorded.
```

```
; Default =0. This option has only effect if above option UseTrending=1
```
UseTrendRefFile=0

Read trend data in notification mode (from trend-log object)

A trend-log object notifies BacDriver when its buffer is full. BacDriver reads subsequently the trend data in and sends them to the PDBS.
In order for BacDriver to receive this message (notification), the following option must be activated:

[Alarming]
AlarmNotification=1

Read trend data in polling mode (from trend-log object)

BacDriver queries the trend data (and sends them to the PDBS) that have not yet been obtained.
The read process can be triggered by following actions:

- After the BacDriver start, if the option **ReadTrendsAtStartup=1** is activated.
- If in DMS the data point **_Pull** is set to **= ON** either in the trend-log object structure or in the corresponding referenced object itself.
  This data point can be used in the visualization, for example, for a "Refresh" button.
- If in a viewer (GE/PWA/PWEB) a screen is opened which contains a trend.
- If the option **PollTrends=1** is activated. The trend data not yet received are then pulled periodically.
  The periode must be defined in `[Settings] PollPeriod=3600` e.g. one hour.

The trend data recording can also function solely in the polling mode, when the option AlarmNotification=0 is deactivated.

In principle, only trend data are queried which are not yet contained in the corresponding PDBS hdb file.

The trend data are sent by the BacDriver directly into the PDBS, without interfacing the HdaMng.

Supported DMS data points in trend-log objects: _TRDready

- In every trend-log object in DMS is an auxiliary data point "**_TRDready**" of type BIT.
- The "**_TRDready**" is set → **OFF** → **ON** whenever new trend data from the trend-log object is read from the buffer and written into the PDBS (no matter if triggered by the user or client or via BACnet notification).
- This DP does not have to be contained in the BMO TrendLog object, it is generated automatically by the BacDriver.
- **_TRDready** can be used to optimize subsequent trend-log buffer read accesses.

9.3.19.3 Configuration of the trend-log object

The automation developer should pay close attention to the defined properties in the following list:

- **start-time** property – Start logging process on a specific date and time.
- **stop-time** property – Stop logging process on a specific date and time.
- **record-count** property – By indicating “0” will clear the buffer.
- **log-interval** property – Specifies the logging interval.
- **stop-when-full** property – Stop logging when the buffer is Full.
- **buffer-size** property – Indicates the maximum number of records in the buffer.
- **log-buffer** property – Details of the buffer itself.

If the trend recording should be continuously, then the **stop-when-full** property must be set **FALSE:**

- then the trend-log buffer is operated as a ring buffer
- the trend data are then either automatically received via notification (**AlarmNotification = 1**) or periodic polling
- or can be received anytime by (manual) triggering of the **_Pull** data point

If rather a unique event data recording is considered, then the **stop-when-full** property must be set **TRUE:**

- then the trend-log buffer is operated as a fixed, linear buffer
- the trend data are then received **only once** either automatically via notification (**AlarmNotification = 1**) or periodic polling
- or can be received anytime by (manual) triggering of the **_Pull** data point
- the trend data can then also be received via the DP trigger **_TRDpullFix**; at the same time the buffer is then emptied and the trend-log object is restarted

9.3.19.4 Useful DMS data points for trends

Following auxiliary data points are presented in DMS trend-log object:

_TRDready (type BIT): notifies the user / client that new trend data have been inserted into PDBS

- **_TRDready** will be set: → **OFF** → **ON** whenever new trend data from the trend-log object is read from the buffer and written into the PDBS
  (no matter if the read in action was triggered by the user or client or via BACnet notification).
- This DP does not have to be contained in the BMO TrendLog object, it is generated

automatically by the BacDriver.
- **_TRDready** can be used to optimize subsequent trend-log buffer read accesses.

_EnabledTL (type BIT): indicates whether the object is recorded by a trend log (set = ON).
- can be used for display control

_Pull (type BIT): forces the trend data to be read in (and sent to PDBS) if triggered (set = ON) by the user / client
- only such trend data will be queried by BacDriver which have not yet been obtained, i.e. which are not yet contained in PDBS

_TRDpullFix (type BIT): only for **fix / non-circular** trend-log buffer (**stop-when-full** = **TRUE**):
if triggered (set = ON) by the user / client then it forces the trend data to be read in (and sent to PDBS)
- only such trend data will be queried by BacDriver which have not yet been obtained, i.e. which are not yet contained in PDBS
- concurrently the trend-log buffer will be reset and the trend-log object re-enable (restarted) again

Visualization of the trend data in trend-log object

➔ Following option must be set: **[Trending] UseTrendRefFile=1**
In this case, the data are recorded in the hdb(x) file "<Trd-obj-DMS-Name>:<Trd-logged-var-type.hdb(x)>" in dependency of data type of the logged property.
The data access (visualization in GE/PWA) occures then via this DP "<Trd-logged-var-type.hdb(x)>".

In the trend object are then following auxiliary DPs set, which reflects the data type of the logged property (BIT, DWU or FLT):
- BIT: _logged-variableIsBIT = ON (_logged-variableIsDWU = OFF and _logged-variableIsFLT = OFF)
- DWU: _logged-variableIsDWU = ON (_logged-variableIsBIT = OFF and _logged-variableIsFLT = OFF)
- FLT: _logged-variableIsFLT = ON (_logged-variableIsBIT = OFF and _logged-variableIsDWU = OFF)

The visualization DPs are (again in dependency of the logged property data type):
- BIT: _logged-variableBIT (that is file name is e.g. "System26_PCD_VAL-Sinus__logged-variableBIT.hdbx")
- DWU: _logged-variableDWU (that is file name is e.g. "System26_PCD_VAL-Sinus__logged-variableDWU.hdbx")
- FLT: _logged-variableFLT (that is file name is e.g. "System26_PCD_VAL-Sinus__logged-variableFLT.hdbx")

These visualization DPs keep also the last logged value.

9.4 SDriver (sdriver.exe)

The SDriver is used for communication with Saia PCD® controllers (SBus protocol). All types of communication, such as serial interface, modem, USB and TCP/IP, are supported.

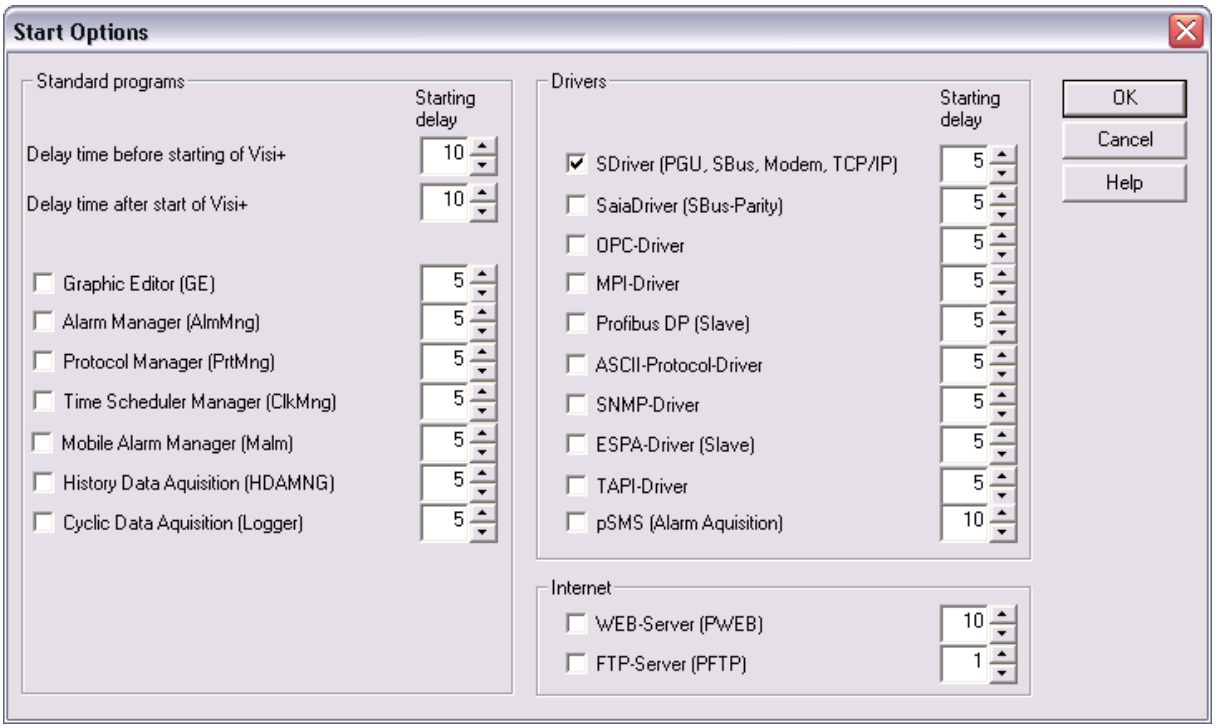
SDriver is based on SCOMM-DLL from Saia Burgess Controls. Therefore all PG5 tools can be used in parallel with the SDriver.

9.4.1 Starting the program SDriver

The **SDriver** can be started at any time by double-clicking on its name in a file management application. The program file is "**Sdriver.exe**" in the default installation directory **C:*Visi.Plus**bin***.

Of course, a shortcut created on the desktop is even more convenient. Alternatively, you can start the SDriver through the **Project Manager** module**,** where the programs to be started for each project can be selected. To do so, proceed as follows:

1. Start the **Project Manager** module.
2. Select the menu item or the **<Start Options>** button.
3. Mark the checkbox **SDriver (PGU, SBus, Modem, TCPIP)**.
4. Confirm with **<OK>.**
5.



If the SDriver is running, the icon  appears on the Windows taskbar in combination with the SBC Driver  (bottom right corner of the screen).

Depending on the number of channels, the telegrams to be communicated and possible startup log settings (see Log Settings), SDriver startup may take up to a few minutes. In this case, the Windows taskbar icon is animated if the startup process is not yet completed.

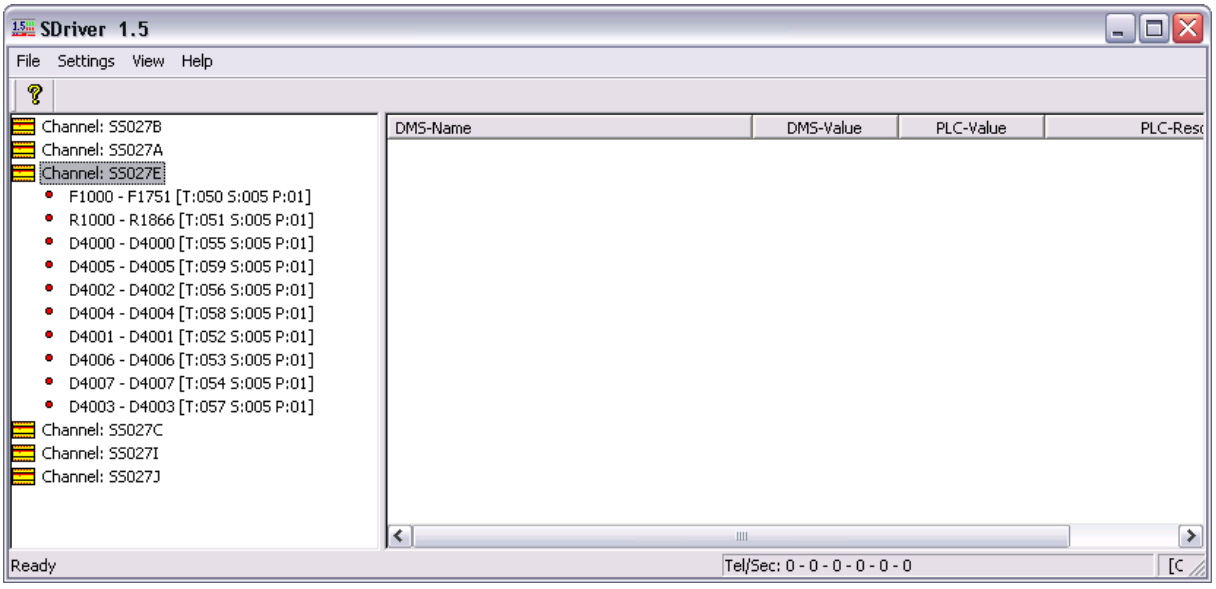
The module normally runs as a background process.

If the program SDriver is not running, no data is communicated to the connected PLC.

9.4.2 The SDriver User Interface

With the SDriver loaded, the user interface shown below can be opened as follows:

Double-click with the **left** mouse button on the SDriver icon in the Windows taskbar at the bottom edge of the screen.



*Alternatively, this can be done by clicking the **right** mouse button: **click once** on the SDriver icon and then select "**Display**".*

Title bar
Contains the name of the current module.

Menu bar
The menus are arranged according to basic functions and are explained in the following.

Toolbar
Displays icon buttons for some important commands and functions available in **SDriver**.

DMS name
Name of the data point to be communicated

DMS value
Shows the recalculated value.

PLC Value
Displays the effective PLC value. With binary signals, the two values match. With analogue signals the analogue value is shown deviating according to the recalculation factors set in the PET.

PLC Resource
Displays the PLC resource: type and address. See PET, Template objects:

Designation	Type
I	Input
O	Output
F	Flag
R	Register
T	Timer
C	Counter
D	Data
X	Text

Status bar
Shows the meaning of the buttons, the number of telegrams per second and the status of communications.

Output Field
Shows the communicated data points (DMS name) and their values.

Channel Info
In the left pane the available channels are displayed. Each icon represents a PLC. Double-clicking on an icon with the left mouse button displays all communicating PLC elements (telegrams).

PLC icons shown in light green represent an active driver (online) with all stations communicating.
PLC icons shown in dark/light green represent an active driver (online) with some stations not communicating.
White: PLC is inactive (offline).
Red: Communications error: Connection interrupted or a telegram cannot be communicated.
Yellow: A connection is being established with the PLC.
Dark blue: Connection to PLC is overloaded, communication is interrupted.

Double-clicking on a channel icon expands or collapses the telegram view.

Telegram Info
In the right pane, underneath a channel icon, its associated telegrams are displayed. Each icon represents a telegram.

The telegram icons for an inactive channel are shown in black (offline).
The telegram icons for an active channel are shown in green (online).
The telegram icons for an active channel which is not communicating are shown in red (error status).

These icons are crossed out when the associated station is deactivated. See Channel Stations.

Deactivated Station: The telegram icons for an inactive channel (offline).
Deactivated Station: The telegram icons for an active channel (online).
Deactivated Station: The telegram icons for an active online channel.

The resources are sorted according to type and telegram.
The telegram and station numbers as well as telegram priorities are shown in square brackets,
e.g.: [T:050 S:012 P:01]

In order to change the telegram priority, mark the desired telegram and press the **<ENTER>** key.
The following dialog window appears:

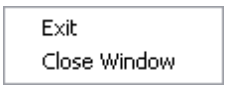
Any priority can be entered between 0 and 99, where 1 indicates the highest priority and 99 is the lowest priority.

Data that has the priority 0 is only read once when the driver is started (e.g. target values).

9.4.3 SDriver menus

9.4.3.1 File menu

The "**File**" menu contains the following functions:

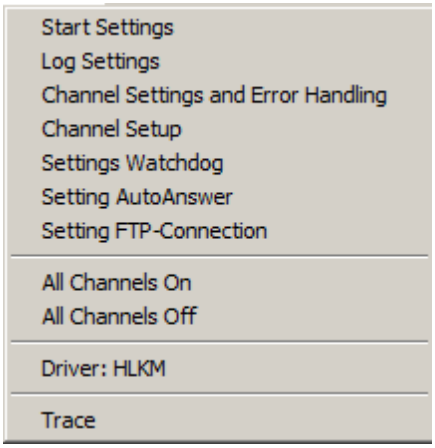


Exit
Closes the current window and removes the driver from the computer's main memory. The driver is then no longer active.

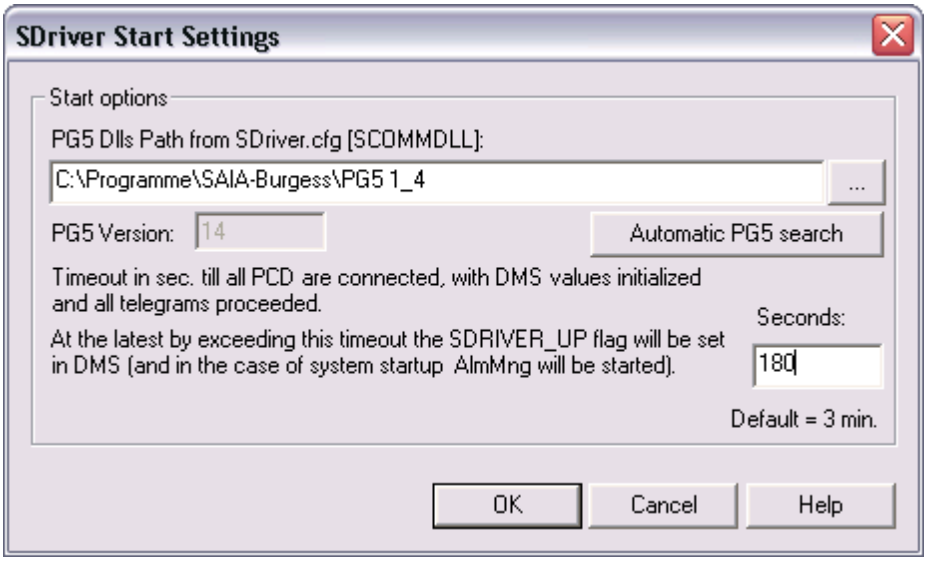
Close Window
Minimises the control window of the **SDriver** to the taskbar. The driver is not exited by doing so.

9.4.3.2 Settings menu

It is possible to select from various communication types (so-called channels) and settings in the "**Settings**" menu:



Startup Settings
Under Startup Settings, you can set which SCOMMDLL the SDriver should use. This depends on the PG5 version being used.



By default, a PG5 path need not be entered. In this case the SDriver searches the SCOMMDLL according to the following procedure:

1. First, a determination is made of which PG5 version is installed in the system (this registry entry must be set in the PET):

   ```
   "HKEY_CURRENT_USER\Software\Saia Visi.Plus 16_Promos16\PET\Settings" "PG5" =
   "20"
   ```
2. The "20" which is determined becomes "2.0" and then
   ```
   "HKEY_LOCAL_MACHINE\SOFTWARE\SBC\PG5\2.1\Global" "AppsDir" = "C:\Programs\SBC
   \PG5 V2.1\"
   ```
3. If no path can be determined, then Visi.Plus \BIN-path + \PG5 is used.

Log Settings
In the control window you can define which actions of the SDriver are logged.

Channel Setup

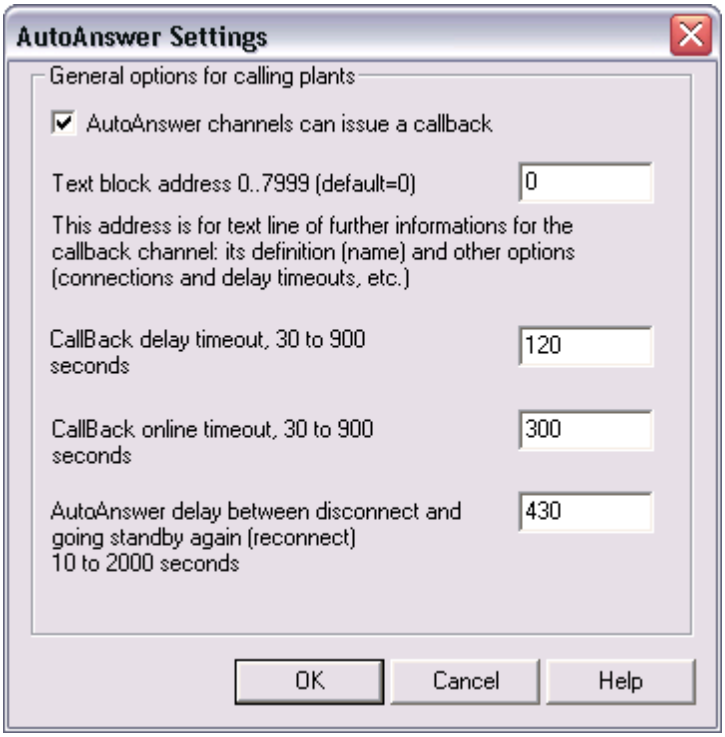
All settings in this dialog apply to all channels jointly.
The reactions of SDriver to errors (restart options, timeouts) and Number of Channels - Data Point Distribution can be set here.
The error log can also be viewed here.

Watchdog Settings

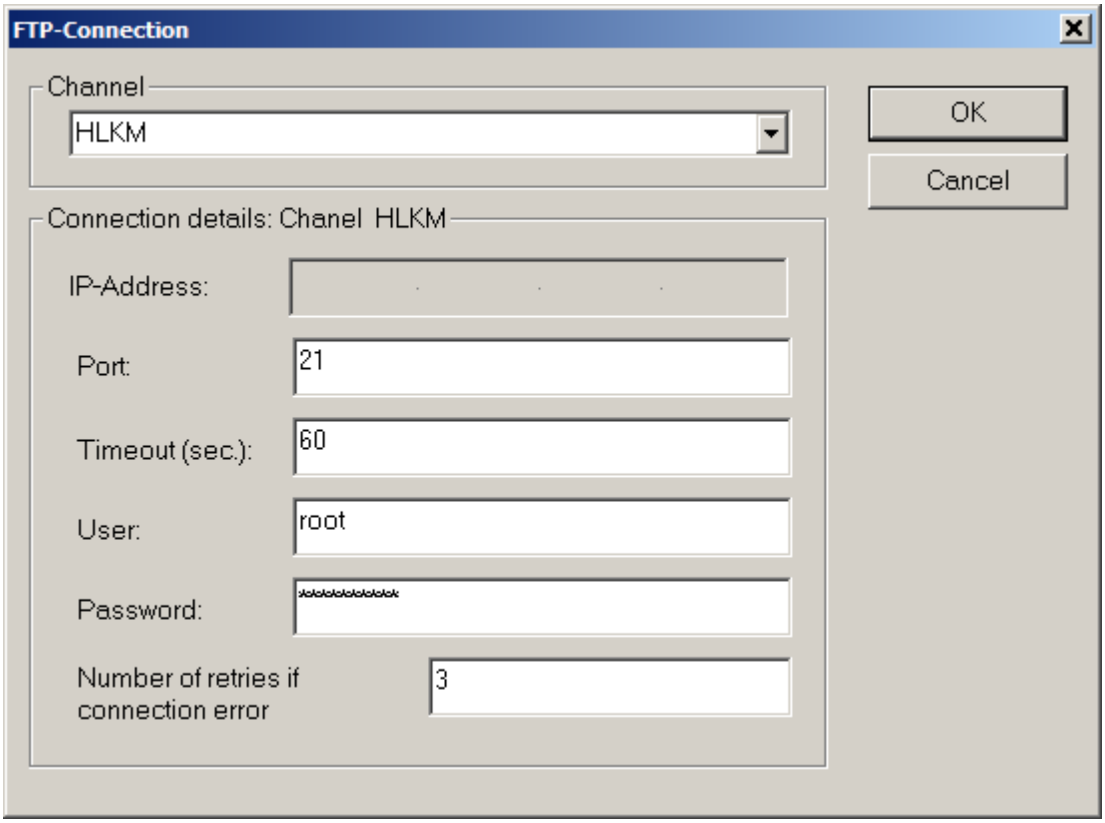
The SDriver can be monitored by the Data Management System (DMS).

AutoAnswer Settings

In this dialog window you can define how and whether the system has an AutoAnswer function in place for calls from systems.
See the chapter AutoAnswer.



FTP Connection Settings



All Channels On
All channels are switched online.

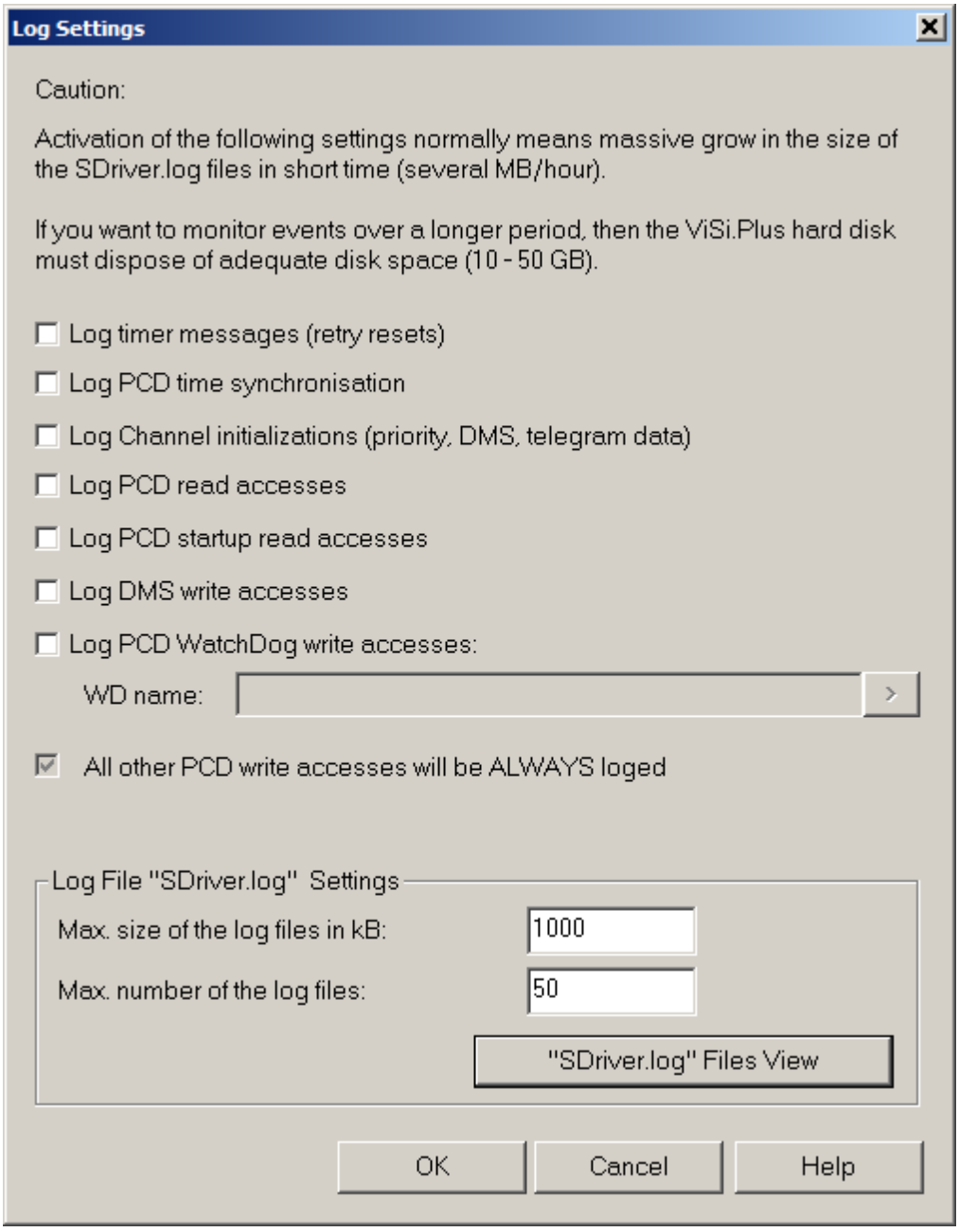
All Channels Off
All channels are switched offline.

Trace
When this menu item is selected, additional processes in the debug window of the program TraceWin.exe are output (error analysis).

9.4.3.2.1 Log Settings

In the control window you can define which actions of the SDriver are logged.
The SDriver produces multiple log files (see the chapter SDriver logs):

- SDriver.log — General log, Start/Exit of the SDriver, channel error reports, Faulty telegrams/channel definitions
- SDriver-XYZ.log — XYZ Channel-Log, Online/Offline behaviour, Error accesses and **all write accesses**



Log timer messages (retry resets)
Timeout management for read error accesses of the individual channels is logged in SDriver.log. These settings apply universally for all channels.

Log PLC time synchronisation
These settings apply universally for all channels. If the channel option "Synchronise PCD with PC system time" is selected, the time synchronisation is logged in the relevant SDriver-XYZ.log for this channel.

Log channel initialisations (priority, DMS, telegram data)
These settings apply universally for all channels. When selected, all information about the respective channel telegrams is logged in SDriver.log.

Log PLC read accesses

These settings apply universally for all channels. When selected, all PLC read accesses are logged in the relevant SDriver-XYZ.log.

We do not recommend leaving this option selected permanently or even over a long period of time, because the SDriver could be overloaded as a result of the SDriver-XYZ.log quickly becoming very large.

Log PLC startup read accesses
These settings apply universally for all channels. When selected, all PLC read accesses are logged in the relevant SDriver-XYZ.log, however only when the channel is opened.

Log DMS write accesses
These settings apply universally for all channels. When selected, all DMS write accesses are logged in the relevant SDriver-XYZ.log, i.e. a log entry is made each time a PLC value has changed.

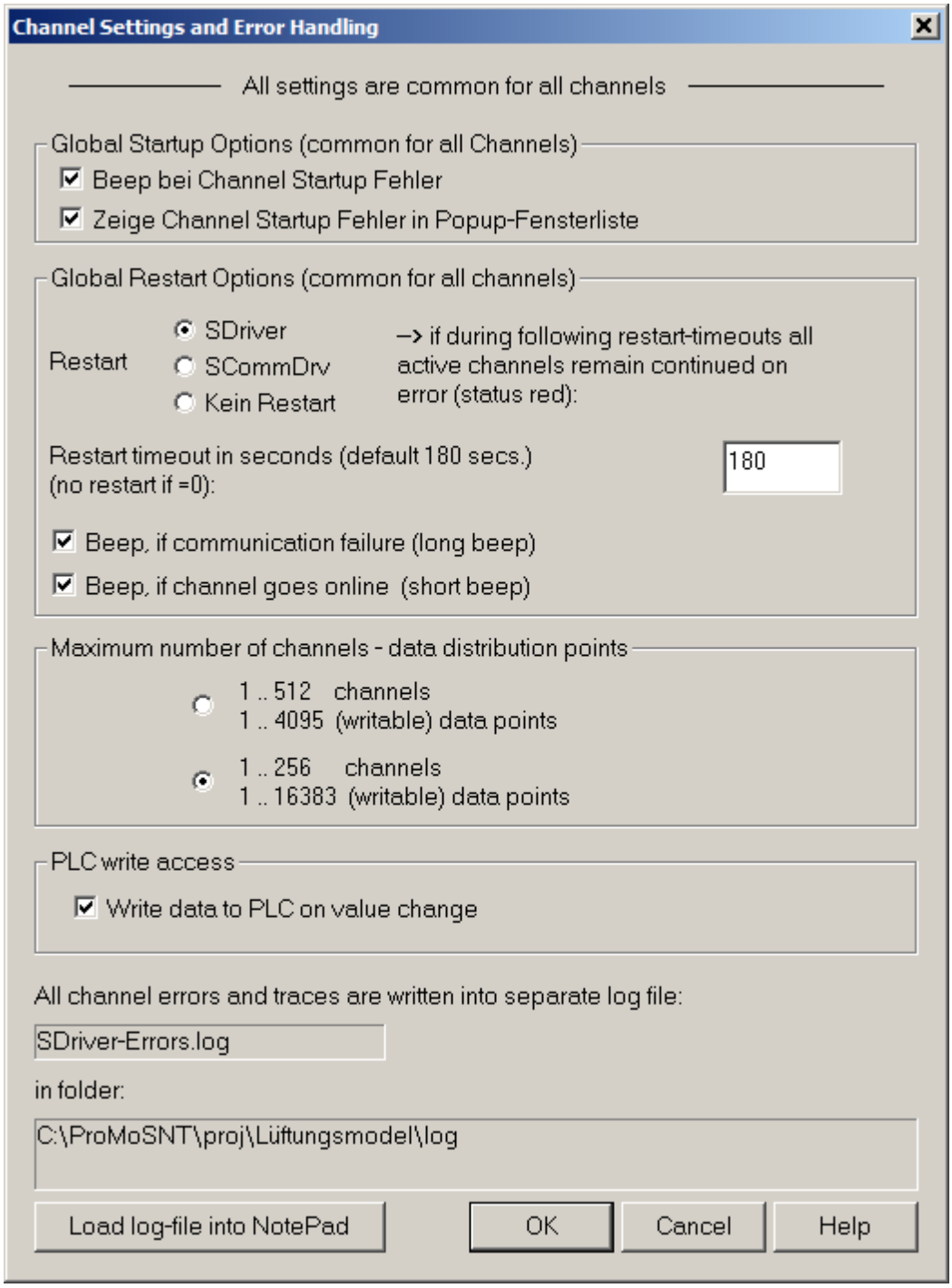
Log PLC watchdog write accesses
These settings apply universally for all channels. When selected, each change of the watchdog data point is logged in the relevant SDriver-XYZ.log, i.e. a log entry is made each time a respective PLC value has changed.
In addition, the standard watchdog signal to the DMS for SDriver monitoring is logged in SDriver.log.

WD Name
If this selection box is marked, the **Log PLC watchdog write accesses** option is activated. Here the watchdog data point can be selected in the DMS structure.

All other PLC write accesses are ALWAYS logged.
These settings are the default and cannot be changed (deactivated). All changes in value which are written to PLC are logged in the relevant SDriver-XYZ.log.

9.4.3.2.2 Channel settings and error handling

All settings in this dialog are **Global Options** (apply to all channels jointly).



Beep if channel is switched to online status (short beep)
The SDriver beeps (short) if the PLC channel goes online (connection established).

Beep in case of communications error (long beep)
The SDriver beeps (long) with a channel read or write access error.

Restart options: Restart is activated if all active channels exhibit uninterrupted communications (red status) errors during the configured **Restart wait time**.
The available options are:

- SDriver Restart (incl. SCommDrv) -> recommended setting.
- SCommDrv Restart.
- no restart options activated.

Restart wait time (in seconds, recommended value=3 mins)
The restart option is only activated if all active channels exhibit uninterrupted communications errors (red status) during this period.

Maximum number of channels - Data point distribution
The SDriver can be operated in two different modes:

- The number of channels in the SDriver is limited to **512**.
- Every channel can write a max. of **4096** data points (reading unlimited)

or (this is the default setting):

- The number of channels in the SDriver is limited to **256**.
- Every channel can write a max. of **16384** data points (reading unlimited)

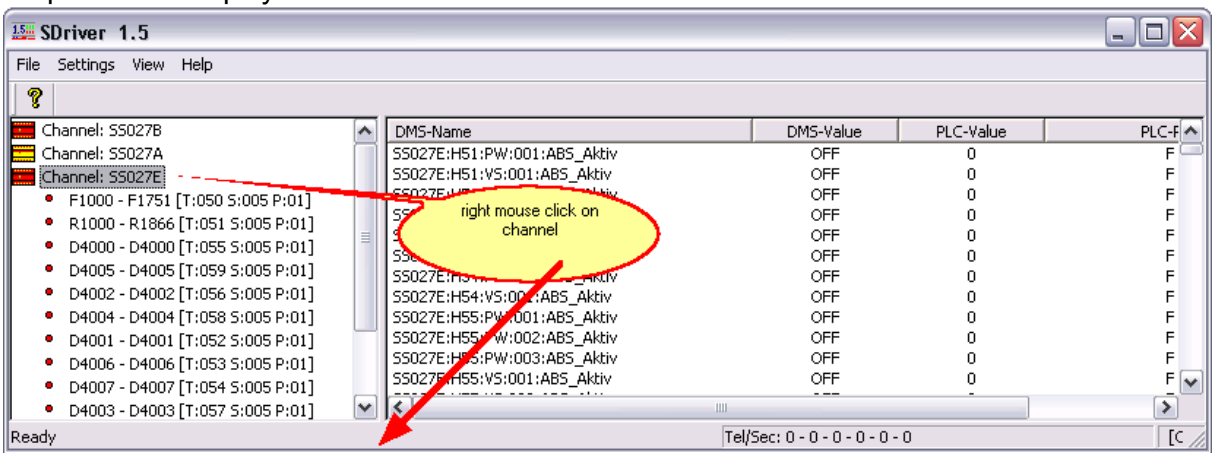
In both cases, every channel can manage a max. of **512** telegrams.

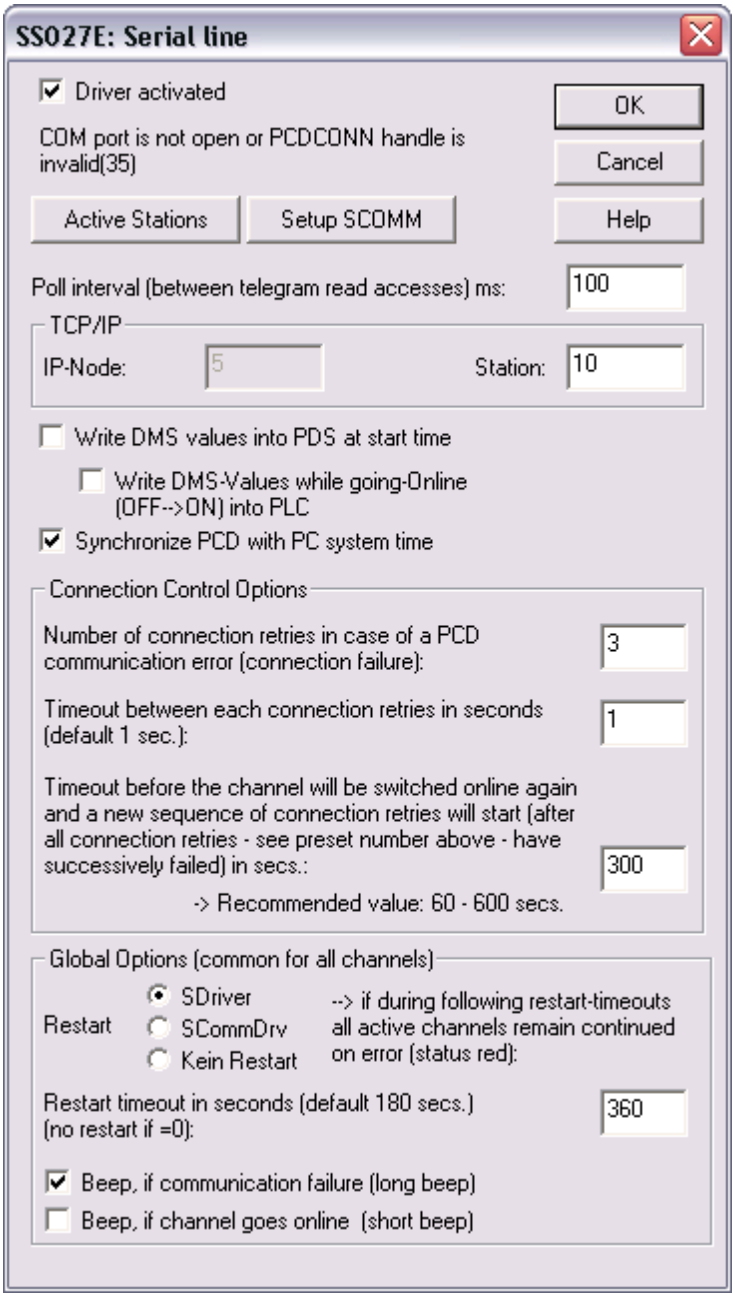
Load the log file in Notepad
All SDriver and channel errors are summarised in a separate log file "**SDriver-Errors.log**" in the log directory of the project. The SDriver-Errors.log file for the current project is displayed in Notepad.

9.4.3.2.3 Channel Settings (driver menus)

There is a menu entry for each channel: "**Driver: <Channel Name>**", which shows the corresponding "Channel Settings" dialog.

The same function can be activated by right clicking on the respective channel in the left part of the display window.





Driver Activated
Activates/deactivates the corresponding channel. This data point is also communicated to the DMS and can be used for visualisation of the channel status:
"**System:Driver:SDriver:SOCKET:Activated**" = "**ON/OFF**", cf. DMS system data points.

Status/Error Message
This can also be visualised with the data point "**System:Driver:SDriver:SOCKET:LastMsg**" = "**No response (6)**", cf. DMS system data points.

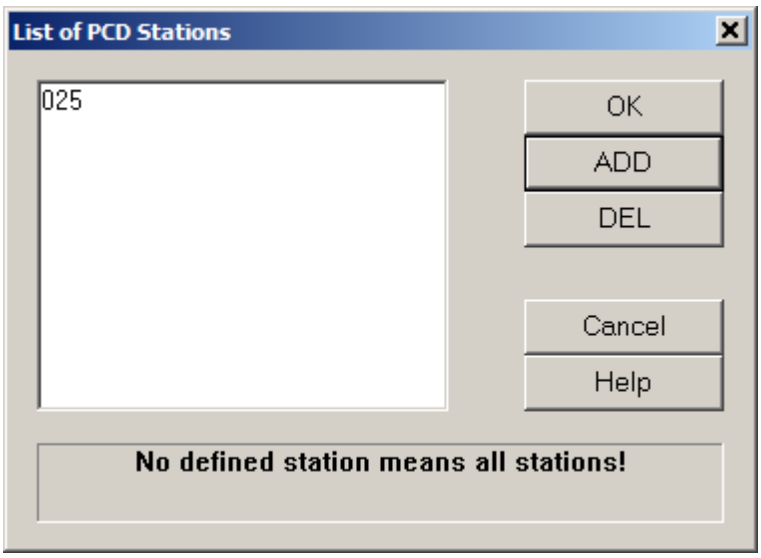
Channel start number
For many channels, the starting sequence can be selected, which makes it possible, for example, to connect the master stations first and then the sub-stations.

PCD backup

Automatically creates a backup of all data on the PCD (flags, registers, data blocks) which can be loaded to the controller again with the data transfer in the PG5.
Also see Channel PCD Backup.

Active Stations

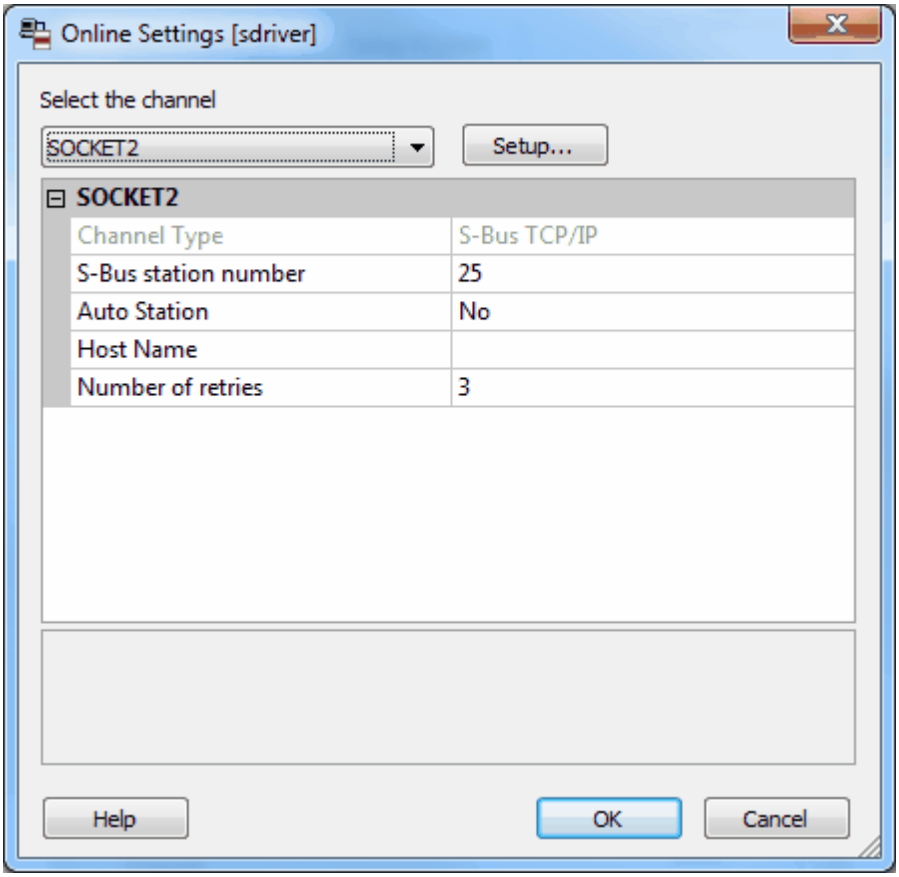
List of PLC stations:



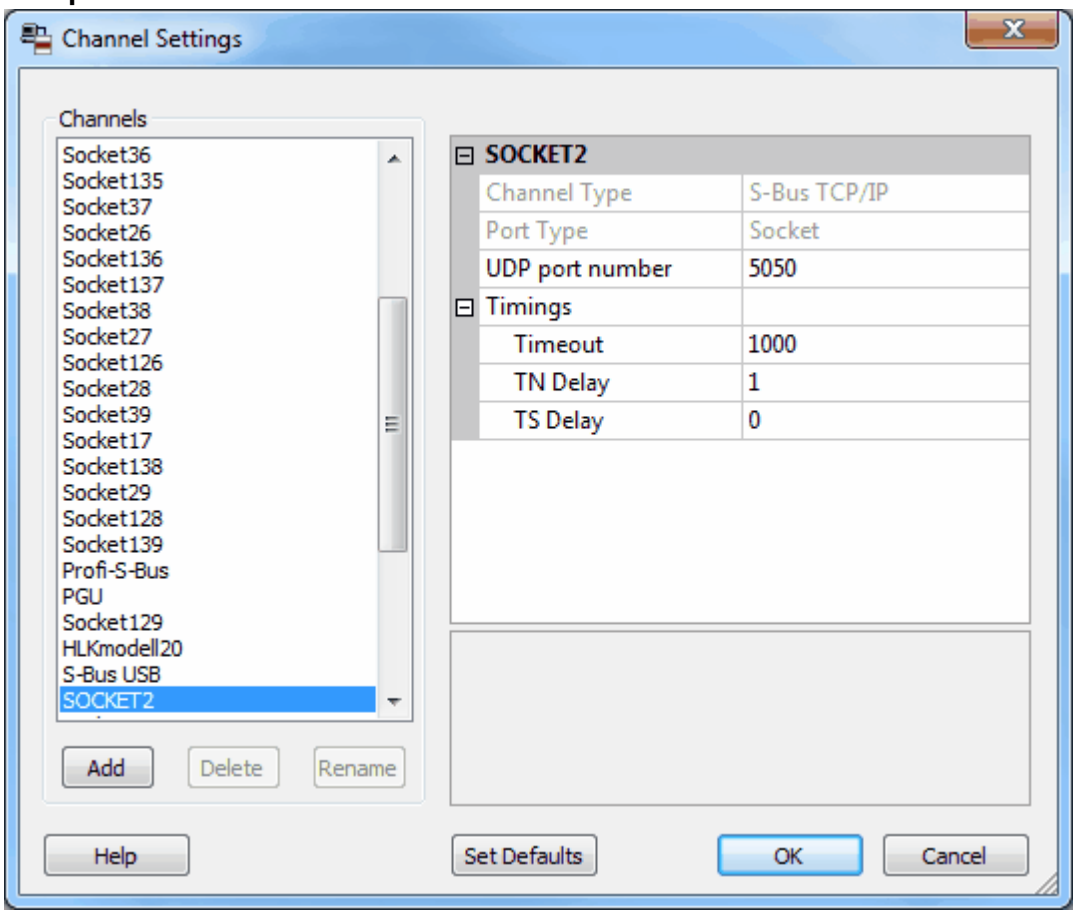
If no stations are entered, this means that all stations are active.

SCOMM Setup

Shows the settings dialog of the channel:



Setup...



Poll Interval (interval between telegram read accesses) in ms
Recommended value: 100ms.

TCP/IP: IP Node

TCP/IP: Station

Write DMS values to PLC on first PLC startup
All registered DMS values are written to the PLC on the first channel startup.

Write DMS values to PLC when going online \n(OFF–>ON changeover)
All registered DMS values are written to the PLC on each channel startup.

Synchronise PCD with PC System Time
The PCD time is synchronised with the PC system time.

Connection Control Options (- only applies to this channel!)
(also see the chapter Communications error handling)
With the following options, the behaviour of the SDriver is determined in the case of a connection error or interruption or for a telegram communications error.

Number of reconnection attempts...
In case of a communications error, the operation is repeated as often as indicated. If this number of attempts is reached, the channel is deactivated (offline status). After the set wait time has elapsed (see settings below), the channel is reactivated (online status).

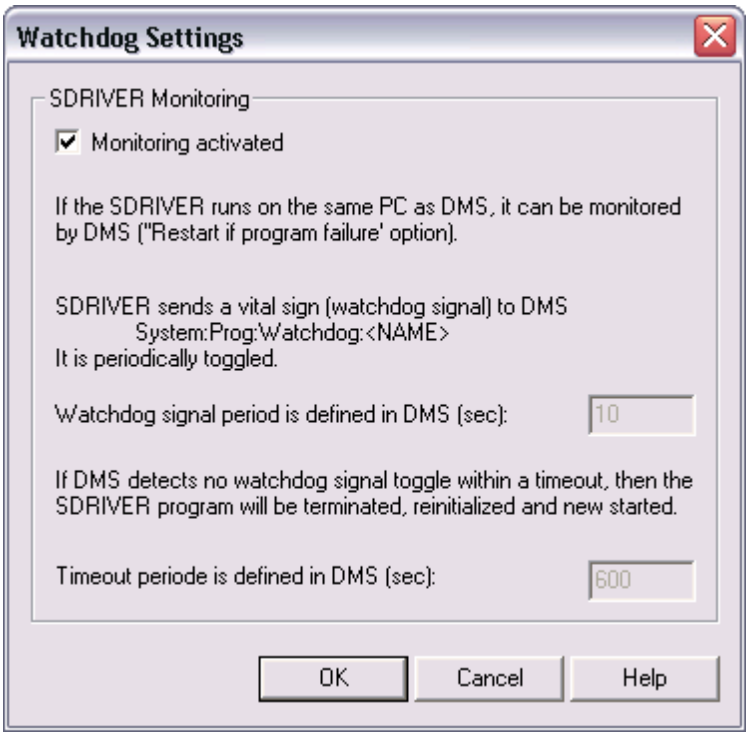
Wait time between individual connection attempts...
Wait time in seconds (default=1 second) until the failed read or write access is repeated.

Wait time until the channel is switched to online status again...
Wait time in seconds (default=300 seconds (5 mins)) until the channel is switched to online status again and a new cycle of connection attempts is started.

Registered data points
Displays number of current DMS data points for this channel which are registered with the DMS for write operations (DMS -> PLC).
This display is only for monitoring purposes. The number of registrations is limited and depends on the channel mode.
See the chapter Number of Channels, Telegrams and Data Points.

9.4.3.2.4 Watchdog Settings

The SDriver can be monitored by the Data Management System (DMS). The SDriver and the DMS exchange a live bit upon activation of the function.



9.4.3.3 View menu

In this menu the bars at the top and bottom edge of the SDriver window can be switched on and off.

Toolbar
Switches the toolbar on or off.

Status bar
Switches the status bar on or off.

9.4.3.4 Help menu

The "**?**" menu contains the Visi.Plus online help and current information about the **SDriver** version.

Help Topics
The Visi.Plus online help is accessed through the **"Help"** menu, via the corresponding **[Help]** button or by pressing the **<F1>** key.

About SDriver
Displays the current version of **SDriver**. The version must always be given in case of any support queries!

Should support queries become necessary, the internet forum is preferential to contact via telephone or e-mail. This offers two advantages: First, help is available more quickly; and secondly, other users also benefit from the entries.

9.4.3.5 Number of channels, telegrams and data points

The SDriver can be operated in two different modes:

- The number of channels in the SDriver is limited to **512**.
- Every channel can write a max. of **4096** data points (reading unlimited)

or (this is the default setting):

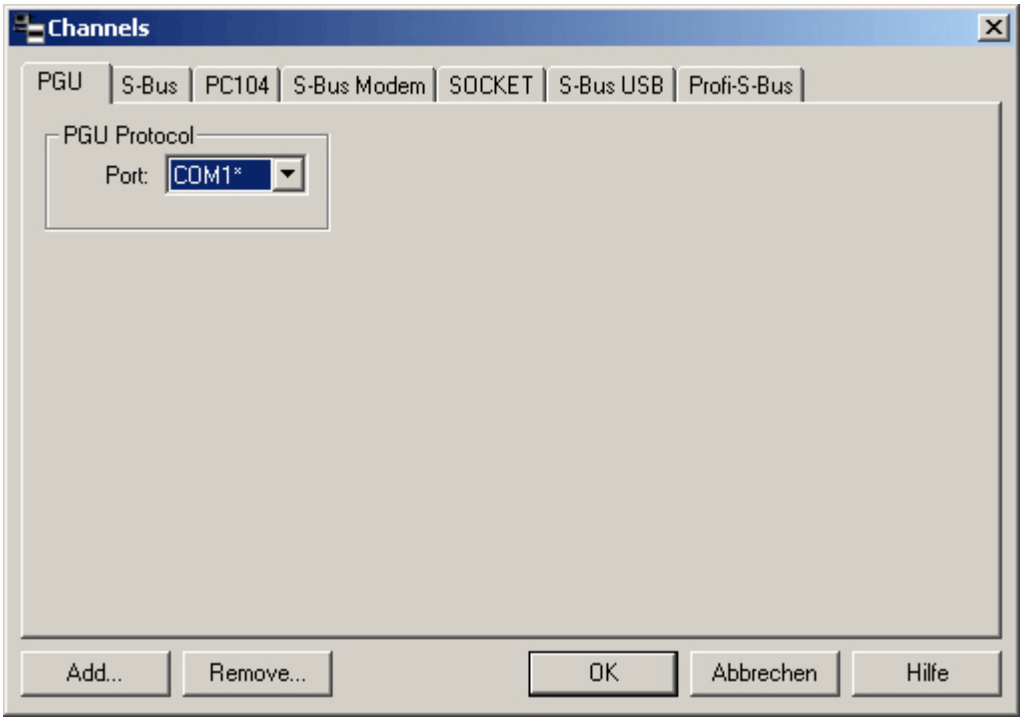
- The number of channels in the SDriver is limited to **256**.
- Every channel can write a max. of **16384** data points (reading unlimited)

In both cases, every channel can manage a max. of **512** telegrams.

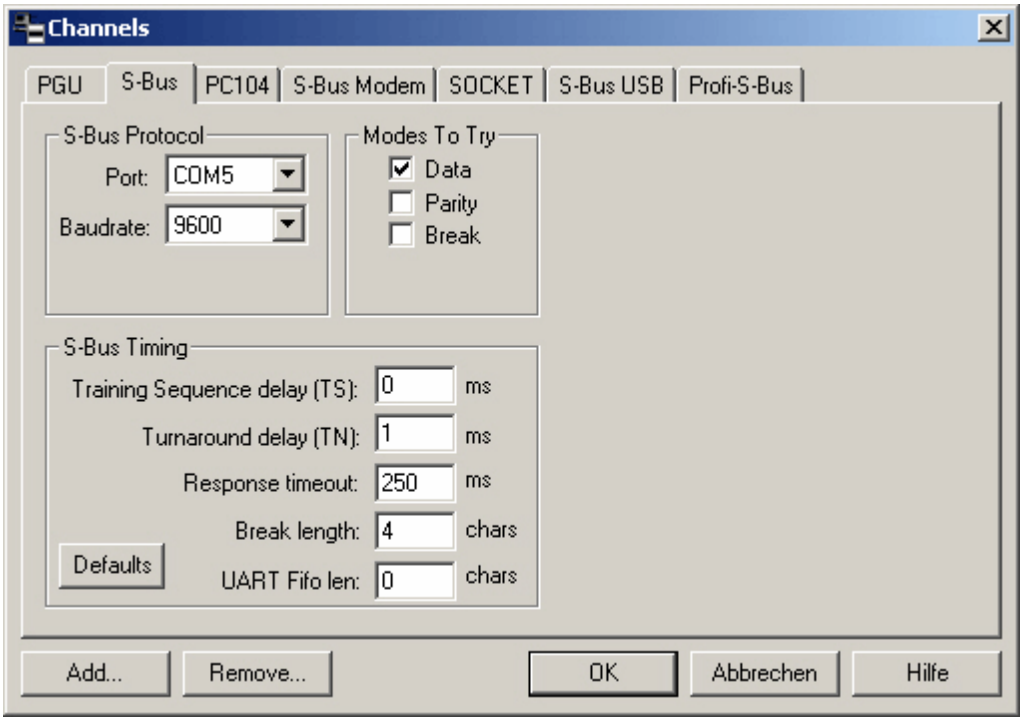
9.4.4 Possible types of communication

In the following chapter the types of communication supported by the Saia PCD® controllers and the SDriver and their attributes are explained.

PGU
PGU pertains to the communications settings for the PGU interface of Saia PCD®. It is only possible to select between the available COM interfaces. The baud rate and the protocol are automatically detected by the **SDriver**.



SBUS
SBUS contains the necessary data for the communications settings of the protocol.



SBUS Protocol
Settings for the desired COM interface and baud rate.

Modes To Try
Selects the SBUS communications type. Multiple modes can be selected.

Data mode

The SBUS telegram is packed into a data mode telegram and sent. No control signals are required for this mode. The control is based on characters specially reserved for this purpose.

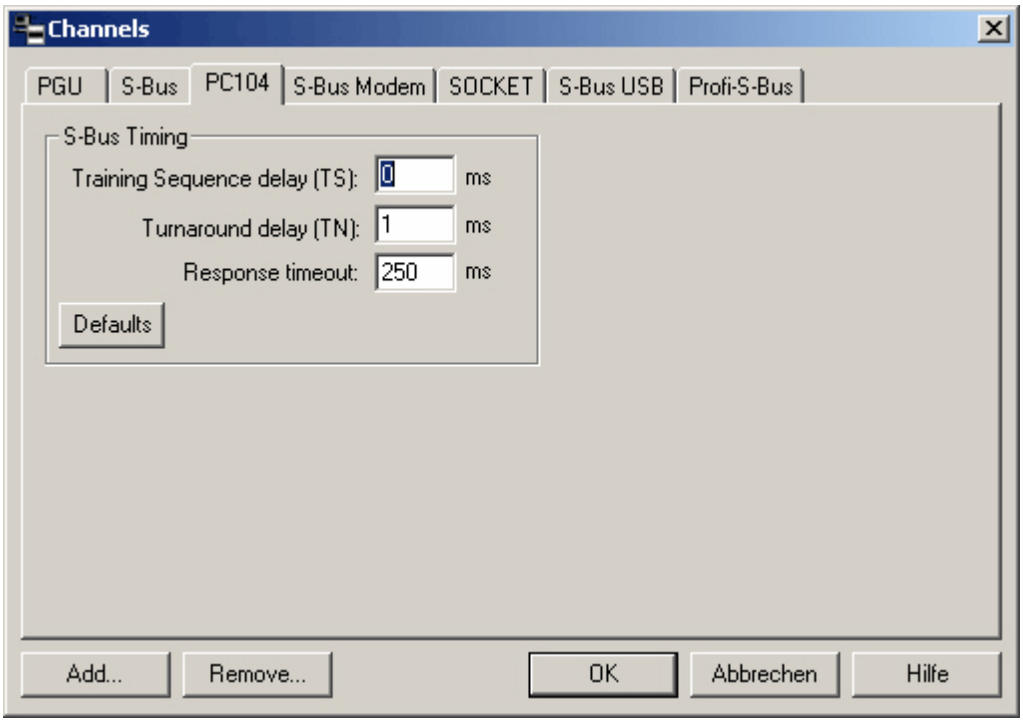
Parity mode
The start of a telegram is initiated by setting the parity bit.

Break mode
The start of a telegram is marked by the break signal.

SBUS Timing
The default values are normally used here. The **<Defaults>** button restores the default settings of the **SBUS Timing** parameters.

Changes to the **SBUS Timing** parameters should only be carried out after consulting the **SBUS Manual**.

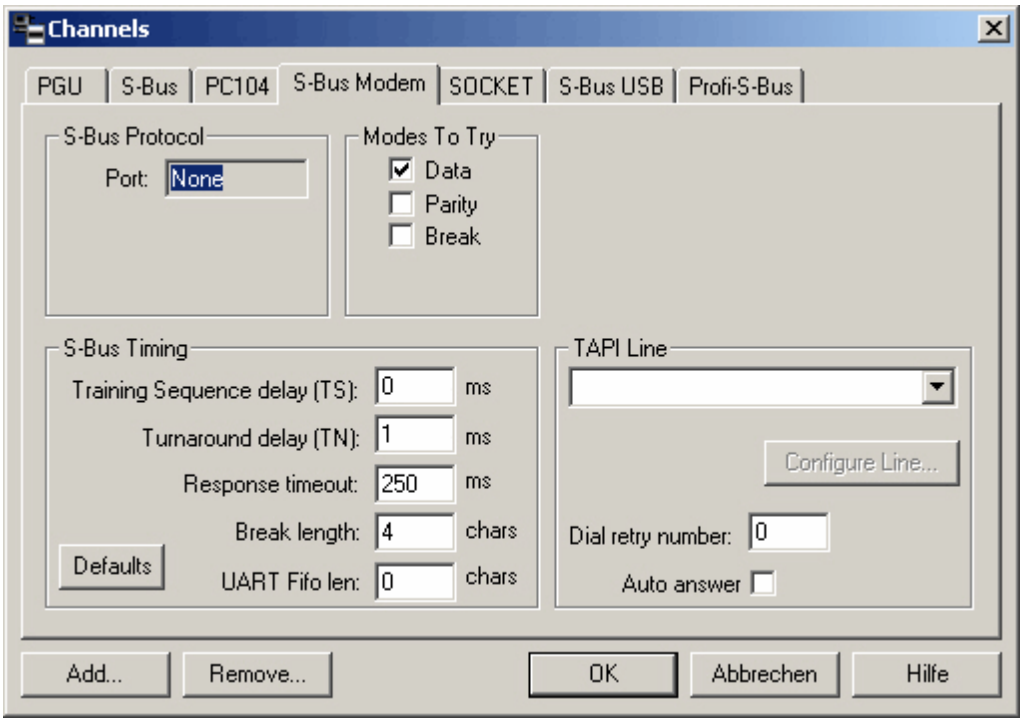
PC104
This register pertains to the communications settings between the PCD2.M250 special version and a plugged-in, conventional PC104 from a computer circuit board manufacturer.



SBUS Timing
The default values are normally used here. The **<Defaults>** button restores the default settings of the **SBUS Timing** parameters.

Changes to the **SBUS Timing** parameters should only be carried out after consulting the SBUS Manual.

SBUS Modem communications settings



SBUS Protocol: It is not possible to select a port

Modes To Try
Selects the SBUS communications type. Multiple modes can be selected.

Data mode:
The SBUS telegram is packed into a data mode telegram and sent. No control signals are required for this mode. The control is based on characters specially reserved for this purpose.

Parity mode:
The start of a telegram is initiated by setting the parity bit.

Break mode:
The start of a telegram is marked by the break signal.

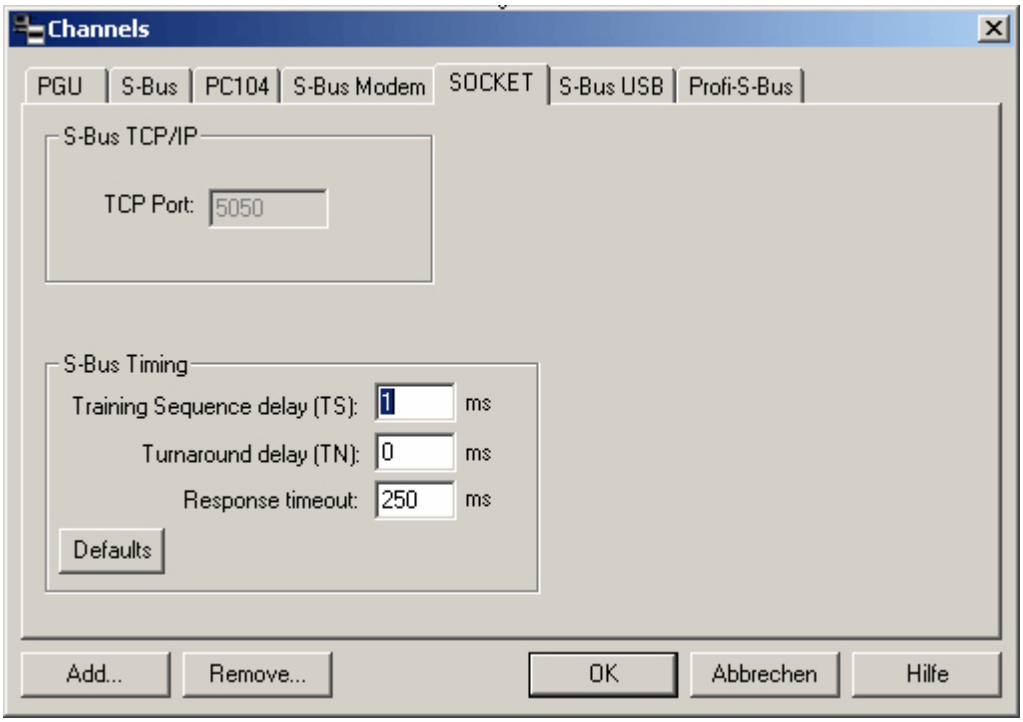
SBUS Timing:
The default values are normally used here.
The **<Defaults>** button restores the default settings of the **SBUS Timing** parameters.

Changes to the **SBUS Timing** parameters should only be carried out after consulting the **SBUS** Manual.

TAPI Line
The modem settings of the operating system should be selected from the dropdown menu. If none are available, they should be installed in accordance with the operating system.

SOCKET

Communications settings, mostly for ethernet.

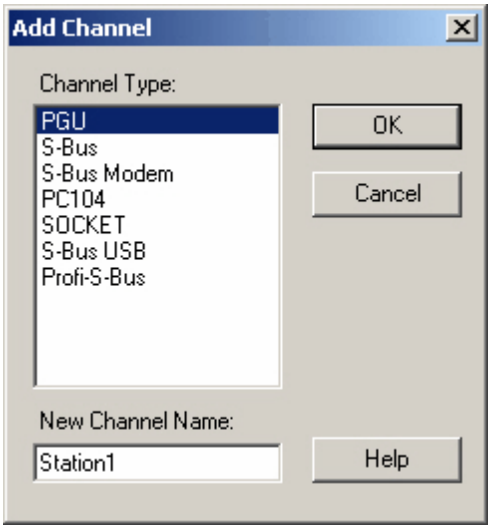


SBUS TCP/IP: Port number (default "**5050**")

SBUS Timing
The default values are normally used here. The **<Defaults>** button restores the default settings of the **SBUS Timing** parameters.

*No changes should be carried out to the **TCP Port** settings (default settings).*

<Add...> Add Channel
With this button a new communications channel can be created.



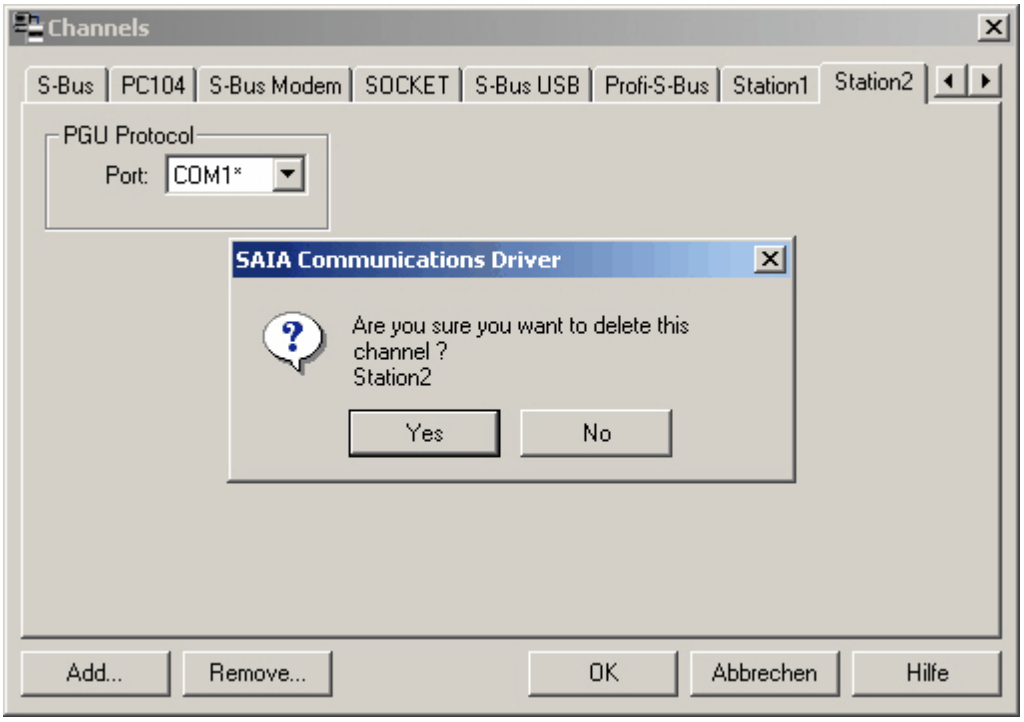
Channel Type

Shows already-defined channels as a template for a new channel which may, however, still be modified.

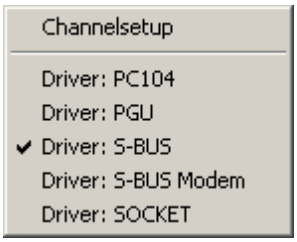
New Channel Name
The name of the new channel is entered here.
In the example this is "Station1".

<OK>: Confirms the entry.
The new channel is displayed in the **Channels** dialog window in a new tab called **Station1**, and can be modified accordingly.

<Remove...>: Remove Channel
Removes the selected tab (channel). In the following image the selected tab, **Station2** is removed.



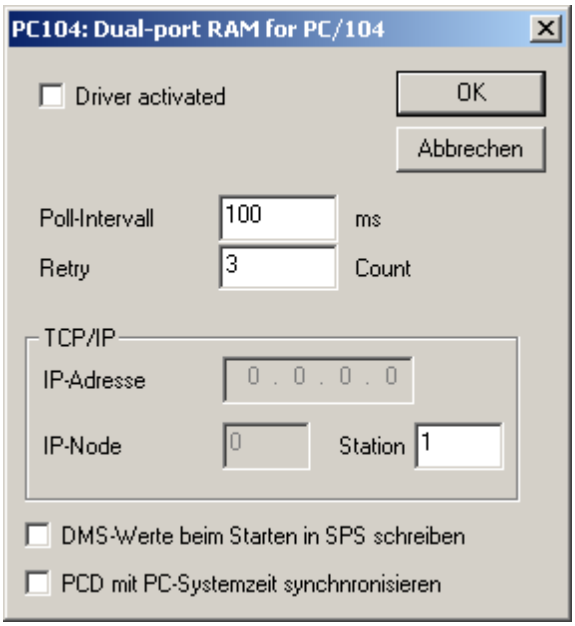
"**Settings**" menu (continued)



The control screens of the following communications drivers are all used the same way.

Driver: PC104
Driver: PGU
Driver: SBUS
Driver: SBUS Modem

The control screen for the PC104 driver is explained in the following:



Driver Activated
This checkbox starts the communications driver. A checkmark preceding the driver name in the **Settings** menu indicates that it is active.

Poll Interval
Defines the interval for the communication driver.

Retry
Number of repeat attempts for a telegram.

TCP/IP
It is only possible to indicate the PCD station number in the "**Station**" input field.

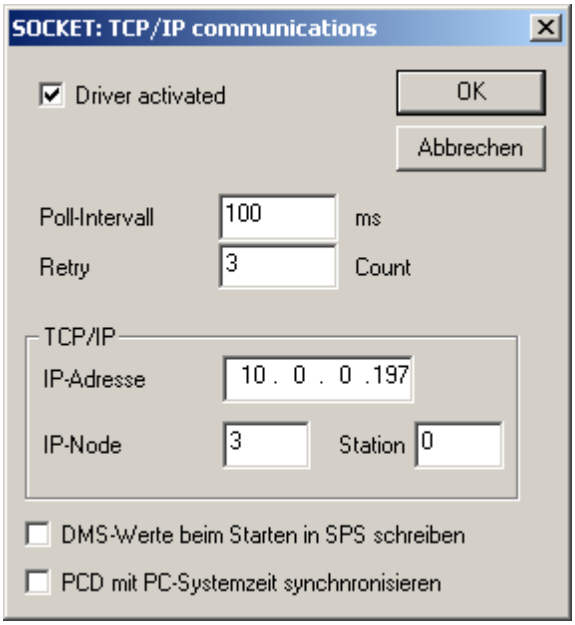
Write DMS Values to PLC on Startup
All values defined in the DMS are written to the PLC.

Synchronise PCD with PC System Time
Synchronises the PCD's internal time with the PC system time.

***Poll Interval** and the **Retry** number should not be changed, insofar as possible.*

*An image is also opened if the right mouse button is clicked on the PLCs "**PC104**", "**PGU**", "**SBUS**" or "**SBUS Modem**" in the SDriver user interface.*

Driver: SOCKET



The dialog is operated in the same way as the previously described drivers.

Differences:

TCP/IP
IP address of the PLC in the input field with the same name. The IP address **must** correspond to the settings in the PLC.

IP Node
This is the number used in the PLC program in order to communicate with another network station.

The IP address is not indicated in an SBUS telegram on an Ethernet network, only the IP node. The IP node is a reference to the IP address.

***Poll Interval** and the **Retry** number should not be changed, insofar as possible.*

When in the SDriver user interface, the image shown above also opens if the right mouse button is clicked on the PLC "Socket".

9.4.5 Communications error handling

The channel is either highlighted red ( ) or dark green ( ) for errors, in this case, there is only an error on one or more stations. See the section Channel info in the chapter *The SDriver user interface*.

The error message and the error code are written to the DMS system data points:

System:Driver:SDriver:xxx:LastErr	Error code of the last (error) message

System:Driver:SDriver:xxx:LastMsg	Last (error) message

Cyclical reconnection attempts
As soon as a telegram cannot be communicated, the connection to the respective PLC is interrupted (disconnect). If the option "Number of reconnection attempts..." > 0, then the connection is reestablished either immediately or after the set wait time ("**Wait time between individual connection attempts...**"), and if successful, the telegram communication is repeated.

With a renewed communication error, the entire cycle (disconnection, reestablishment of connection and renewed attempt) is repeated the number of times that set ("**Number of reconnection attempts...**").

If this cycle ends without success, the channel is switched to offline, a set timeout ("**Wait time until the channel is switched to online status again...**") is waited and then an attempt is made to switch the channel to online again.

This mechanism is always active.

Error analysis
There are basically two types of errors:

- Connection error: the PLC cannot be connected. The **PG5 Online Debugger** can be a valuable troubleshooting tool.
- Telegram error: incorrect settings, e.g. address 0. Here a telegram test with the PET can be helpful.

Optimal settings
If multiple channels are configured in the project, we recommend not setting the wait time between the channel off and online cycles under at least 1 minute (default: 5 minutes). Otherwise there is a significant risk that the SCommDriver (the driver between SDriver and the PLC) will block all other channel communications for too long of a time during the connection calls.

9.4.6 Suspended communication

This functionality is only available in the SDriver as of version 1.6.

The following DMS DP "**System:Driver:SDriver:Suspend**" allows SDriver communication to be suspended for a maximum of 300 seconds.
This timeout of 300 seconds can be set in \drv\sdriver.cfg: [Settings] SuspendTimeout=300

Example:
In the PET, "System:Driver:SDriver:Suspend" is set at the beginning of the code generation. SDriver then suspends all communication. This significantly reduces the CPU/DMS load. Once code generation has been completed, PET sets "System:Driver:SDriver:Suspend" =0, and SDriver resumes communication immediately afterward.

9.4.7 SDriver logs

See also Log Settings.

All log files are located by default in the log directory of the project.

The SDriver maintains multiple log files (see SDriver Log Settings):

- SDriver.log — General log, Start/Exit of the SDriver, channel error reports, faulty telegrams/channel definitions

In addition, each channel has its own log:

- SDriver-XYZ.log — XYZ Channel-Log, Online/Offline behaviour, Error accesses and **all write accesses**

All SDriver and channel errors are summarised in a separate log file:

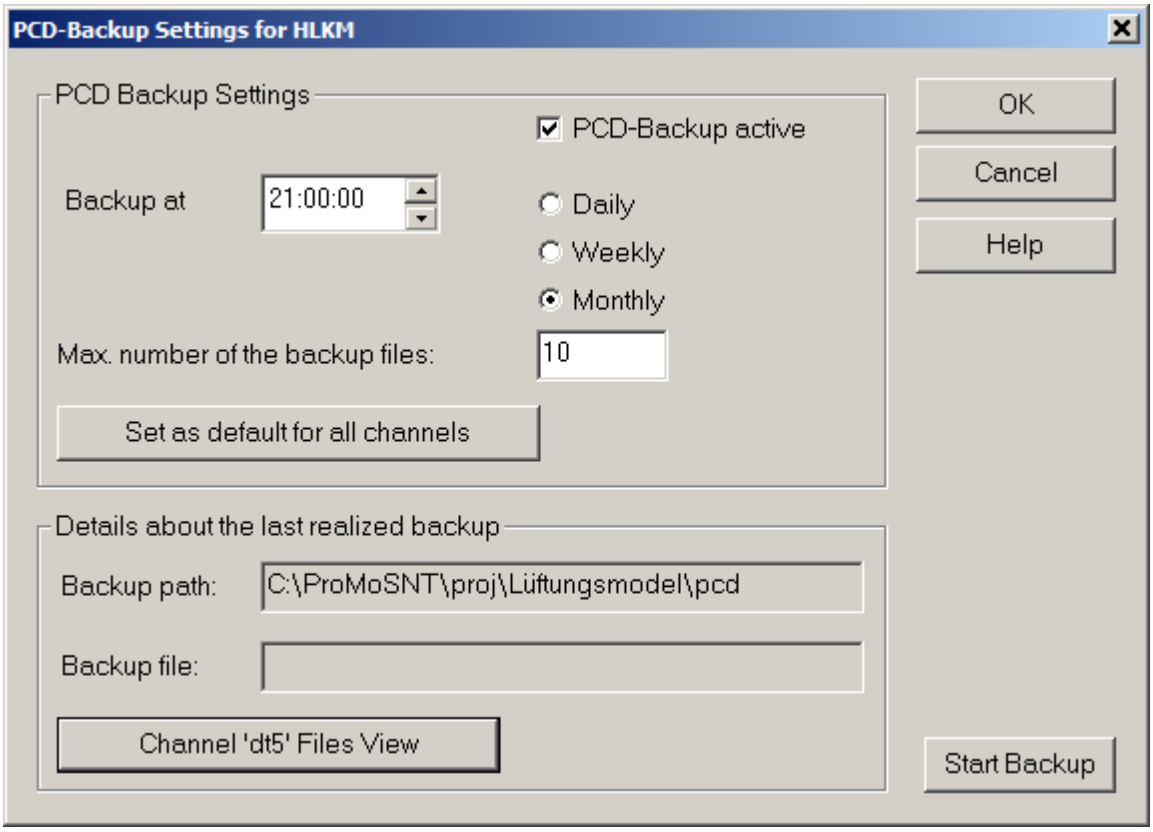
- SDriver-Errors.log

For troubleshooting, you should analyse SDriver-Errors.log, the SDriver.log and the relevant channel.

9.4.8 Channel PCD backup

This functionality is only available in the SDriver as of version 1.6.
In this version, only those controllers which are directly connected to the channel are processed;
controllers operating via the gateway function of the PCD (e.g. Ethernet to RS485) cannot be backed up.

The Data Transfer program from PG5 must be used to load the files back to the PCD.



Backup at
Select the time for the automatic backup. It is recommended not to use the same time here as for the PDBS and other backups.

PCD backup active
Switches the backup function on and off.

Options
Interval settings
Number of backup files which remain saved. After this, the oldest backups are deleted automatically.

9.4.9 PLC error messages

Designation	Code	Error message
PCD_OK	0	Communications OK
PCD_NAK	1	NAK response, command not accepted
PCD_BCCERR	3	Invalid BCC or CRC, no other errors
PCD_INVALID	4	Invalid response
PCD_P8CONN	5	Bad connection between P800 and PCD6
PCD_NORESP	6	No response
PCD_NORESP2	7	No response after PcdRestart in PcdWrAllocation
PCD_NODCD	8	No Data Carrier Detect
PCD_INUSE	9	Port in use in another mode or baud rate

PCD_BADPARM	10	Bad parameter
PCD_VERIFERR	11	PCD memory write verify error
PCD_OUTOFMEM	12	IBM PC out of memory
PCD_WRONGCPU	13	Physically connected to wrong CPU on PCD6
PCD_NOCPU	14	CPU not present
PCD_TOFLASH	15	PCD to flash failed on M170
PCD_DIFF	16	Files & CPU don't compare, see <f PcdCompare>
PCD_CANTOPEN	17	Can't open file
PCD_WRERROR	18	Error writing to file
PCD_RDERROR	19	Error reading file
PCD_BADFILE	20	Invalid PCD file
PCD_CSERROR	21	PCD file checksum error
PCD_BADHDR	22	Invalid header in PCD
PCD_CODESIZE	23	Not enough space in code segment
PCD_TEXTSIZE	24	Not enough space in text segment
PCD_EXTSIZE	25	Extension memory segment too small
PCD_NOTINIT	26	Header not initialised
PCD_EPROM	27	Can't write to EPROM memory
PCD_CANCEL	28	Operation cancelled by user
PCD_NOTDEF	29	Block not defined
PCD_RUNNING	30	CPU must be in Stop or Halt
PCD_NOTPROG	31	CPU not programmed
PCD_DRVOVER	32	No response, driver overloaded
PCD_NOCHGEXT	34	Can't change size of extension memory
PCD_NOTOPEN	35	COM port is not open or PCDCONN handle is invalid
PCD_TXERROR	36	Write to PCD failed
PCD_COMBUSY	37	Port request timeout
PCD_LOCKED	40	Fatal internal error
PCD_BUG	44	
PCD_NOTSUP	45	Operation not supported by PCD F/W version
PCD_ALRDYDEF	46	Station number already defined
PCD_EXTHDR	47	Not enough space for extended header
PCD_BADPWRD	48	Invalid password
PCD_EXTHDRCS	49	Extended header checksum error
PCD_WRONGSTN	50	Download to wrong S-Bus station ($STN)
PCD_WRONGFW	51	Download to PCD with wrong firmware ($PCDVER)
PCD_NOPARITY	52	S-Bus Parity mode is not supported for this COM port
PCD_NORTSCTS	54	RTS/CTS mode is not supported for this COM port
PCD_SQLERR	56	(BUES) Error in Sql Statement
PCD_INTDATA	63	Internal data invalid
PCD_EXCEPTION	64	C++ Exception in DLL
PCD_ASKPWRD	65	Ask for password
PCD_NOTIMEOUT	67	Timeout is not supported for this COM port
PCD_NODRIVER	68	Driver not available
PCD_BUES_E	69	Received a BUES CMD_E response
PCD_BUES_A	70	Received a BUES CMD_A response
PCD_FDL_SAP	71	Cannot Activate SAP

DRV_RETRY	100	(INTERNAL) Retry
DRV_IOPENDING	101	(INTERNAL) Input/Output pending
DRV_CALLBACK	102	(INTERNAL) Callback message from the driver
DRV_NOIMP	103	(INTERNAL) Not implemented function
DRV_PASSWORD	104	(INTERNAL) Driver ask for password
DRV_DISCONNECT	105	(INTERNAL) Port disconnected
DRV_ENQUERY	106	(INTERNAL) Acknowledge in P800 ACK-ENQ
DRV_SEQNUM	107	(INTERNAL) Udp sequence number fault

9.4.10 Status of channels and stations in the DMS

The extended status of all channels is defined in the DMS:

"**System:Driver:SDriver:<Channelname>:Status**"
- **0** = off
- **1** = on
- **4** = off->on
- **5** = error
- **6** = Callback on
- **7** = Callback off->on
- **8** = Callback error
- **9** = AutoAnswer on
- **10** = AutoAnswer off->on
- **11** = AutoAnswer error
- **12** = on with StationError
- **13** = AutoAnswer on with StationError
- **14** = Callback on with StationError

The status of all channels is defined in the DMS:

"**System:Driver:SDriver:<Channelname>:Station<No>:Status**"
- **0** = off (error)
- **1** = on (OK)

9.4.11 Monitoring of the SDriver

If not present, then the SDriver builds up in DMS following two alarm data points:

DMS:System:Driver:SDriver:<Chanelname>:ALM

and

DMS:System:Driver:SDriver:<Chanelname>:<StationNr>:Status:ALM

Example for channel name "SCAC":

Following values can be configured in ..\drv\\sdriver.cfg in section [**CHANNEL_ALM**] :

```
      Group=998                          ; this is default for SDriver / PCDDriver
      ChannelFormat=#c / #N / #VLastMsg / #Z(Kommt:Geht:Quit) #u          ; this is
for System:Driver:SDriver:<Chanelname>:ALM:Alarm1:Format
      StationFormat=#c / #N / #V^LastMsg / #Z(Kommt:Geht:Quit) #u         ; this is
for System:Driver:SDriver:<Chanelname>:<StationNr>:Status:ALM:Alarm1:Format
      Pri=3
      Text=         ; default is empty string
```

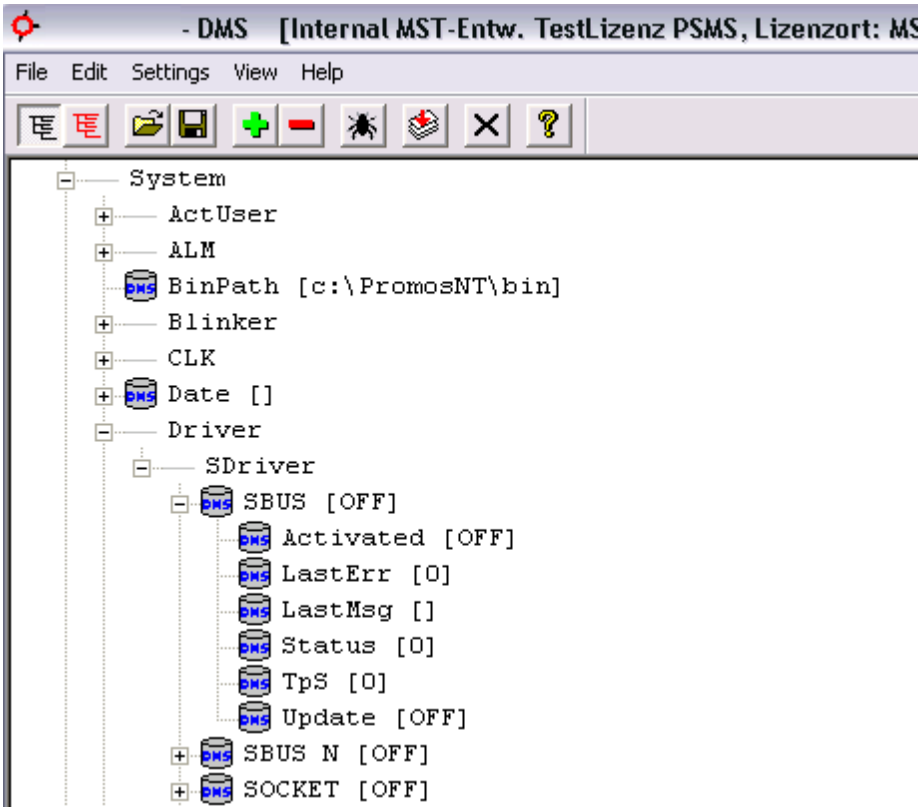
Other values are "hardwired" and hava following fixed defaults:

```
      Activ=1
      DBGRP=0
      Prot=ALARM
      Type=<=
      Value=0             ; DMS type = FLT
```

9.4.12 SDriver <-> PLC communication via modem

In order to be able to access a PLC with the SDriver via modem, the following adjustments must be made in the DMS:

Under "**System:Driver:SDriver:SBUS Modem:Activated**" (BIT), a 1 must be set (the data point must be inserted manually). Under "**System:Driver:SDriver:SBUS Modem:PhoneNumber**" (STR), the telephone number must be entered. This data point must also be defined manually. Only the DATA_MODE is supported (no BREAK mode).

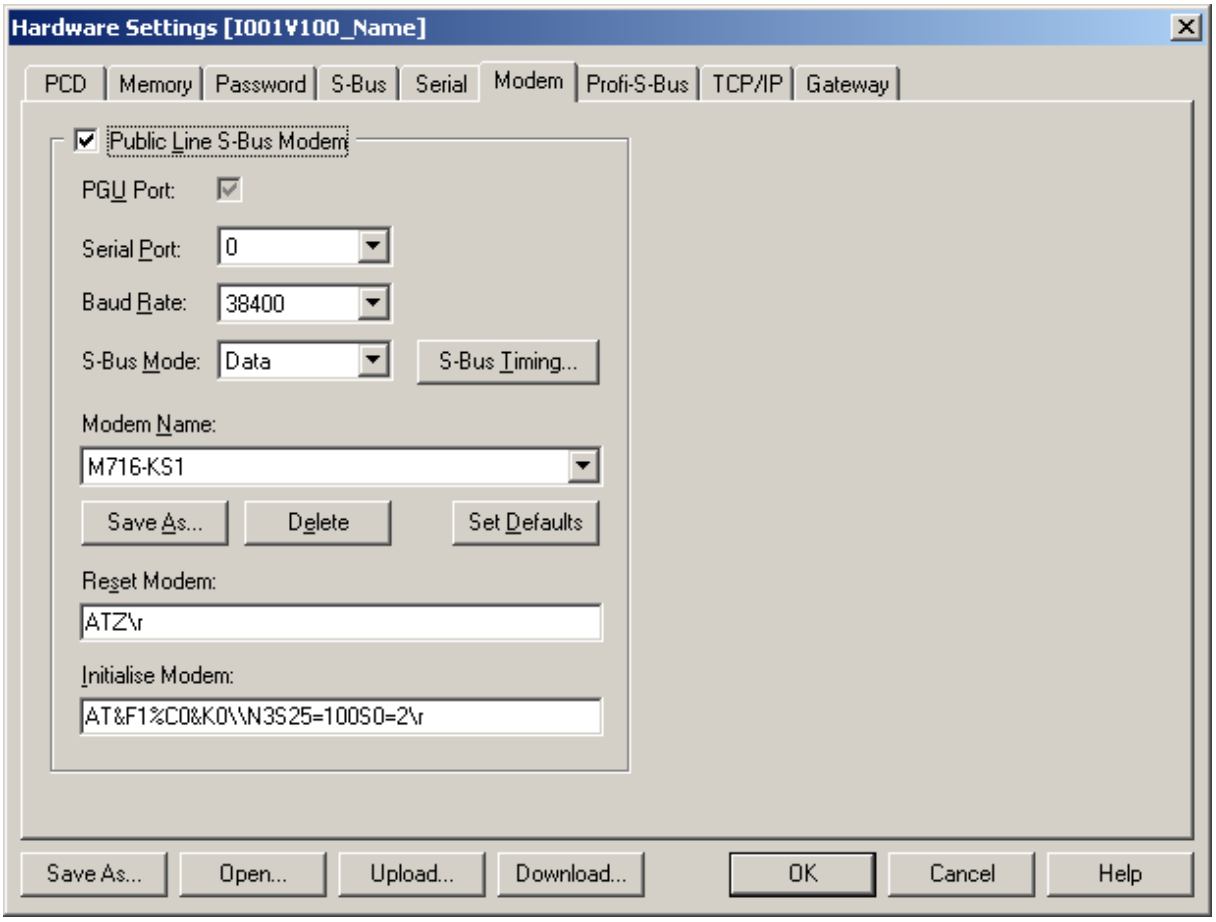


9.4.13 AutoAnswer

"AutoAnswer mode" enables incoming calls to be received directly by a PCD in Visi.Plus. An intermediate PCD is no longer necessary.

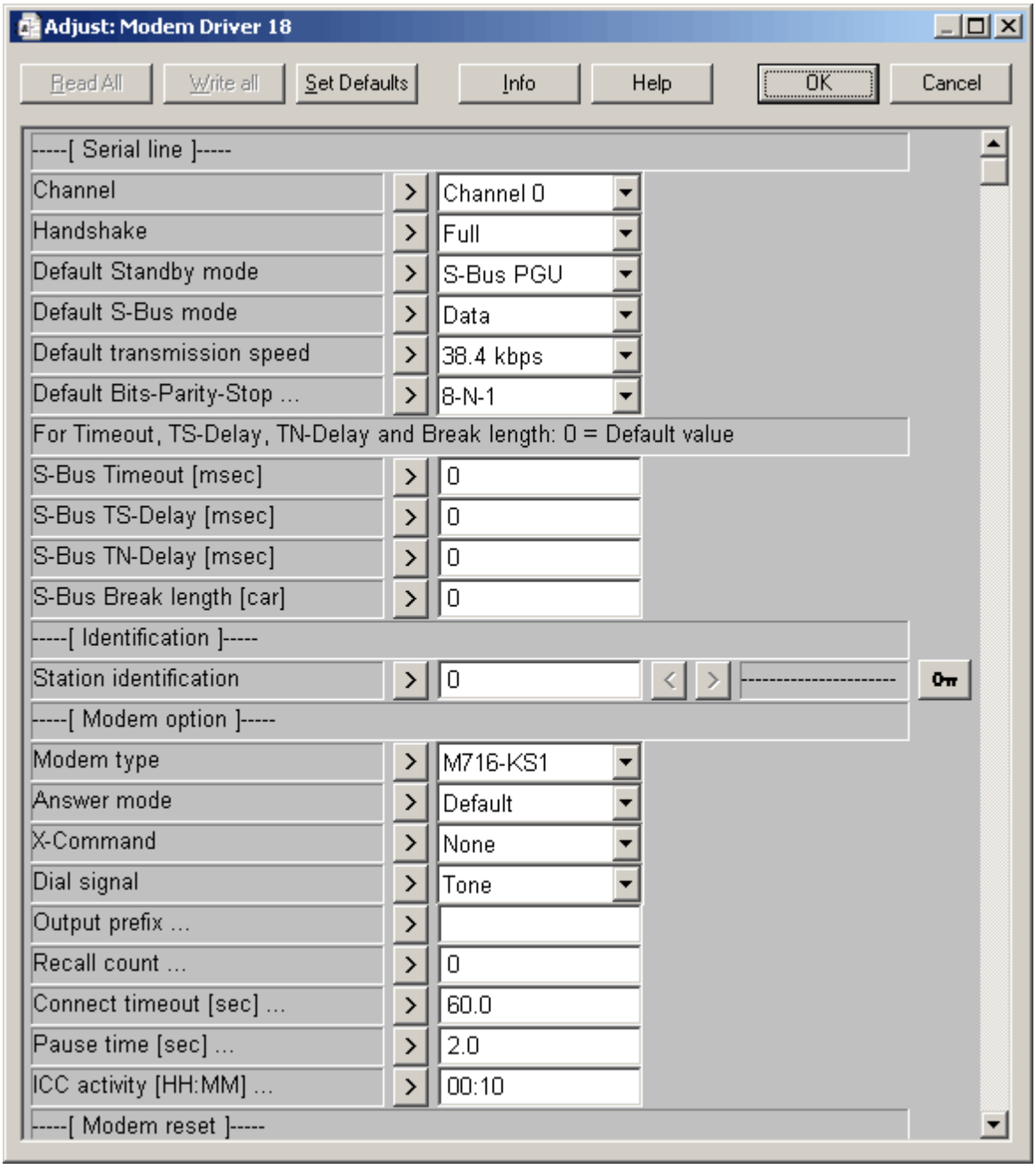
9.4.13.1 Controller Hardware Settings

Make the following settings in the PG5 Project Manager under Settings/Hardware/Modem:



9.4.13.2 Using FUPLA

The two FBoxes "**Modem 18**" and "**Call PGU**" are used in Fupla.
The settings in "Modem 18" are as follows:

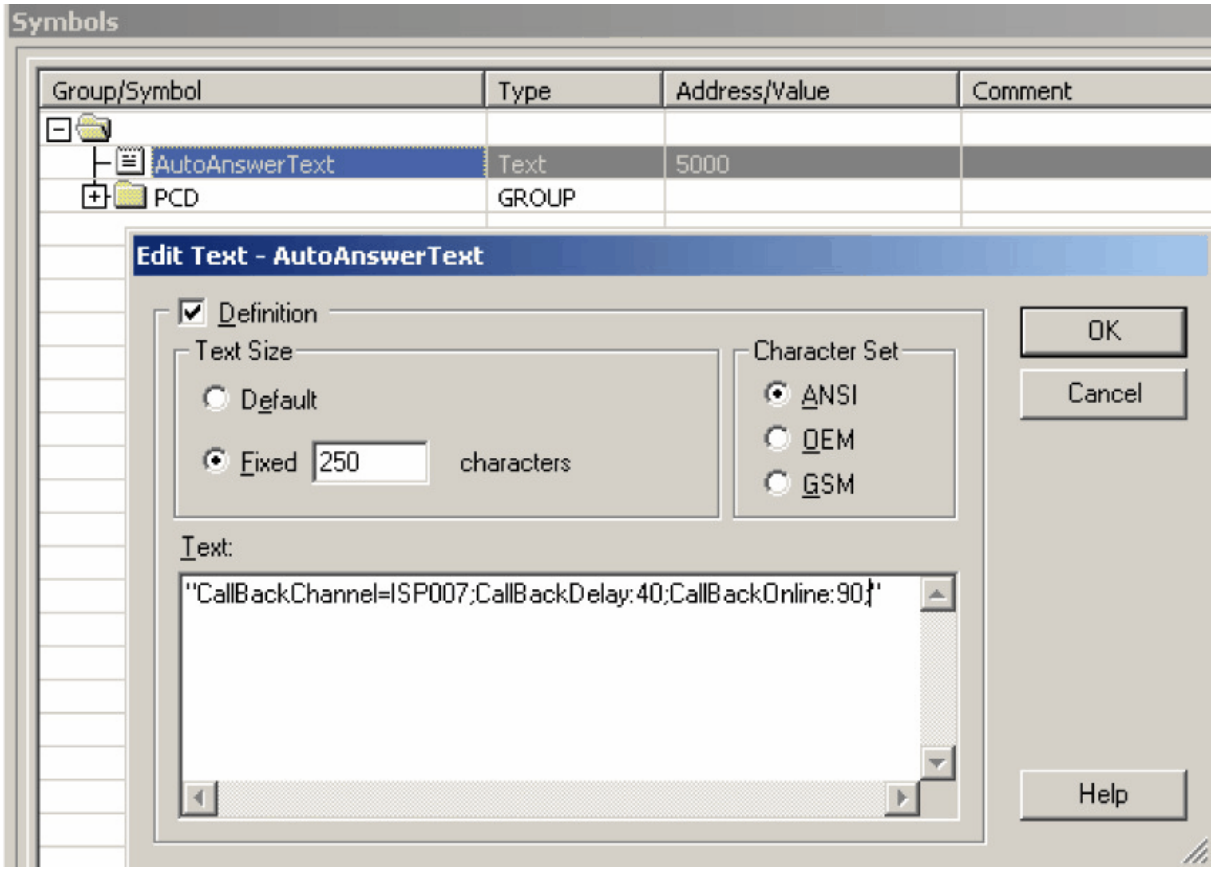


The parameters in the module "**Call PGU**" are default values; only the telephone number to call should be added.
The call is then initiated in Visi.Plus through the FBox "**Call PGU**".

9.4.13.3 Information for Visi.Plus

In order to inform Visi.Plus which action should be triggered with a dial-in, a text should be defined. In this case it makes no difference whether it is global or local, because the text is only saved in the PCD and is read by Visi.Plus.

The text number must be the same in all PCDs attached to the AutoAnswer channel! It is not possible to use different text numbers!



The text must at least contain the information "**CallBackChannel=xxxx;**", where "**xxxx**" corresponds to the name of the channel in Visi.Plus (SDriver) that should be called back. In this example the **Channel ISP007** should be called back.

The parameter "**CallBackDelay=40**" indicates how many seconds a callback should take place after the incoming call. This time span should enable the PCD to reinitialise the modem and wait on an incoming call (callback of Visi.Plus).

The parameter "**CallBackOnline=90**" indicates how many seconds a callback is active; this is the online duration after a connection is established.

9.4.13.4 Requirements from Visi.Plus

AutoAnswer can be implemented with a single modem. This means that the same modem receives incoming calls and is also used for callbacks. This works, but can lead to longer "down times" when a so-called callback (callback to the PCD) fails (poor or busy telephone line, etc.).

Therefore we recommend working with 2 modems: one modem is used for the incoming calls and the other is used for the callback.

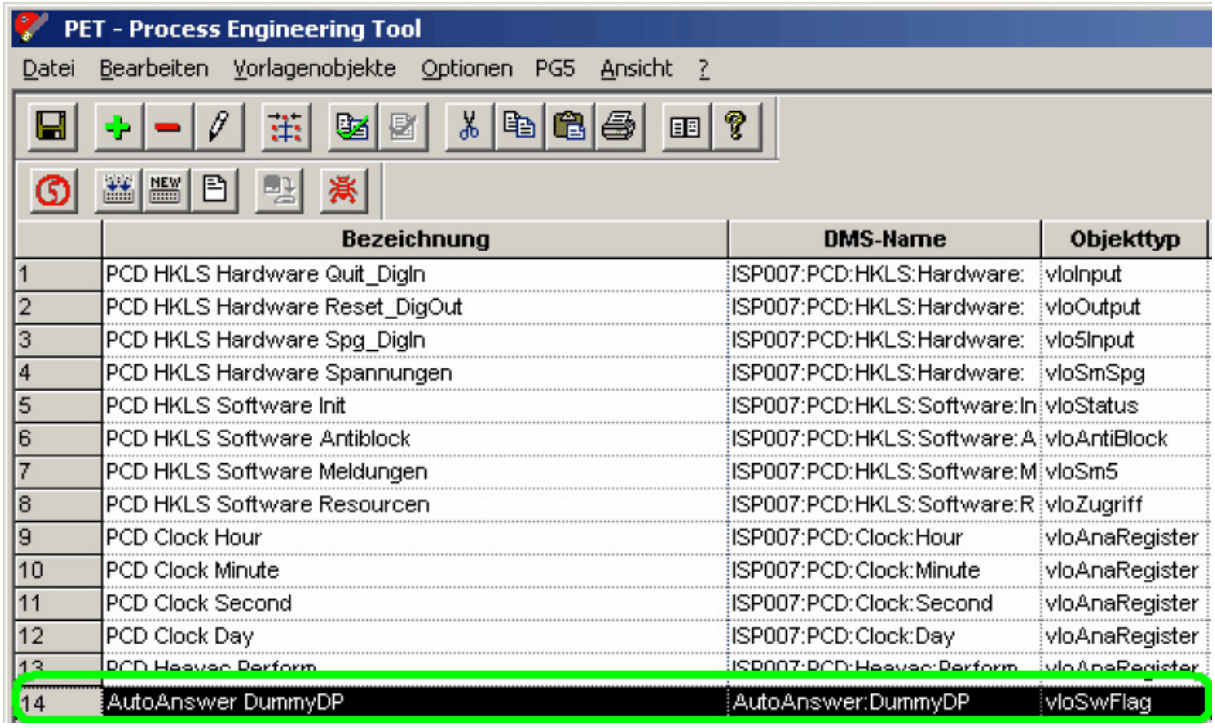
9.4.13.5 Definitions in the PET

The AutoAnswer channel in the SDriver is handled mostly just like a normal communications channel. That means to receive an AutoAnswer channel, a data point must be defined. In so doing, please note the following:

All PCDs connected with AutoAnswer must have the same SBus station number. Differing SBus station numbers are not permitted!

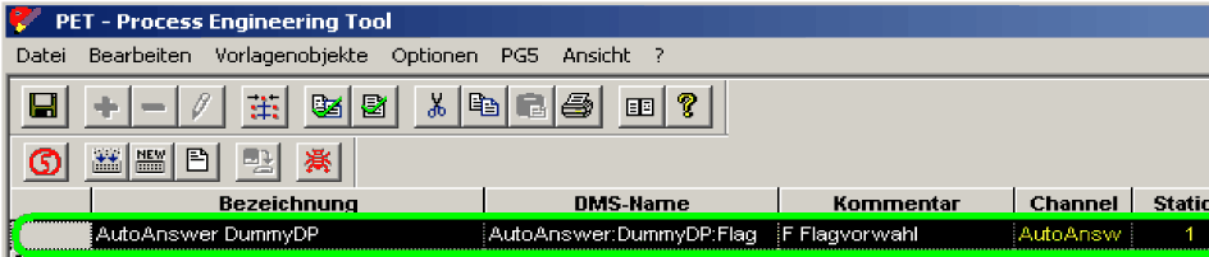
The minimum configuration is for a single flag/register. Here a system object of the type vloSwFlag was defined, containing a single flag.

Be sure not to use any specific information in the AutoAnswer channel, because you will never know for an incoming call which property you are connected to! The AutoAnswer channel is only used for receiving an incoming call and to initiate other actions (callback to the respective station).



	Bezeichnung	DMS-Name	Objekttyp
1	PCD HKLS Hardware Quit_DigIn	ISP007:PCD:HKLS:Hardware:	vloInput
2	PCD HKLS Hardware Reset_DigOut	ISP007:PCD:HKLS:Hardware:	vloOutput
3	PCD HKLS Hardware Spg_DigIn	ISP007:PCD:HKLS:Hardware:	vlo5Input
4	PCD HKLS Hardware Spannungen	ISP007:PCD:HKLS:Hardware:	vloSmSpg
5	PCD HKLS Software Init	ISP007:PCD:HKLS:Software:In	vloStatus
6	PCD HKLS Software Antiblock	ISP007:PCD:HKLS:Software:A	vloAntiBlock
7	PCD HKLS Software Meldungen	ISP007:PCD:HKLS:Software:M	vloSm5
8	PCD HKLS Software Resourcen	ISP007:PCD:HKLS:Software:R	vloZugriff
9	PCD Clock Hour	ISP007:PCD:Clock:Hour	vloAnaRegister
10	PCD Clock Minute	ISP007:PCD:Clock:Minute	vloAnaRegister
11	PCD Clock Second	ISP007:PCD:Clock:Second	vloAnaRegister
12	PCD Clock Day	ISP007:PCD:Clock:Day	vloAnaRegister
13	PCD Heavac Perform	ISP007:PCD:Heavac:Perform	vloAnaRegister
14	AutoAnswer DummyDP	AutoAnswer:DummyDP	vloSwFlag

The "**Digital Signals**" view:

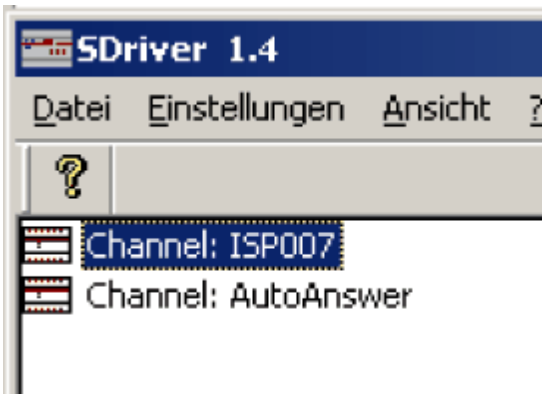


Bezeichnung	DMS-Name	Kommentar	Channel	Static
AutoAnswer DummyDP	AutoAnswer:DummyDP:Flag	F Flagvorwahl	AutoAnsw	1

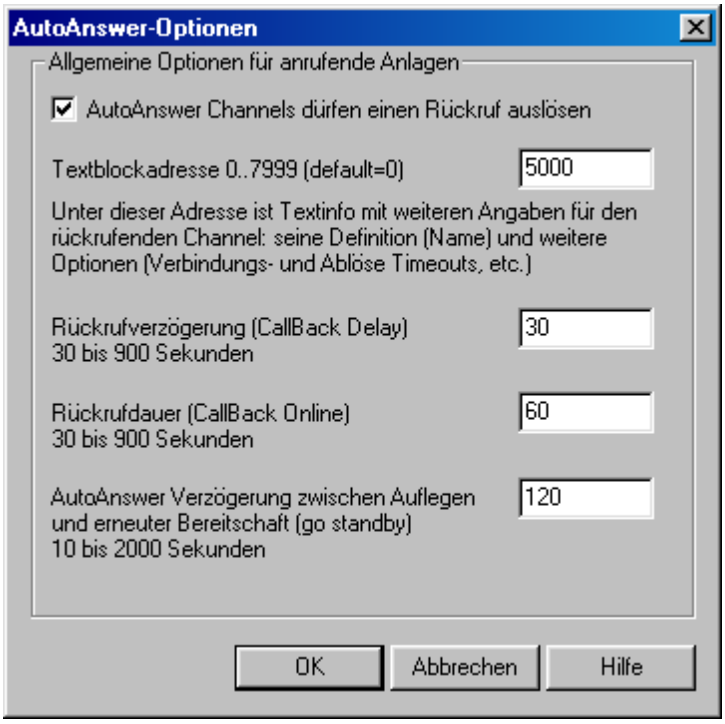
We recommend 999 as a telegram number.

9.4.13.6 Definitions in the SDriver

The SDriver assumes an AutoAnswer function. At least 2 channels should be displayed, one on which the PCD is normally called back (the actual channel) and a straight AutoAnswer channel:



Under Settings/AutoAnswer Settings you can indicate default values:



The checkbox "**AutoAnswer channels may trigger a callback**" should always be selected. The ability to use the incoming telephone call has not yet been implemented, so a callback is always used.

Under "**Text Block Address**" the text number must be indicated (the same for all PCDs!) which contains the information for the AutoAnswer function (see Information for Visi.Plus)

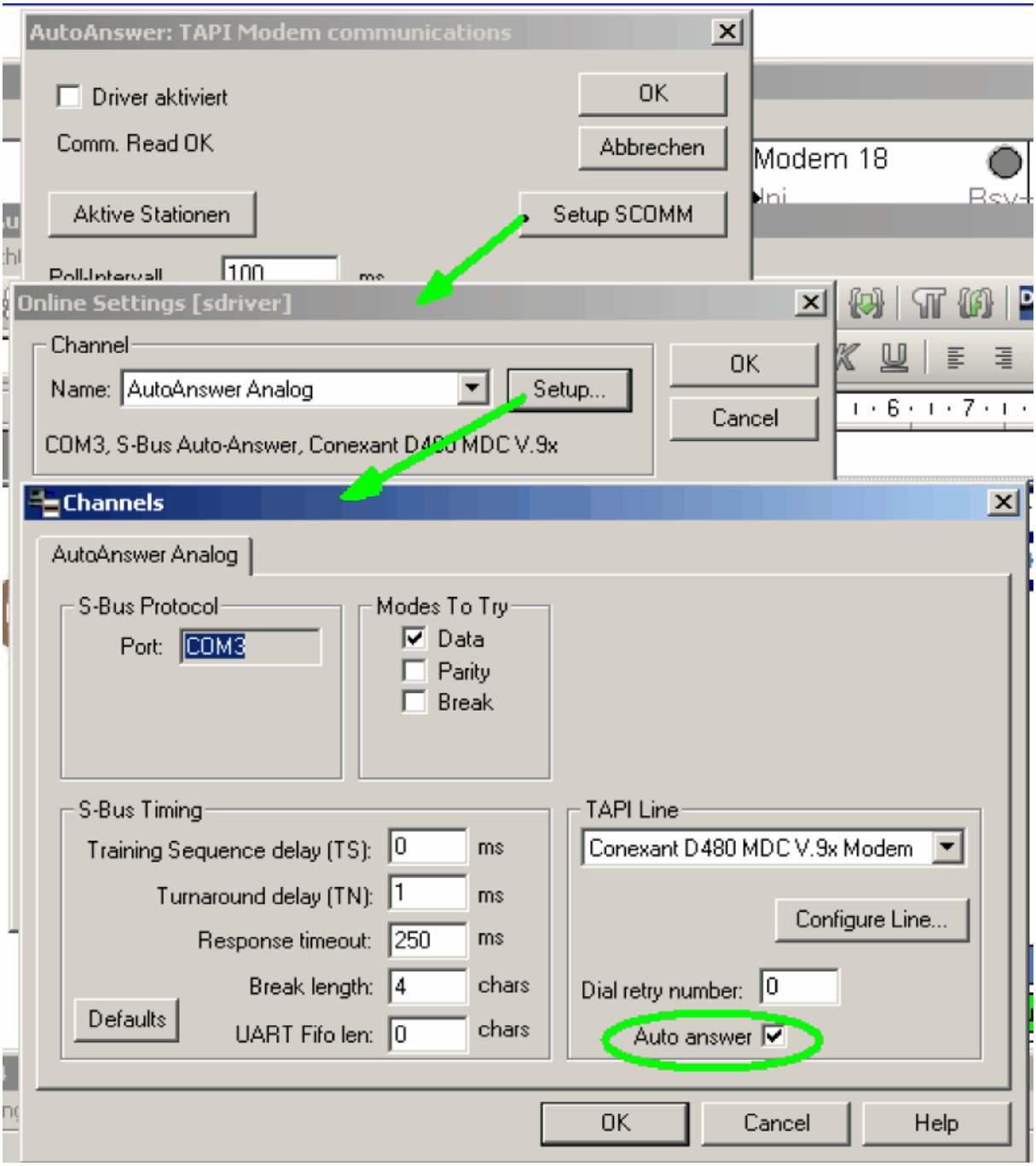
The "**Callback Delay**" parameter has the same function as the "**CallBackDelay**" parameter described under Information for Visi.Plus. If this is not indicated in the text, this default value is used.

The "**Callback Duration**" parameter has the same function as the "**CallBackOnline**" parameter described under Information for Visi.Plus. If this is not indicated in the text, this default value is used.

The "**AutoAnswer Delay**" parameter indicates how many seconds after an incoming call the AutoAnswer channel should be reactivated. If 2 modems are used, 10 seconds can be entered here; if only a single modem is being used, a significantly higher valve must be entered. This is calculated from:

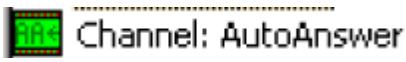
CallBackDelay + CallBackOnline + 10

The AutoAnswer channel must still be configured in accordance with the modem used and AutoAnswer must be selected:



It should therefore be taken into account that the modem is also used for the callback, and that the AutoAnswer function will be unavailable during this time.

indicates that the AutoAnswer channel is waiting for an incoming call.



indicates an active incoming call (connected).

indicates an existing callback (CB).

indicates an outgoing connected callback (CB); data is being read.

9.5 The OPC Driver (opcdriver.exe)

OPC = Open Process Control

Visi.Plus provides an OPC Driver, which reads the data from an OPC Server of a third-party provider and automatically enters it into the DMS.

Visi.Plus currently supports the following types:

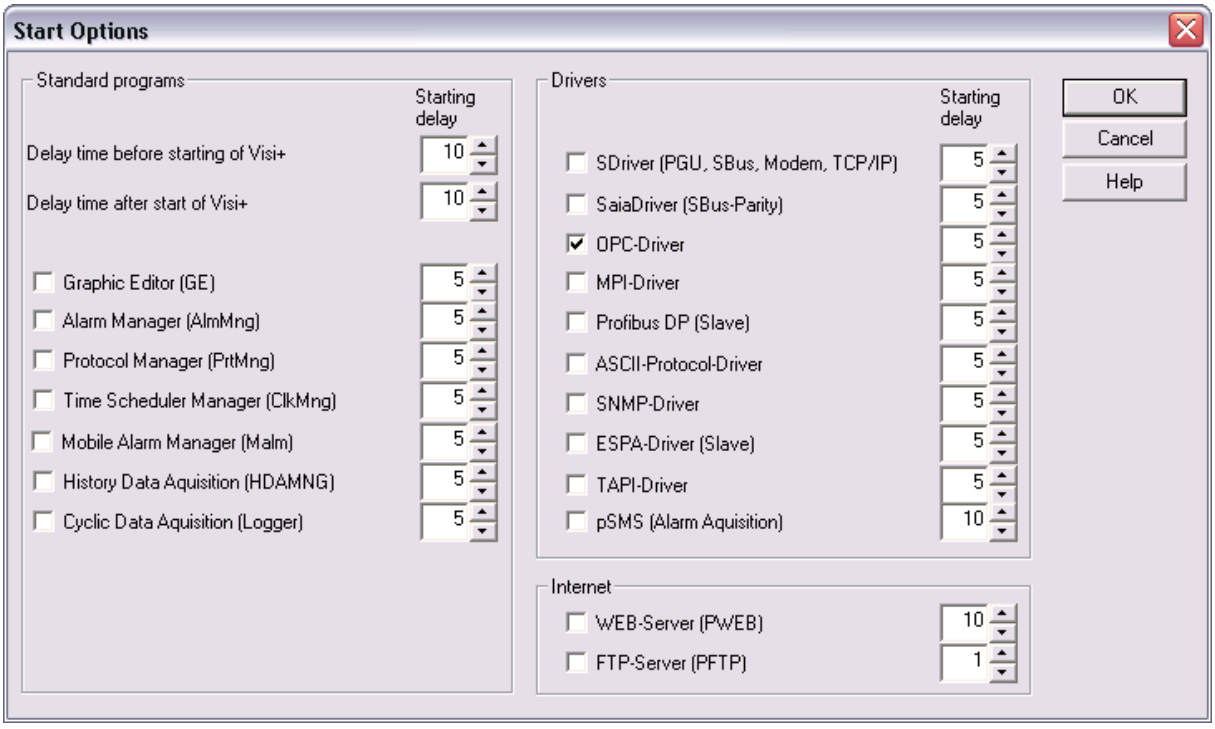
- Telemecanique (Schneider) (type 0)
- SBC OPC-Server (Merz) (type 1)
- Melsoft.MCSOPC (type 2)
- BacNet (Cimetrics) -> Caution: this type has not yet been tested! (type 3)
- General: Rockwell (RSLinx) (type 4)
- BACnet (MBR) (type 5)

9.5.1 Starting the program "OPCDriver"

This program should be started up automatically when a project is opened.

This can be configured as follows:

- Open the **Project Settings** window (Start > Programs > Saia Visi.Plus > Project Settings)
- Click on **<Start Options>**
- Mark the checkbox **OPC Driver**
- Confirm with **<OK>**



The next time the project starts, the OPC Driver will automatically be started up along with it.

The OPC Driver normally runs as a background process.

This is visible on the taskbar. By clicking on the OPC Driver the OPC Driver user interface opens.

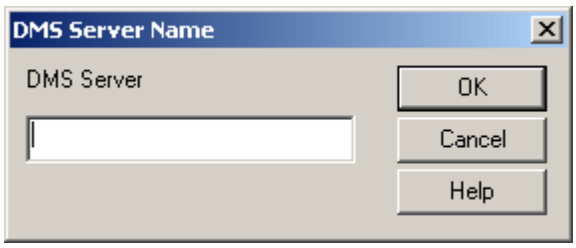
If the OPC Driver program is not started, no data is communicated to the connected PLC.
The OPC Driver is located in the default installation directory "c:\Visi.Plus\bin" and can be started from there directly at any time.

9.5.2 Starting the OPCDriver in remote mode

The OPCDriver can also be started on a remote PC and connected to the DMS server.

The project must first be copied to the remote PC and the project paths should be set accordingly in the ProjectCfg.

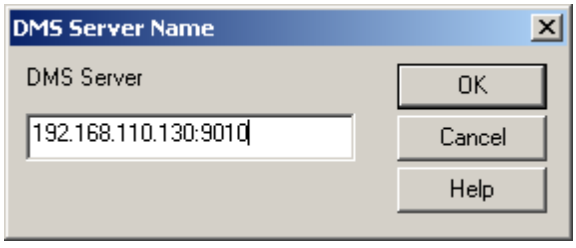
In this case, the following dialog is displayed during the first run:



Here the IP address of the DMS server must be entered, together with the DMS port number, e.g. if DMS IP adr. is 192.168.110.130:

192.168.110.130:9010

There will be always a TCP / IP connection to the DMS server established.
The DMS port number is default always 9010 (configurable in DMS menu, Settings...):

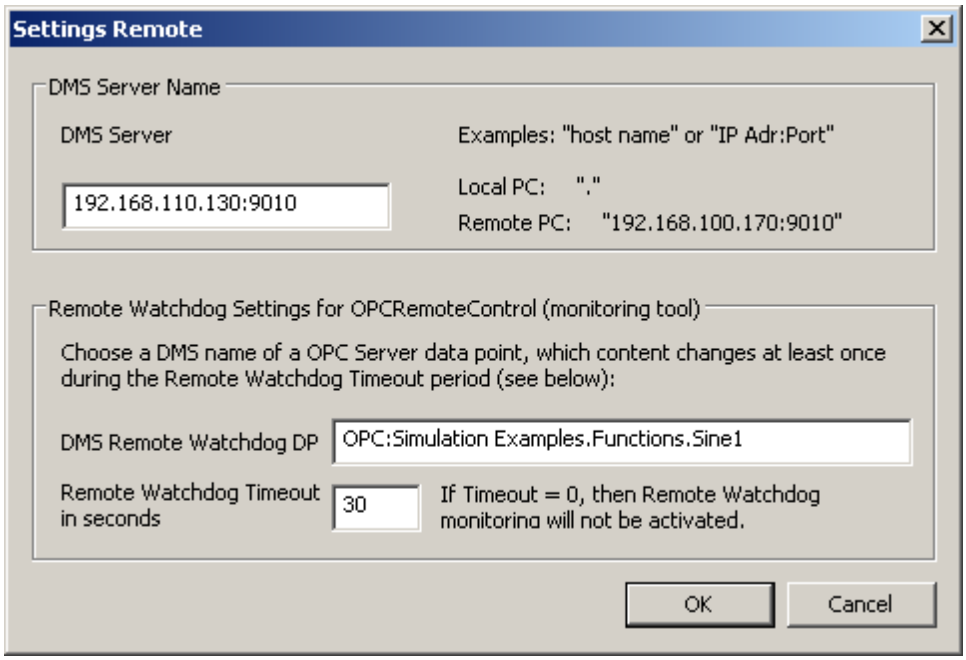


The DMS server address is saved in the opcdriver.cfg configuration file (in **..\proj\<Project name>\cfg**).

During the next OPCDriver start, OPCDriver automatically connects to DMS server without the dialog being displayed again:

```
[Settings]
RemoteDMSServer=192.168.110.130:9010
```

These settings can be modified also directly in OPCDriver menu, Settings, **Settings Remote (DMS Server + Watchdog)**:



9.5.2.1 Monitoring of the OPCDriver in remote mode

The OPCDriver can also be monitored remotely. This requires a tool program running on the same remote PC as the remote OPCDriver:

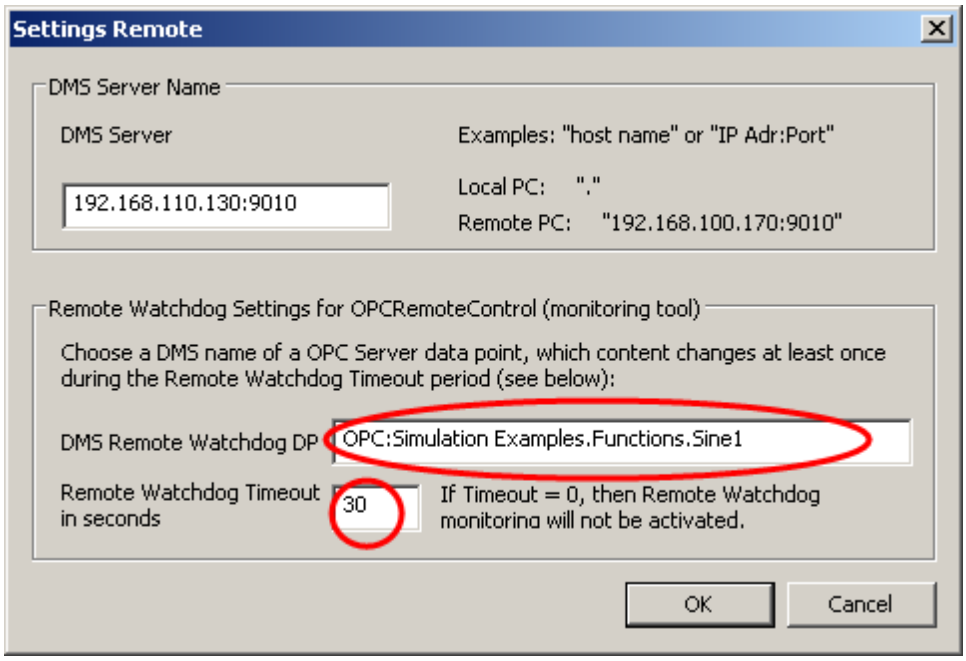
OPCRemoteControl.exe

A shortcut to this OPCRemote.exe tool should be placed into the Windows Autostart location, so that it is started automatically with the system.

On startup, OPCRemoteControl connects to the DMS to the following watchdog data points (if configured accordingly):

- DMS system watchdog, configurable via the existing commands Menu, Settings, Setting Watchdog.
  This watchdog toggelt periodically (default is 20 sec) the data point in DMS System: Prog: WatchDogRemote: OPCDRIVER between 1 and -1
  and primarily monitors the standard Windows program functionality of the OPCDriver.

- DMS remote watchdog data point, which is freely selectable and whose value is preferably written by the OPCDriver and will certainly change within the selected timeout.
  This watchdog can be used primarily for monitoring the OPC server connection.
  Setting the timeout is also done in the Remote Settings dialog:

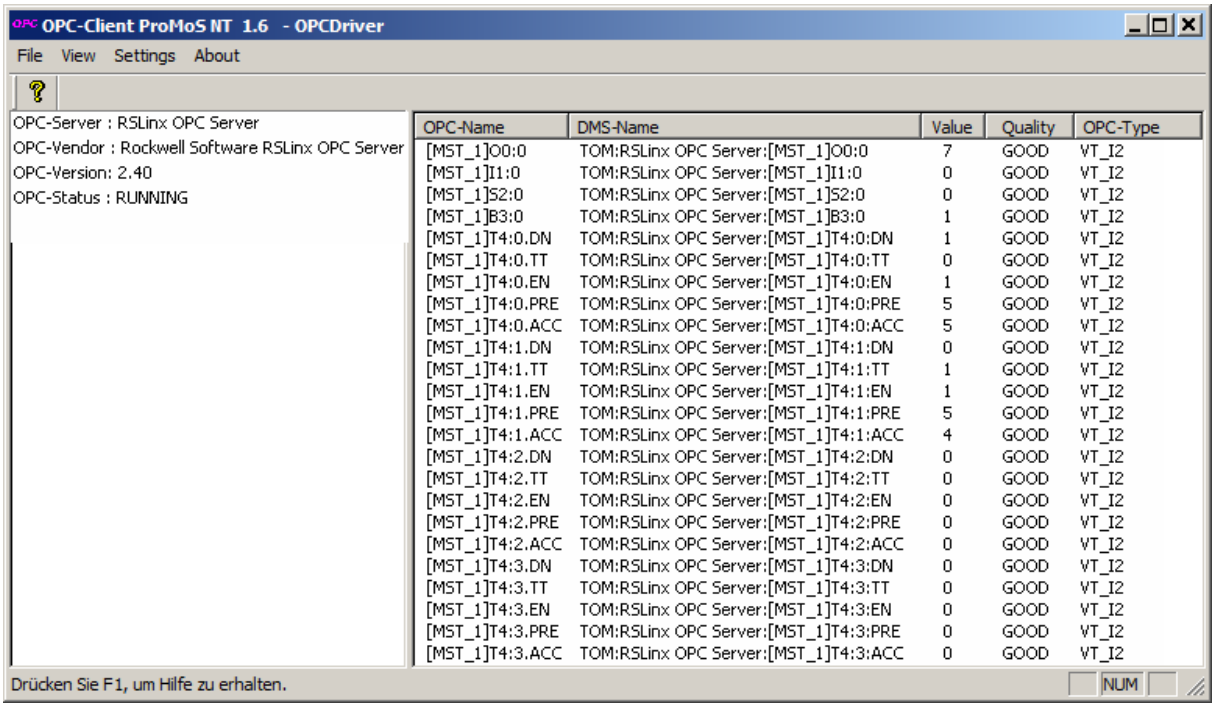


If within the configured timeout no toggle, resp. no value change of the DMS watchdog data point is detected anymore, then
- the existing OPCDriver process will be terminated
- the OPCDriver will be restarted and consequently also connected to the OPC server

All restarts will be logged into the OPCRemoteControl.log.

All settings are saved in the OPCDriver.cfg configuration file.

9.5.3 The OPC Driver user interface



Title bar
Program name and selected OPC Server

Menu bar
The menus are explained in the following.

Toolbar
Displays icon buttons for important commands and functions available in the **OPC Driver**.

Column labelling
OPC Name
OPC name of the communicated data point

DMS Name
DMS name of the communicated data point

Value
Current PLC value

Quality
Shows the correctness of the communicated signal.

OPC Type
Type of OPC Driver

Value Illustrations
Displays all communicating data points and their values.

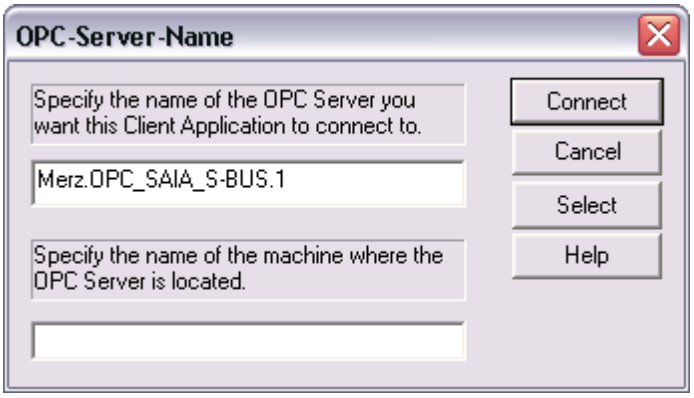
Status bar
Information and indications for the user

OPC Server Data
Displays the name of the selected OPC Server. The status and the current version number are also displayed.

9.5.4 Configuring the OPC Driver

We recommend selecting the three menu items in the "**File**" menu consecutively in order to carry out or check the necessary settings.

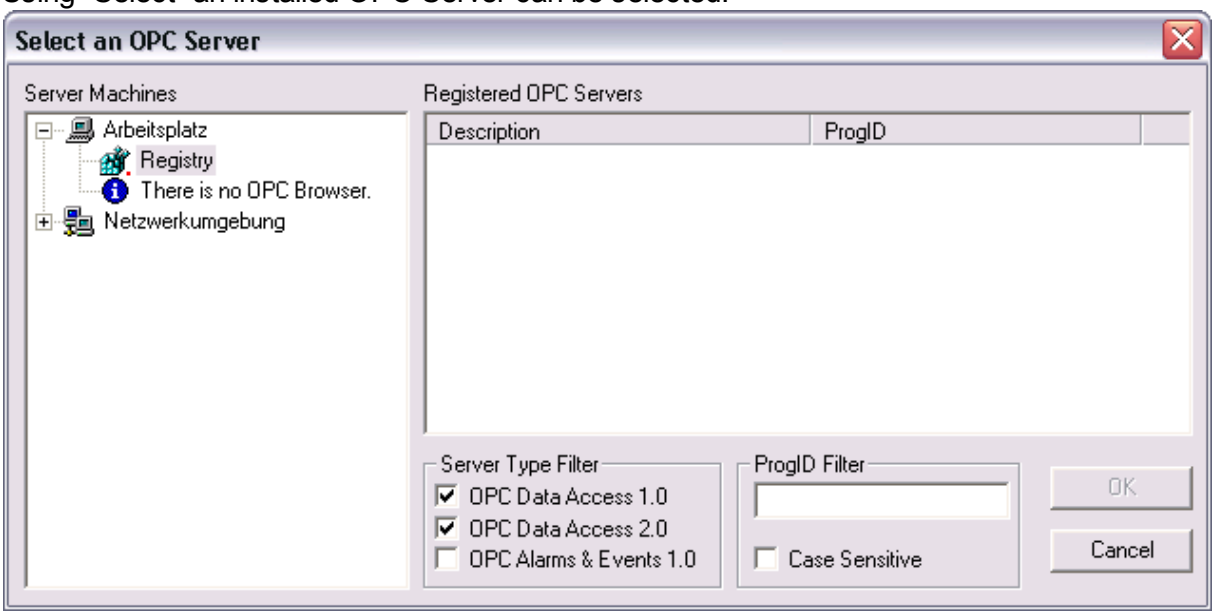
OPC Server Name



The local PC or a remote PC can be entered for the PC name.

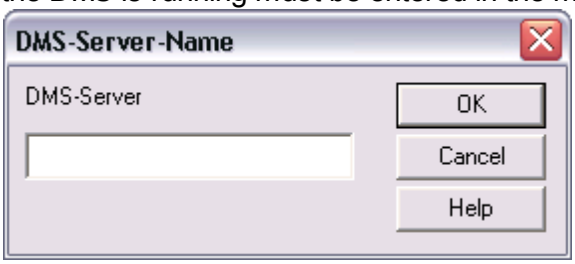
Example:

Using "Select" an installed OPC Server can be selected:



The OPC Server must be installed on the system in advance. Instructions for this installation can be found in the OPC Server manual.

If the DMS is not running on the local PC, the name or the IP address of the computer where the DMS is running must be entered in the menu item "DMS Server Name".



A dot (.) means that a connection should be established with the local DMS (on the same PC).

9.5.5 The OPC.INI file

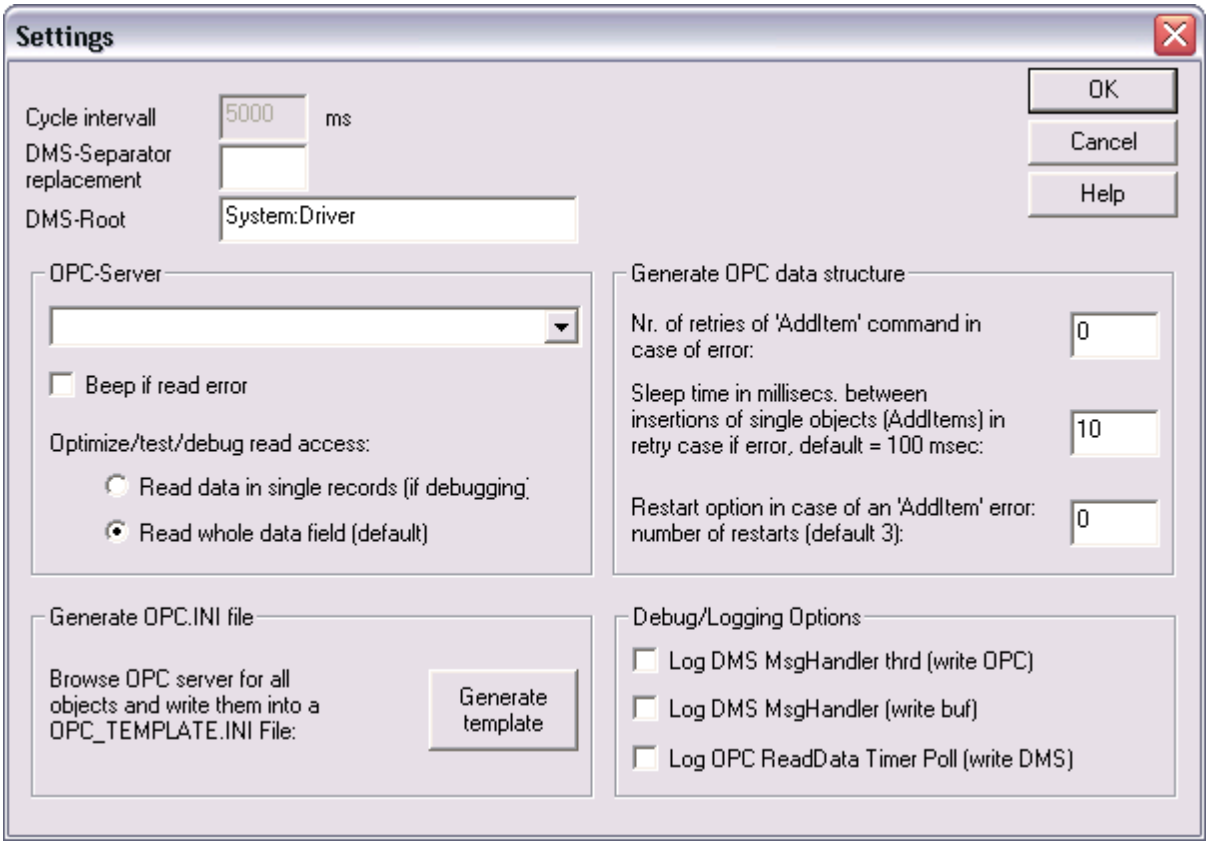
The individual communications data points can be defined with a configuration file in the project's \DRV directory.

Structure of the OPC.INI file

```
OPC-Dayname1== DMS-Name1
OPC-Dayname2== DMS-Name2
OPC-Dayname3== DMS-Name3
OPC-Dayname4== DMS-Name4
```

The file can be created manually using an ASCII editor. Not all data points available in the OPC Server have to be listed. OPC data points that are not shown in the list are discarded (and not communicated to the DMS).

A template of the file can be generated through the menu item "Settings" > "General Settings":



A file with the name OPC_TEMPLATE.INI is created, which should be manually adjusted – unless you want all data point names in the DMS to be identical to those on the OPC Server. However, this usually prevents an object-oriented representation in the DMS and the benefits of inheritance are lost, e.g. in the GE.

Example of an OPC INI file (Honeywell OPC Server):

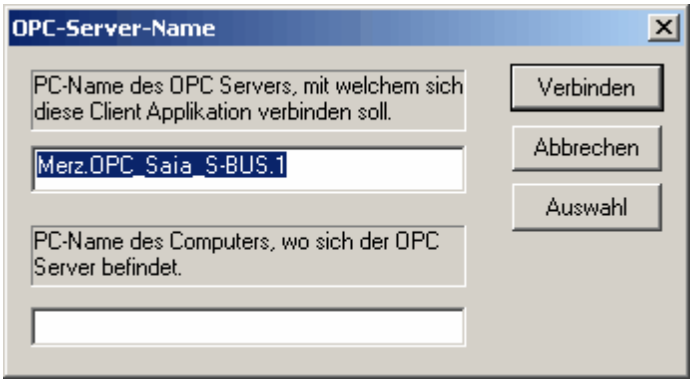
```
c102_par.DH03SS_BOILER==ZW042:H04:VS:001:Value
c102_par.DH06SS_PRIM_VE==ZW042:H15:VS:001:Value
c102_par.DH06SS_SEK_VE==ZW042:H15:VS:002:Value
c102_par.DE01AL_SECURITY==ZW042:H09:YZ:503:Status
c102_par.DE01BQ_ALARM==ZW042:H09:YZ:501:Status
c102_par.DH01AL_ENTH_PU==ZW042:H02:YZ:500:Status/INV
c102_par.DH01AL_FREON==ZW042:H02:YZ:501:Status/INV
c102_par.DH01AL_COND_PU==ZW042:H02:YZ:502:Status/INV
```

With the /INV parameter a binary signal can be inverted.

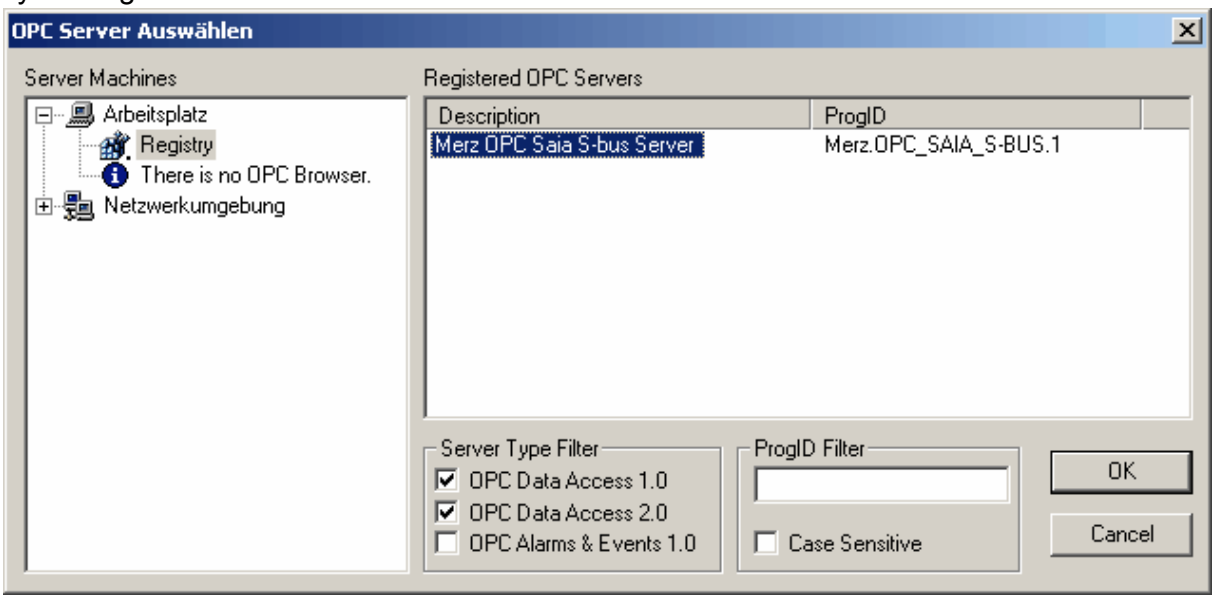
9.5.6 The OPC menus

9.5.6.1 File menu

OPC Server Name
Before data can be communicated, an OPC Server must be selected with the command "**OPC Server Name**". The following dialog window appears:



The installed OPC Server must be indicated in the input field or selected from the list provided by clicking on the **<Select>** button:



After selecting the desired OPC Driver, confirm with **<OK>**.
The screen shown above appears again and the selected OPC Server appears in the input field. By clicking on **<OK>** the settings are adopted.

DMS Server Name

With this menu item, the **network name of the computer** on which the DMS is working on the network is specified (e.g. "DMS_Server").
By default, the name is a dot "**.**". This means that the DMS is located on the current local computer.

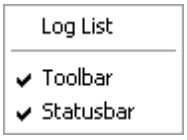
If the DMS is not running on the current computer, it must have access to the server PC on the network, otherwise no connection can be established (drive must be connected).

After modifications to the OPC Server Name and the OPC Server, the OPC Driver must be restarted.

If everything has been adjusted properly, the OPC Driver will install the OPC Server data under the specified DMS name.

9.5.6.2 View menu

In this menu the bars at the top and bottom edge of the OPC Server window can be switched on and off. A log list can also be created.



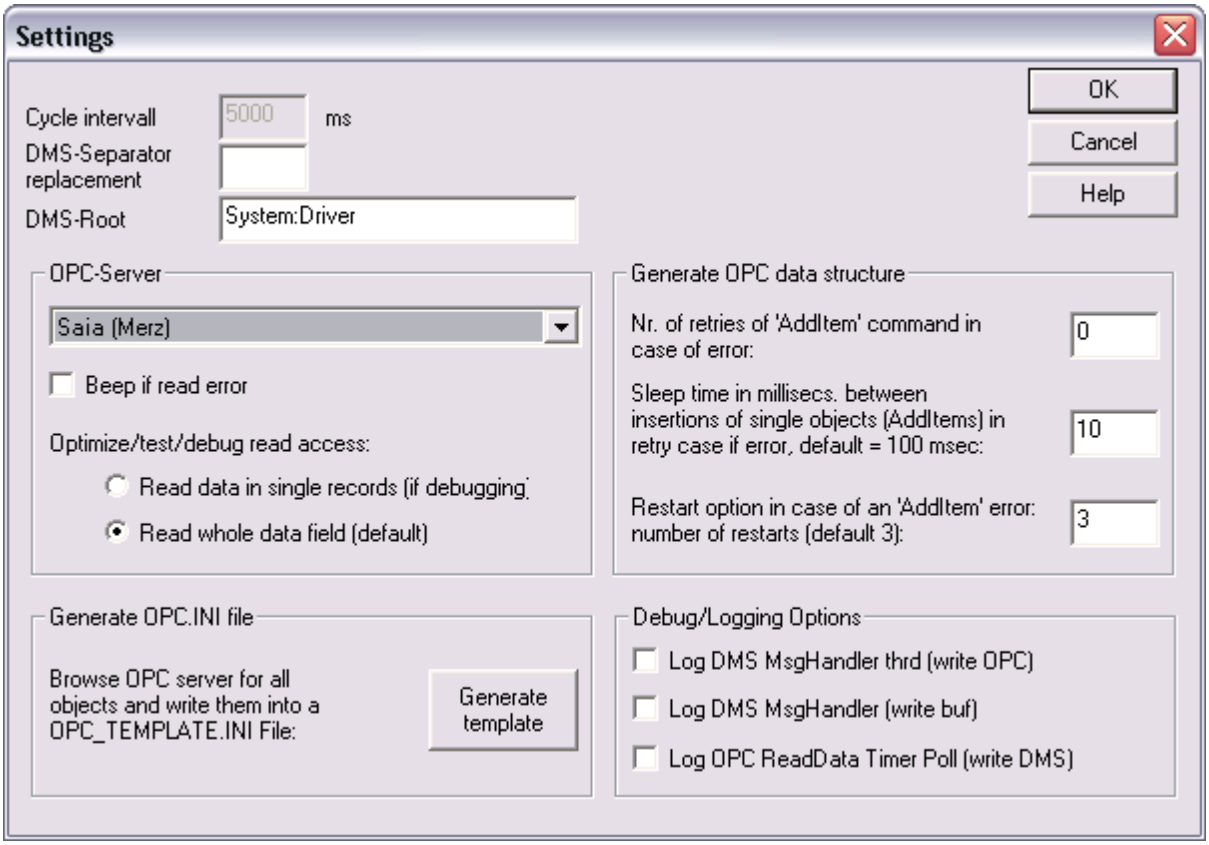
Log List
Switches the log list on or off.

Toolbar
Switches the toolbar on or off.

Status bar
Switches the status bar on or off.

9.5.6.3 Settings

General Settings
Here it is defined in which location the data should be installed in the DMS of the OPC Driver.



Cycle Interval
In this input field the frequency with which the OPC Driver updates values can be indicated.

The shorter the time, the greater the resources that are required. The suggested default value of 500 ms represents a reasonable period of time. The PC may lock up entirely with very short cycle values.

DMS Character Replacement
If a separator is used in the OPC day name, it can automatically be converted to a colon. This separator must be specified.

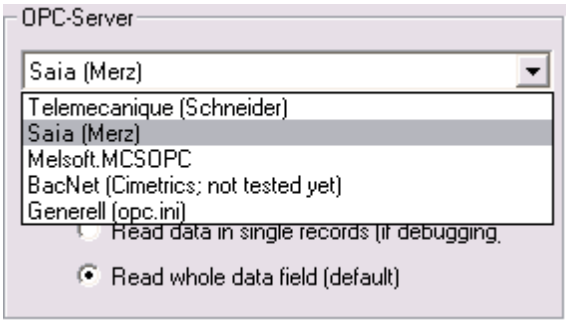
DMS Root
In the input field, the DMS name should be entered in which the OPC Server should install the PLC data. This allows a starting point to be defined in the DMS.

Example:

All of the OPC Server data points are installed under OPC:...

When working with the file OPC.INI, the DMS root has no influence.

OPC Server
The OPC Server can be selected by clicking on the dropdown menu.



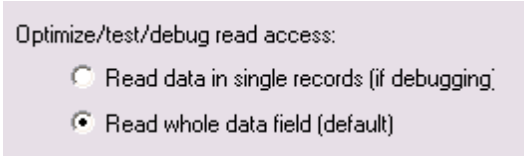
There are four different OPC Servers available for selection. If one of these servers is selected, the OPC Driver attempts to automatically store the server data points in a usable format in the DMS. In the process, all OPC tags are carried over into the DMS.

With the option "General (opc.ini)", the OPC tags are taken from the OPC.INI file (every data point can be individually defined in the file).

We recommend the option with the configuration file OPC.INI, since it is otherwise almost impossible to realise an object-oriented data structure in the DMS.

Read Error
With the option "Beep in case of read error", potential problems can be indicated audibly. In case of communications problems, a beep is issued through the PC speakers.

Optimise/Test/Debug Read Accesses
This option is only used for optimisation of new OPC Servers and should not be altered by the user. The option "Read Entire Data Field (Default)" must be activated.



Creating an OPC data structure
Some OPC Servers are incapable of initialising (inserting) large quantities of data (i.e. several thousand data points) at the same time and processing them via the "Add Item" command. For this situation, the following options exist:

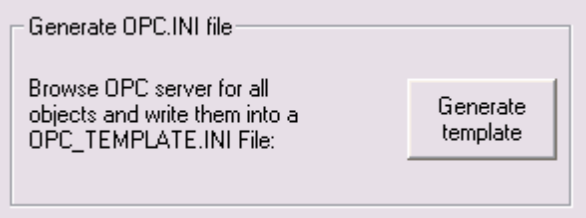
Number of repetitions of the "Add Item" command in the event of an error
When an error occurs, the initialisation sequence for the respective data point is repeated accordingly.

Sleep time (waiting time) in millisec. between individual objects being inserted (Add Items) and a repetition event when an error occurs (default = 100 msec)
An appropriate delay is inserted between repetition sequences.

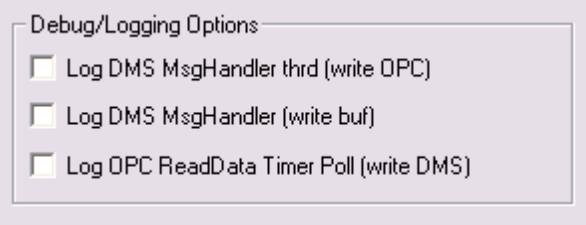
Restart option for 'Add Item' error: Number of restarts (default 3):
After all repetition sequeces have failed, the OPC Server is restarted. The restart of the OPC Server is usually more successful, because most data points (objects) have already

been initialised.

Create OPC.INI File
With this option a template for the OPC.INI file can be created. The program reads all OPC tags from the server and creates a file with the name OPC_TEMPLATE.INI in the directory \DRV, which can (and should) be modified by the user.



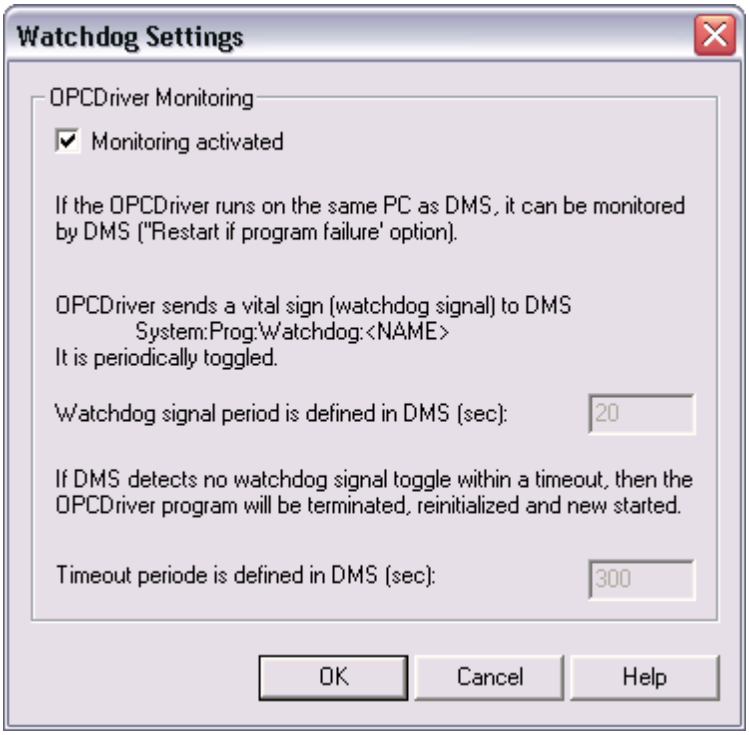
Debug/logging options
By activation of the individual options, the debug output (log file) can be made significantly more meaningful. These options are only to be activated if problems are encountered. The Visi.Plus product support department can zero in on significantly more problems with the enhanced log functions if these options are activated before the log files are sent to the support department.



<OK>
Adopts the settings.

<Cancel>
The application is exited.

Watchdog Settings
Through monitoring, the DMS can recognise a process which is not responding and restart as necessary.



9.5.6.4 Help menu

The "**?**" menu contains the Visi.Plus online help and current information on the **OPC Driver** and the version.

Help Topics
The Visi.Plus online help is accessed through the **"Help Topics"** menu, via the corresponding **[Help]** button or by pressing the **<F1>** key.

About the OPC Driver
Displays the current version of **OPC Driver**. The version must always be given in case of any support queries!

Should support queries become necessary, the internet forum is preferential to contact via telephone or e-mail. This offers two advantages: First, help is available more quickly; and secondly, other users also benefit from the entries.

9.5.7 Scaling

The scaling option is only possible for the **OPC type General** (type 4 - Rockwell).

All data types except BIT and STR can be scaled.

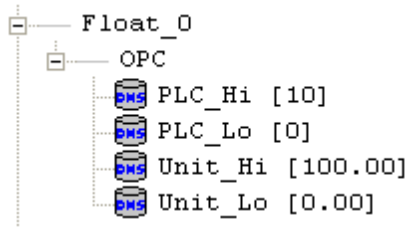
Scaling for a data point is defined by the following scaling parameters: **PLC_Lo, PLC_Hi, Unit_Lo, Unit_Hi**.
These scaling parameters must be defined in the OPC.INI for every relevant data point by appending the following information to the respective line: "<space>/SCA <PLC_Lo>-<PLC-

Hi>=<Unit_Lo>-<Unit_Hi>"

Example: "Channel_4.Device_6.Float_0==Test_4:Flt_6 **/SCA 0-10=0-100**".

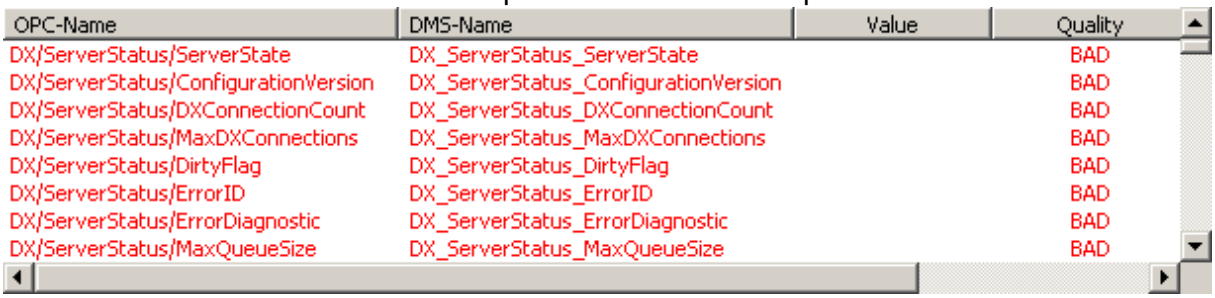
Formatting: Use at least 1 space between the DMS name and "/SCA"; spaces in the scaling definition are irrelevant, "/SCA0-10=0-100" or "/SCA 0 - 10 = 0 - 100".

After the OPC driver is started, the scaling parameters are subsequently also created in the DMS in the corresponding DP:



9.5.8 Diagnostics options

The OPC Driver shows individual data points in red in case of problems.



OPC-Name	DMS-Name	Value	Quality
DX/ServerStatus/ServerState	DX_ServerStatus_ServerState		BAD
DX/ServerStatus/ConfigurationVersion	DX_ServerStatus_ConfigurationVersion		BAD
DX/ServerStatus/DXConnectionCount	DX_ServerStatus_DXConnectionCount		BAD
DX/ServerStatus/MaxDXConnections	DX_ServerStatus_MaxDXConnections		BAD
DX/ServerStatus/DirtyFlag	DX_ServerStatus_DirtyFlag		BAD
DX/ServerStatus/ErrorID	DX_ServerStatus_ErrorID		BAD
DX/ServerStatus/ErrorDiagnostic	DX_ServerStatus_ErrorDiagnostic		BAD
DX/ServerStatus/MaxQueueSize	DX_ServerStatus_MaxQueueSize		BAD

The status (quality) of each individual data point is shown and can include the following content:

OPC Driver display	OPC designation	Comments
GOOD	OPC_QUALITY_GOOD	Data point can be read without errors
BAD	OPC_QUALITY_BAD	Data point cannot be read without errors
UNCERTAIN	OPC_QUALITY_UNCERTAIN	Data point uncertain
CONFIG ERROR	OPC_QUALITY_CONFIG_ERROR	Configuration error
NOT CONNECTED	OPC_QUALITY_NOT_CONNECTED	Data point not connected
DEVICE FAILURE	OPC_QUALITY_DEVICE_FAILURE	Hardware communications error
SENSOR FAILURE	OPC_QUALITY_SENSOR_FAILURE	
LAST KNOWN	OPC_QUALITY_LAST_KNOW	

	N	
COMM FAILURE	OPC_QUALITY_COMM_FAILURE	Communications errors
LAST USABLE	OPC_QUALITY_LAST_USABLE	
SENSOR CAL	OPC_QUALITY_SENSOR_CAL	
OUT OF SERVICE	OPC_QUALITY_OUT_OF_SERVICE	Out of service
SUB NORMAL	OPC_QUALITY_SUB_NORMAL	
ERROR: x		Other error (error code is issued)

9.5.9 Supported data types

The OPC Driver supports the following data types:

VT_BOOL	Binary data (TRUE, FALSE)
VT_BSTR	String
VT_I1	Byte signed
VT_UI1	Byte unsigned
VT_I2	Word signed (16 Bit)
VT_UI2	Word unsigned (16 Bit)
VT_I4	Doubleword signed (32 Bit)
VT_UI4	Doubleword unsigned (32 Bit)
VT_R4	Floating point number (4 bytes real)
VT_R8	Floating point number (8 bytes real)
VT_DATE	Date

No other data types are supported by the OPC Driver, and can lead to problems if used.

List of OPC data types with the respective code (used in log file):

```
VT NULL       1
VT I2         2
VT I4         3
VT R4         4
VT R8         5
VT CY         6
VT DATE       7
VT BSTR       8
VT DISPATCH   9
VT ERROR      10
VT BOOL       11
VT VARIANT    12
VT UNKNOWN    13
VT DECIMAL    14
VT I1         16
VT_UI1        17
VT_UI2        18
VT_UI4        19
VT_I8         20
```

```
VT_UI8          21
VT_INT          22
VT_UINT         23
VT_VOID         24
VT_HRESULT      25
VT_PTR          26
VT_SAFEARRAY    27
VT_CARRAY       28
VT_USERDEFINED  29
VT_LPSTR        30
VT_LPWSTR       31
VT_RECORD       36
```

9.5.10 Monitoring of the OPC Server

OPC Server Status

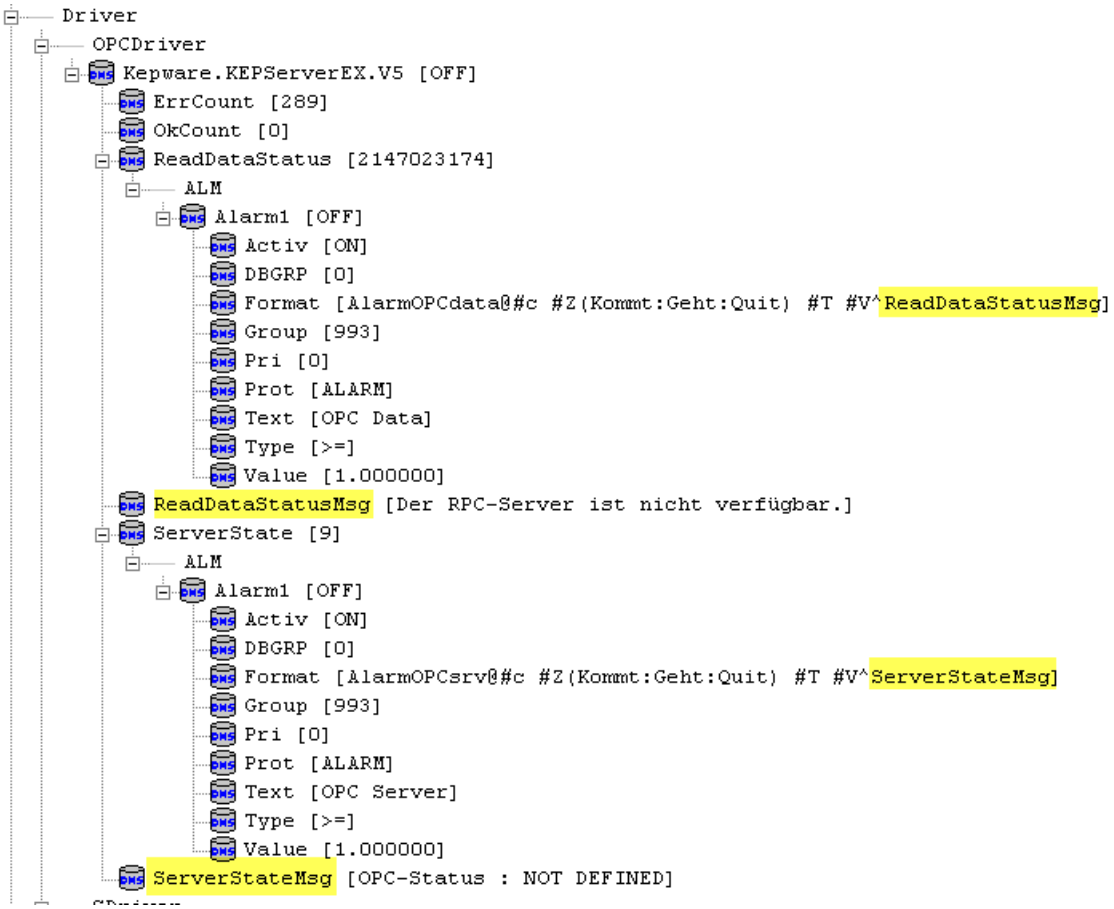
1) The OPC server is each WatchDog time tick monitored by the OPCDriver, the result - server state - is written into DMS:

- System:Driver:OPCDriver:<OPC-Manufacturer>:ServerState
  and
- System:Driver:OPCDriver:<OPC-Manufacturer>:ServerStateMsg

2) There is a second OPC server status, which is written by the OPCDriver as result of each read data cycle:

- System:Driver:OPCDriver:<OPC-Manufacturer>:ReadDataStatus
  and
- System:Driver:OPCDriver:<OPC-Manufacturer>:ReadDataStatusMsg

Example with detected OPC server errors:



Alarm ALM add-on's definitions

To the both status DP's an alarm ALM DP add-on is added (at the OPCDriver startup time):

- The only one difference between them is the source error message DP in the "Format"
- Following ALM DP's will be with these default values fixed predefined (hardcoded):
  - Status:ALM:Alarm1:**Activ**=1
  - **DBGRP**=0
  - **Prot**="ALARM"
  - **Type**=">="
  - **Value**=1.000
- The default values of the other ALM DP's can be configured in <proj>\cfg\opcdriver.cfg **[OPC_ALM_SRV]** section under following keywords:
  - **Group**=993
  - **Pri**=0
  - **Text**=OPC Server
  - **Format**=AlarmOPCsrv@#c #Z(Kommt:Geht:Quit) #T #V^ServerStateMsg

    and

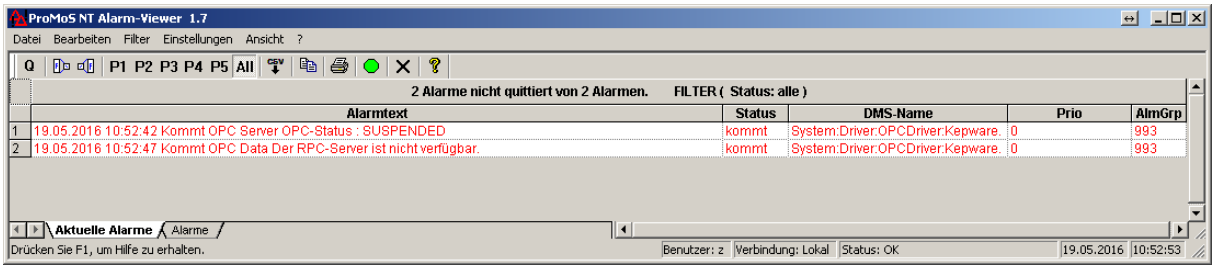
- **[OPC_ALM_DATA]** section under following keywords:
  - **Group**=993
  - **Pri**=0

- **Text**=OPC Data
- **Format**=AlarmOPCdata@#c #Z(Kommt:Geht:Quit) #T #V^ReadDataStatusMsg

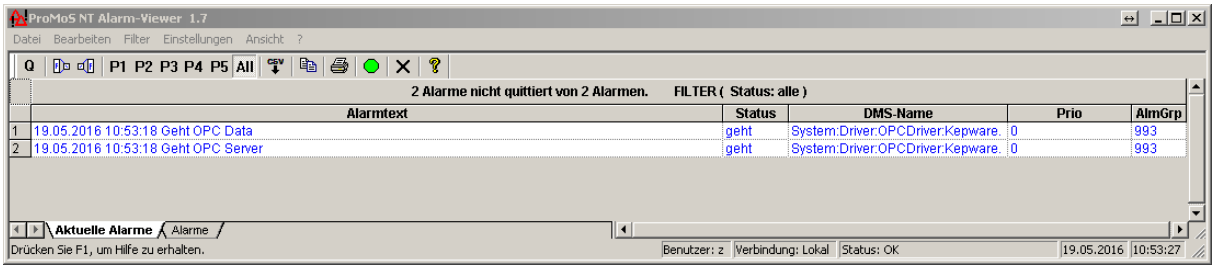
The default values will be automatically written into the opcdriver.cfg if they do not exist yet.

In AlarmView then the OPC alarms are displayed as follows:

In case of error:



After the errors went away:



9.6 The ESPA driver (espadriver.exe)

ESPA = European Selective Paging Association

The ESPA driver is used for connecting external systems, e.g. light call systems, fire alarm systems, staff paging systems, telephone exchanges, etc.

- The ESPA driver is always operated as a ESPA slave (ESPA interface settings are described in depth in the following).
- A received ESPA message is compared against an allocation list (= ESPADRIVER.INI file in the project's \DRV directory).
- When the address and the starting text match, the respective instruction/function is executed in the DMS.

The following functions are available:

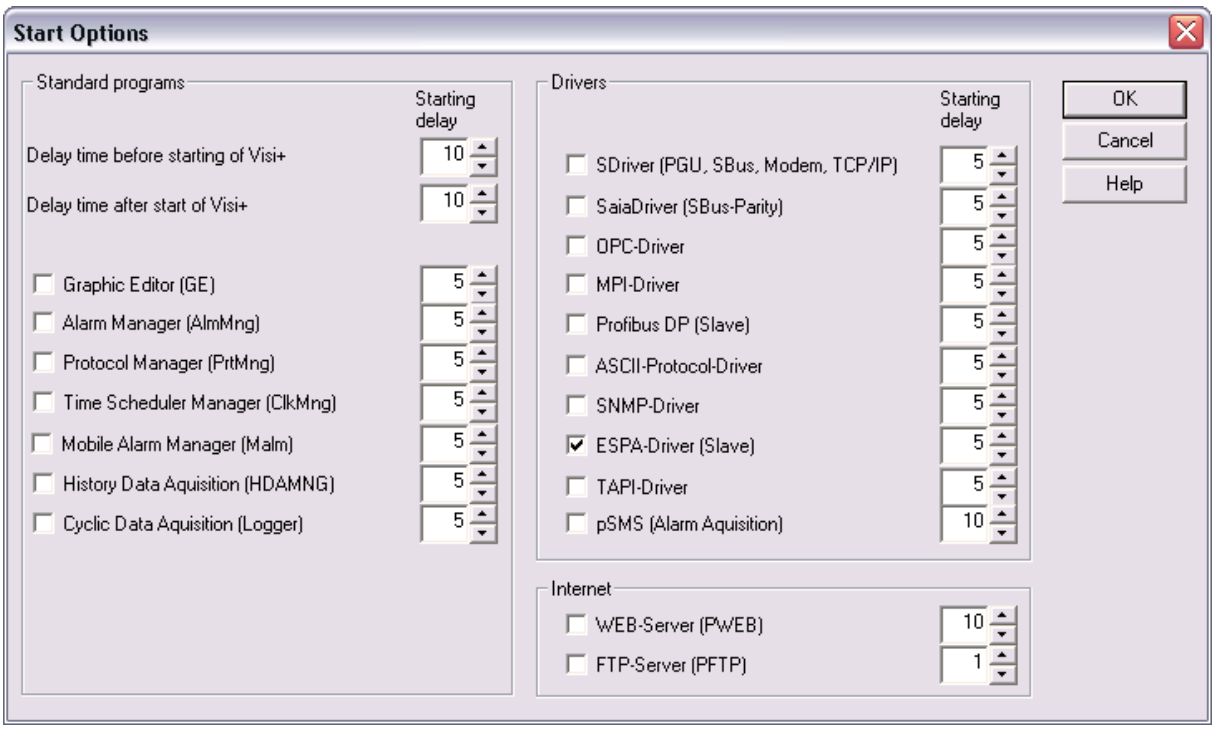
- Set or reset bit / produce impulse
- Set register
- Enter received text in the DMS

9.6.1 Starting the program "ESPADriver"

The **ESPA driver** can be started at any time by double-clicking on the icon in a file manager program. The filename is "**ESPADriver.exe**" in the default installation directory **C:*Visi.Plus**bin***.

Of course, your own shortcut created on the desktop is even more convenient. Alternatively, the ESPA driver can be started automatically through the **Project Manager** module, where the program to be started for each project can be selected. To do so, proceed as follows:

1. Starting the **Project Manager module**
2. Select the menu item or the **<Start Options> button**
3. Mark the checkbox **ESPA Driver (Slave)**
4. Confirm with **<OK>**



If the ESPA driver has been started, the icon appears on the Windows taskbar (bottom right corner of the screen).

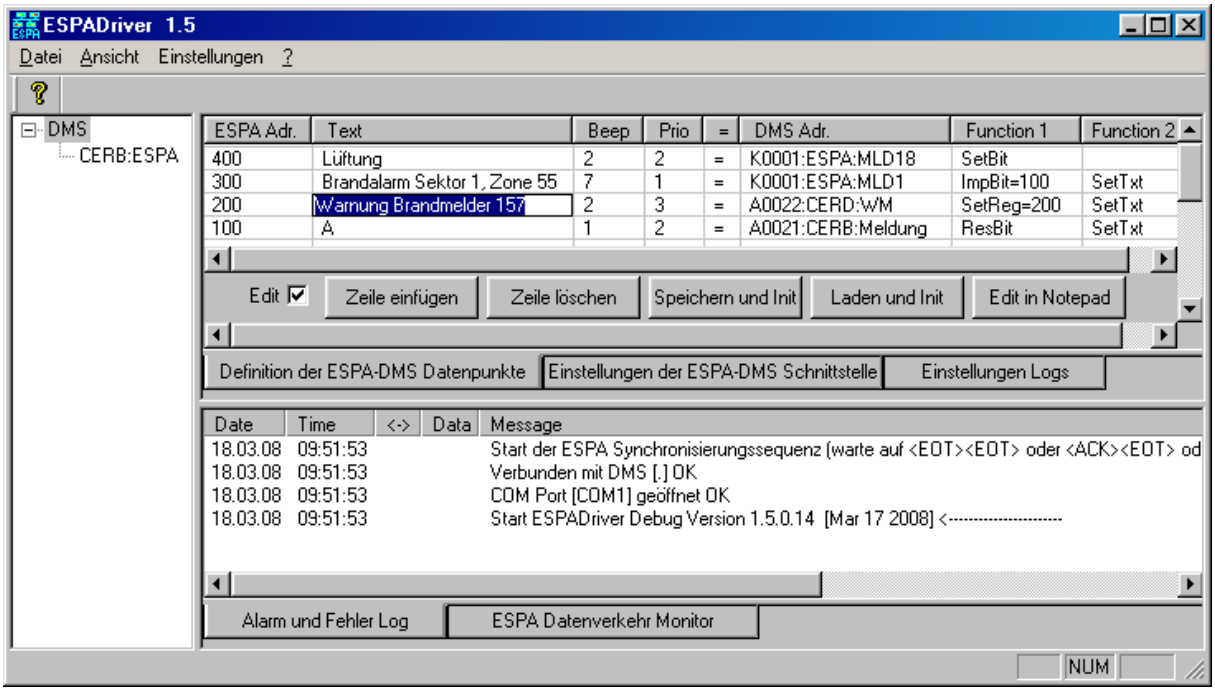
The module normally runs as a background process.

If the program ESPADriver is not started, no data is communicated to the connected device.

9.6.2 The ESPA driver user interface

With the ESPADriver loaded, the user interface shown below can be opened as follows:

Double-click with the left mouse button on the ESPADriver icon on the Windows taskbar at the buttom of the screen.



Title bar
Contains the name of the current module.

Menu bar
The menus are explained in the following.

Display window
The main window is composed of 3 panes:

- Left: DMS Root Display: shows the root DMS name under which all communicated data points are integrated.
- Top right: Configuration tab window for data point definitions and ESPA and log settings
- Bottom right: Log and ESPA monitoring window

9.6.3 The ESPA driver menus

9.6.3.1 File menu

The "**File**" menu contains the following functions:

Exit
Closes the current window and removes the driver from the computer's main memory. The driver is then no longer active.

Close Window
Minimises the main window of the **ESPAdriver** to the taskbar. The driver is not exited by doing so.

9.6.3.2 View menu

In this menu the bars at the top and bottom edge of the ESPA Driver window can be switched on and off. Log and monitor windows can also be deleted.

Delete Alarm Error Log window
Deletes the log list.

Delete ESPA Monitor window
Deletes the ESPA data communication monitor list.

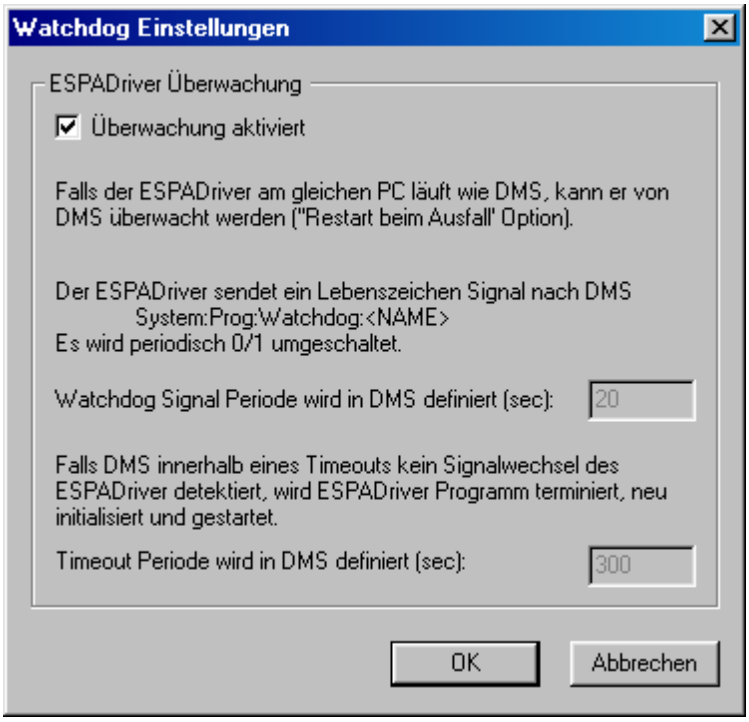
Toolbar
Switches the toolbar on or off.

Status bar
Switches the status bar on or off.

9.6.3.3 Settings menu

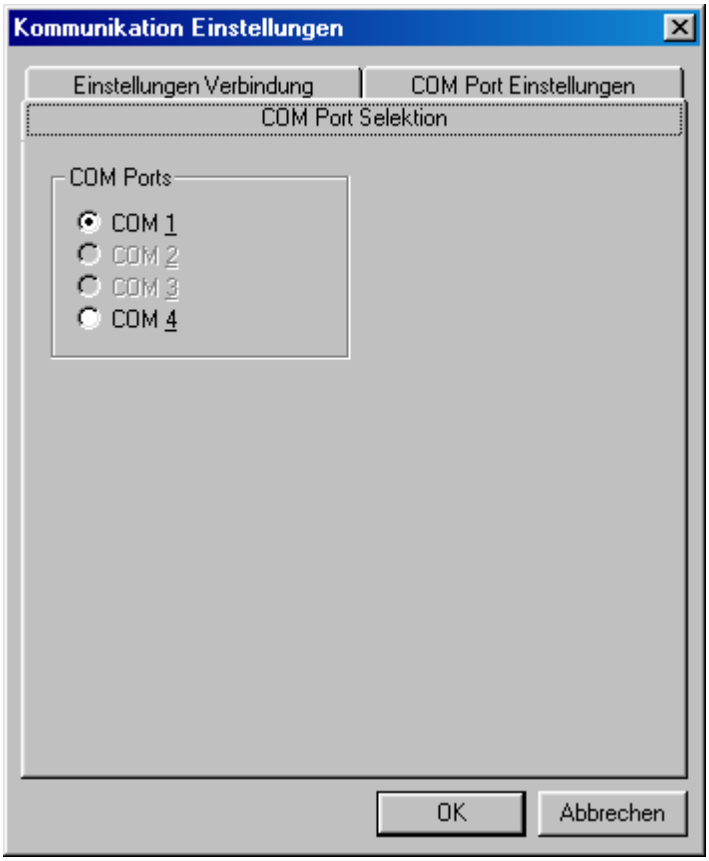
The watchdog, **communications** and debug settings are available in the "Settings" menu.

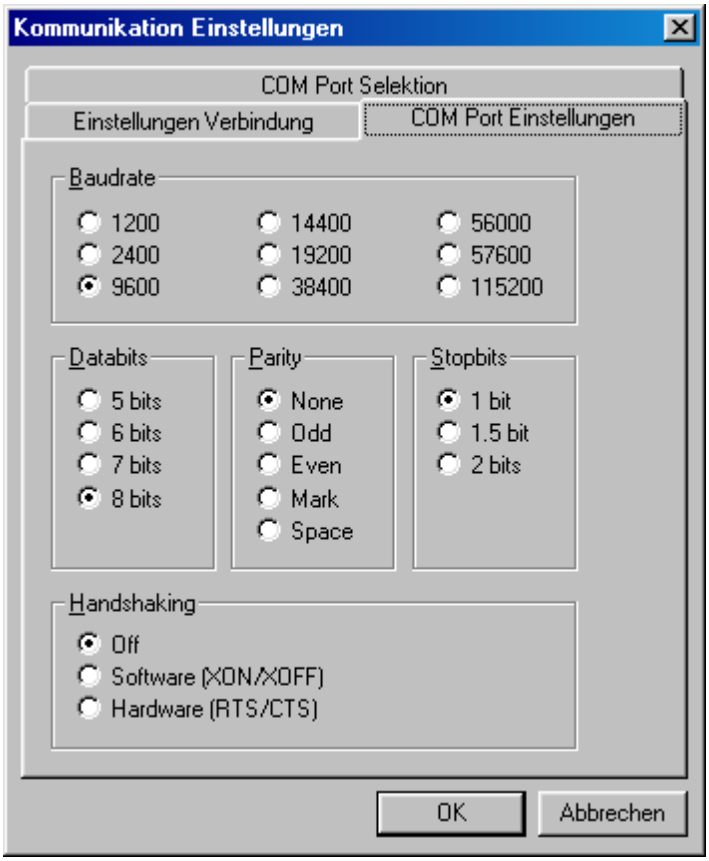
Watchdog Settings
The ESPADriver can be monitored by the Data Management System (DMS). The ESPADriver and the DMS exchange a live bit upon activation of the function.

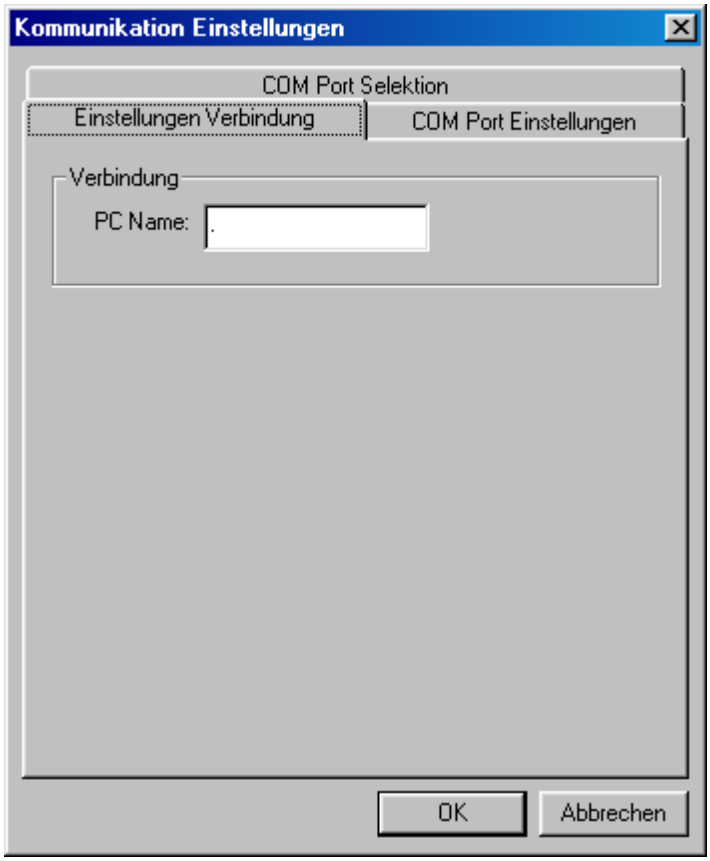


Communications Settings

In the tabs of the main window, the serial COM port interface as well as the connection to the DMS server can be defined:







Trace Settings
You can define whether actions of the ESPA driver are logged in detail in the special Trace.exe program.
Trace.exe is located in the default installation directory **C:*Visi.Plus**bin*** and must be started manually.

9.6.4 Settings and configuration

9.6.4.1 Configuring the ESPA driver (ESPADRIVER.INI)

A received ESPA message is compared against an allocation list (= ESPADRIVER.INI file in the project's \DRV directory). When the address and the starting text match, the respective instruction/function is executed in the DMS.

The complete configuration of all data point definitions can be carried out in the upper tab window "Definition of the ESPA/DMS Data Points". Alternatively, the data point definitions can be edited in the file ESPADRIVER.INI (in the project's \DRV directory).

Specification of an ESPA message

- ESPA address (0000–9999)
- Message text (1–160 chars)
- Beep code (0–9)
- Priority (0–9)

Specification of a DMS instruction

- DMS address
- Function 1
- Function 2

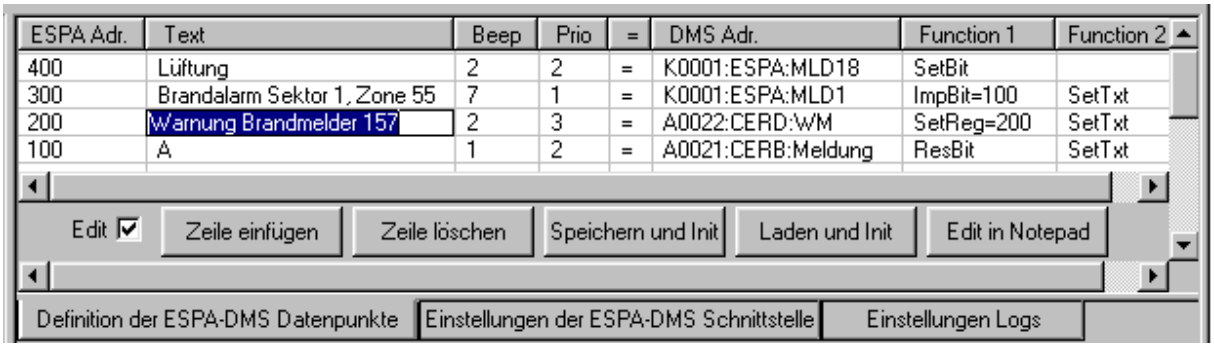
Example DMS structure of a data point:

K0001:ESPA:MLD1 = DMS name
K0001:ESPA:MLD1:**BIT** = Bit value for set/reset (type BIT)
K0001:ESPA:MLD1:**REG** = Register value for set (type FLT)
K0001:ESPA:MLD1:**ADR** = ESPA address (type FLT)
K0001:ESPA:MLD1:**TXT** = ESPA text (type STR)
K0001:ESPA:MLD1:**BC** = ESPA beep code (type FLT)
K0001:ESPA:MLD1:**PRI** = ESPA priority (type FLT)

Functions

SetBit: sets DMS name:BIT = 1
ResBit: sets DMS name:BIT = 0
SetReg=900: sets DMS name:REG= 900
SetTxt: writes received address, text, beep code and priority to DMS
ImpBit=500: makes 500 ms impulse of DMS name:BIT, meaning the DP is reset=0, set=1
and reset again=0
ImpBit: (without impulse duration parameter) makes the shortest possible impulse

Example of some ESPA DMS definitions:



Normally, the definition table is write-protected. The **Edit** checkbox must be marked for editing, then the subsequent buttons **Insert/Delete Line** and **Save and Init** are activated:

- **Insert Line**: a new blank line is added to the end of the table and marked with "**=**" in the assignment column.
- **Delete Line:** the line that the cursor is on is deleted.
- **Save and Init**: the table is saved as ESPADRIVER.INI, the file path (<Project>\DRV) is displayed and internal ESPA driver definitions are initialised accordingly, meaning the changes are instantaneously activated.
- **Load and Init**: if an ESPADRIVER.INI file was modified in an external editor, it is loaded, displayed in the definition list and the ESPA driver is initialised accordingly.
- **Edit in Notepad**: a previously saved ESPADRIVER.INI is loaded in Notepad.

Resulting ESPADRIVER.INI file:

```
400|Ventilation|2|2==K0001:ESPA:MLD18|SetBit|
300|FireAlarm Sector 1, Zone 55|7|1==K0001:ESPA:MLD1|ImpBit=100|SetTxt
200|Fire detection warning 157|2|3==A0022:CERD:WM|SetReg=200|SetTxt
100|A|1|2==A0021:CERB:Message|ResBit|SetTxt
```

Important: the number of positions that are marked with the separator "|" must be strictly adhered to.

9.6.4.2 Settings for the ESPA/DMS interface

The settings are stored in the file ESPADRIVER.CFG in the project's \CFG directory.

ESPA interface

The ESPA driver is defined as an ESPA slave station through the following settings:

- Station Address: default 2–9. The address 1 is reserved for the control station (master).
- Header Type: default 1 ('Call to pager').

  ESPA Headers:
  - '1' Call to pager
  - '2' Status information
  - '3' Status request
  - '4' Call to subscriber line
- Response Time Delay: Default 100 msec. After issuing a command, the ESPA driver only begins to evaluate the serial data line after this waiting period.

The DMS interface

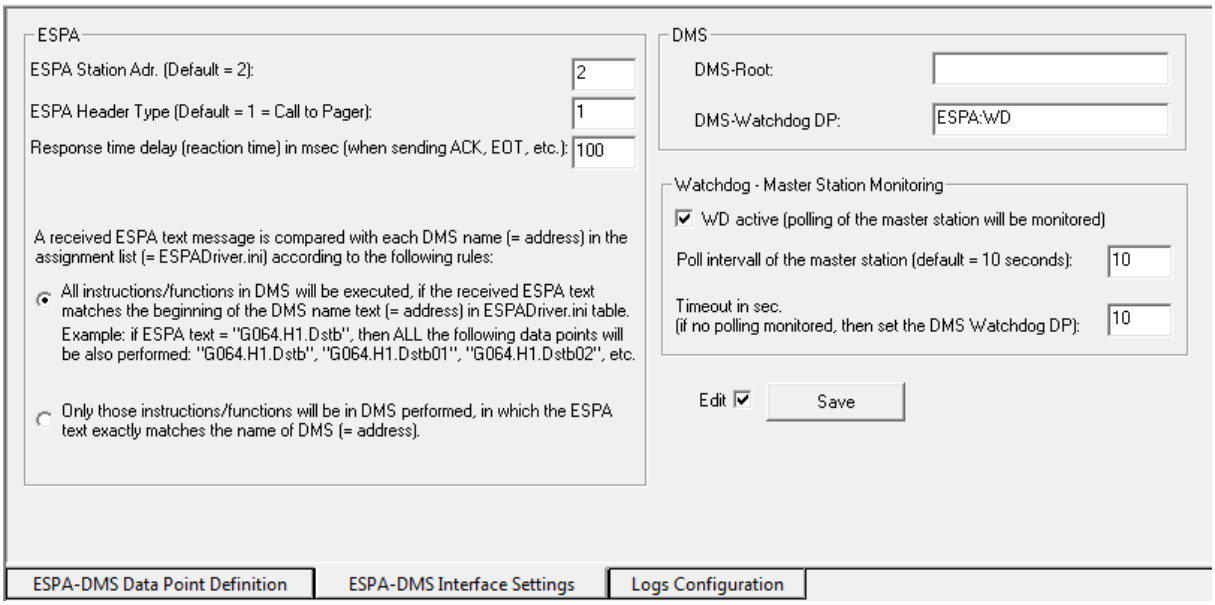
- DMS Root: when filled out, this term is prefixed to all DMS addresses.
- DMS Watchdog DP: this data point is set to 1 (= alarm) when the watchdog monitoring (see below) discovers an interruption in the connection to the control station.

ESPA master watchdog

Normally the control station can be adjusted so that it can monitor the connection to the slave. If it cannot discover a valid transaction on the line for a period of 10 seconds, it sends an EOT, then polls each station with ENQ and checks the response.

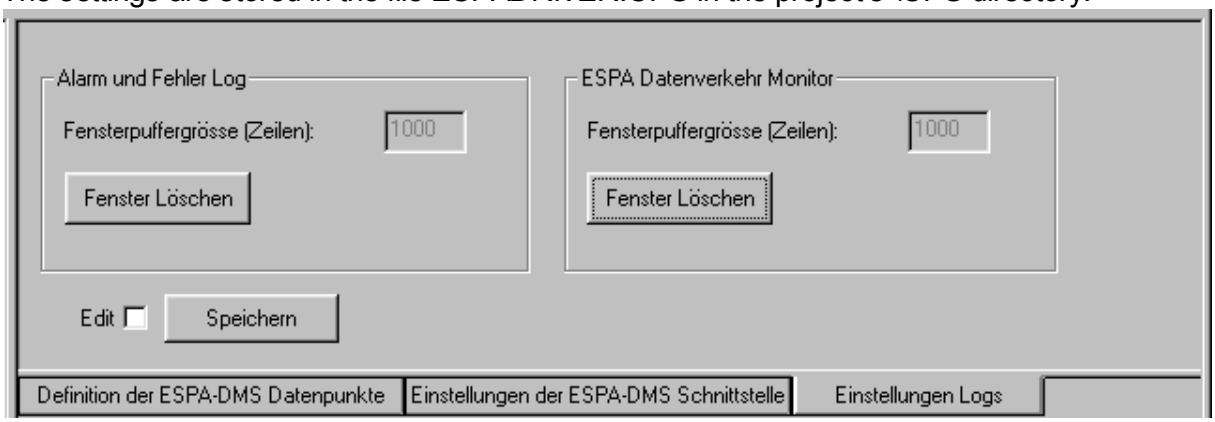
The ESPA driver can also be adjusted so that it checks the connection to the master from its end. To do so, the following settings must be made:

- WD Active must be selected.
- The poll interval of the master station should be determined and entered here (default 10 seconds).
- Timeout in seconds, after which the alarm is triggered (DMS watchdog DP is set to 1 - see above) as long as no master station polling is found during this time.



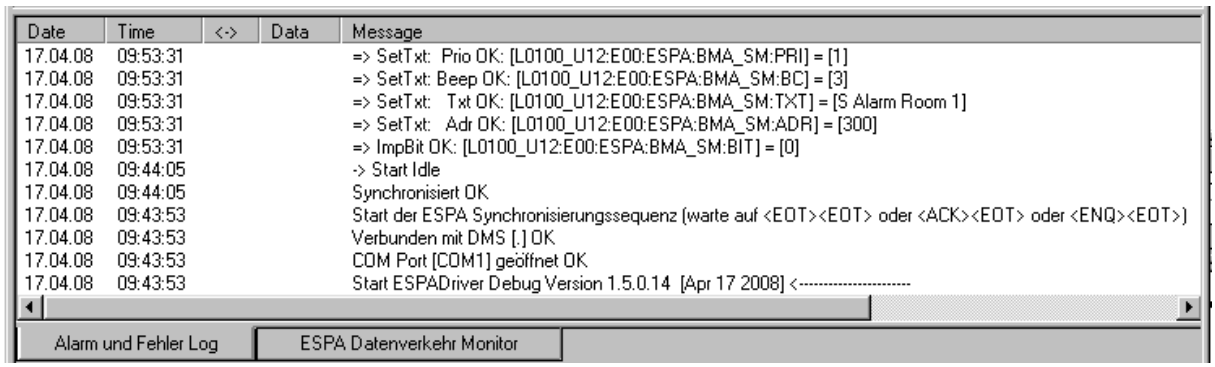
9.6.4.3 Settings for the log entries

The settings are stored in the file ESPADRIVER.CFG in the project's \CFG directory:



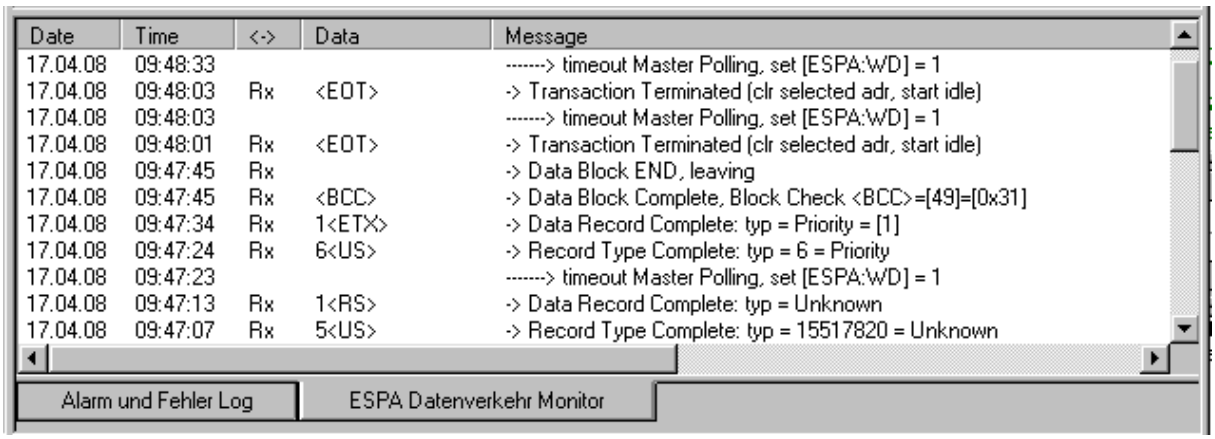
ESPA Alarm and Error Log List

All errors as well as alarm inputs and outputs are logged here.



ESPA Data Traffic Monitor

Significant ESPA log events are logged here.
This is useful for incoming error analysis.



Date	Time	<->	Data	Message
17.04.08	09:48:33			-------> timeout Master Polling, set [ESPA:WD] = 1
17.04.08	09:48:03	Rx	<EOT>	-> Transaction Terminated (clr selected adr, start idle)
17.04.08	09:48:03			-------> timeout Master Polling, set [ESPA:WD] = 1
17.04.08	09:48:01	Rx	<EOT>	-> Transaction Terminated (clr selected adr, start idle)
17.04.08	09:47:45	Rx		-> Data Block END, leaving
17.04.08	09:47:45	Rx	<BCC>	-> Data Block Complete, Block Check <BCC>=[49]=[0x31]
17.04.08	09:47:34	Rx	1<ETX>	-> Data Record Complete: typ = Priority = [1]
17.04.08	09:47:24	Rx	6<US>	-> Record Type Complete: typ = 6 = Priority
17.04.08	09:47:23			-------> timeout Master Polling, set [ESPA:WD] = 1
17.04.08	09:47:13	Rx	1<RS>	-> Data Record Complete: typ = Unknown
17.04.08	09:47:07	Rx	5<US>	-> Record Type Complete: typ = 15517820 = Unknown

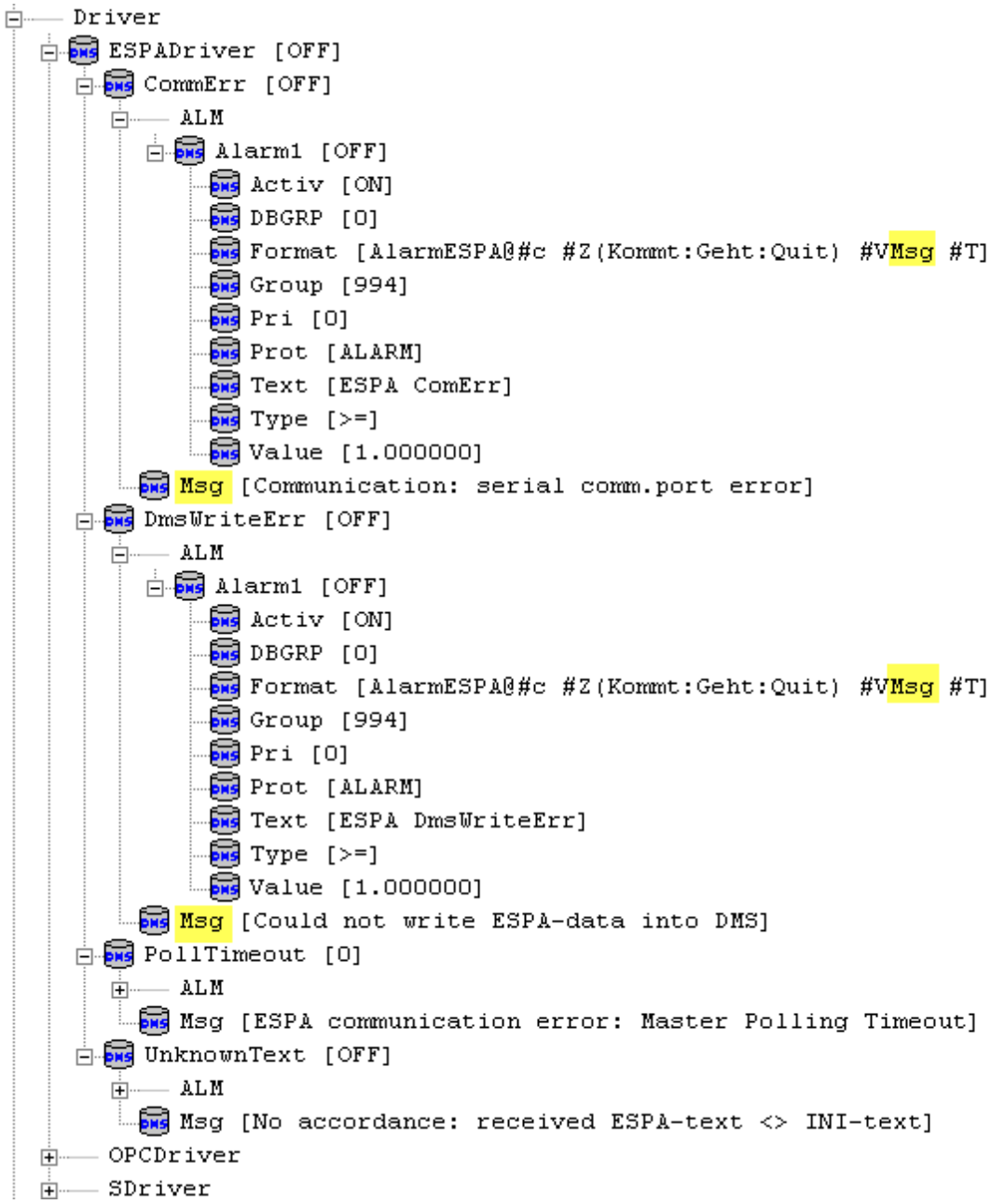
Alarm und Fehler Log | ESPA Datenverkehr Monitor

9.6.5 Monitoring of the ESPADriver

There are 4 status DPs written into **DMS:System:Driver:ESPADriver** which monitor the state and quality of the serial ESPA - DMS connection.

- System:Driver:ESPADriver:**CommErr** : (BIT) - true (1) when communication port could not be opened
- System:Driver:ESPADriver:**DmsWriteErr** : (BIT) - true (1) when failed to write DMS data point
- System:Driver:ESPADriver:**UnknownText** : (BIT) - true (1) when no text string matched
- System:Driver:ESPADriver:**PollTimeout** : (DWS) - 0=no timeout / 1=timeout / -1=not activated

To each one an ALM add-on is appended at ESPADriver startup:



Alarm ALM add-on's definitions

- The only one difference between them is the source **Text** message DP
- Following ALM DP's will be with these default values fixed predefined (hardcoded):
  - Status:ALM:Alarm1:**Activ**=1
  - **DBGRP**=0
  - **Prot**="ALARM"
  - **Type**=">="
  - **Value**=1.000

- The default values of the other ALM DP's can be configured in <proj>\cfg\espadriver.cfg **[ALM_CommErr]** section under following keywords:

- **Group**=994
- **Pri**=0
- **Text**=ESPA ComErr
- **Format**=AlarmESPA@#c #Z(Kommt:Geht:Quit) #VMsg #T
- **ErrMsg**=Communication: serial comm.port error
- **ErrMsg_Ge**=Kommunikation: serieller ComPort Fehler

- **[ALM_DmsWriteErr]** section under following keywords:
  - **Group**=994
  - **Pri**=0
  - **Text**=ESPA DmsWriteErr
  - **Format**=AlarmESPA@#c #Z(Kommt:Geht:Quit) #VMsg #T
  - **ErrMsg**=Could not write ESPA-data into DMS
  - **ErrMsg_Ge**=DMS-Schreibfehler von ESPA-Daten -> DMS

- **[ALM_PollTimeout]** section under following keywords:
  - **Group**=994
  - **Pri**=0
  - **Text**=ESPA PollTimeout
  - **Format**=AlarmESPA@#c #Z(Kommt:Geht:Quit) #VMsg #T
  - **ErrMsg**=ESPA communication error: Master Polling Timeout
  - **ErrMsg_Ge**=ESPA-Verbindungsfehler: Master Polling Timeout

- **[ALM_UnknownText]** section under following keywords:
  - **Group**=994
  - **Pri**=0
  - **Text**=ESPA UnknownText
  - **Format**=AlarmESPA@#c #Z(Kommt:Geht:Quit) #VMsg #T
  - **ErrMsg**=No accordance: received ESPA-text <> INI-text
  - **ErrMsg_Ge**=Keine übereinstimmung: empf.ESPA-Text <> INI-Text

The default values will be automatically written into the espadriver.cfg if they do not exist yet.

Example of an espadriver.cfg:

```
[ALM_CommErr]
Group=994
Pri=0
Text=ESPA ComErr
Format=AlarmESPA@#c #Z(Kommt:Geht:Quit) #VMsg #T
ErrMsg=Communication: serial comm.port error
ErrMsg_Ge=Kommunikation: serieller ComPort Fehler

[ALM_DmsWriteErr]
Group=994
Pri=0
Text=ESPA DmsWriteErr
Format=AlarmESPA@#c #Z(Kommt:Geht:Quit) #VMsg #T
ErrMsg=Could not write ESPA-data into DMS
ErrMsg_Ge=DMS-Schreibfehler von ESPA-Daten -> DMS

[ALM_PollTimeout]
```

```
Group=994
Pri=0
Text=ESPA PollTimeout
Format=AlarmESPA@#c #Z(Kommt:Geht:Quit) #VMsg #T
ErrMsg=ESPA communication error: Master Polling Timeout
ErrMsg_Ge=ESPA-Verbindungsfehler: Master Polling Timeout

[ALM_UnknownText]
Group=994
Pri=0
Text=ESPA UnknownText
Format=AlarmESPA@#c #Z(Kommt:Geht:Quit) #VMsg #T
ErrMsg=No accordance: received ESPA-text <> INI-text
ErrMsg_Ge=Keine übereinstimmung: empf.ESPA-Text <> INI-Text
```

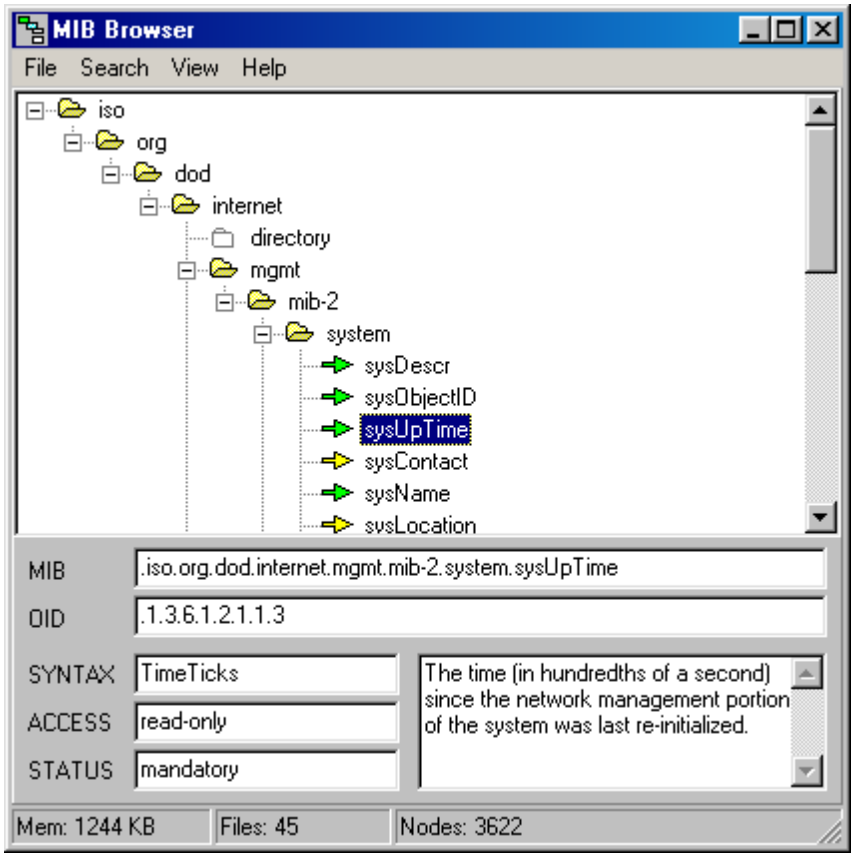
9.7 The SNMP Driver (snmpdriver.exe)

The SNMP Driver (**Simple Network Management Protocol**) can monitor all software and hardware components ("**managed objects**") on the local PC or on the network (e.g. router, server, switches, printer, computer, etc.). These managed objects must be entered in a Management Information Base (**MIB** for short) and uniquely identified by a series of numbers (**OID** for short), e.g. 1.3.6.1.2.1.

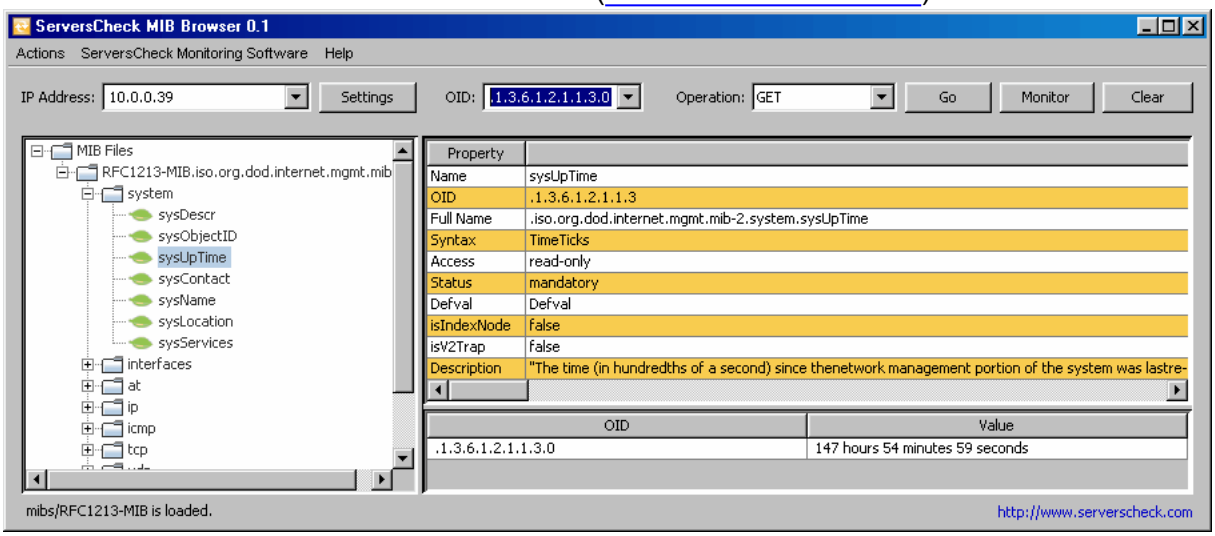
Example:
The object **.iso.org.dod.internet.mgmt.mib-2.system.sysUpTime = .1.3.6.1.2.1.1.3.0**, gives the time (in hundreths of a second) since the network management component of the system was last reinitialised.

MIB Browsers
For inspecting/displaying an MIB file or to find or verify an OID the use of an MIB brower start is recommended, e.g.:

- Freeware MIB Browser from KS-Soft (www.ks-soft.net)



- Freeware MIB Browser from ServersCheck (www.serverscheck.com)

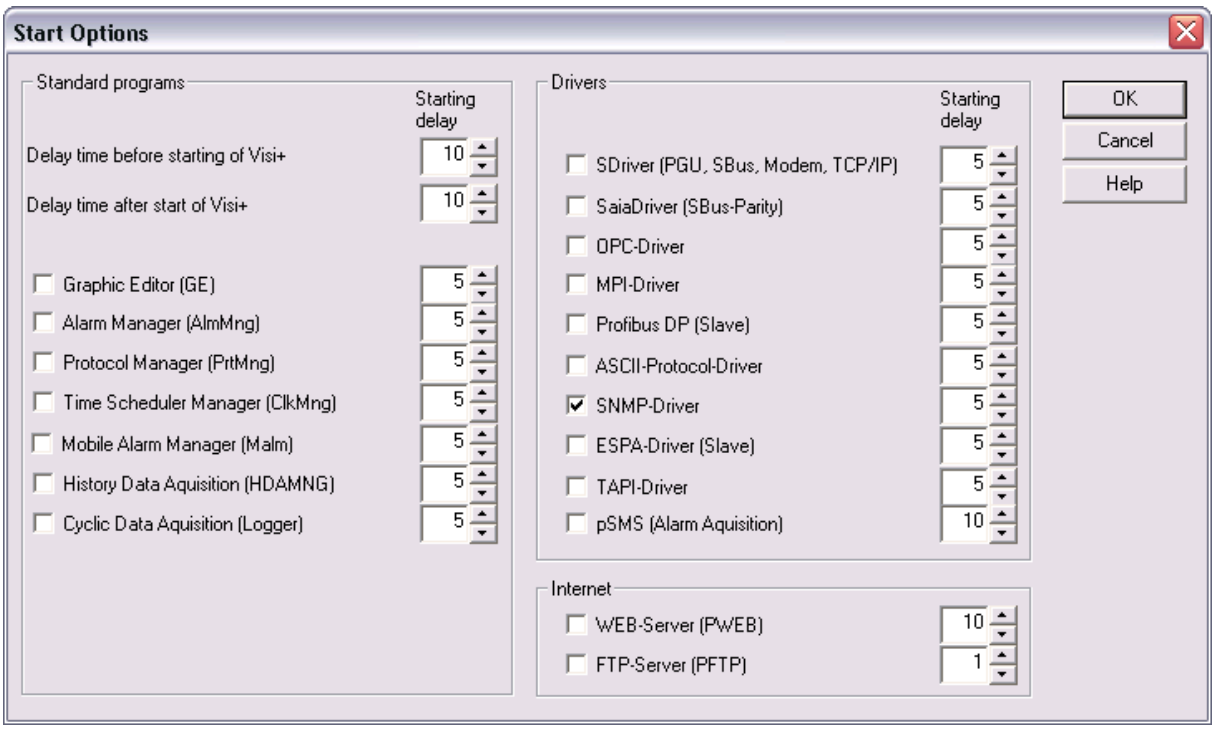


9.7.1 Starting the program "SNMPDriver"

The **SNMPDriver** can be started at any time by double-clicking on its name in a file management application. The filename is "**SNMPDriver.exe**" in the default installation directory **C:*Visi.Plus**bin***.

Of course, your own shortcut created on the desktop is even more convenient. Alternatively, the SNMPDriver can be started automatically through the **Project Manager**module, where the programs to be started for each project can be selected. To do so, proceed as follows:

1. Start the **Project Manager** module.
2. Select the menu item or the **<Start Options>** button.
3. Mark the checkbox **SNMP Driver**.
4. Confirm with **<OK>.**



If the SNMPDriver has been started, the icon 

appears on the Windows taskbar (bottom right corner of the screen).

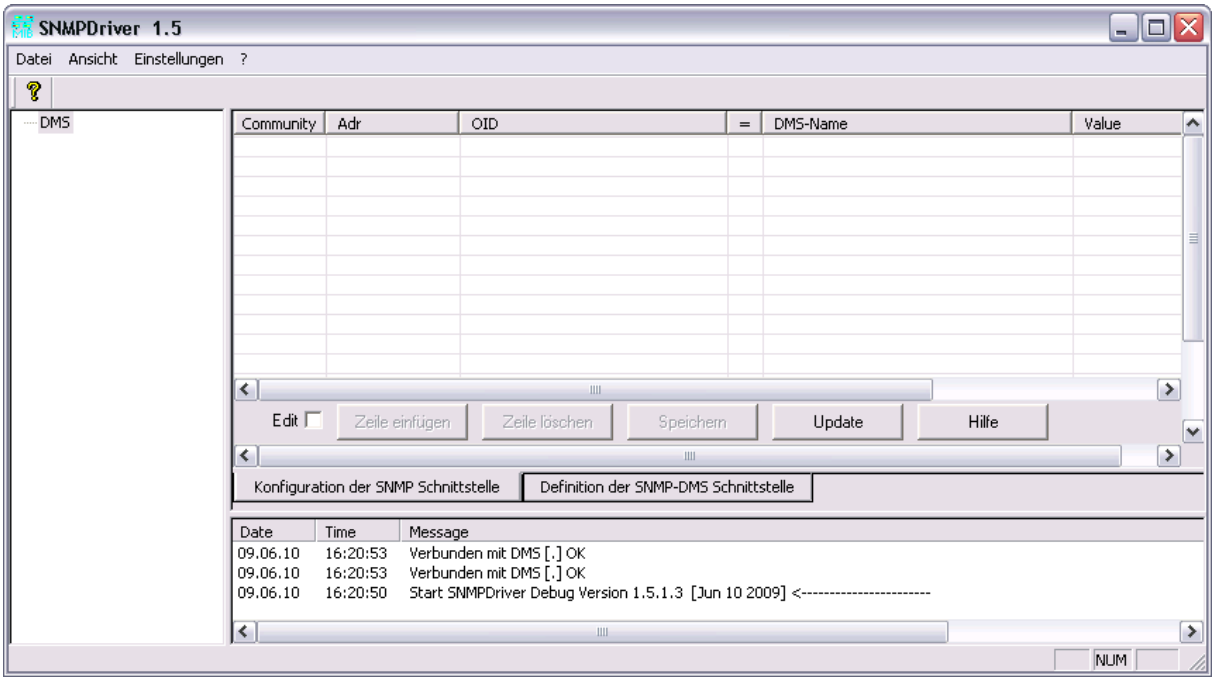
The module normally runs as a background process.

 *If the program SNMPDriver is not started, no data is communicated to the* Management Information Base *by selected objects.*

9.7.2 The SNMP Driver user interface

With the SNMPDriver loaded, the user interface shown below can be opened as follows:

Double-click with the left mouse button on the SNMPDriver icon on the Windows taskbar at the buttom of the screen.



Title bar
Contains the name of the current module.

Menu bar
The menus are explained in the following.

Display window
The main window is composed of 3 panes:

- Left: DMS Root Display: shows the root DMS name under which all communicated data points are integrated.
- Top right: configuration tab window for data point definition and SNMP, DMS and log settings
- Bottom right: log window

Tabs

- SNMP interface configuration: for defining the SNMP objects and their respective DMS data points (snmpdriver.ini).
- SNMP interface definition: SNMPDriver settings (snmpdriver.cfg).

9.7.3 The SNMPDriver menus

9.7.3.1 File menu

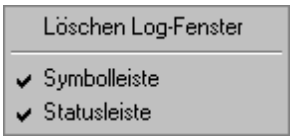
The "File" menu contains the following functions:

Exit
Closes the current window and removes the driver from the computer's main memory. The driver is then no longer active.

Close Window
Minimises the control window of the SNMPDriver to the taskbar. The driver is not exited by doing so.

9.7.3.2 View menu

In this menu the bars at the top and bottom edge of the SNMPDriver window can be switched on and off. Log and monitor windows can also be deleted.



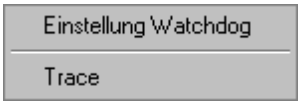
Delete Log Window 
Deletes the log list.

Toolbar
Switches the toolbar on or off.

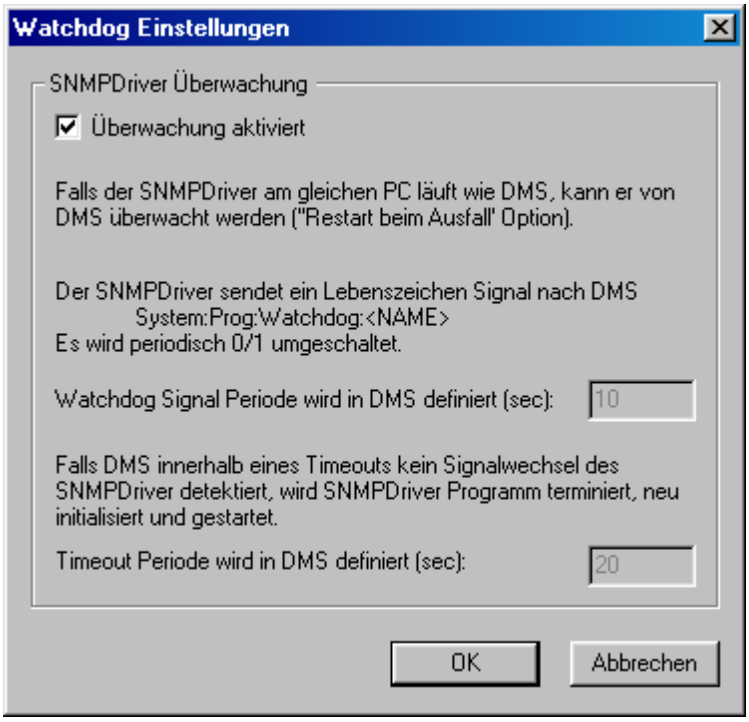
Status bar
Switches the status bar on or off.

9.7.3.3 Settings menu

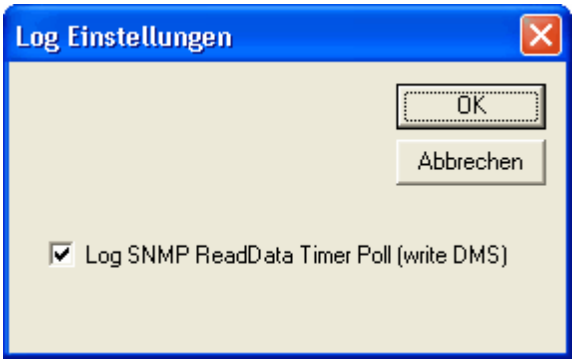
The Watchdog Settings and Debug Settings are available in the "**Settings**" menu.



Watchdog Settings
The SNMPDriver can be monitored by the Data Management System (DMS). The SNMPDriver and the DMS exchange a live bit upon activation of the function.



Log Setting
A log option for all "ReadData" read cycles of the SNMPDriver can be activated; all SNMP read accesses (in the poll timer) are logged: SNMP name and type, DMS name and type, SNMP value and old value.



Trace Settings
You can define whether actions of the SNMPDriver are logged in detail in the special Trace.exe program.
Trace.exe is located in the default installation directory **C:*Visi.Plus**bin*** and must be started manually.

9.7.4 Settings and configuration

9.7.4.1 Configuring the SNMP interface

The SNMP Driver polls the SNMP Management Console cyclically for entered objects.

If the received SNMP value has changed, the corresponding DMS data point is set on the basis of an assignment list (= SNMPDRIVER.INI file in the project's \DRV directory).

The complete configuration of all data point definitions can be carried out in the upper tab window "SNMP interface configuration". Alternatively, the data point definitions can be edited in the file SNMPDRIVER.INI (in the project's \DRV directory).

Specification of an SNMP message

- SNMP Community ('public', 'private')
- Network address (location of the MIB – Management Information Base)
- Object Identifier (OID) No. (= path for the components in the MIB)

Specification of a DMS instruction

- DMS address

Example of some SNMP/DMS definitions (see example)**:**

Community	Adr	OID	=	DMS-Name	Value	SNMP-Type
public	127.0.0.1	.1.3.6.1.2.1.1.5.0	=	SNMP:NAME1	NODE30	str
public	10.0.0.45	.1.3.6.1.2.1.1.5.0	=	SNMP:NAME45	WS001	str
public	10.0.0.55	.1.3.6.1.2.1.1.5.0	=	SNMP:NAME55	NODE12	str

Normally the definition table is write-protected. The **Edit** checkbox must be marked for editing, then the subsequent buttons **Insert/Delete Line** and **Save and Init** are activated:

- **Insert Line**: a new blank line is added to the end of the table and marked with **"="** in the Assignment column.
- **Delete Line**: the line that the cursor is on is deleted.
- **Save**: the table is saved as SNMPDRIVER.INI, the file path (<Project>\DRV) is displayed and internal SNMPDriver definitions are initialised accordingly, meaning the changes are immediately active.

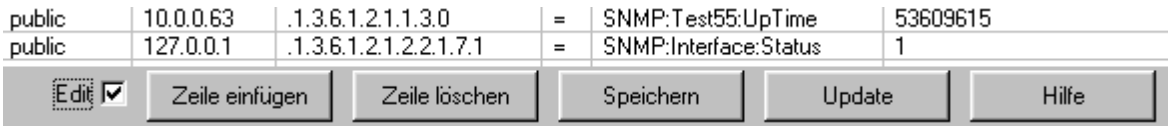
After starting the SNMPDriver, all DMS data points and their type are created.

Resulting SNMPDRIVER.INI file:

```
[OID]
.1.3.6.1.2.1.1.5.0|public|127.0.0.1==SNMP:NAME1
.1.3.6.1.2.1.1.5.0|public|10.0.0.45==SNMP:NAME45
.1.3.6.1.2.1.1.5.0|public|10.0.0.55==SNMP:NAME55
.1.3.6.1.4.1.77.1.2.1.0|public|127.0.0.1==SNMP:DESCR1
```

Important: the number of positions that are marked with the separator "|" must be strictly adhered to.

9.7.4.1.1 Controls



In order to be able to make modifications to the SNMP objects and/or DMS data points, you must:

1. Place a check in the **Edit** checkbox and then the information can be edited from right in the tab. By double-clicking you can reach the appropriate field (Not:. **Copy** with Ctrl. C and/or the right mouse button, **Paste** only with right mouse button). The modifications only become active when the "**Save**" button is pressed.

or

2. Edit the snmpdriver.ini file directly in a text editor (Notepad). The modifications only become active when, after saving snmpdriver.ini, the "**Update**" button is pressed.

For wide-ranging changes or redefinitions/imports, we recommend method 2.

Update
Reimports snmpdriver.ini and forces a refresh of all SNMP values in the DMS tree and in the SNMPDriver display.

9.7.4.2 Example of an SNMP configuration

With the **snmpdriver.ini** file listed below, SNMPDriver can be tested immediately on a network:

```
; SNMPDriver Definitions
; 08.06.2009 16:55:33

[OID]
.1.3.6.1.2.1.1.5.0|public|127.0.0.1==SNMP:NAME1
.1.3.6.1.2.1.1.5.0|public|10.0.0.45==SNMP:NAME45
.1.3.6.1.2.1.1.5.0|public|10.0.0.63==SNMP:NAME55
.1.3.6.1.4.1.77.1.2.1.0|public|127.0.0.1==SNMP:DESCR1
.1.3.6.1.4.1.77.1.2.1.0|public|10.0.0.63==SNMP:DESCR55
.1.3.6.1.4.1.77.1.2.1.0|public|10.0.0.45==SNMP:DESCR45
.1.3.6.1.2.1.1.1.0|public|127.0.0.1==SNMP:Test127:Hardware
.1.3.6.1.2.1.1.1.0|public|10.0.0.45==SNMP:Test45:Hardware
.1.3.6.1.2.1.1.1.0|public|10.0.0.63==SNMP:Test55:Hardware
.1.3.6.1.2.1.2.2.1.7.1|public|127.0.0.1==SNMP:Interface:Status
```

In addition, the following **public** SNMP objects are used which can be found on any PC. These are accessed by the local PC (IP 127.0.0.1) and 2 other PCs (10.0.0.45 and 10.0.0.63).

- .1.3.6.1.2.1.1.5.0 (=.iso.org.dod.internet.mgmt.mib-2.system.sysName): Node Name (computer name).
- .1.3.6.1.4.1.77.1.2.1.0 (=.iso.org.dod.internet.private.enterprises.lanmanager.lanmgr-2.server.svDescription): Computer Description.
- .1.3.6.1.2.1.1.1.0 (=.iso.org.dod.internet.mgmt.mib-2.system.sysDescr): PC Hardware and Sofrware Description.

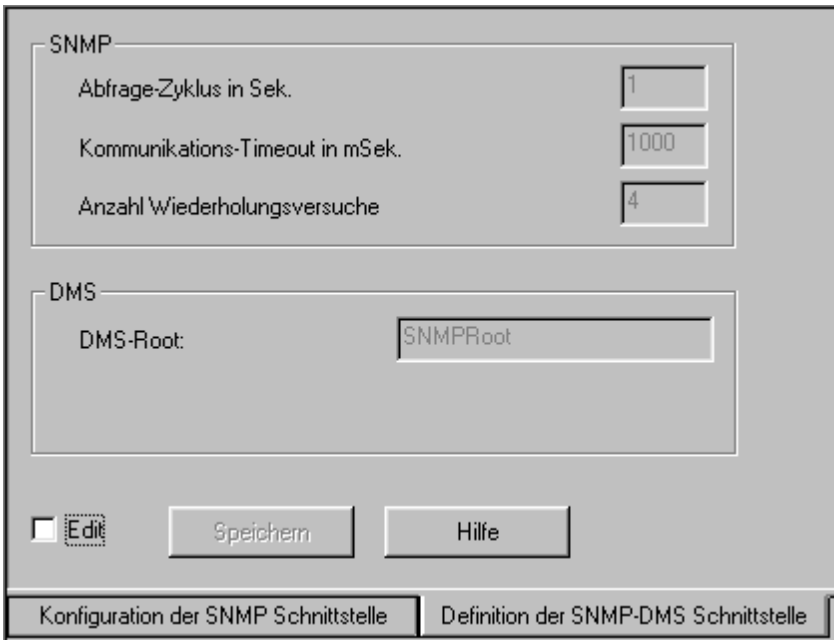
- .1.3.6.1.2.1.1.3.0 (=.iso.org.dod.internet.mgmt.mib-2.system.sysUpTime): gives the time (in hundredths of a second) since the system's Network Management component was last reinitialised.

9.7.4.3 Settings for the SNMP/DMS interface

The settings are stored in the file SNMPDRIVER.CFG in the project's \CFG directory.

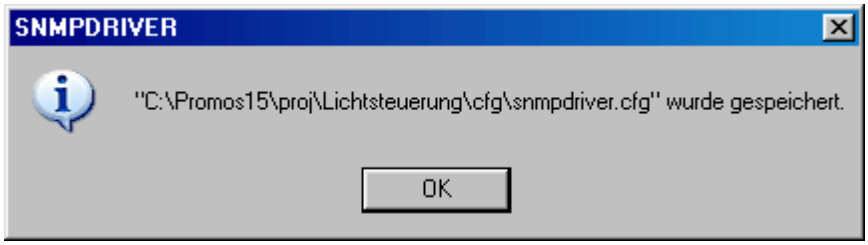
The SNMP interface
- Query cycle of the SNMP management console: Time in seconds (recommended: 5 to 10 sec.)
- Communications timeout in milliseconds (recommended: 100 to 1000 msec)
- Number of repeat attempts (if an SNMP query cannot be executed)

The DMS interface
- DMS Root: when filled out, this term is prefixed to all DMS addresses.



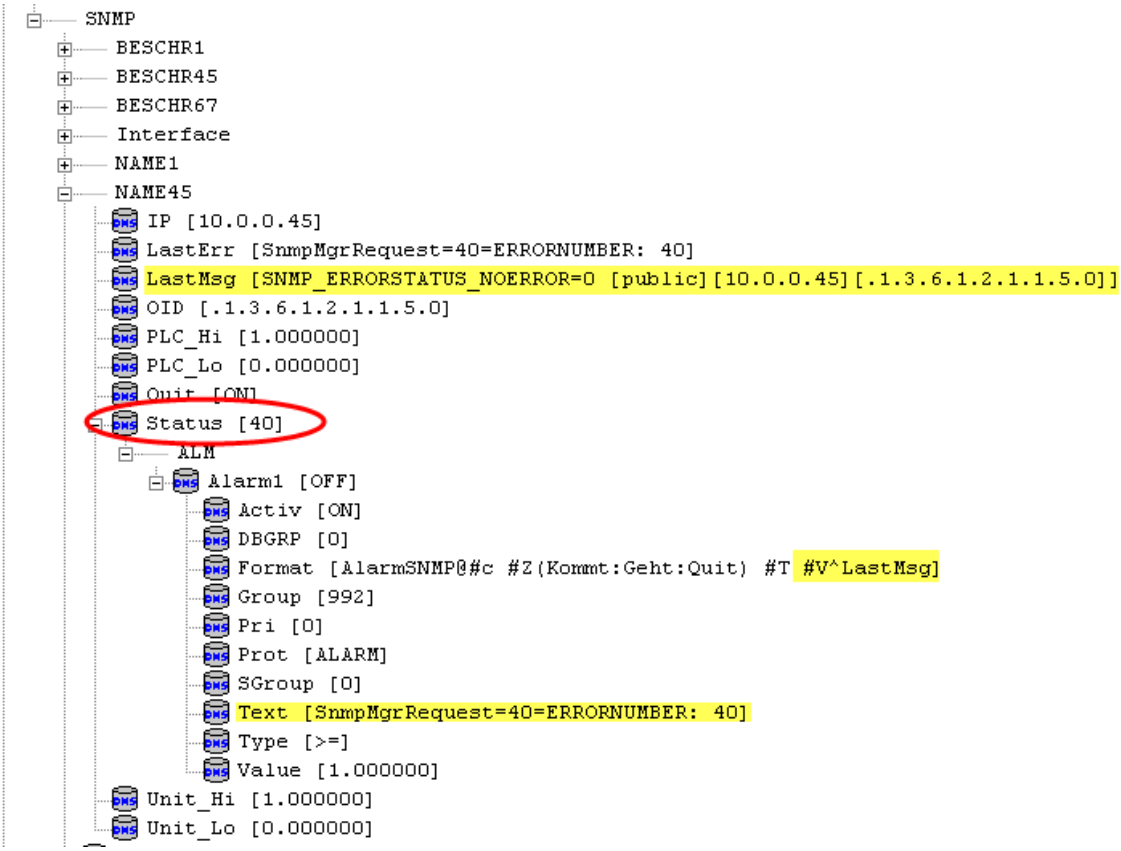
Edit (checkbox)
In order to modify the top settings, a checkmark must be set here.

Save (button)
Modified settings are saved. A confirmation dialog showing the path information for the configuration file then appears:



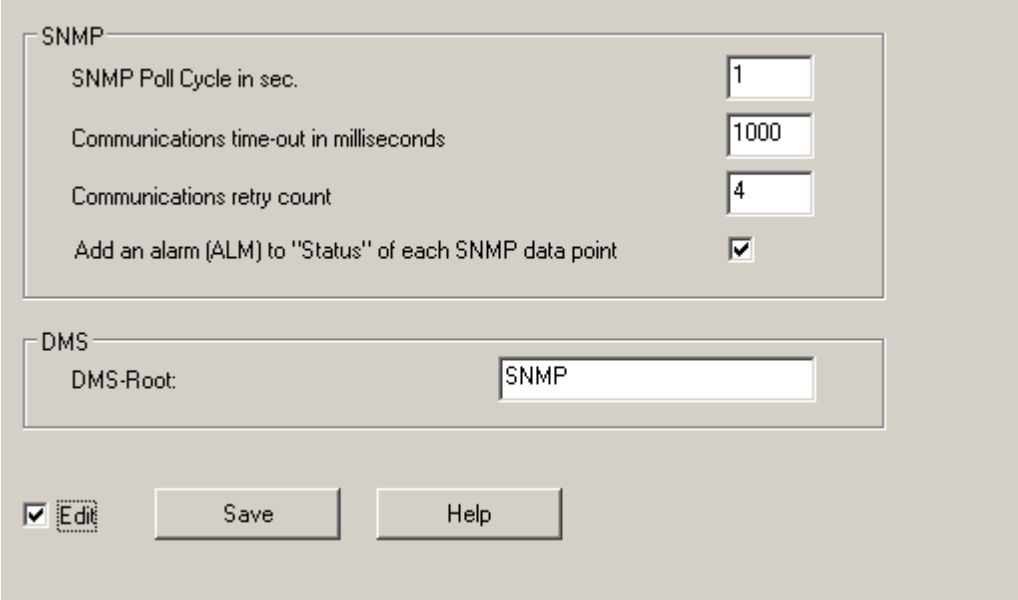
9.7.5 Monitoring of the SNMP Data Points

Each SNMP data point can be monitored with an alarm which is attached to its status. For this purpose, on the "Status" data point (this has any SNMP-DP) is an ALM add-on appended:



All the ALM add-ons will be appended by SNMPDriver during the startup automatically (if they aren't already), if following configuration option is selected:

- in the configuration pane the check box "**Add an alarm (ALM) to "Status" of each SNMP data point**"

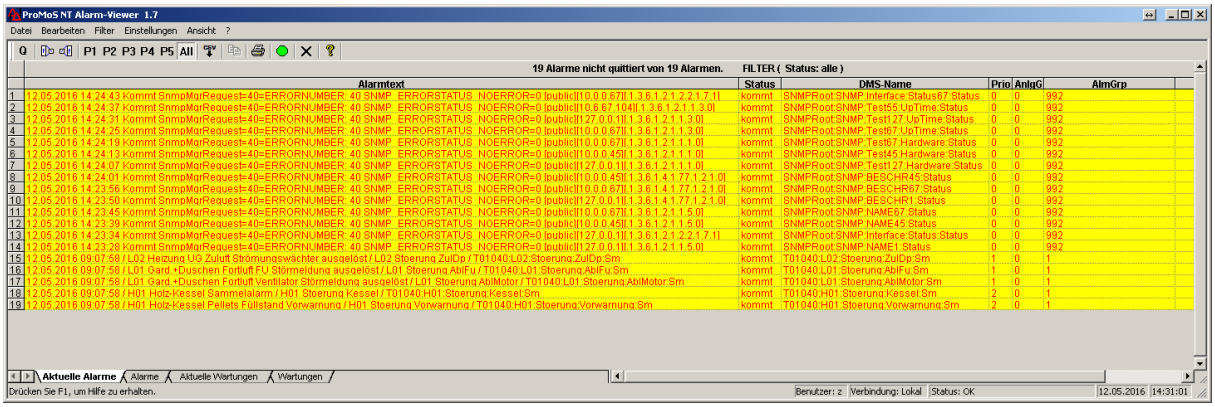


The settings are saved in **snmpdriver.cfg**.

- If activated, at SNMPDriver startup time there will be automatically to each SNMP "**Status**" DP an ALM add-on appended.
- Following ALM DP's will be with these default values fixed predefined (hardcoded):
  - Status:ALM:Alarm1:**Activ**=1
  - **DBGRP**=0
  - **Prot**="ALARM"
  - **Type**=">="
  - **Value**=1.000
- The default values of the other ALM DP's can be configured in <proj>\cfg \snmpdriver.cfg **[SNMP_ALM]** section under following keywords:
  - **Group**=992
  - **Pri**=0
  - **Text**=SNMP
  - **Format**=AlarmSNMP@#c #Z(Kommt:Geht:Quit) #T #V^LastMsg

The default values will be automatically written into the snmpdriver.cfg if they do not exist yet.

In AlarmView then the SNMP alarms are displayed as follows:



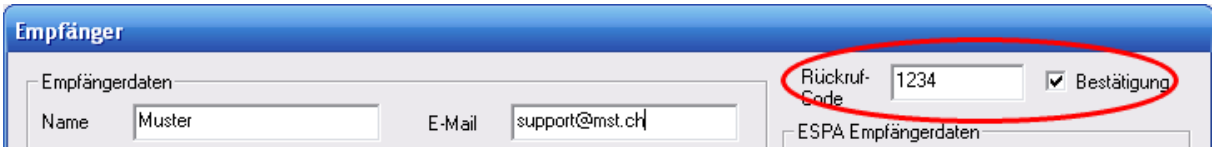
9.8 The TAPI Driver (tapidriver.exe)

Available in Version 1.5 or higher.

TAPI = Telephone Application Interface

The TAPI Driver accepts telephone calls and, after successful verification of the caller (code input check), can acknowledge all alarms that were sent via MalmMng (SMS, e-mail, pager, ESPA, Tel.Voice).

The caller must be defined as a user (=receiver) in MalmCfg and the callback information must be entered in their recipient data (3 to 10 digits), see Operating programs: "**The MAlm Configurator (MalmCfg.exe)**":

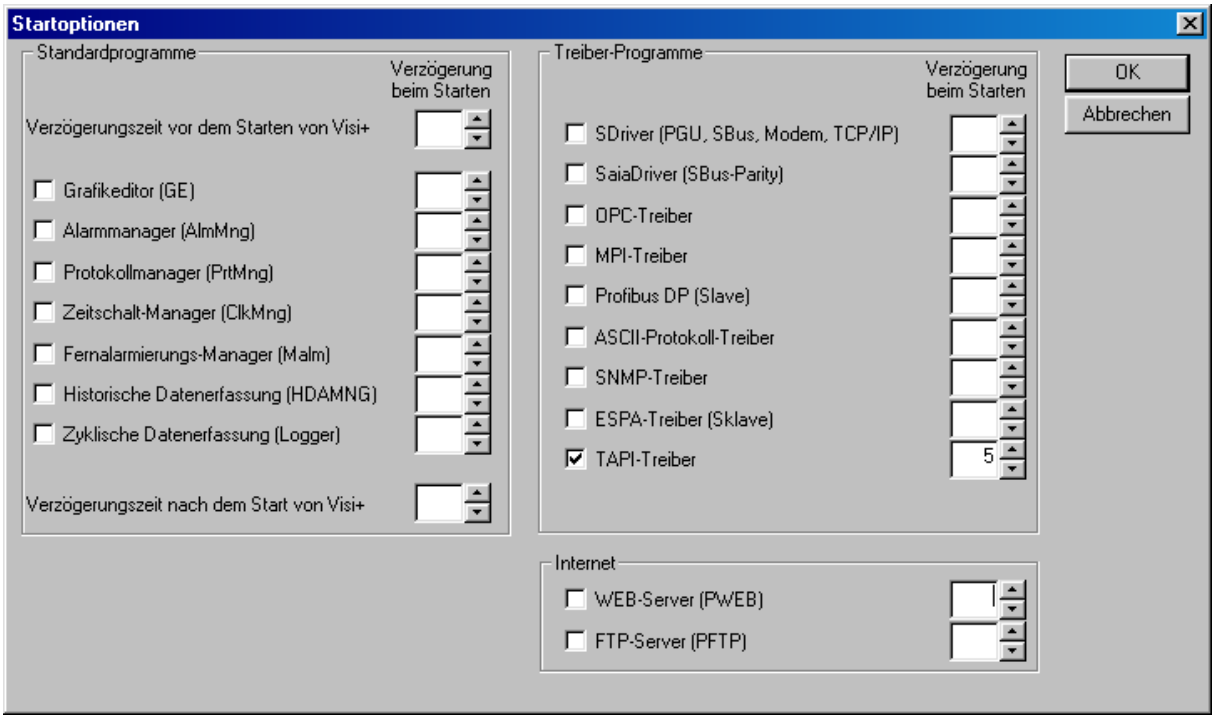


9.8.1 Starting the program "TAPIDriver"

The **TAPIDriver** can be started at any time by double-clicking on the icon in a file manager program. The filename is "**TAPIdriver.exe**" in the default installation directory **C:*Visi.Plus*** ***bin***.

Of course, your own shortcut created on the desktop is even more convenient. Alternatively, the TAPIDriver can be started automatically through the **Project Manager** module, where the programs to be started for each project can be selected. To do so, proceed as follows:

1. Start the **Project Manager** module.
2. Select the menu item or the **<Start Options>** button.
3. Mark the checkbox **TAPI Driver (Slave)**.
4. Confirm with **<OK>.**



If the TAPIDriver has been started, the icon appears on the Windows task bar (bottom right corner of the screen).

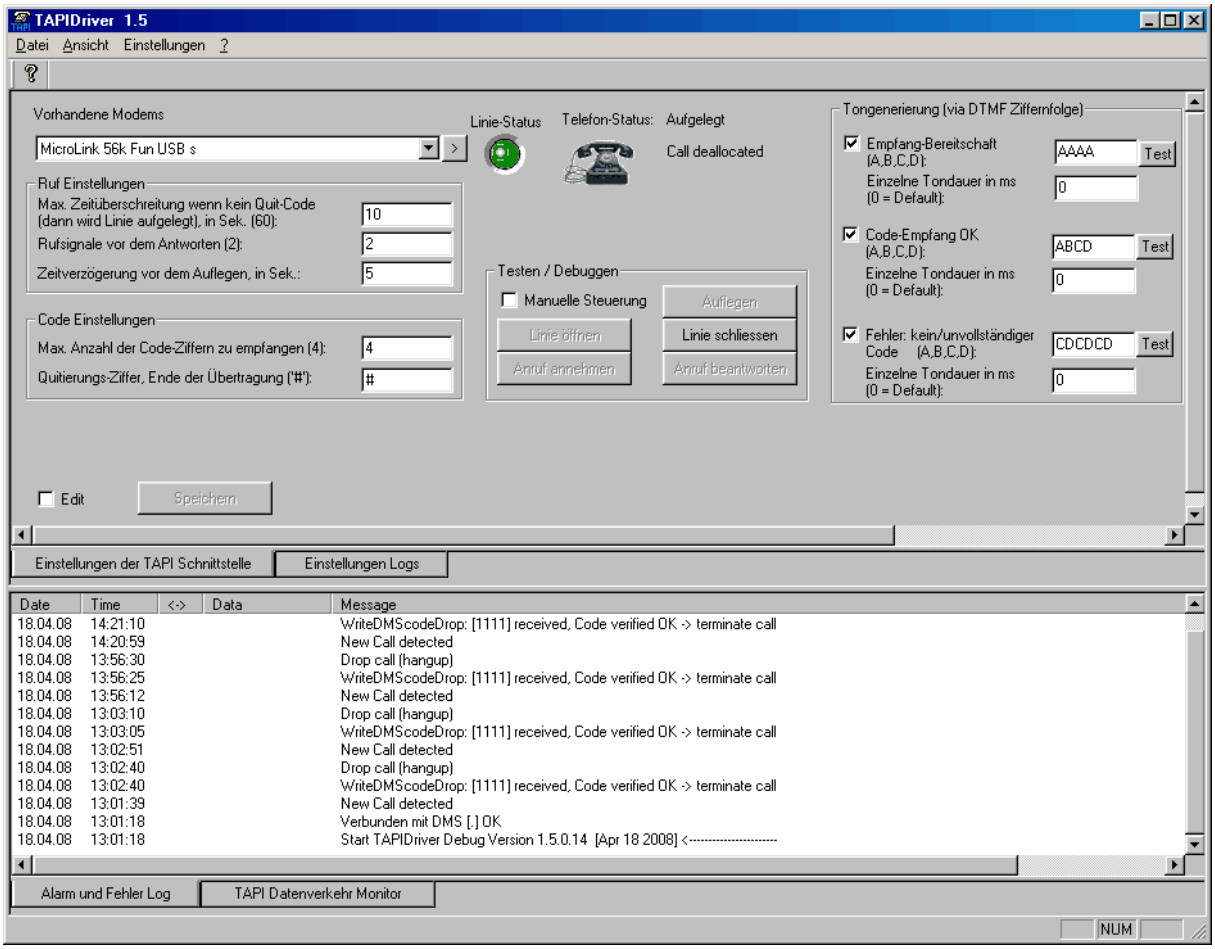
The module normally runs as a background process.

If the TAPIDriver is not started, no calls can be received and alarms sent via telephone cannot be acknowledged.

9.8.2 The TAPI Driver user interface

With the TAPIDriver loaded, the user interface shown below can be opened as follows:

Double-click with the left mouse button on the ESPADriver icon on the Windows taskbar at the buttom of the screen.



Title bar
Contains the name of the current module.

Menu bar
The menus are explained in the following.

Display window
The main window is composed of 2 panes:

- Top: configuration and status display tab window for TAPI interface and log settings
- Bottom: alarm and error log and TAPI monitoring window

9.8.3 The TAPIDriver menus

9.8.3.1 File menu

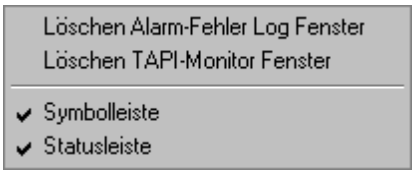
The "**File**" menu contains the following functions:

Exit
Closes the current window and removes the driver from the computer's main memory. The driver is then no longer active.

Close Window
Minimises the control window of the **TAPIDriver** to the taskbar. The driver is not exited by doing so.

9.8.3.2 View menu

In this menu the bars at the top and bottom edge of the TAPIDriver window can be switched on and off. Log and monitor windows can also be deleted.



Delete Alarm Error Log window
Deletes the log list.

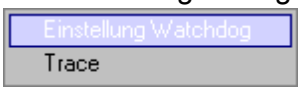
Delete ESPA Monitor window
Deletes the TAPI data traffic monitor list.

Toolbar
Switches the toolbar on or off.

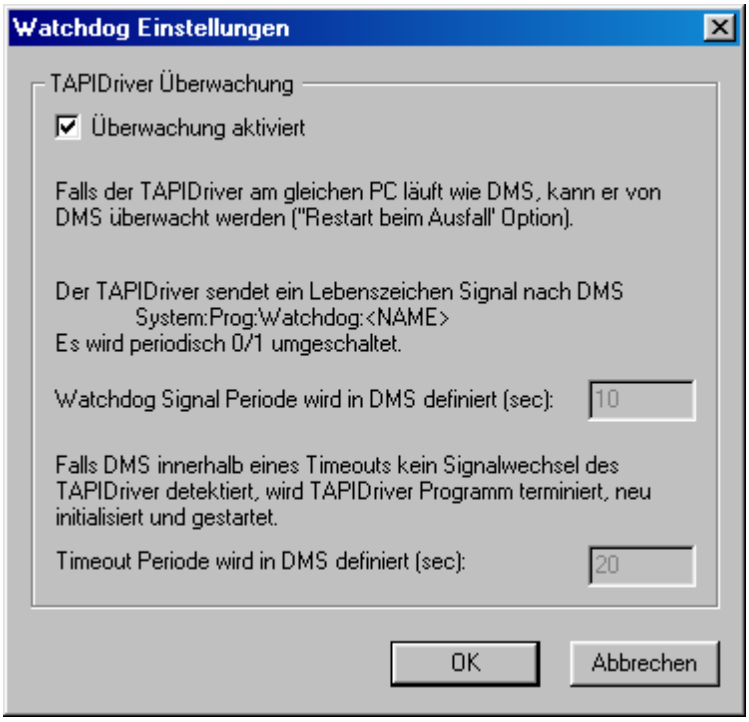
Status bar
Switches the status bar on or off.

9.8.3.3 Settings menu

The Watchdog Settings and Debug Settings are available in the "**Settings**" menu.



Watchdog Settings
The TAPIDriver can be monitored by the Data Management System (DMS). The TAPIDriver and the DMS exchange a live bit upon activation of the function.

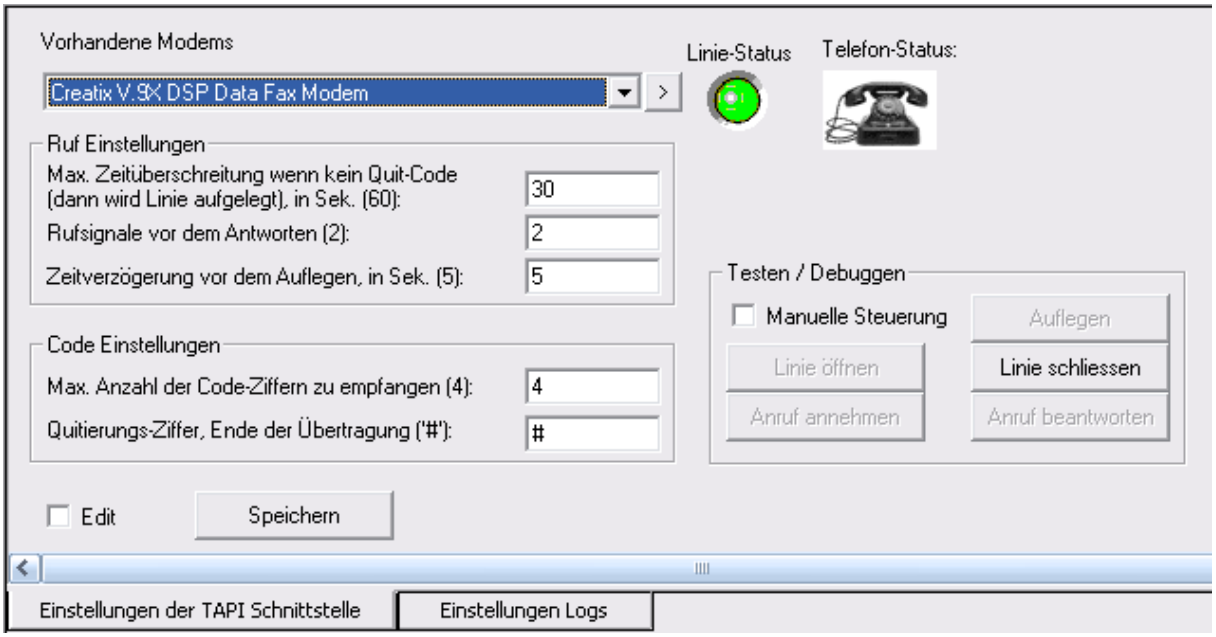


Trace Settings
You can define whether actions of the TAPIDriver are logged in detail in the special Trace.exe program.
Trace.exe is located in the default installation directory **C:*Visi.Plus**bin*** and must be started manually.

9.8.4 Settings and configuration

9.8.4.1 Configuration of the TAPI Driver

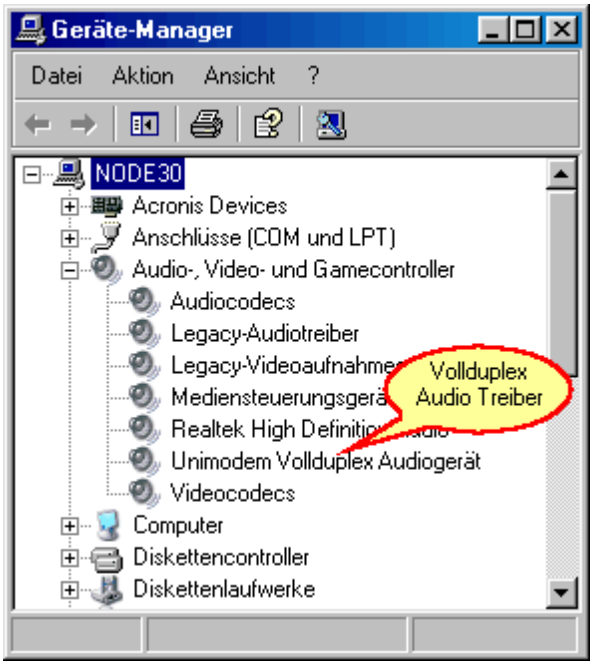
All data point definitions can be fully configured in the upper tab window "TAPI Interface Settings".



Modem Selection
The TAPI Driver only works with voice modems. Only voice modems are available for selection in the combo box. If modems are newly connected or removed for the runtime of the TAPI Driver, the "**>**" button (on the far right) rescans all modems into the list.

After a modem has been selected, it is immediately connected with the TAPI Driver and the result of this action is displayed as a line status:
- Green: Connection OK
- Red: Connection not possible (in voice mode)

Attention!
If the TAPI driver does not function correctly with a green LED (no telephone acknowledgement possible), then an a required modem property must be checked in the system device manager (Control Panel > System > Hardware > Device Manager):



Call Settings

- Max. time exceeded if no acknowledgment code is provided (then line is hung up), in sec. (60): if no user input takes place within this time (acknowledgment key code), the call is hung up.
- Rings before answering (2): the line is only picked up after reaching the set number of rings.
- Time delay before hanging up in sec. (5): after successful acknowledgment the line is only hung up after this period has elapsed.

Code Settings

- Max. number of code digits to receive (4): = max. length of the acknowledgment code to be received (can also have fewer digits). If more key digits are received, the line is hung up.
- Acknowledge digit, end of the transmission ('#"): the caller can terminate the call with this digit; digits entered up to this point were interpreted as part of the acknowledgment code.

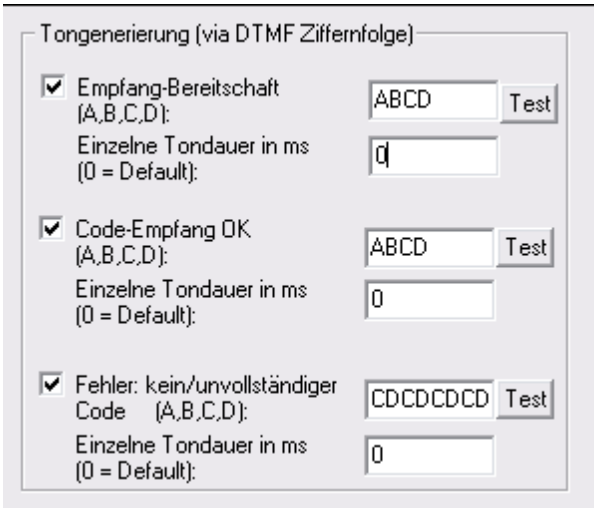
Testing/debugging

The modem/the line can be connected manually with the following switch functions:

- Manual Control: when activated, the line is no longer picked up or hung up automatically.
- Hang up: The line is disconnected, the call terminated and the line is ready for calling.
- Open Line: The selected modem is initialised and can be called.
- Close Line: The connection to the current modem is ended, calling no longer possible.
- Accept Call: Connect incoming call, pick up telephone. Depending on modem type, same function as: Answer Call.
- Answer Call: Connect incoming call, pick up telephone. Depending on modem type, same function as: Accept Call.

Tone Generation (via DTMF digit sequence)

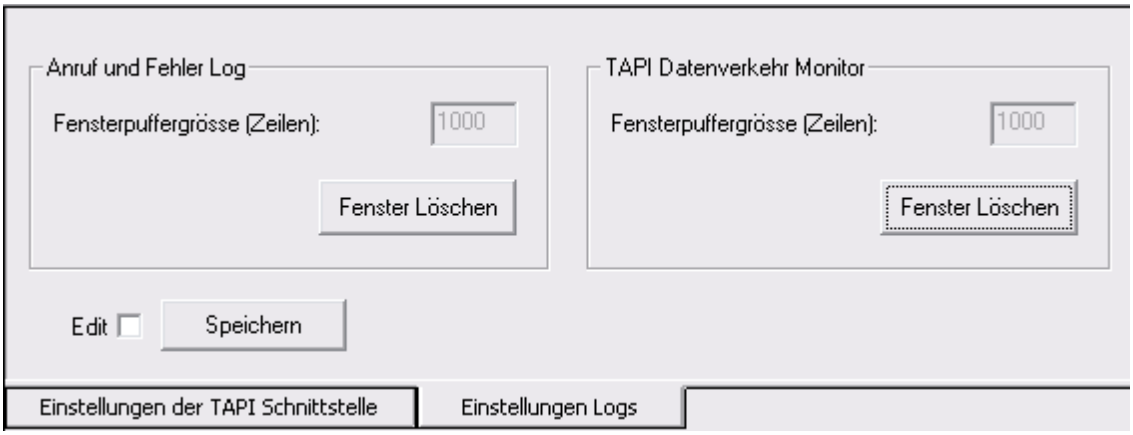
When picking up or before hanging up, the following audible signals can be generated as a response for the caller.
Only the tone of the keys A, B, C, D can be used to generate the signals.
For most modems the tone length cannot be freely determined; in this case the tone length value is ignored.



- Ready to Receive: After this signal (call picked up) the caller can begin inputting the code.
- Code Received OK: indicates that the acknowledgment code was received and correctly identified.
- Either no code or an incomplete code was received (time exceeded) or the code entered is not correct, meaning no receiver is defined with this code in MalmCfg.

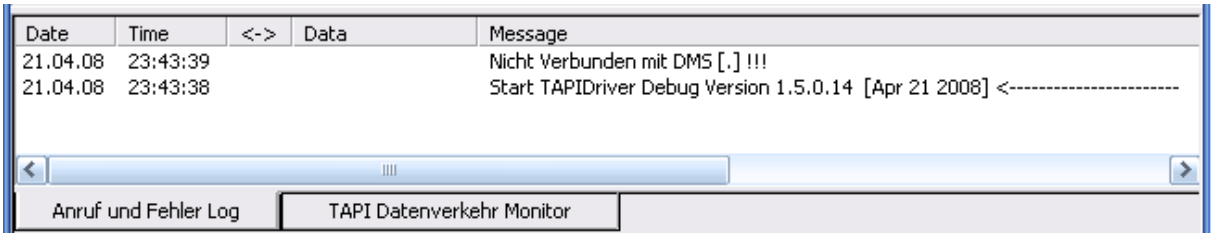
9.8.4.2 Settings for the log entries

The settings are stored in the file TAPIDRIVER.CFG in the project's \CFG directory.



Call and Error Log List

All calls, code entries as well as line errors are logged here.



TAPI Data Traffic Monitor

Significant line events are logged here.
This is useful for incoming error analysis.

Saia Visi.Plus

Operating programs

Chapter 10

10 Operating programs

10.1 Summary

This chapter deals with the user interfaces (operating programs) of Visi.Plus.

Along with the Graphical Editor (GE), there are several program modules that allow you to solve the various tasks of visualisation.

This gives the whole system a sense of clarity and user-friendliness. Therefore, if a function is improved or further developed, only the software module in question needs to be updated. This applies to the extent that all modules of Visi.Plus can be updated to the latest version over the internet, at the press of a button.

Visi.Plus has the following control modules:

Graphical Editor (GE)
For the user this is the actual user interface in execution mode and for the programmer this is the design tool for process and system images.

Alarm Viewer (AlmView.exe)
The Alarm Viewer shows all alarms in tabular form.

Log Viewer (Prtview.exe)
Recorded logs can be displayed with the Log Viewer.

pList (pList.exe)
The DMS data can be managed and edited with pList.

Time Switch Program (ClkCfg.exe)
Switches can be configured on a time-dependent basis with the Time Switch Program.

Graph diagrams (pChart.exe)
This module can display, edit and export graph diagrams.

User Management (pUser.exe)
Here you can create users and define their rights.

Password Change (changepwd.exe)
This module serves for changing user passwords.

Remote Alarm Program (MalmCfg.exe)/(older version MalmConfig.exe)
With MalmCfg, remote alarms and their recipients can be configured.

ASCII Export
The DMS data can be exported to an ASCII file with ASCII Export.

Text Editor (pEdit.exe)
Text can be freely composed with this editor.

Login (pLogin.exe)
Opens the Login window of the visualisation.

IMPORTANT:
In this chapter the demo project "Promos" is used in order to familiarise you on an experimental basis with the functions of the GE. We recommend saving a copy of this project, because when switching from edit mode to runtime mode the respective process diagrams must be saved and the project is changed as a result.

10.2 Object editor (oList.exe)

oList.exe is a debugging tool that can be used for editing all project data in a list mode. The displayed list can also be limited by using the filter function on the DMS name.

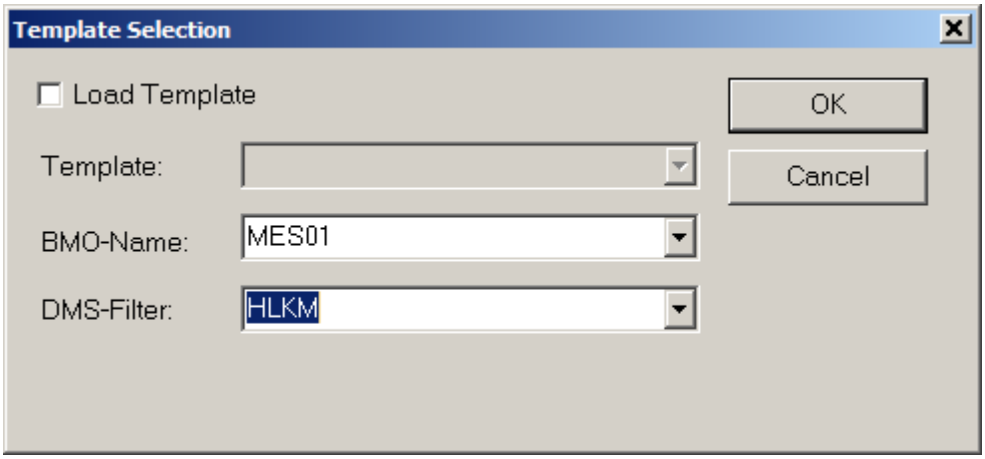
10.2.1 Starting the program

oList offers a list view of all objects in the project. The use of oList in projects which only contain the objects DIGI01 and ANA01 does not make much sense.
The main application is for projects which have been created on the basis of DDC Suite or Code Generator.
oList can only used if a user is logged in. Manipulating datapoints is only possible with administrator rights.

***All** data in Visi.Plus can be edited with oList (thresholds, alarms, etc.). Therefore the following information is important:*
If a button that starts oList is created in the Graphical Editor, it must be ensured that the button can only be selected if the user has corresponding rights.
***Careless handling can cause data of security relevance to be changed in oList**.*

10.2.2 The oList User Interface

When starting oList, the DMS data to be edited must be selected if this was not already given as a parameter on startup.

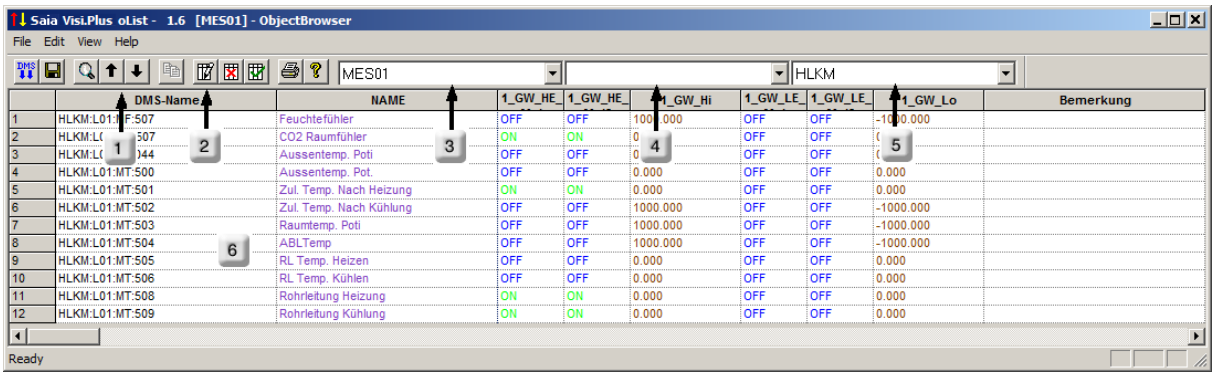


Template:
By clicking on the dropdown menu, all templates are shown.
Templates offer the opportunity to display a user-specific selection, e.g. only the alarm texts of an object or the target values.

BMO Name:
Selection of the respective BMO Object

DMS Filter
The list can be narrowed down by specifying a part of the DMS name. The PCD stations are suggested by default.

The oList Window Contents



1	**Search / Sort**	Create new search / sort by column (select column beforehand)
2	**Column Functions**	Reload columns from DMS, delete or add columns
3	**Object Selection**	Selection of the template objects (BMO)
4	**Template Selection**	Selection of a personalised template
5	**Filter Selection**	Selection of the filter (PCD selection)

6	**Data Field**	Data are changed directly in the field and are written to the DMS. Caution: when the S-Driver is running, the data are also sent directly to the PLC!

oList *is particularly well suited to making project configurations with a good overview. Target values, delay times and names can be changed simply and conveniently.*

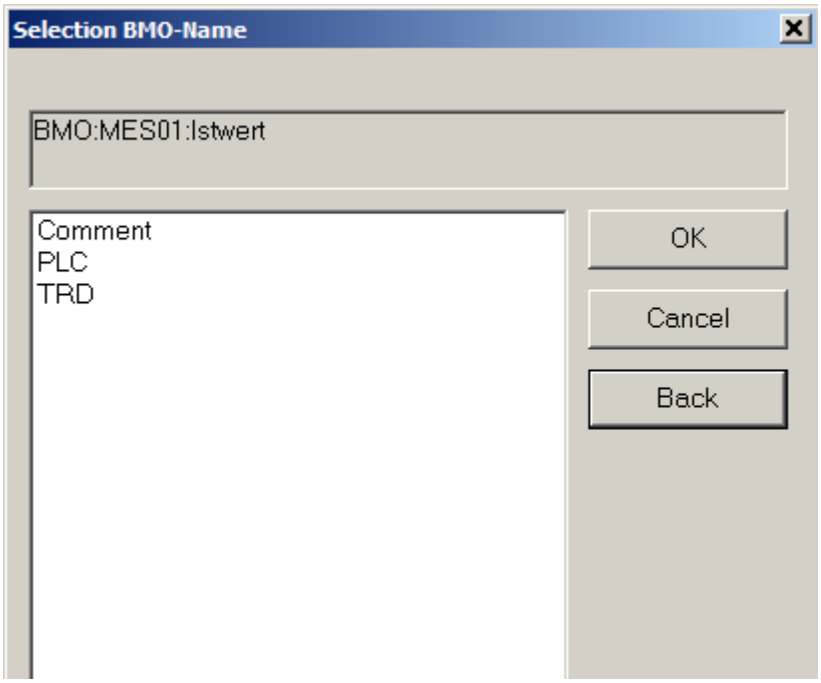
10.2.3 Inserting new columns

A great advantage of oList is that the DMS data to be edited can be selected freely, right down to every data point in the object.

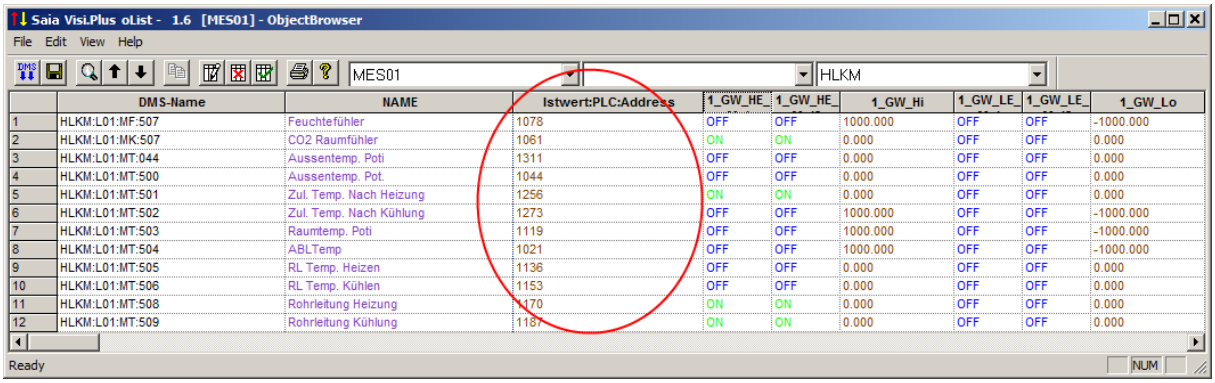
Delete column:
The active column is deleted when this button is clicked.

Insert column:
A new column is inserted when this button is clicked.

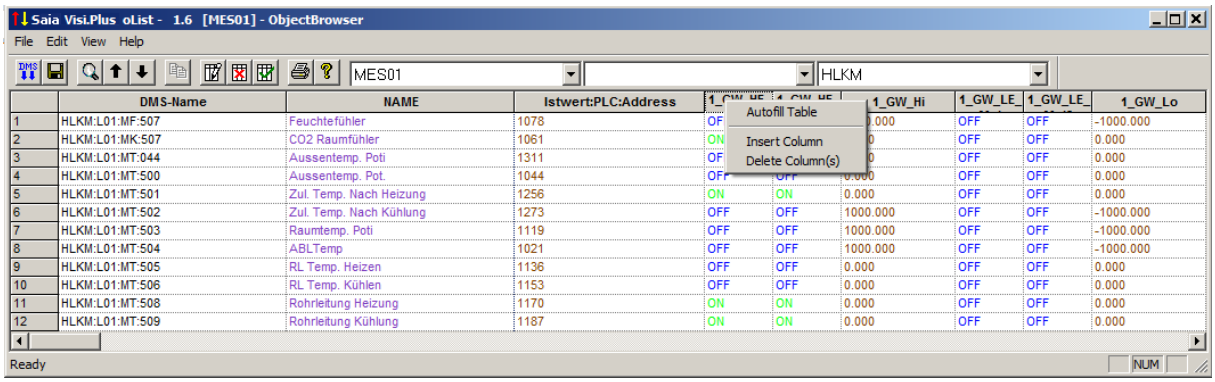
Select the data point in the usual manner:



This means it is possible, for example, to display the address of the register in the PCD where this target value is stored.

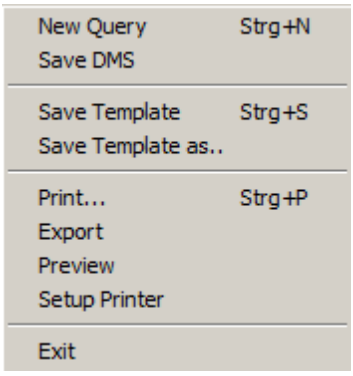


The actions Insert/Delete can also be performed through the context menu using a right click on the column header. This causes the selection to be started already in the subgroup of the corresponding DMS data point.



10.2.4 File menu

The "**File**" menu provides various options for opening, logging in, filtering and exiting the object editor (oList).



10.2.4.1 New Query <CTRL+N>

 In order to start a new query, select the command "**File > New Query**" or press the key combination **<CTRL+N>**.

In order to be able to start a new query, the user must have administrator rights.

10.2.4.2 Save

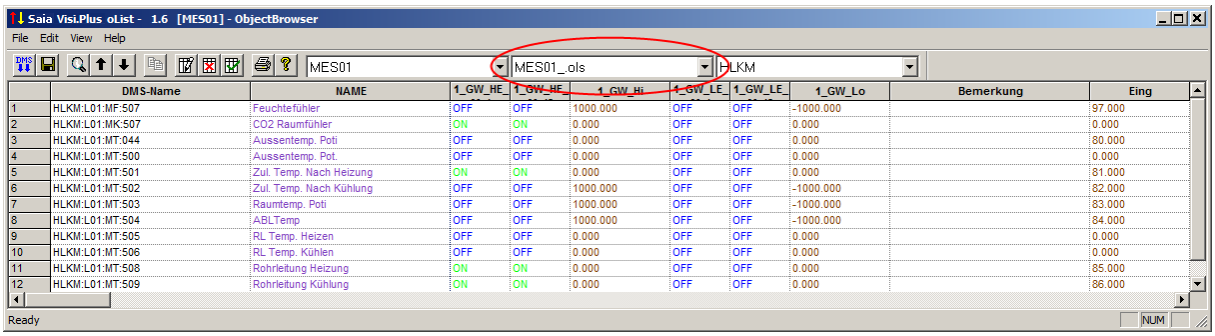
In order to save the project data edited in the DMS (in the computer's main memory), use the "**Save**" command. All values in the DMS are saved to the hard disk.

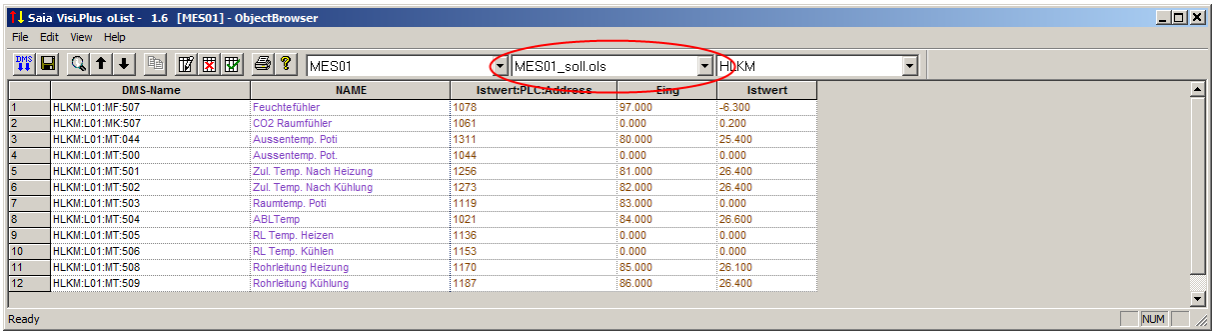
*If oList is exited **without saving**, the edited data points are **not** saved.*

As an alternative, the DMS save button can also be used 

10.2.4.3 Save template <CTRL+S>

The selection of the columns can be saved in the template, which makes it possible to create simple and clear lists that can be used to change and control selected parameters via identical objects.
Several templates are possible per object; the selection is made in the drop-down menu.

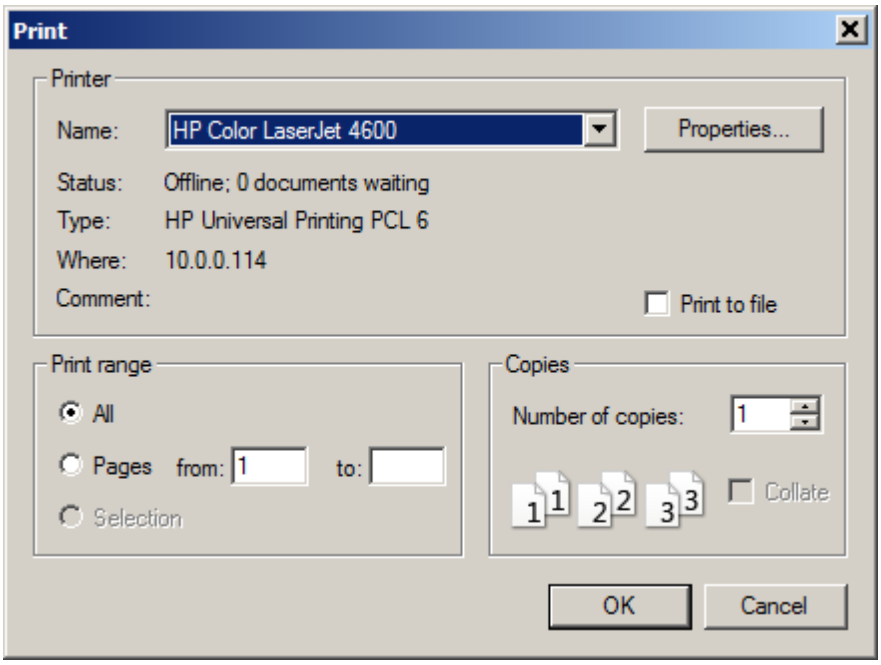




This is a PC operating system control window and should be used accordingly.

10.2.4.4 Print... <CTRL+P>

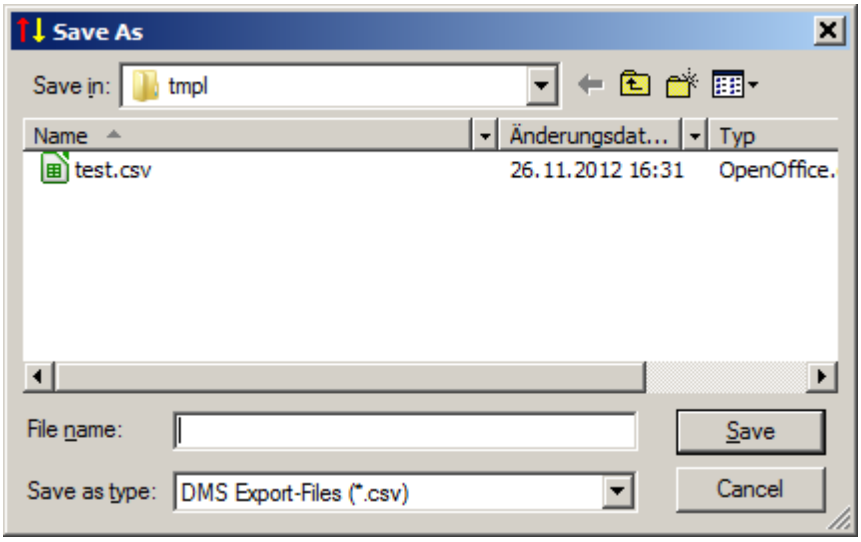
In order to print out the data displayed in oList, click on the  icon on the toolbar. Alternatively, select the "**Print**" command in the **"File"** menu or press the key combination **<CTRL+P>**.



This is a PC operating system control window and should be used accordingly.

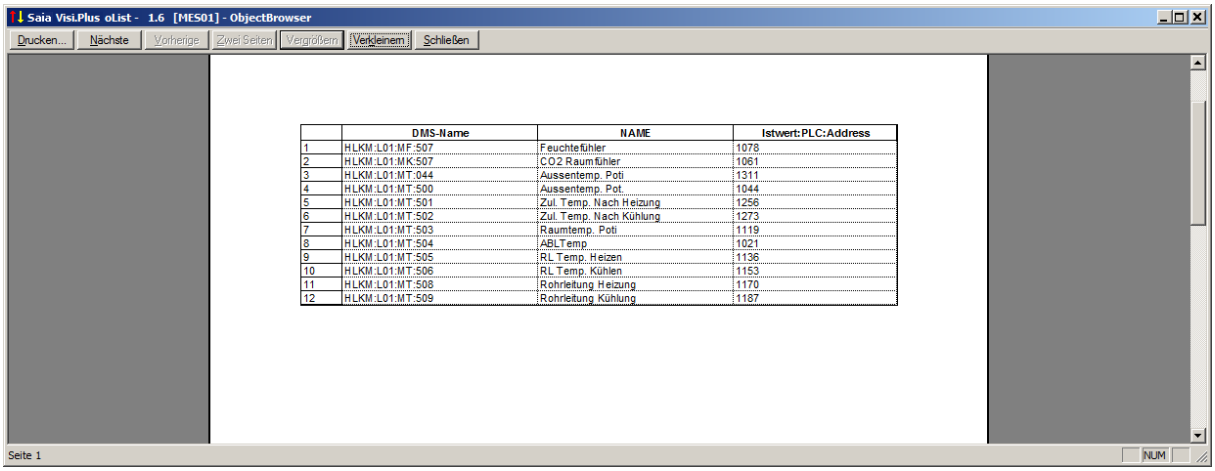
10.2.4.5 Export

With this function the o**List list** can be saved to a file. The data is stored in **CSV table format** and can thus be viewed and processed further with OpenOffice, Excel, any text editor or a similar program.



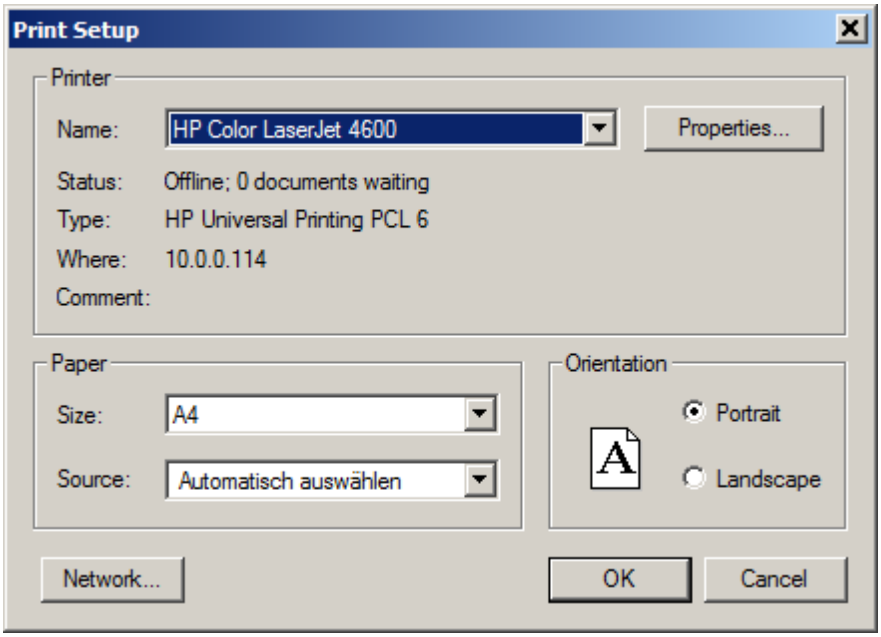
10.2.4.6 Print Preview

Before an oList list is to be printed, it makes sense to preview the printout.
This can be achieved with the "**Print Preview**" command. This may be helpful for adjusting and optimising the margins, for example.



	DMS-Name	NAME	Istwert:PLC:Address
1	HLKM:L01:MF:507	Feuchtefühler	1078
2	HLKM:L01:MK:507	CO2 Raumfühler	1061
3	HLKM:L01:MT:044	Aussentemp. Poti	1311
4	HLKM:L01:MT:500	Aussentemp. Pot.	1044
5	HLKM:L01:MT:501	Zul. Temp. Nach Heizung	1256
6	HLKM:L01:MT:502	Zul. Temp. Nach Kühlung	1273
7	HLKM:L01:MT:503	Raumtemp. Poti	1119
8	HLKM:L01:MT:504	ABLTemp	1021
9	HLKM:L01:MT:505	RL Temp. Heizen	1136
10	HLKM:L01:MT:506	RL Temp. Kühlen	1153
11	HLKM:L01:MT:508	Rohrleitung Heizung	1170
12	HLKM:L01:MT:509	Rohrleitung Kühlung	1187

10.2.4.7 Printer Settings



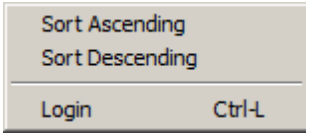
This is a PC operating system control window and should be used accordingly.

A network printer can be added by clicking on the **<Network...>** button.

10.2.4.8 Exit

This menu item exits oList.

10.2.5 Edit menu



10.2.5.1 Sort Up

 Sorts the displayed list in ascending order based on the middle column.

10.2.5.2 Sort Down

 Sorts the displayed list in descending order based on the middle column.

10.2.5.3 User Login <CTRL+L>

The window for user login appears.

10.2.5.4 Additional functions via keyboard

Copy <CTRL+C>

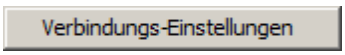
 Copies a selected oList range to the clipboard.

Paste <CTRL+V>

Inserts the contents of the clipboard at the cursor position.

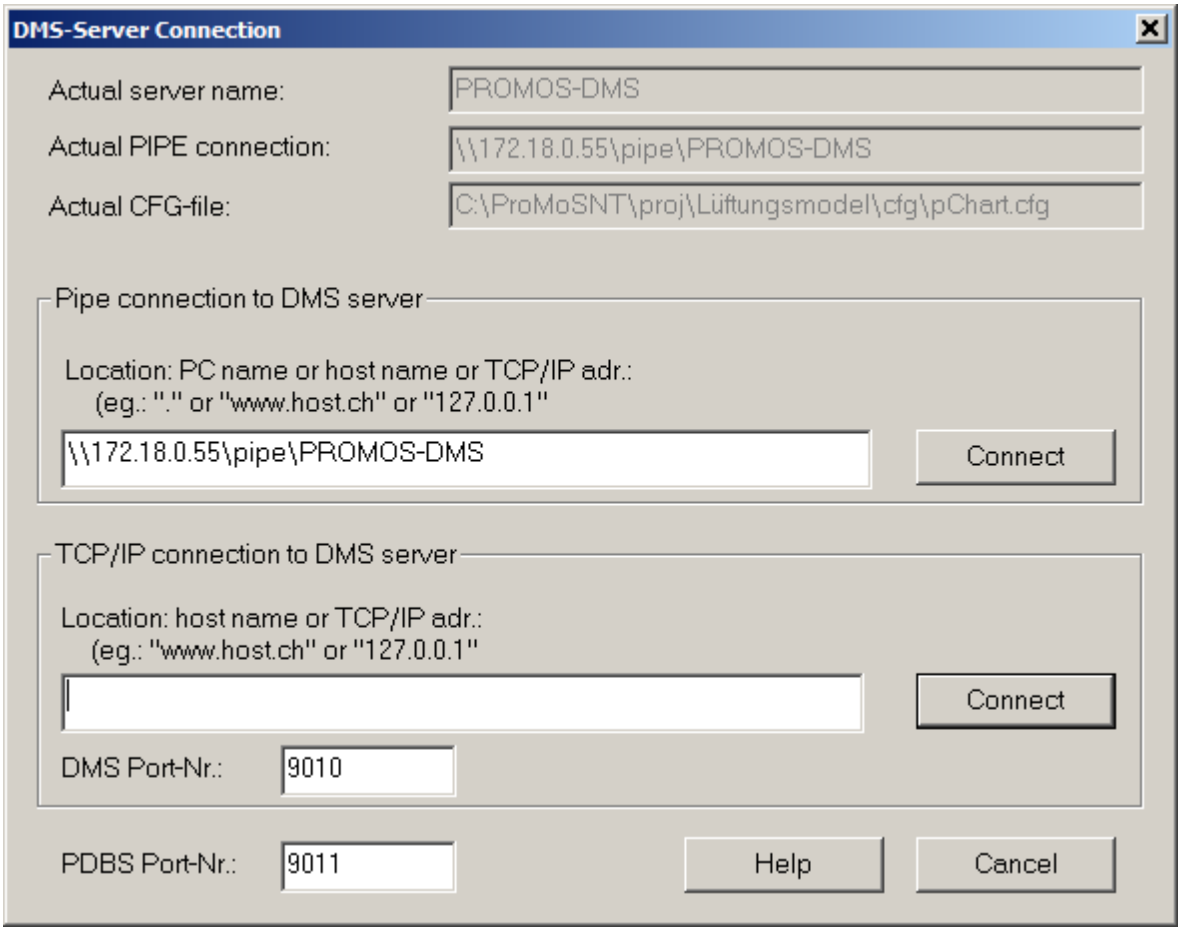
 *To sort a pList list according to the columns "**Resource**" or "**DMS Name**", double-click on the respective column or header.*

10.2.6 Settings menu



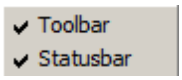
10.2.6.1 Connection Settings

oList can be connected to the DMS via a network.
The settings must match those of the ports from the DMS.



10.2.7 View menu

The toolbar and status bar can be switched on or off in this menu.



10.2.7.1 Toolbar

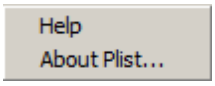
Switches the toolbar on or off.

10.2.7.2 Status bar

Switches the status bar on or off.

10.2.8 Help menu

The **Visi.Plus online help** and current information about the oList version are found in the "**?**" menu.



10.2.8.1 Help

The Visi.Plus online help is accessed via "**? > Help**" in the menu, via the  button or by pressing the **<F1>** key.

10.2.8.2 About oList

Indicates the version of oList being used.

Important
The version number must always be indicated in case of support queries.

We recommend our internet forum for support queries. This offers two advantages: First, help is available more quickly; and secondly, other users can benefit from the entries.

10.3 Graphical Editor (GE)

With the **Graphical Editor** (hereafter abbreviated as **GE**), process diagrams, application windows and screens are generated in programming mode (**edit mode**) and displayed during the runtime (**runtime mode**) as an operable frontend (visualisation).

How can we explain the two modes more precisely for the programmer or user?

Edit mode
In this mode, **process diagrams** are generated on a project-specific basis with the assistance of **graphical objects** like **lines**, **boxes**, etc. and distributed over multiple screen pages if necessary.

Control buttons
Buttons are placed in process diagrams for the user so that they can call up and manage specific pages or elements in the controller. Control buttons enable additional programs to be called up within Visi.Plus (e.g. pEdit.exe) as well as from outside of Visi.Plus (e.g. Windows Calculator).

Input Fields
Enable the user to perform parameter changes.

Not all users should have the same **rights**, so the programmer can lock entire screen pages or only individual input fields. For this purpose, it is possible to hide or show operating elements for certain users, depending on their rights.

With the GE's **versatile graphics capabilities**, the programmer can ensure the user will

recognise whether a pump is running or not. The programmer uses so-called **icons** for this purpose, which can be displayed by a data point or alternatively by the GE. The GE also has this possibility for **colour changes, line lengths**, etc.

For example, the GE is so flexible that a vertical and/or horizontal fill level indicator can change its shape and position through a "**Box**" graphical object online depending on a DMS value.

The data required to display colour changes, icon changes, value displays, and value modifications is gathered from the DMS by the GE. In doing so, the programmer **initialises** the data (e.g. the colour of a line) to a DMS value, that is to say, to a flag in the controller. As a result the colour of the line changes with a status change of the flag.

This philosophy extends across all functions of the GE.

Runtime mode
It is not possible for the user or operator to **carry out modifications to the process diagram**. Buttons and input fields are made available to them in the process diagram by the programmer, depending on user rights. These are used by clicking with the **mouse or via the keyboard**.

It is preferable to enter **values** in the input fields with the **keyboard**. There is a special alternative for and changing values on **touch panels** (virtual keyboard).

For security purposes and because it is possible for different users to have different rights, a user authentication process (login) is provided.

10.3.1 Starting the Graphical Editor (GE)

There are three ways to start the Graphical Editor:

1. Right from the Windows Start menu:
Start -> Programs ->Saia Visi.Plus -> Graphical Editor

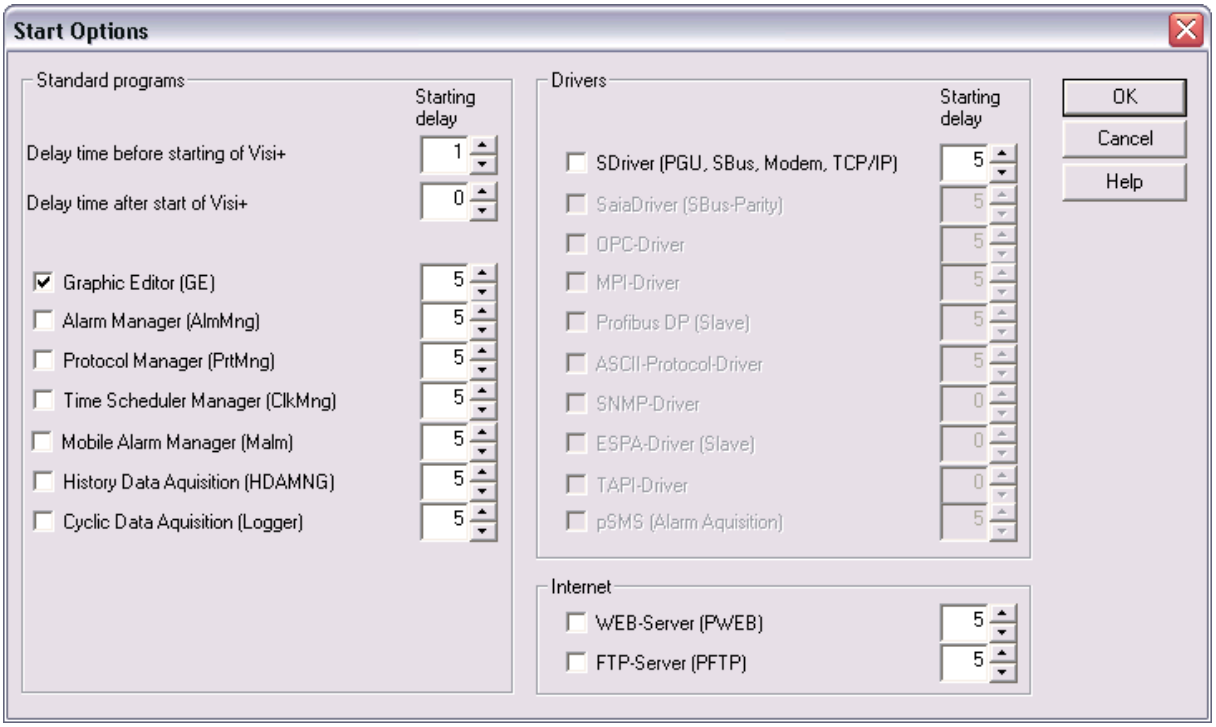
or

2. By opening a project in Visi.Plus

or

3. Proceed as follows to **start the GE automatically in a project**:

- Start the **Project Manager** module and click on the **[Start Options]** button.
- In the open **Start Options** window, mark the checkbox **Graphical Editor (GE)** in the Default Programs area.
- The next time the project starts, the **Graphical Editor (GE)** will automatically be started up along with it.



The fields with the label **Delay on Startup** allow the startup of programs to be delayed by particular periods of time. This is necessary so modules do not attempt to access data from other modules before they are loaded to the main memory of the computer. **Number** indicates how many graphical editors should be opened on startup.

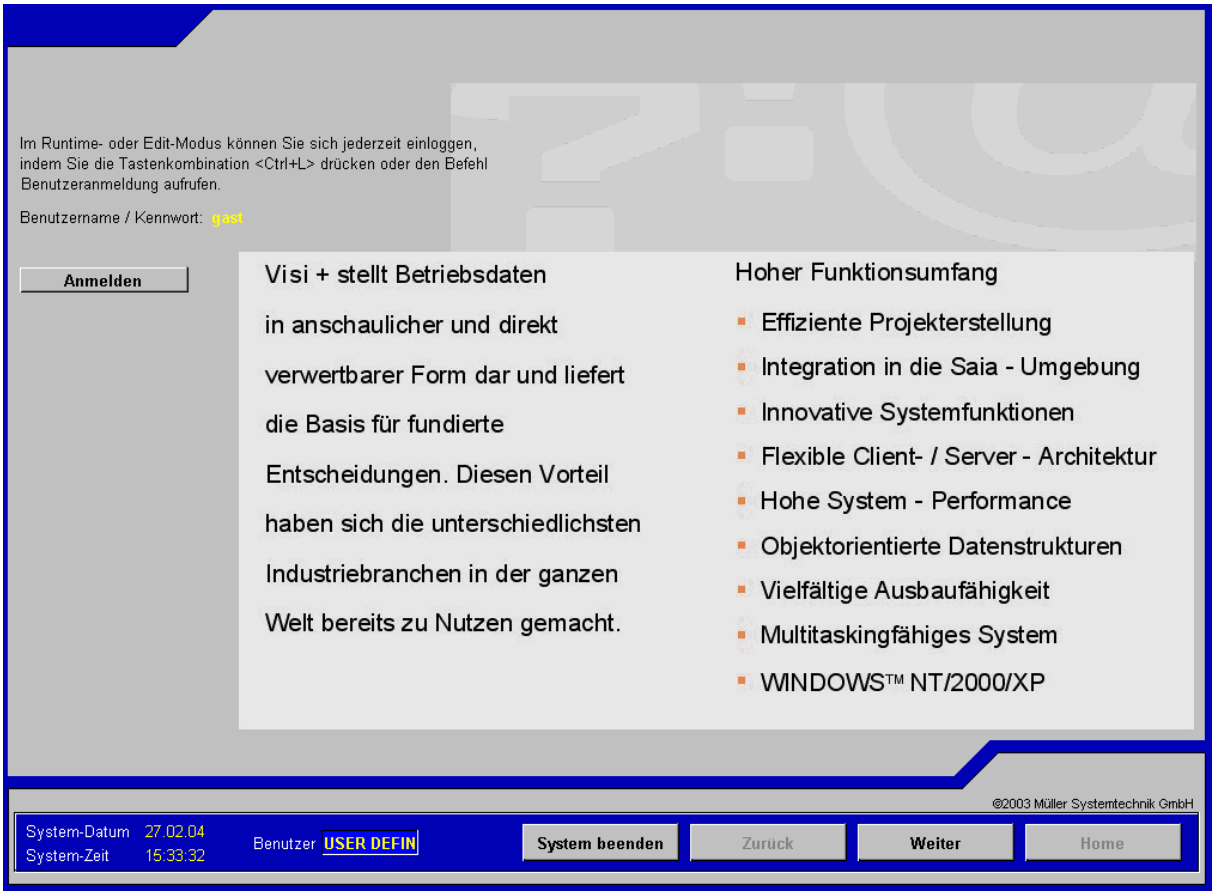
In the future, this will be automatically controlled by DMS.exe (if no module is selected here). It is also possible to swap these for your own entries.

As already described, the **Graphical Editor** (GE) recognises two different display modes:

- **Edit mode** and
- **Runtime mode**

The following descriptions are based on the assumption that Visi.Plus was just installed and the GE has been started for the first time as described above.

The **GE** presents a first control image as shown in the following image, but without menu/ control elements at the top and without the status bar at the bottom edge of the window.
This means that the Graphical Editor is in **Runtime mode** and is now showing the first operator page of the demo project **Promos**.



This view is an example of the status of an actual system for the machine or system operator.

If the instructions on the screen are followed, the GE shows multiple process diagrams with the help of the **<Back>**, **<More>** and **<Home>** buttons. In this case it is a brief introduction to the capabilities of Visi.Plus.

As already mentioned in the introduction, the GE can be switched to edit mode.

In the preceding image this is only possible by logging in with the **<Login>** button displayed in the process diagram and by entering the user "**guest**" and the password "**guest**" in the input fields.

More information about this protective feature, so-called rights, can be found under the chapter User management. With such rights, it is possible to block specific users from switching to edit mode.

Switching between edit/runtime mode
You can switch between these two modes at any time with the **<e>** key, as long as you have sufficient rights. Obviously no changes can be made to the process diagrams in **runtime mode**.

After selecting **edit mode**, additional menus, control elements and the status bars now appear at the top and bottom edges of the window.

*It does not matter which mode the GE is in. The process data (current data from the PLC) is displayed both in **edit** and in **runtime** mode in real time.*

In the following image the **GE** window is shown in **edit mode** with the demo project "Promos" loaded, which is included in the scope of delivery and is a brief introduction showing some of the capabilities of Visi.Plus.

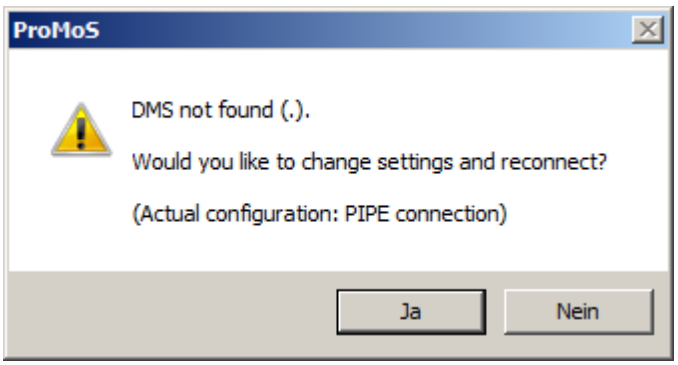
10.3.2 Starting a Remote Graphical Editor

The GE can be started even if the DMS is not running on the same PC as the graphical editor. To do so, a remote connection must be established.
The connection can be established via pipe or TCP/IP: Pipe connections are more reliable, however the ports are suppressed by many firewalls. It is recommended that a pipe connection be used for a local network. For a connection via the Internet, a TCP/IP connection must be established.

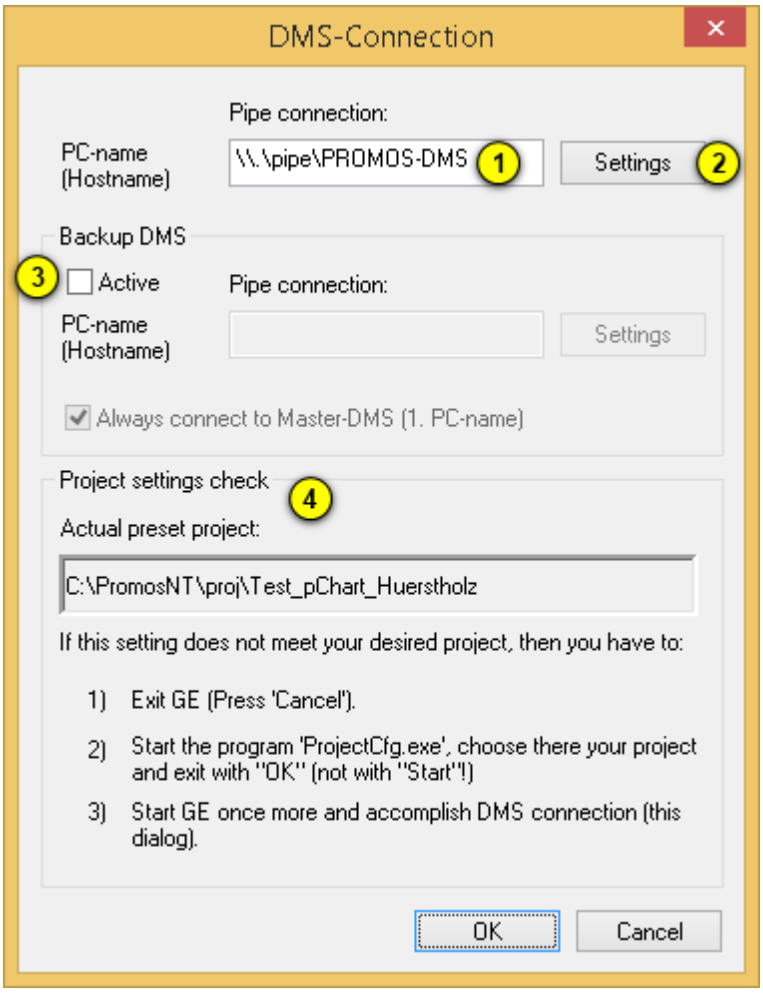
The project can be stored locally or in a network folder. If the project is stored locally on every PC, the corresponding data must be synchronised when a change is made to a process diagram in the GE. If the project is administrated from every PC via a common folder, it is possible that a delay may occur during a screen change in the GE, depending on the access speed of the network.

To start a remote GE, the project must first be selected in the ProjectCfg. Instead of Start, however, **<OK>** must be used to confirm.

Afterward, the GE can be started. The following information window will appear:



When the **<Yes>** button is actuated, the configuration window for the remote connection will be opened.

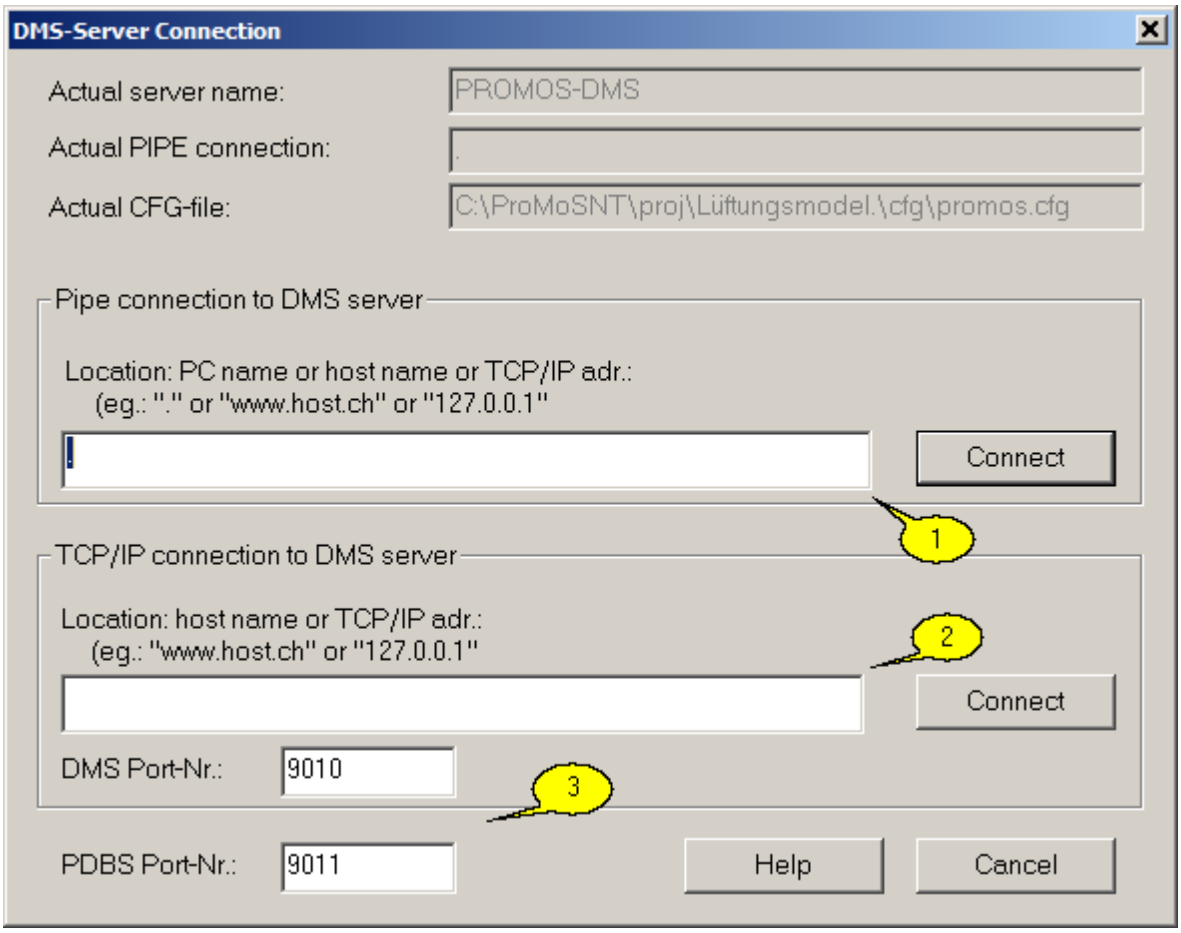


1) Location:
Name or IP address of the computer where the DMS is running. A dot is used here to refer to the local PC.

2) Settings:
Use this button to open the settings window. See the following image in this regard:

3) Backup DMS:
If the connection to the set DMS fails, the 2nd DMS takes over. At the moment, this is a quasi-redundancy. The two DMS do not exchange any data with each other.

4) Project settings monitoring:
The project path which has been set in the ProjectCfg is shown here.



1) Pipe connection to the DMS server
If a pipe connection is established, the IP address or name of the corresponding PC is shown here. Actuating the **<Connect>** button causes the GE to establish a pipe connection to the DMS. If the establishment of the connection fails after 3 tries, the attempt is aborted automatically.

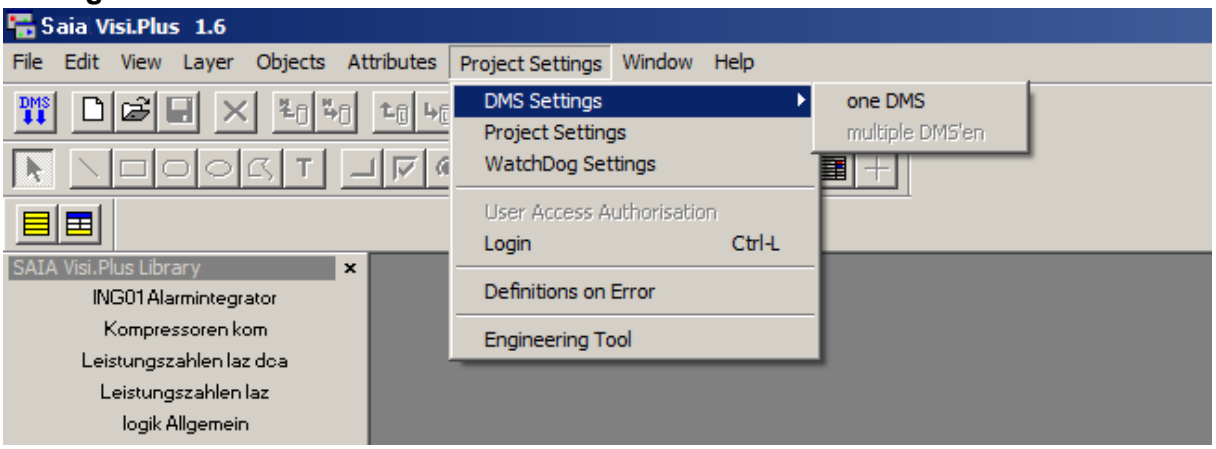
2) TCP/IP connection to the DMS server
If a TCP/IP connection is established, the IP address or name of the corresponding PC is shown here. Actuating the **<Connect>** button causes the GE to establish a TCP/IP connection to the DMS. If the establishment of the connection fails after 3 tries, the attempt is aborted automatically.

3) Port settings

The port number for the DMS as well as PDBS can be specified here. The default given is 9010 for the DMS and 9011 for the PDBS.

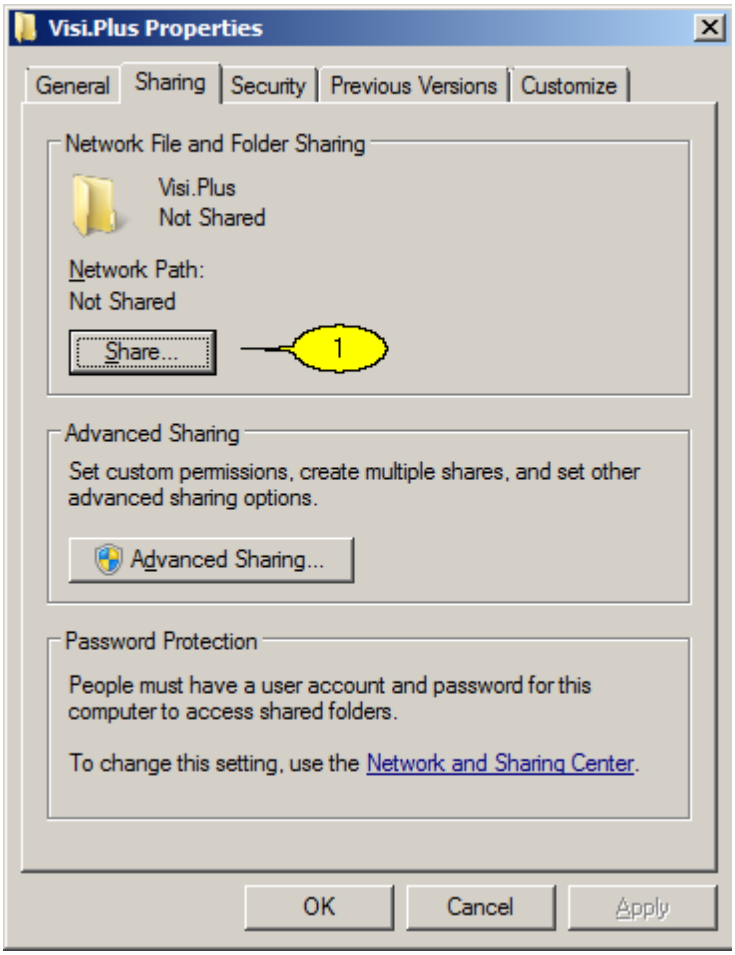
The dot- sign "." for the connection is only working for Pipe. With TCP/IP it won't work. For local connection under TCP/IP the setting localhost or IP-Address 127.0.0.1 is used.

The connection settings can also be made in the GE under **"Project Settings->DMS Settings"**.

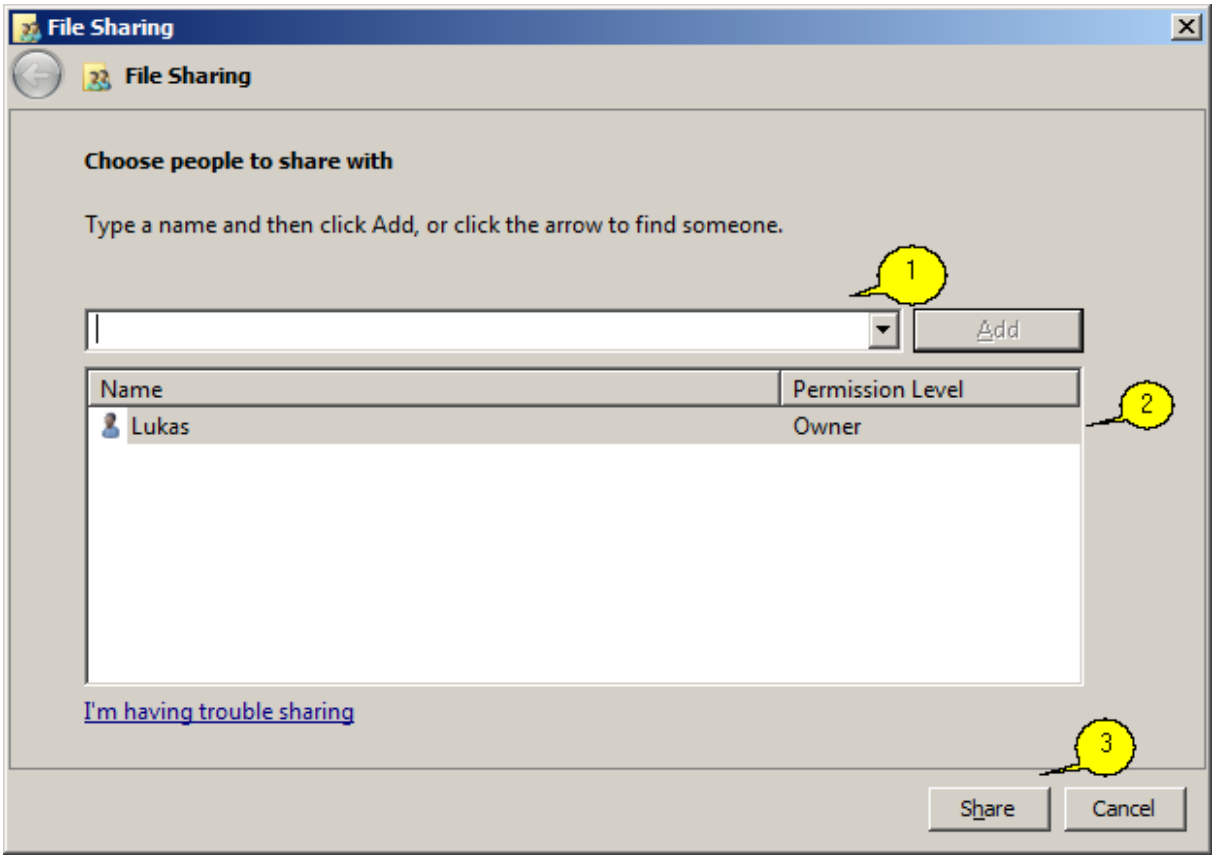


It might be necessary for the drive to be mapped on the master PC in order for the pipe connection to operate trouble free. Here are brief instructions for Windows 7:

First, the folder must be shared. To do so, click on Properties for the project folder with the right mouse button:



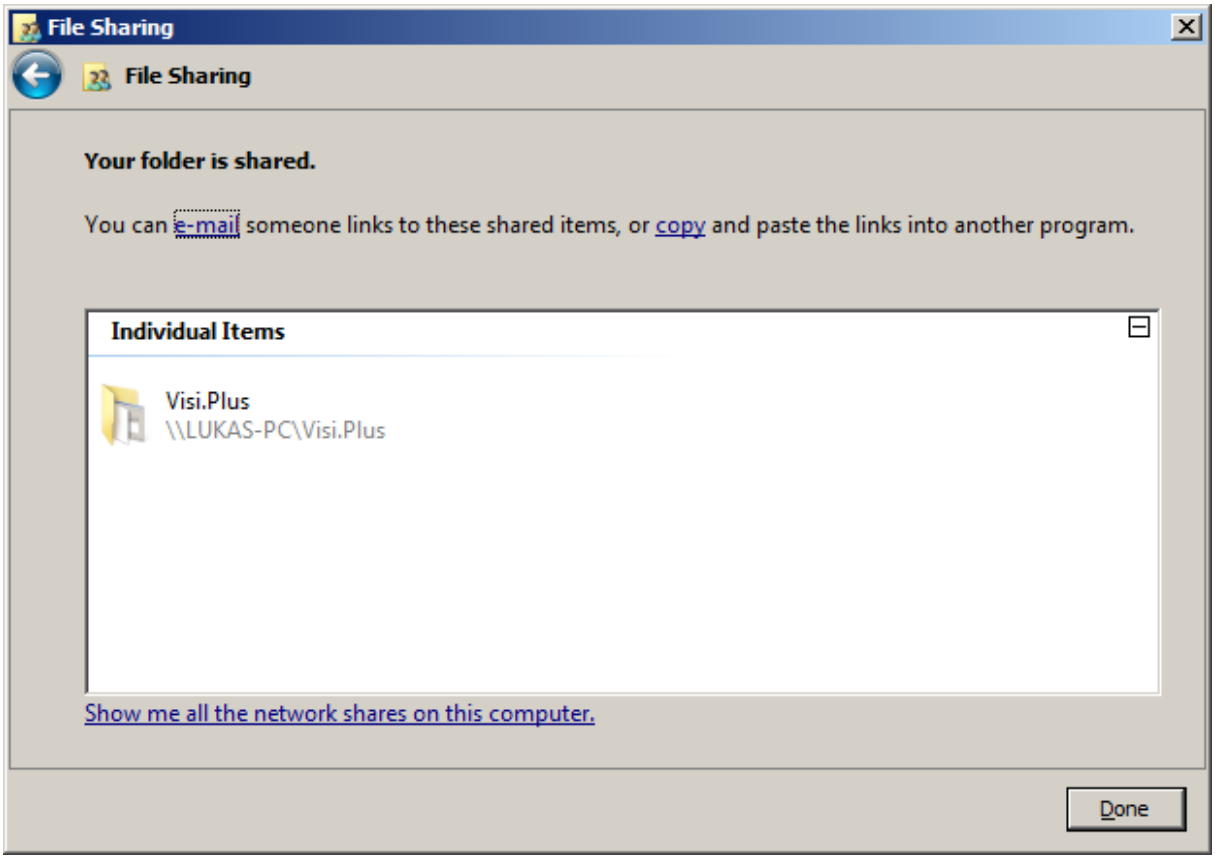
1) Release
When the **<Release...>** button is clicked, the file release window opens:



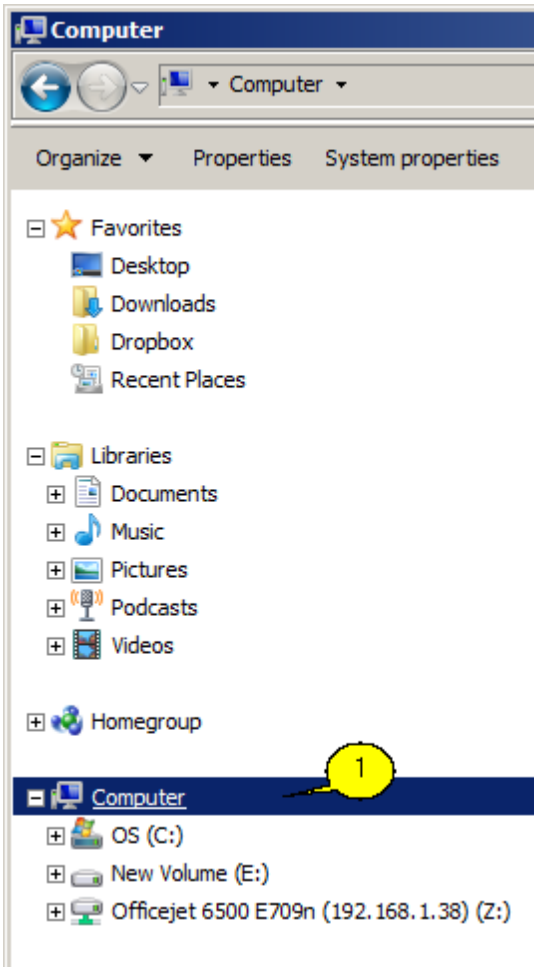
1) Add a User:
Existing users can be added using the drop-down list.

2) Permission Level:
It is mandatory for the desired users to have read and write permission.

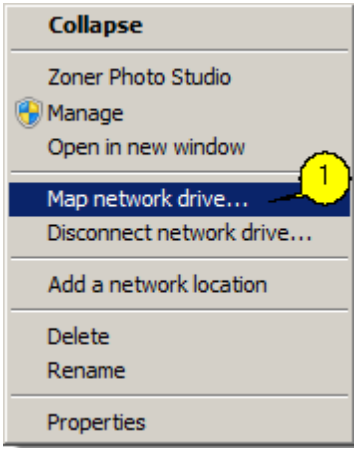
3) Release:
When the **<Release>** button is actuated, the folder is shared with the desired users.



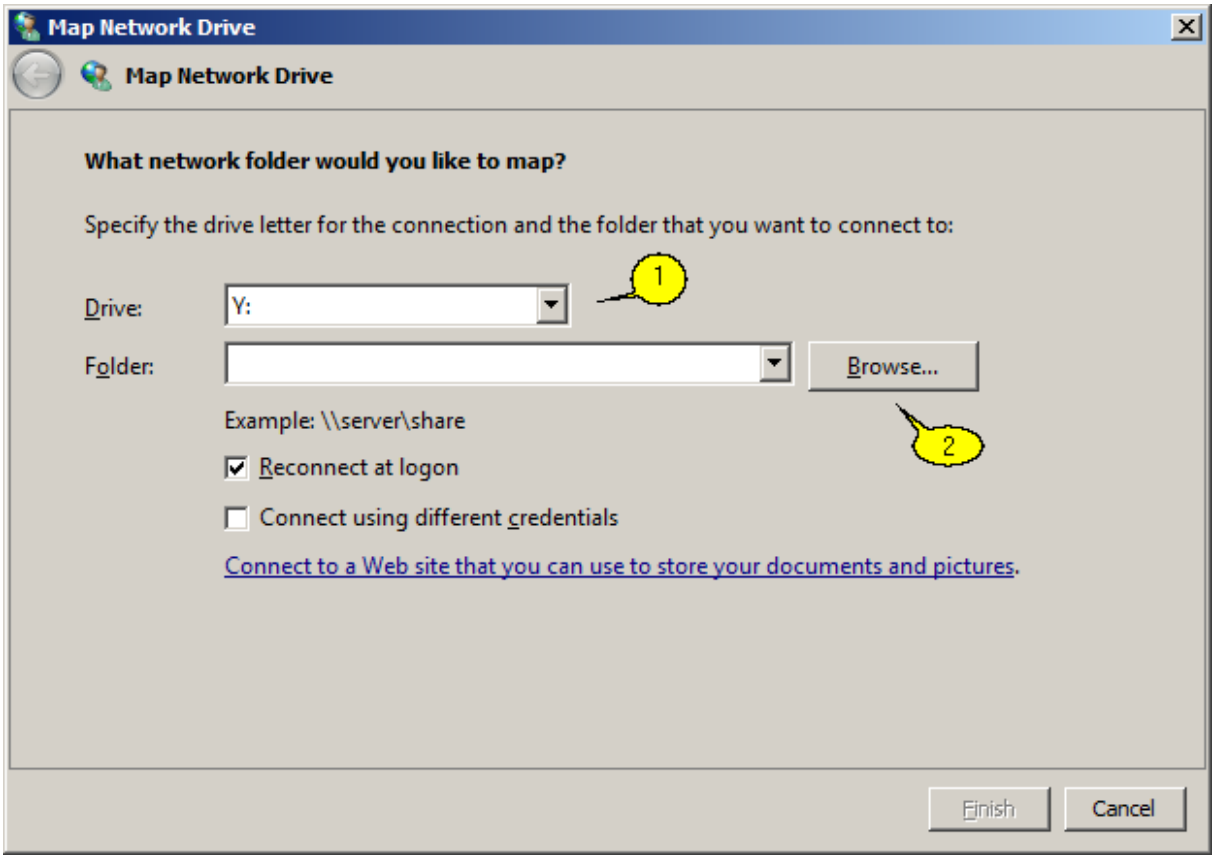
In the Windows menu, open **"Start"** the computer:



1): By right clicking on the computer icon, the shared folder can then be added as a network drive.



1): When the **<Connect network drive>** button is actuated, the following window opens:



1) Drive:
A letter can be selected from the Drive drop-down list.

2) Folder:
The **<Browse>** button can be used to search for and add the shared folder.

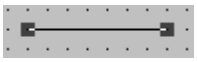
10.3.3 Brief instructions for creating a process diagram

In order to briefly present the **working method** using the **GE**, a line should be drawn on a new process diagram and then one of the end points of the line should be dragged to a new position. Then the colour should be changed.

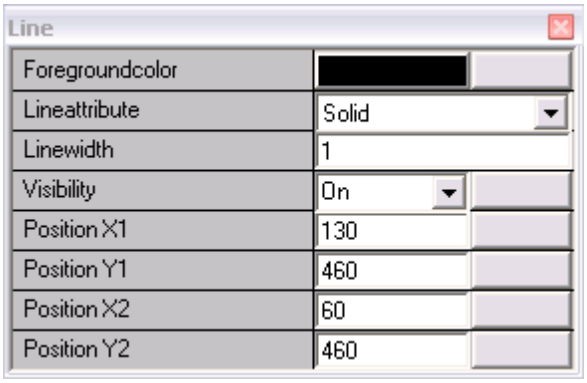
- **Open a new process diagram**.
- Switch the on-screen gridlines on or off with the **<g>** key. Adjust the gridlines with the menu item "**View > Grid Settings**".
- Select **Line Object** .
- Click on the screen position (**line beginning)** with the left mouse button, hold the button down and release it at the desired end position of the line.
- Click on the **screen background**.
- If it is not already selected, click on the **Select icon** .

- **Click on Line** with the left mouse button.
- Select the line end to be changed with the left mouse button, hold the mouse button down and drag the line end to a new position, then release the mouse button.

An active graphical object in edit mode is identified by small black handles



- You can modify the colour and line width of the active line with the help of the **Attributes window,** which can be accessed by right-clicking. (The mouse pointer does not have to point to the object. However, the object must be selected.) Select "**Attributes**" or click on the icon .

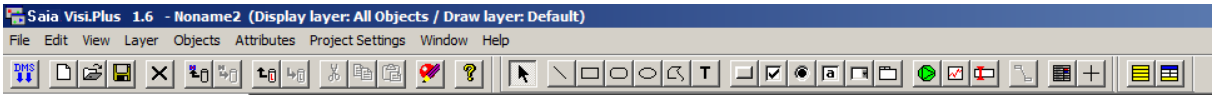


- Click on the black button in the middle column under "**Foreground Colour**" and select the desired colour.
- Click on the white field **Line Width** and change the number **1** as desired. Press the **<Enter>** key (important!).
- Save the diagram.
- Press the **<e>** key in order to switch from **edit mode** to **runtime mode**. The menu bar and the toolbars are no longer visible.
- For modifications to the process diagram, etc., press the **<e>** key.

The process just described works in the same way for all process diagrams and the remaining graphical objects.

The individual graphical objects and the toolbar are described in the following.

10.3.4 The toolbar



10.3.4.1 Menu bar

The menu bar contains 9 different drop-down lists:

File menu:	(See the chapter File menu)
Edit menu:	(See the chapter Edit menu)
View menu:	(See the chapter View menu)
Layer menu:	(See the chapter Layer menu)
Objects menu:	(See the chapter Objects menu)
Attributes menu:	(See the chapter Attributes menu)
Project Settings menu:	(See the chapter Project Settings menu)
Window menu:	(See the chapter Window menu)
Help menu:	(See the chapter Help menu)

10.3.4.2 General toolbar

	Save the DMS	(See Save DMS)
	Opens a new window for the processes	(See New Process Diagram)
	Opens an existing window for the processes	(See Open Process Diagram)
	Saves the process window	(See Save Process Diagram)
	Closes the current process window	(See Close Process Diagram)
	Loads a new macro from the library	(See Load Macro)

	Saves a marked macro to the library	(See Save Macro)
	Loads a new VLO from the library	(See Load VLO Object)
	Saves the marked VLO to the library	(See Save VLO Object)
	Cuts selected object to the clipboard. Key combination <CTRL + X>	(See Cut)
	Copies a selected object. Key combination <CTRL + C>	(See Copy)
	Inserts a selected object. Key combination <CTRL + V>	(See Insert)
	Starting the Project Engineering module	(See Call up PET)
	Calls up the help system	(See Help)

10.3.4.3 Graphical Objects toolbar

The **GE** provides various graphical objects for the creation of process diagrams, whose attributes can be changed or initialised.

In Version 1.4 the toolbar appears as follows:

In Version 1.5 a graph ruler object is additionally available. To facilitate the alignment of objects, a crosshair can also be displayed.

Version 1.6 also offers a tab object. Furthermore, the VLO can be linked to the link object:

	Select	(see Select)
	Line	(see Line)
	Box	(see Box)
	Rounded Box	(see Rounded box)

	Ellipse	(see Ellipse)
	Polygon/Polyline	(see Polygon/Polyline)
	Text/Text Field	(see Text/Text Field)
	Button	(see Button)
	Bitmap	(see Bitmap)
	Trend graph	(see Trend Graph)
	Checkbox	(see Checkbox)
	Radio Button	(see Radio Button)
	Input Field	(see Input Field)
	Combo Box	(see Combo Box)
	Attributes window	(see Attributes window)
	Graph Ruler (Version 1.5 or higher)	(see Graph Ruler)
	Crosshair (Version 1.5 or higher)	(see Crosshair)
	Tabs (Version 1.6 or higher)	(see Tabs)
	Create link (Version 1.6 or higher)	(see Link objects)

.

10.3.4.4 Link object tool bar

	Design view	(see Design view)
	Link Boxes view	(see Link Boxes view)

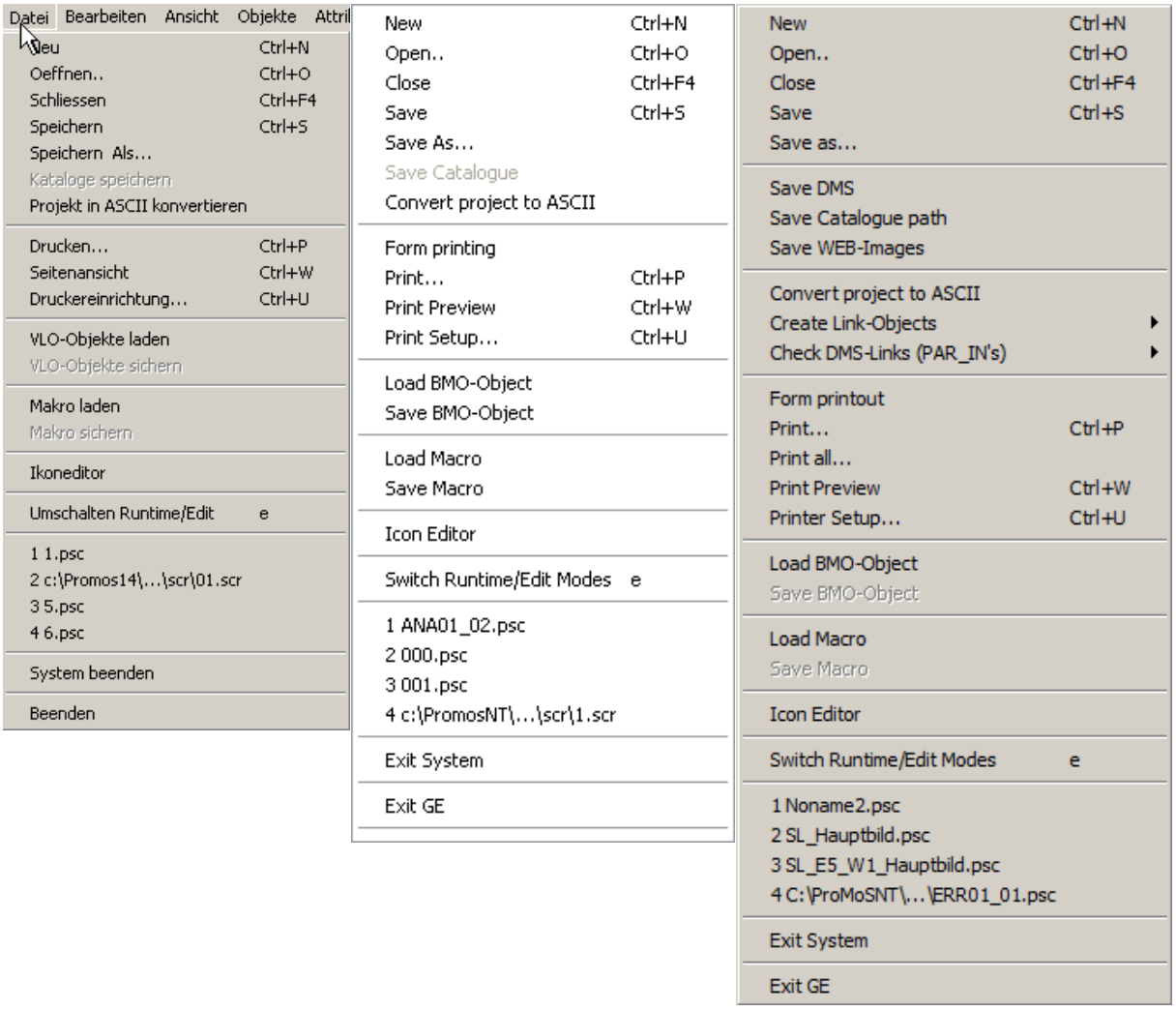
(This funcionality is disabled by default).

10.3.5 File menu

In the "**File**" menu there are functions like loading and saving process diagrams, macros, VLO libraries, printing of process diagrams and for exiting Visi.Plus.

In Version 1.5 or higher it is possible to print forms for the release of process diagrams. Switching between printing of process diagrams and release forms is done by clicking on the menu entry **File > Form Printout**.

File Menu Version 1.4 ***File Menu Version 1.5*** ***File Menu Version 1.6***



*Tip: By using key combinations such as **<CTRL> + <S>**, which saves the current process diagram, you do not always have to search for the "File > Save" command in the menu.*

10.3.5.1 New Process Diagram

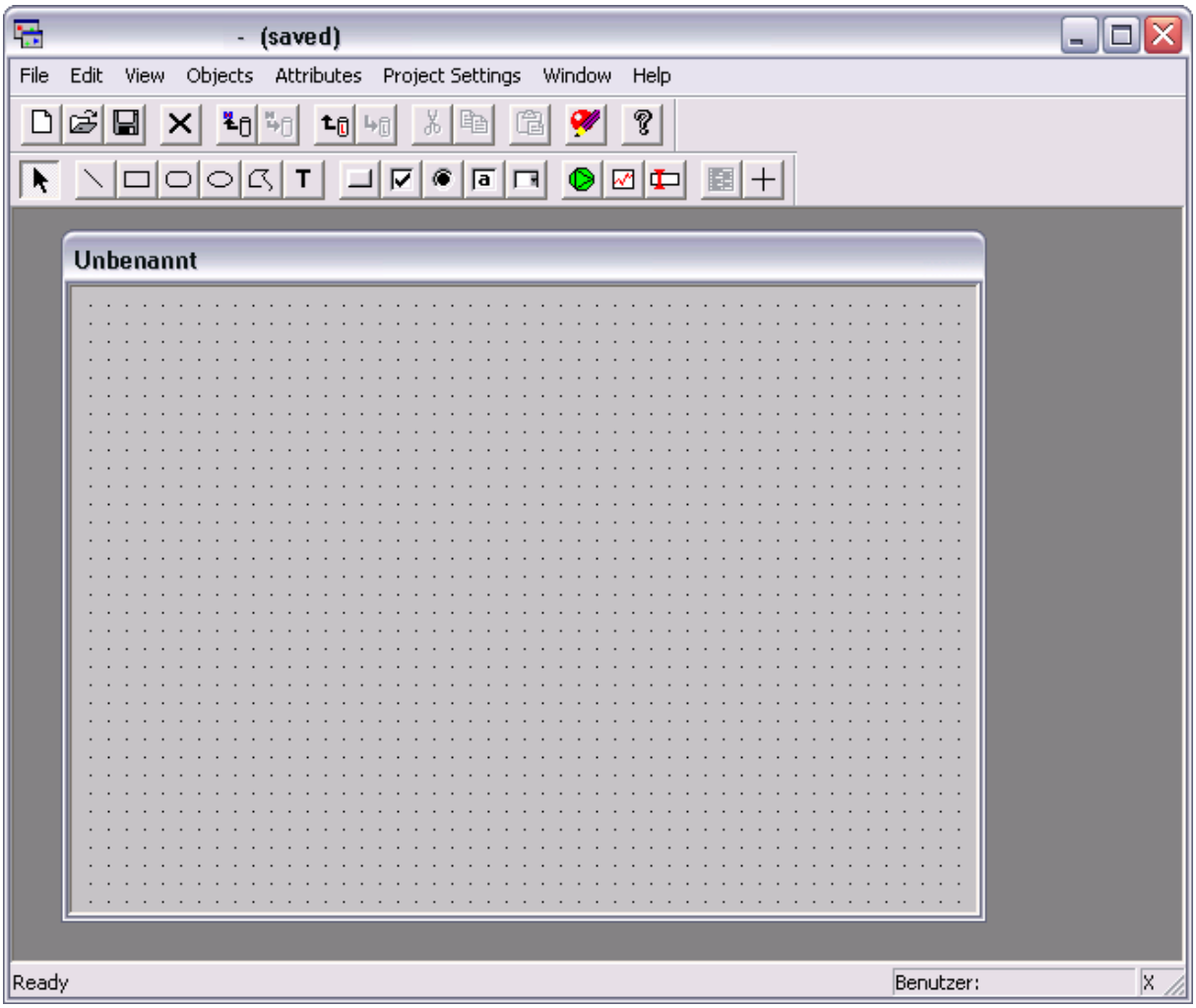
Before a new process diagram is created, the **screen resolution** of the target computer on which the application will run **must be set**. The defined maximum screen size, or process diagram, is then defined for the entire project. Process diagrams can always be reduced in size after their creation, e.g. for popup windows.

Set the **resolution** in the menu under
"Project Settings > Project Settings > Screen Resolution".

If the wrong resolution is selected, the process diagrams will either not be completely shown or the screen will not be completely filled.

A **New Process Diagram** can be created either with the menu item "**File > New**" or the key combination **<CTRL+N>** or by clicking on .

The menu bar, status bar and toolbar are only visible if edit mode is selected in the Graphical Editor.



A new process diagram is shown as a blank window entitled "Unnamed".

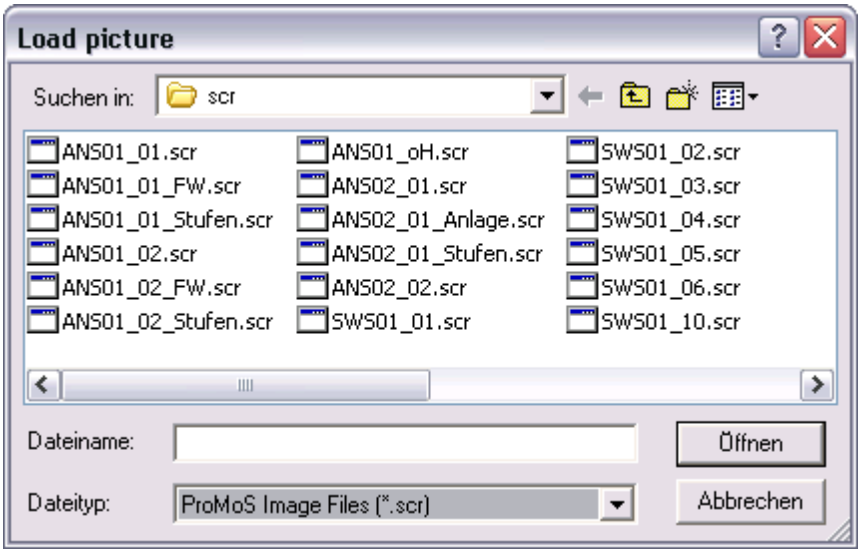
Like any other graphical object, a process diagram can also be viewed as such. Therefore its attributes, such as name, colour, etc. are also modified through the **Attributes window**.

The chapter Process diagram/window/image attributes in this manual thoroughly describes

which attributes can be changed.

10.3.5.2 Open Process Diagram

Select  or **<CTRL+O>** or **in the menu "File > Open"**, in order to open an existing process diagram that was saved to the hard disk or another data carrier.

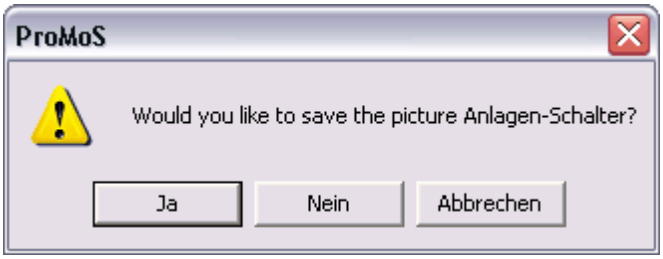


*Diagrams from other projects found in different directories should not be selected, because their bitmaps are saved separately. If, despite this, process diagrams from an existing project are copied into a new project, the associated bitmaps must also be copied. The bitmaps are located in the respective project's "c:\Visi.Plus\proj\project_name**bmp**" directory and the process diagrams in the "...**scr**" directory.*

10.3.5.3 Close Process Diagram

 or **<CTRL>+<F4>** or **in the menu "File > Close"**

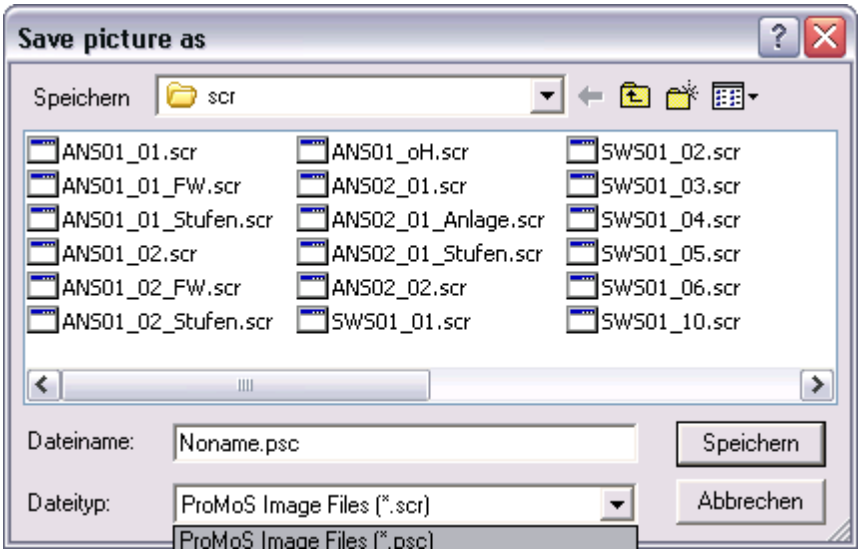
This function closes the active (selected) process diagram. If unsaved changes were made, Visi.Plus asks whether the current process diagram (its name is specified) should be saved.



Confirming this with "**Yes**" will save the process diagram in its current state. If you select "**No**", the changes are not saved. By clicking on **Cancel** the process diagram is not closed.

10.3.5.4 Save Process Diagram

 or **<CTRL+S>** or **in the menu "File > Save"**



When saving a new process diagram, a dialog is displayed for selecting the save location of the process diagram. When assigning the filename, the conditions prescribed by the computer's operating system apply (no characters like /, +, &, etc.).

 *You have the option here of choosing between the *.psc (ASCII) and *.scr (binary) file formats. The binary format is the conventional format in Visi.Plus and ASCII is the newer format in Visi.Plus.*

Advantages of ASCII over binary:

- Smaller file sets
- Source code editing is possible
- *Security aspects*
- *Backward compatibility is assured*

We recommend using the current ASCII format (.psc extension), because the binary format is not longer supported in later versions.

 *IMPORTANT: The process diagram should be saved frequently during editing. This is achieved most easily with the key combination **<CTRL> + <S>**.*

In the menu "File> Save As..."
With "**Save As...**" a different filename and save location can be indicated.
In doing so, different versions of images can be saved without spending time copying.

10.3.5.5 Save DMS

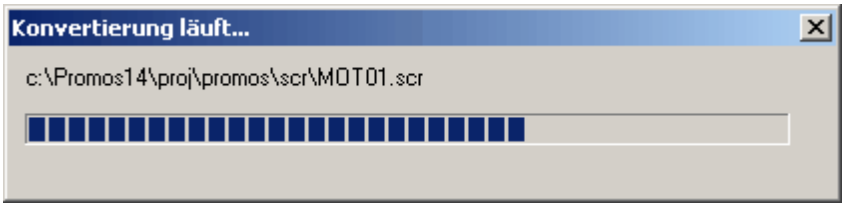
Saves the current DMS.

10.3.5.6 Save catalogues

Saves the visible icons in the catalogue bar/library as a catalogue file.

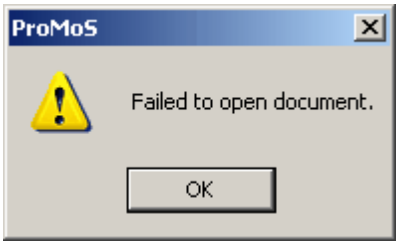
10.3.5.7 Convert Project to ASCII

Converts *.scr image files from binary format to ASCII format (*.psc).
Images, libraries and macros can be saved and used as an ASCII file (*.psc).



Attention:
This function is not backwards compatible. This means: As soon as you have converted your project to ASCII files you can only view the images in GE versions that were released after Version 1.4.0.11.

With older versions, the following error message appears as soon as the project is started:



*Bear in mind that the predefined start image of the project, which must be adjusted in the GE in the Project Settings, should also be converted from *.scr to *.psc. Otherwise, when first starting the project after the ASCII conversion, the start image will not be opened.*

Recommendation: Use the ASCII format in all projects, because the binary format is no longer supported in newer versions.

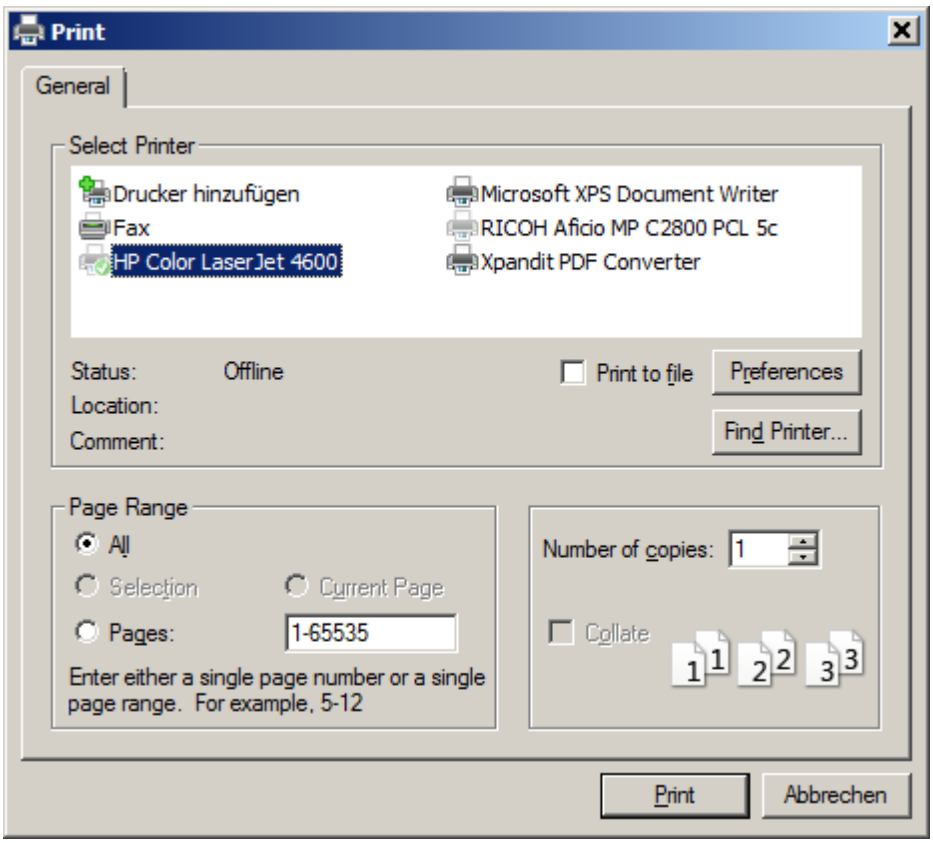
10.3.5.8 Form Printout

In Version 1.5 or higher it is possible to print forms for the release of process diagrams and display them in Print Preview. Switch Form Printout on and off by clicking on the menu entry **"File > Form Printout"**. If the checkmark is set, by clicking on the menu entry **"File > Print..."** the release form is printed or the form is displayed by clicking on **"File > Print Preview".**

The settings are saved in the file **<proj>/cfg/promos.cfg** in the section **[Print Settings]**.

10.3.5.9 Print <CTRL+P>

Process diagrams and forms (in Version 1.5 or higher) for the release of process diagrams can be printed by clicking on the print icon in the toolbar, via the menu command of the same name under "File" or with the key combination **<CTRL+P>**.



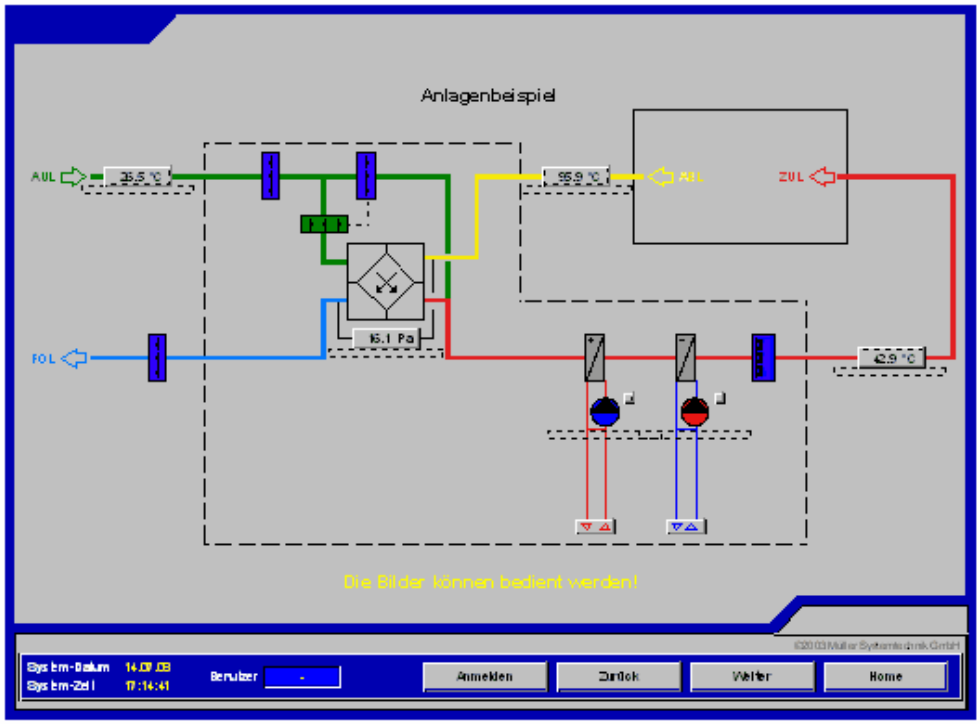
This is a PC operating system dialog and should be interpreted accordingly.

Switch Form Printout on and off by clicking on the menu entry **"File > Form Printout"**. If the checkmark is placed here, the release form is printed.

Release forms have the following appearance:

Bildname **6.scr**

Bilddatum 14.07.08



FREIGABE Prozessbild		
	Datum	Visum
Intern		
Planer		
Kunde		

Projekt c:\Promos14\proj_Handbuch_promos

Lizenz MST Systemtechnik AG
Anlage Internal Use Only!

Druckdatum 14.07.08

Copyright © MST Systemtechnik AG, CH-3123 Belp

The text of the table as well as the copyright (last lines) are saved - in the file **GE_Form_Germ.cfg** for the German version or **GE_Form_Engl.cfg** for the English version - the first time Form Printout or form Print Preview in the Visi.Plus bin directory is selected, and

can now be modified as desired using a text editor.

GE_Form_Germ.cfg file with default values:

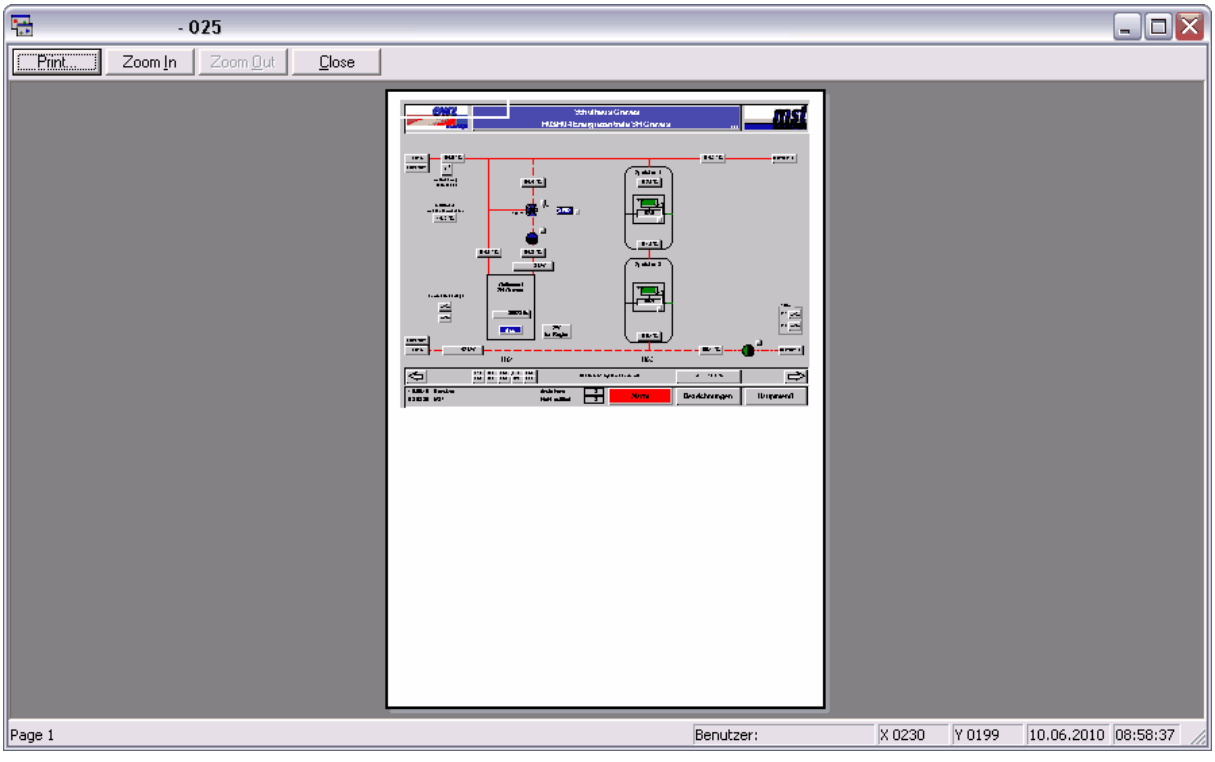
[Settings]
A1=RELEASE process diagram
B2=Date
C2=Initials
A3=Internal
A4=Planner
A5=Client
Copyright=Copyright © MST Systemtechnik AG, CH-3123 Belp

The position of the text within the table is defined by a letter for the column and a number for the line, e.g. A1 for the 1st line of the table. The numbering corresponds to the name in Excel or OpenOffice Calc and may not be changed. The text after the equals sign is freely selectable.

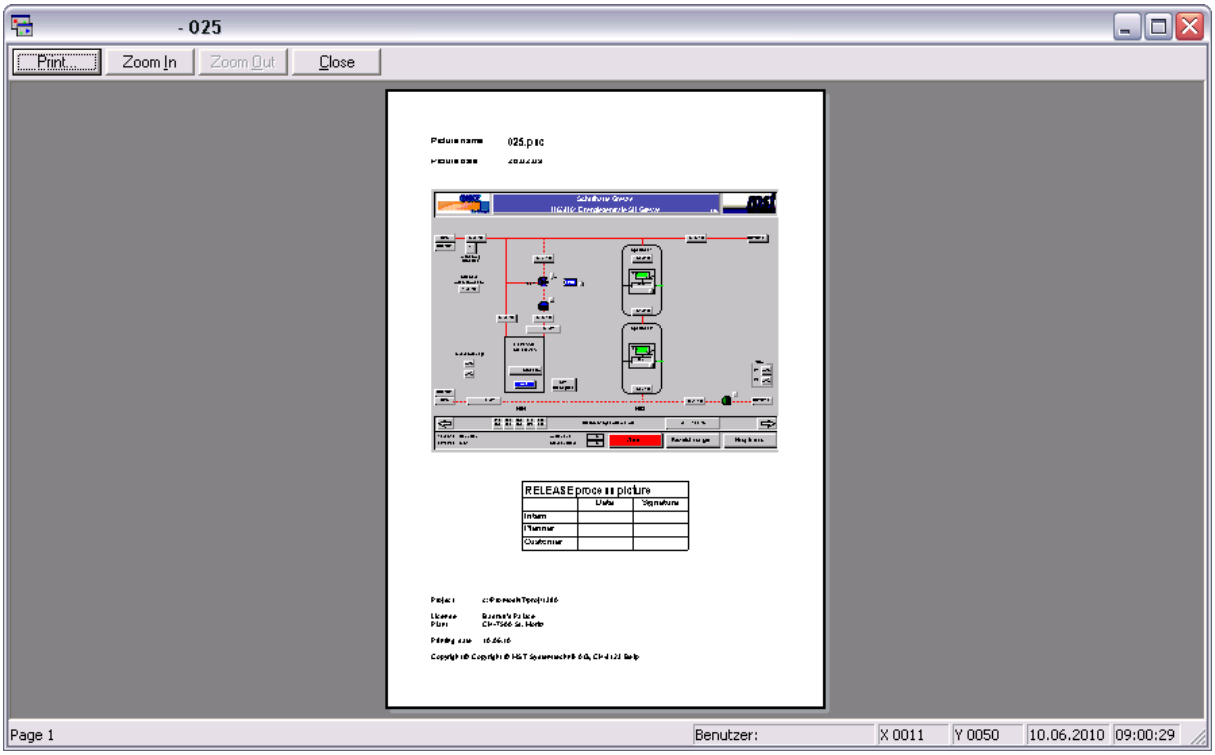
It is possible to initiate printing through the data point System:Node:<PC-Name>:Print. In doing so, it must be ensured that the printer dialog is not open when the flag of the data point System:Node:<PC-Name>:PrintDirect is set. The printer dialog is only displayed if the flag is not set.

10.3.5.10 Print Preview <CTRL+W>

Before printing out an diagram it is possible to show a preview of the upcoming printout. This may be helpful for adjusting and optimising the borders, for example.



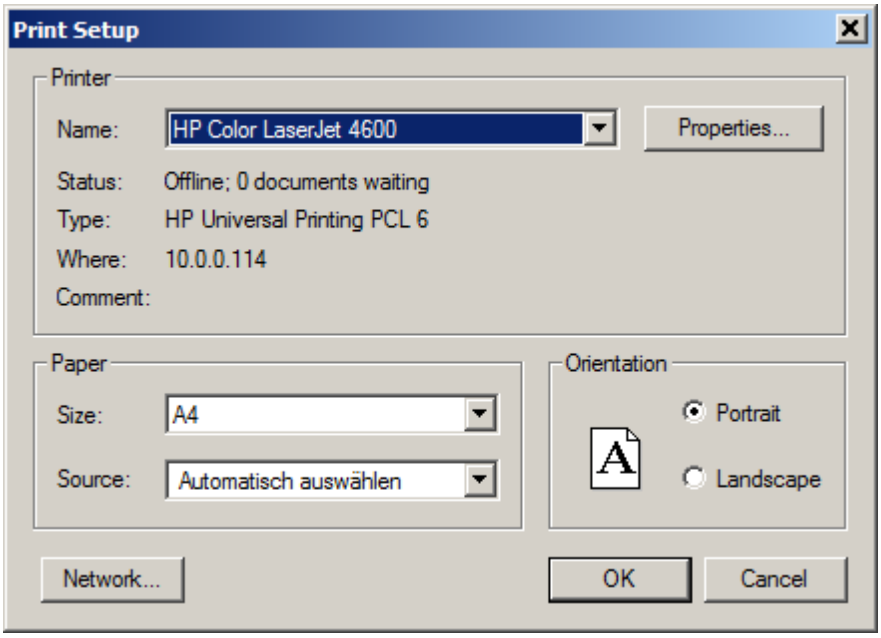
In Version 1.5 or higher it is possible to have the page displayed as a form. You can switch the form Print Preview on and off by clicking on the menu entry **File > Form Printout**. If the checkmark is placed here, the release form is displayed.



It is possible to initiate printing through the data point System:Node:<PC-Name>:Preview.

10.3.5.11 Printer Setup <CTRL+U>

With this menu item, settings can be made for the printer used in Visi.Plus. You can select between portrait and landscape as well as the source and type of paper.



10.3.5.12 Template objects

Each system consists of different object types, such as pumps, valves, analogue measurement points, etc. which, as objects, always have the same attributes.

We will take a pump as an example. A graphic representation of whether the pump is running or not could be easily realised with two icons.

- What about if this pump could be switched from "Automatic" to "Manual mode" with the mouse?
- If "Manual" is set, how should the pump be switched on or off with the mouse?
- What about the performance data?
- Is the pump overheating or is it jammed?
- How should the operating hours be displayed?
- Should the pump be switched to service mode?
- Who is the manufacturer and what item number does the device have?
- Must these visual elements always be redrawn and reprogrammed?

No, because this is precisely what the so-called **VLOs (template objects)** are intended to help with.

VLOs (template objects) are parent objects. That means that all graphical objects are initialised and then grouped together. They are then saved with a meaningful name and can later be loaded or copied as often as necessary. As soon as the **VLO** is loaded, it must be reinitialised to the system objects.

The effort and expense on engineering is thereby significantly reduced. In addition, it is possible to automatically generate a large part of the **PLC source code** (see the chapter PET Code Generator).

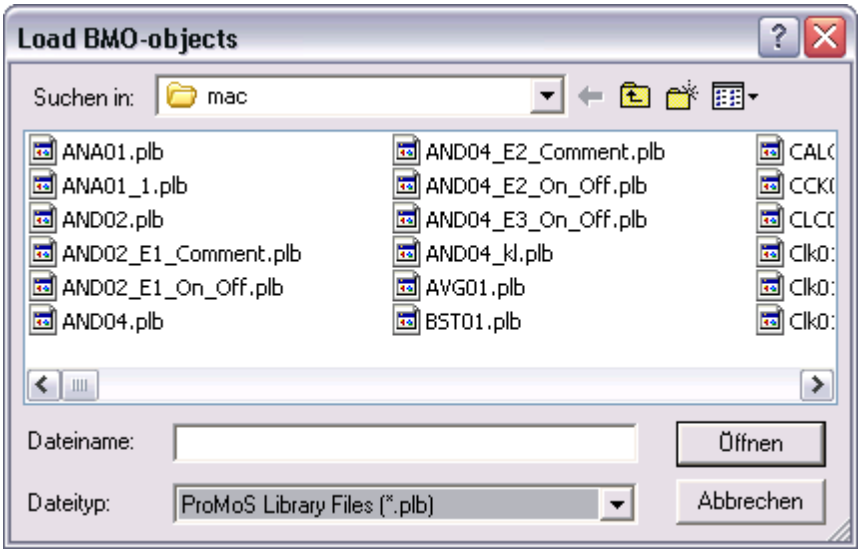
For these reasons, all actuators/sensors of a system should wherever possible be defined through **VLO**s (template objects or parent objects).

Those who value a simple overview will create as few **VLOs** as possible.

Entire groups of signals can be grouped, e.g. heating groups, motors, pumps, measurements, etc.

10.3.5.13 Load VLO Object

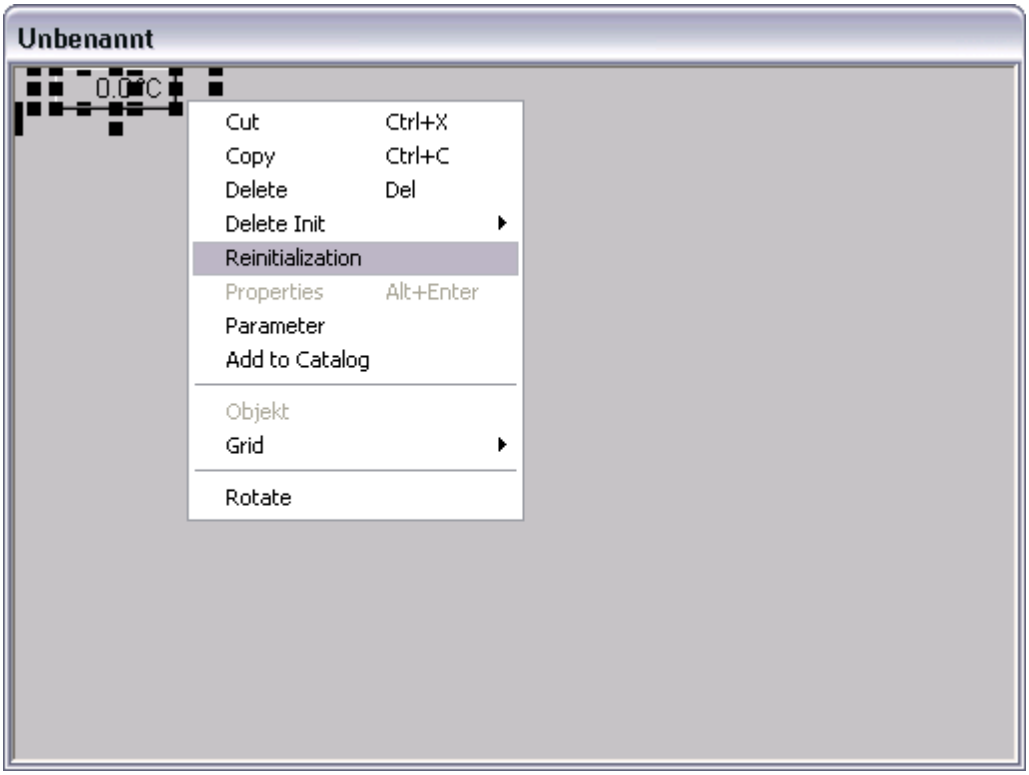
or in the menu "File > Load VLO Object"
Loads a VLO object (template object) from hard disk or another data carrying medium.



If VLO objects are to be copied from an existing project to a new project, the associated bitmaps must also be copied.
The bitmaps are located in the projects ***'"c:\Visi.Plus\proj\project_name\bmp"*** *and the VLOs in the* ***"...\mac"*** *directories.*

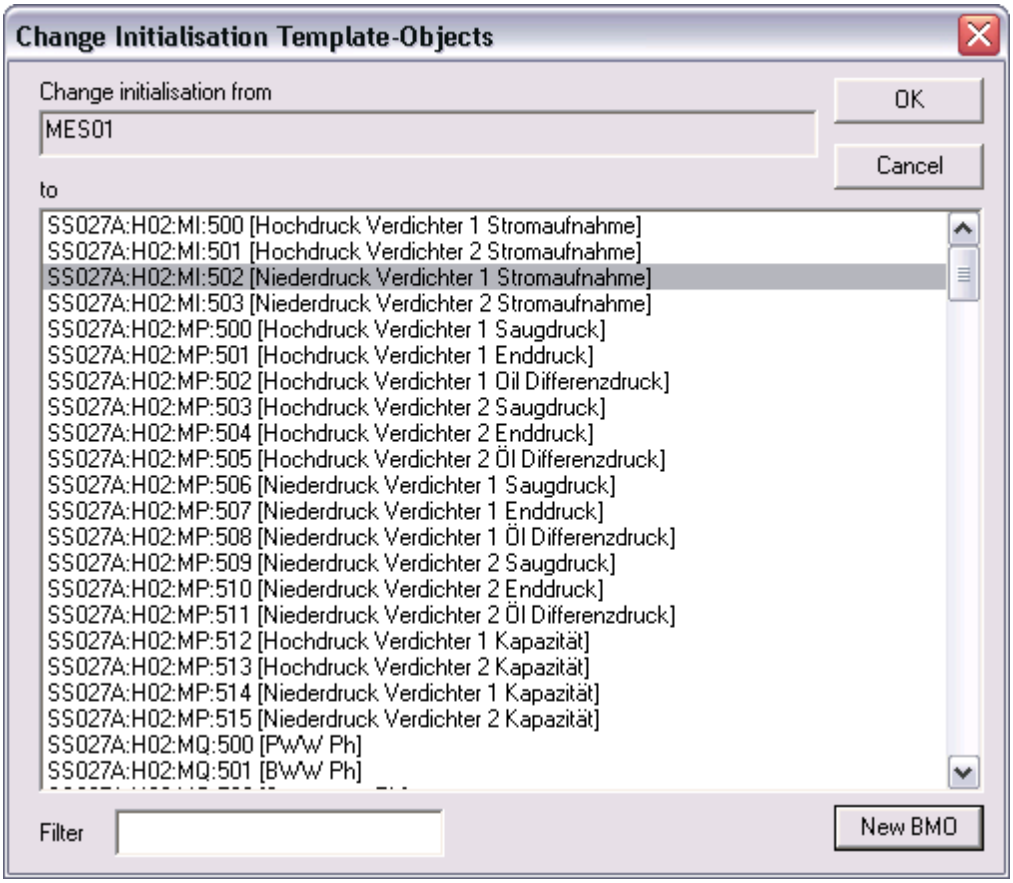
The selected VLO object is inserted into the process diagram at the top left corner of the screen. The VLO object can now be placed anywhere in the process diagram with the mouse. If an object should only be moved a few pixels, the selected object can be positioned with the arrow keys. The object is moved by one pixel per keystroke. If the arrow key is held down, the object is moved until the arrow key is released again.

The object that was just loaded from the data carrier retains its original initialisations and therefore must be reinitialised to a desired data point.



In order to reinitialise a selected VLO object, click the right mouse button. Then select the command "**Reinitialise**" from the popup menu. Alternatively, the **<Space bar>** can be pressed.

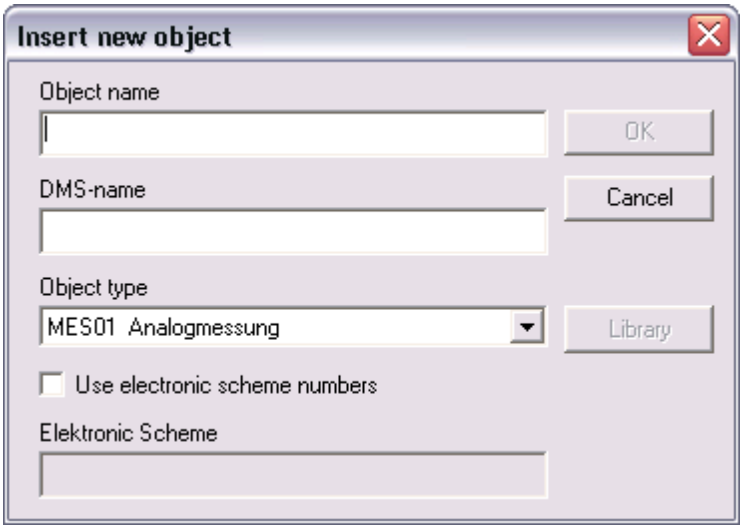
The window **"VLO Objects Reinitialisation"** opens:



From the list that is provided, select the desired DMS name to which the template object should be reinitialised. To confirm, click on the **<OK>** button. The initialised template object immediately displays the values to which it was initialised in the GE.

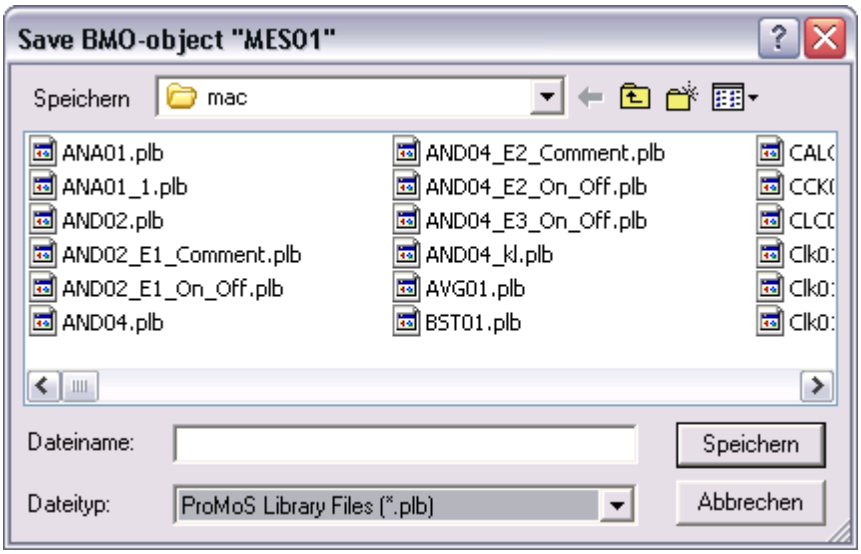
In the example shown, the template object MES01 displays the values of the system object BHS60:MP:501.

"New BMO" button:
Opens the default window **"Insert New Object"** for inputting a new system object.
You can select a new object from your catalogue or your library.



10.3.5.14 Save VLO Object

or in the menu "File > Save VLO Object"



Before saving, carry out or check the following steps.

- *Generate the entire graphic for the planned VLO.*
- *All necessary graphical object attributes, which should be reinitialised later, are to be initialised to the respective parent object.*
- *All graphical objects which are used to represent the VLO must be selected and then grouped.*
  In order group multiple graphical objects into a group the ***<SHIFT>*** *key must be used in conjunction with the mouse. Alternativly, you can draw a box with the mouse around the objects.*
  They are then grouped by pressing the ***<z>*** *key or in the menu via "**Edit > Group Objects**".*
- *After these steps, the VLO can be saved.*

VLO objects can only be reinitialised to the VLO types you have created (system objects entered in the PET) or to Visi.Plus system data such as blinkers.

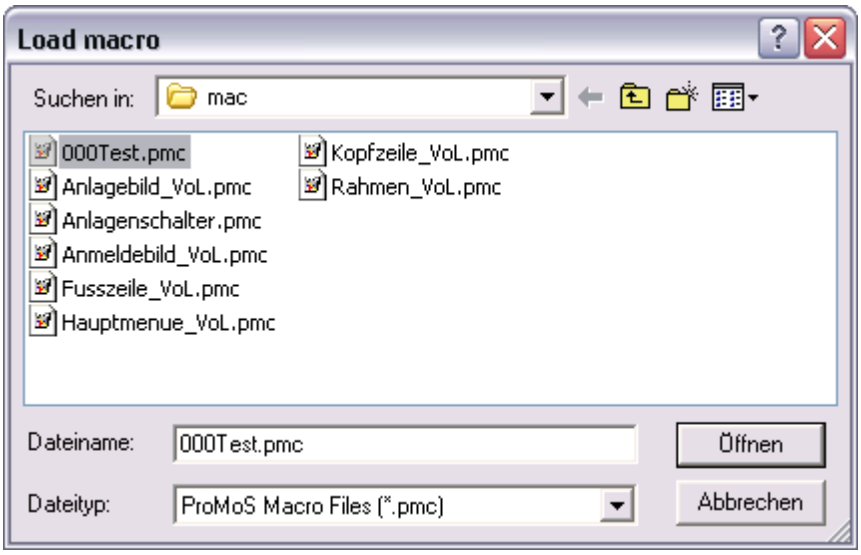
10.3.5.15 Macro

The idea of macros is that multiple graphical objects can be saved as one, with or without initialisation. This allows you to save alignments of graphical objects and their initialisations as a macro. When loading a macro, all initialisations are displayed. Therefore it is possible to change them. A typical example of a macro is a header or status bar as well as a control field that is repeated on all system pages, and thus needs to be the same. Another example would be a heating group which might consist of a valve, a motor and a regulator.

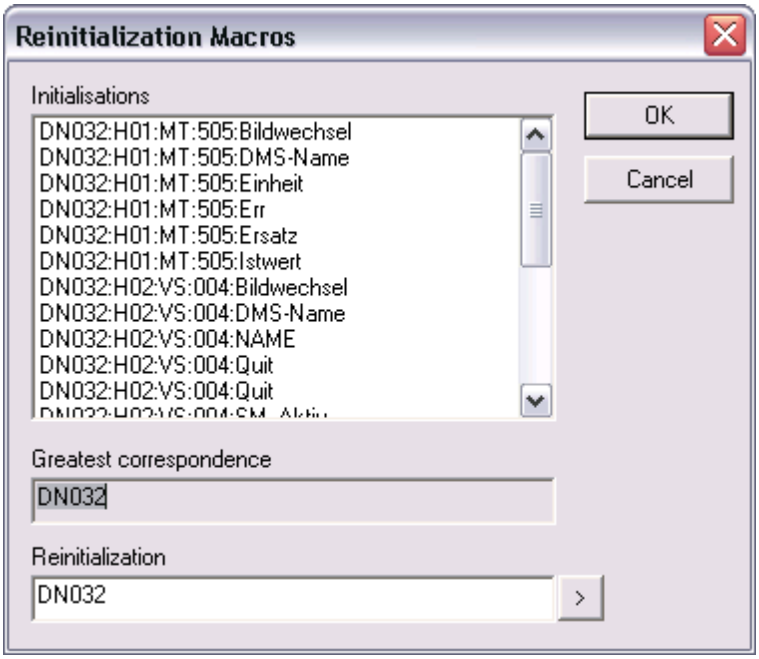
10.3.5.16 Load Macro

 **or in the menu "File > Load Macro"**

The macro is loaded:

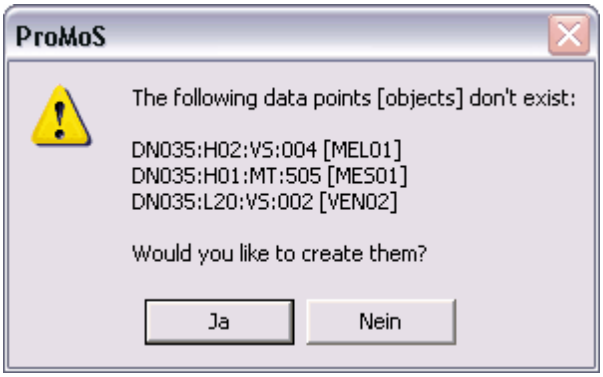


After selecting a macro and clicking on open, a list of all initialised objects is shown.

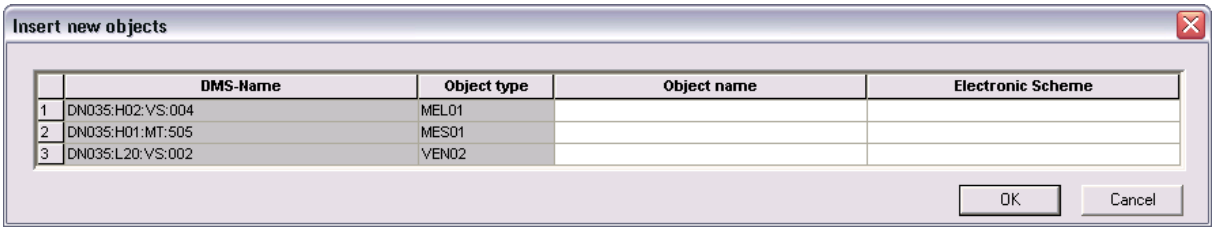


The smallest match can now be adjusted:

The objects are automatically reinitialised, if available in the DMS. If the objects are not found in the DMS, the following dialog appears:



If you click **Yes**, the following dialog is then displayed:

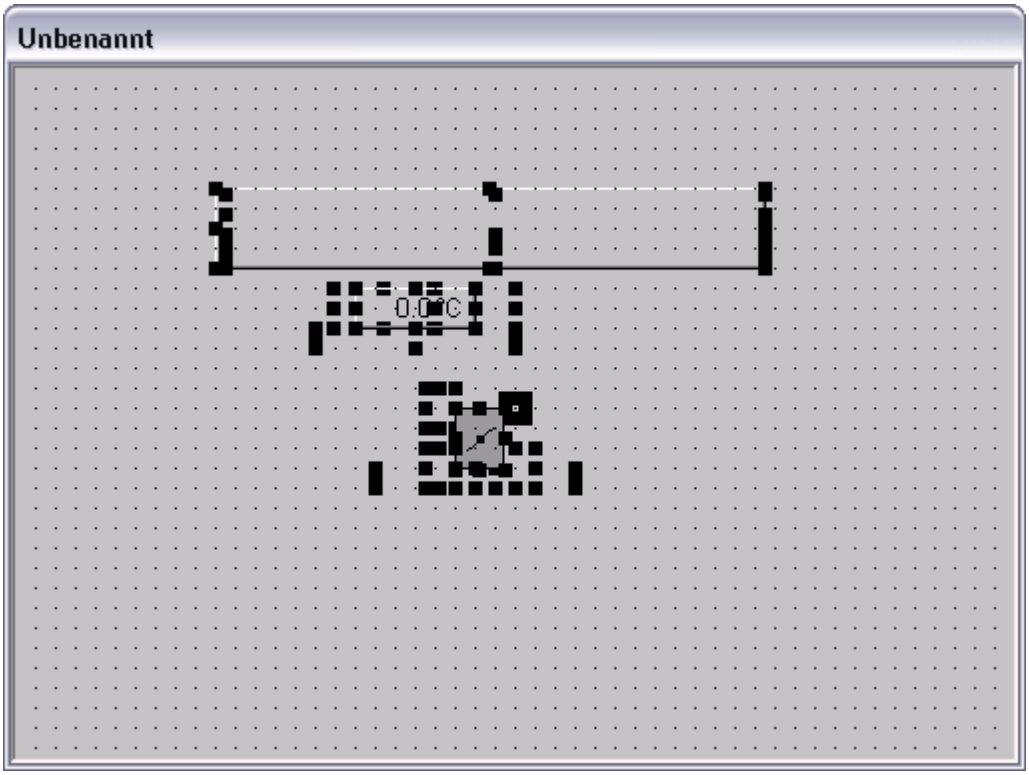


Here you can enter a new object name for the individual sensors/actuators/internal objects. The electrical diagram name can also be changed.

By clicking on **OK** the new objects are created in the DMS. This can take a while, because

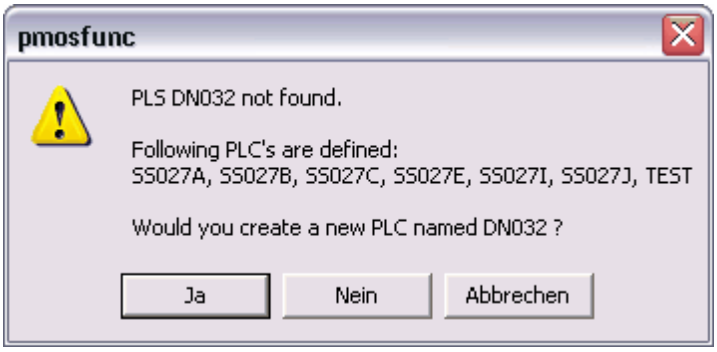
as many as several thousand data points must be created.

All objects are integrated into the GE and the data points are also generated in the DMS.

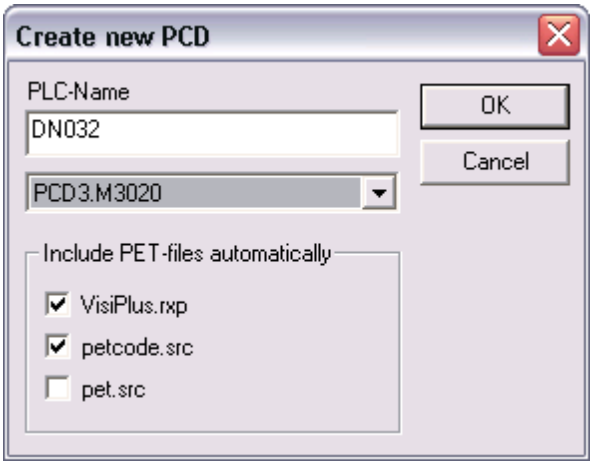


It should be noted that the PLC addressing may no longer be correct and may need changes (this is done most easily in the PET with automatic address assignment).

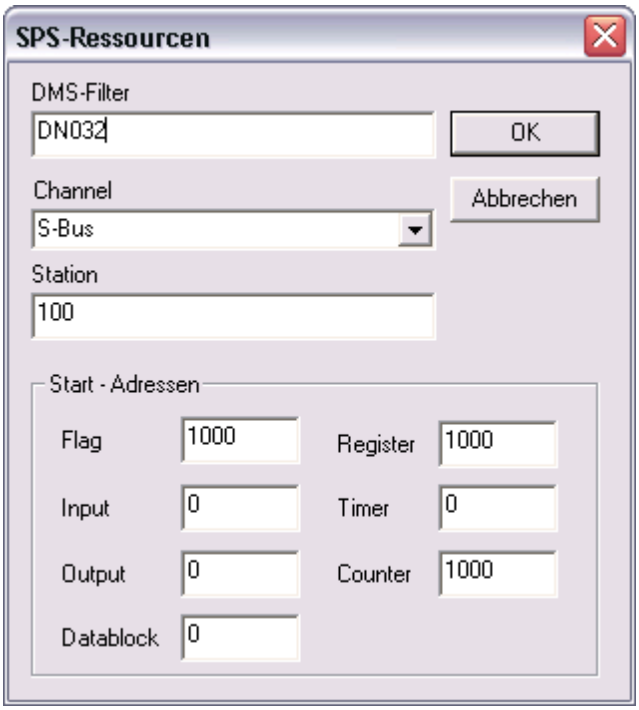
If the checkbox **Automatically Assign PLC Resources** is marked in the **PET** in the menu "PG5 > Export Options", the PLC resources (channel, station, telegram, address) are also automatically generated on reinitialisation of the macro. If the PLC is not present yet, this is recognised by the GE:



You can now recreate the PLC. To do so, click **Yes.** Then select the PLC type and the PET files which should be integrated automatically.



After clicking **OK**, you must enter the channel, station and the start addresses in another dialog.
Then exit the dialog with **OK**.



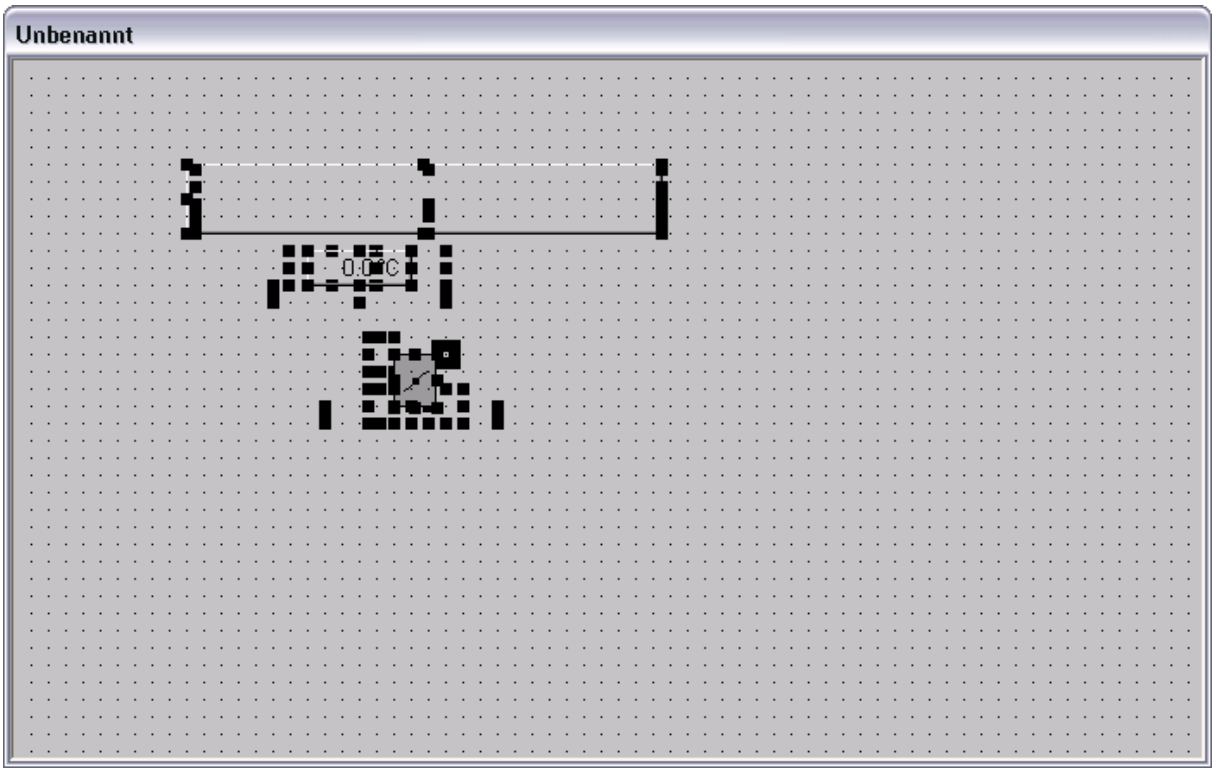
10.3.5.17 Save Macro

or in the menu "File > Save macro"

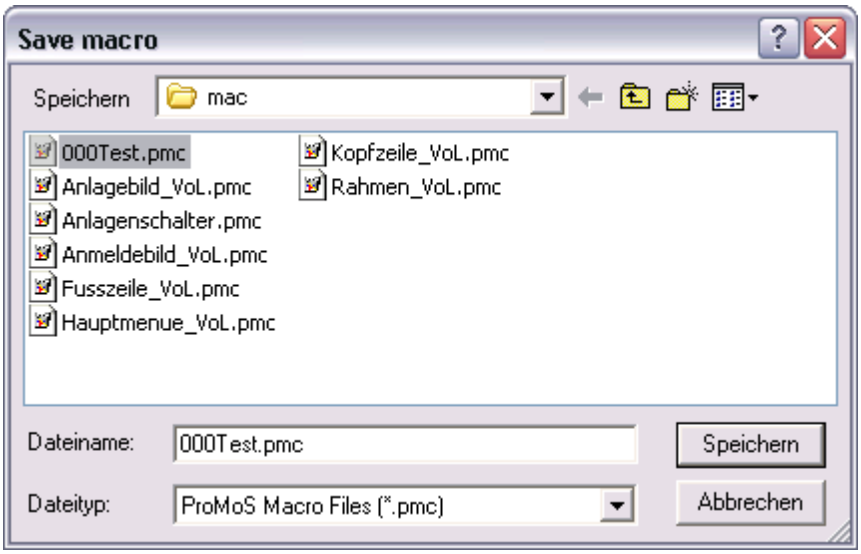
In the process, it must be ensured that the elements to be saved (VLO object, button, texts, bitmaps, etc.) are all selected.
Multiple elements can be selected with the mouse while also holding down the **<SHIFT>** key or by creating a box around the desired elements with the mouse. Then press the **<z>** key to

group the objects or select the menu item "**Edit > Group Objects**".

A macro consists of multiple graphical objects that are saved together.

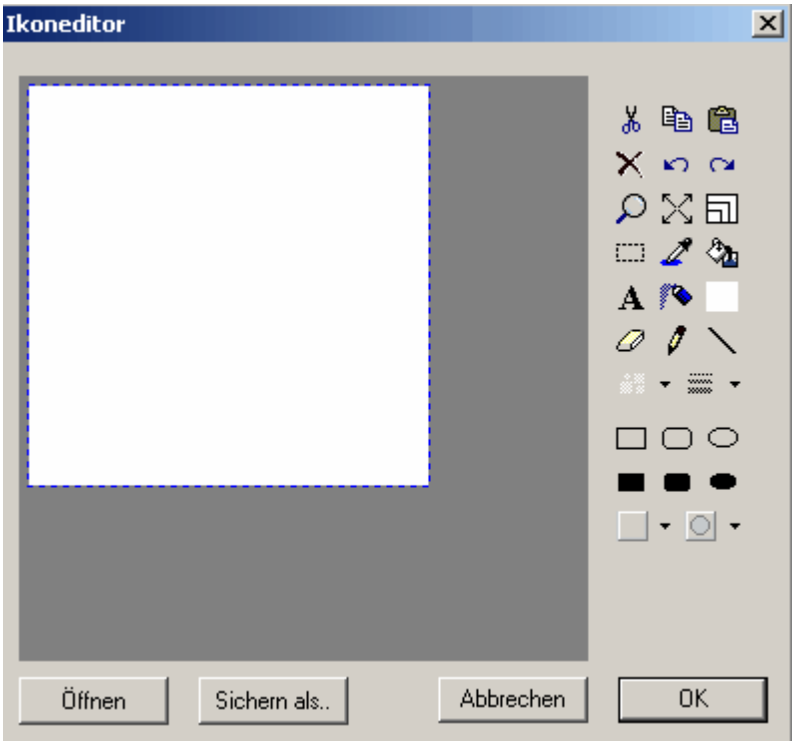


The grouped objects are saved as a macro.



10.3.5.18 The Icon Editor

The Icon Editor allows you to draw and edit small bitmap images.



The icon editor is an easily operated image editor. All commands are self-explanatory and typical for Windows.
The file formats used are the standard BMP and JPG formats. Any other image creation program may also be used (e.g. PaintShop Pro).

If a bitmap which is already being used on a process diagram, etc. is modified, the Graphical Editor must be restarted, as the changes will otherwise not be visible in the GE. On startup, the GE loads all bitmaps into its internal memory (cache) for performance reasons.

*All JPG and BMP files which are opened in the GE must be located in the current project directory in the "..**bmp**" folder!*

10.3.5.19 Mode change and exiting the GE

Switching from runtime to edit
The mode that is currently selected is recognisable from the hidden/displayed menu bar, toolbar and status bar.

Edit mode = bars displayed
Runtime mode = bars hidden

You can switch back and forth between the two modes with the **<e>** key.

If no administrator rights have been assigned to the user in the User Rights module (pUser.exe), they can no longer get back into edit mode.

Recently Opened Files
Shows the files of the process diagrams opened most recently.

Quit system
Exits the entire Visi.Plus system, meaning all associated programs and add-on modules (DMS, PDBS, etc., recognisable from the coloured dots to the bottom right on the Windows taskbar) are exited. In other words, the entire visualisation system is shut down.

Exit
This menu item only exits the GE (Graphical Editor). This allows you to exit the representation or display on a computer without losing the data (e.g. trend data, alarms).

If additional changes were made without saving them, Visi.Plus asks whether the current process diagram should be saved.

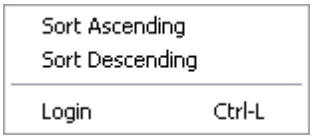
 *Exiting the Graphical Editor, does not exit the other Visi.Plus programs, such as the DMS, PDBS, the PLC driver, etc.!*

10.3.5.20 Save WEB images

As of version 1.6, it is possible to save all images and to create the WEB images directly this way.
This replaces the program SaveWeb.exe.

A re-saving of the images is necessary if, for example, changes have been made to the master image. It is recommended that all images be saved after initial startup so that the first initialisation does not display a fault before the image is fully set up.

10.3.6 Edit menu



10.3.6.1 Cut menu

 or **<CTRL> + <X>** or in the menu "**Edit > Cut**"

The "**Cut**" command removes the elements previously selected from the current process diagram and copies them to the clipboard. From there they can then be reinserted into the process diagram using **<CTRL> + <V>**.

10.3.6.2 Duplicate menu

<CTRL> + <D> or in the menu "**Edit > Duplicate**"
After being selected, elements can be duplicated with this command. The duplicated elements appear right in the current process diagram, 10 pixels below and to the right of the base element in each case. In principle, this command does the same thing as **<CTRL> +**

<C>(copy) and **<CTRL> + <V>** (paste).

10.3.6.3 Copy menu

 or **<CTRL> + <C>** or in the menu "**Edit > Copy**"

The "**Copy**" command is used to store a copy of a previously selected element in the clipboard.

10.3.6.4 Paste menu

 or **<CTRL> + <V>** or in the menu "**Edit > Paste**"

With the "**Paste**" command, the content of the clipboard is inserted into the process diagram. In the process, all properties of the elements are retained.

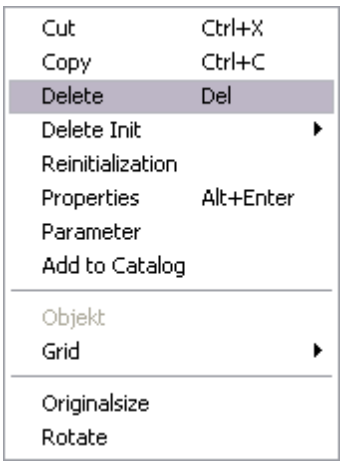
10.3.6.5 Delete menu

<DEL> or in the menu "**Edit > Delete**"
Previously selected elements are permanently deleted from the process diagram with the "**Delete**" command.

10.3.6.6 Select All

<CTRL> + <A> or in the menu "**Edit > Select All**"
This command selects all elements in the current process diagram.

 *The commands "**Cut**", "**Copy**" and "**Delete**" can also be accessed via the right mouse button. A context menu appears in which the desired function can be selected.*

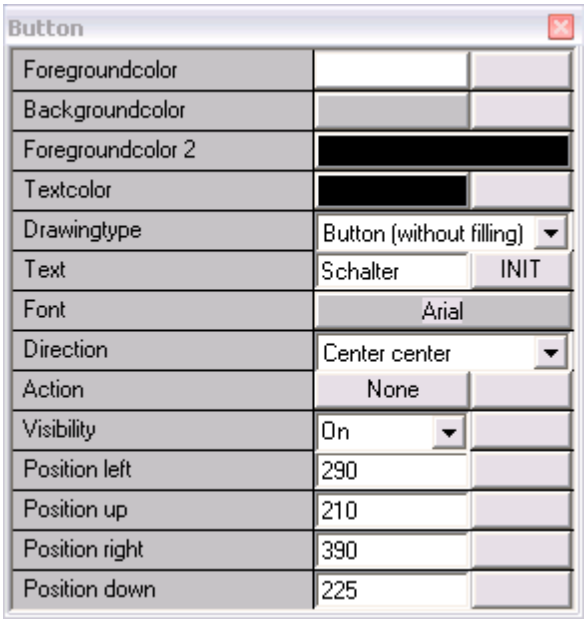


10.3.6.7 Attributes

 or **<ALT>+<ENTER>** or the menu "**Edit > Attributes ...**" or holding down the left mouse button for more than a second on an element.

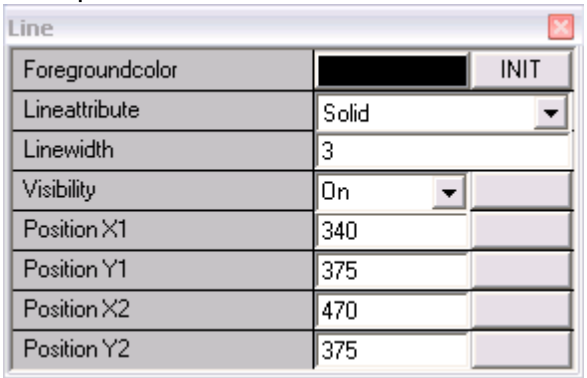
By using this command the dialog window **Attributes** appears for the selected element. The object type is indicated in the title bar.

 *The **<Tab>, <Shift + Tab>, <B> or <F>** keys or key combinations can be used to switch the attributes window between the individual elements!*

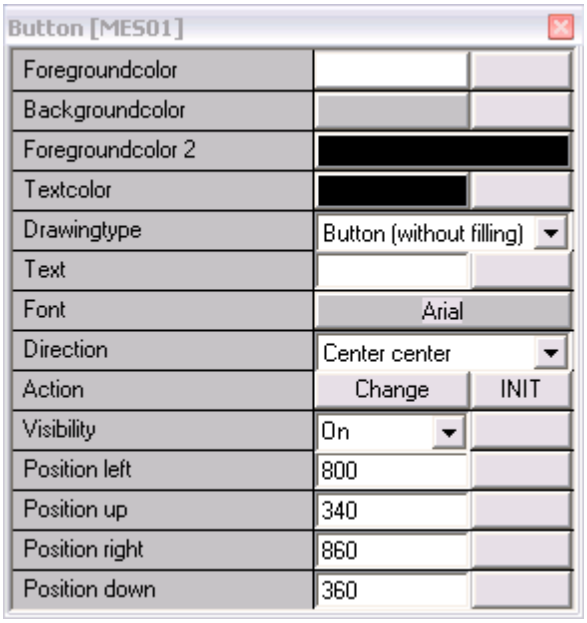


The content of the Attributes window depends on the type of graphical object.

Example attributes - Line:



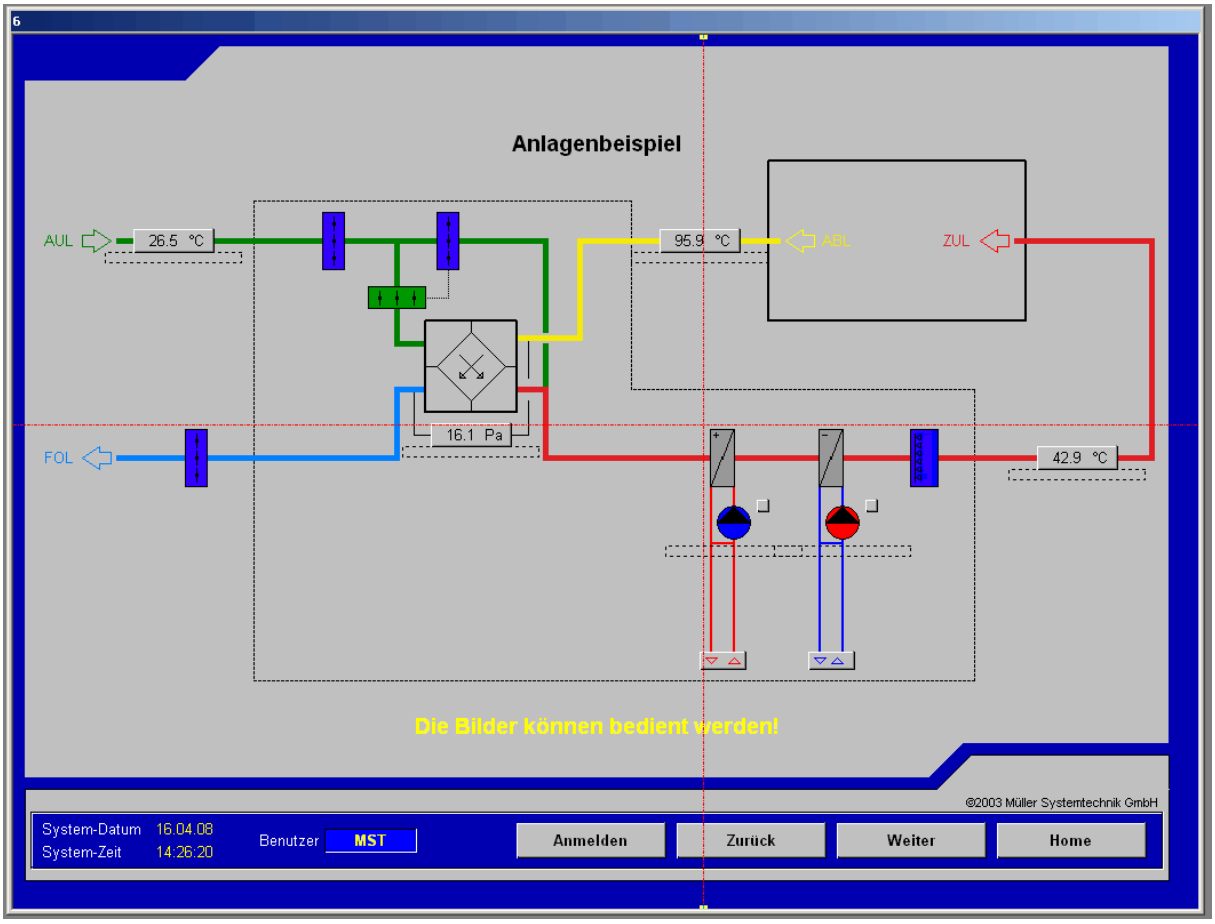
Example attributes - Button:



10.3.6.8 Crosshair

 or **<ALT>+<+>** or in the menu "**Edit > Crosshair**"

By calling up this command a crosshair is shown on the process diagram in the form of a red horizontal and red vertical line. The crosshair facilitates the alignment of objects above and next to one another, because the objects can now also be aligned with the guidelines.



The guidelines can be moved. To do so, place the mouse pointer on the vertical or horizontal guideline. When the mouse pointer changes into two parallel lines with an arrow, click the left mouse button. The guideline is now selected and can be moved to any position on the process diagram with the mouse or the left and right arrow keys.

By clicking on the window background the selection of the guidelines is cancelled. Clicking again on  switches the crosshair off again.

10.3.6.9 Group Objects

<z> or in the menu **"Edit > Group Objects"**
Multiple objects or drawing elements can be combined to form a group.

The selection of objects takes place as usual with **<SHIFT> + left mouse button**.
Then select the command "**Group Objects**" or press the **<z>** key.

All subsequent changes carried out on the group affect all its parts.
Grouped objects can be moved as a single object.

10.3.6.10 Ungroup Objects

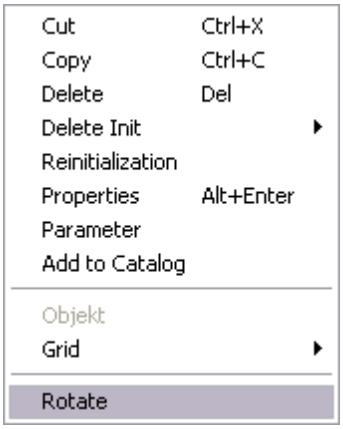
<t> or in the menu "**Edit > Ungroup Objects**"
Separates an object group back into the individual graphical objects.

10.3.6.11 Rotate Objects

Lines, boxes, text fields, buttons and icons can all be rotated 90° clockwise. When either multiple objects or grouped objects are selected, rotation takes place around the common centre point.

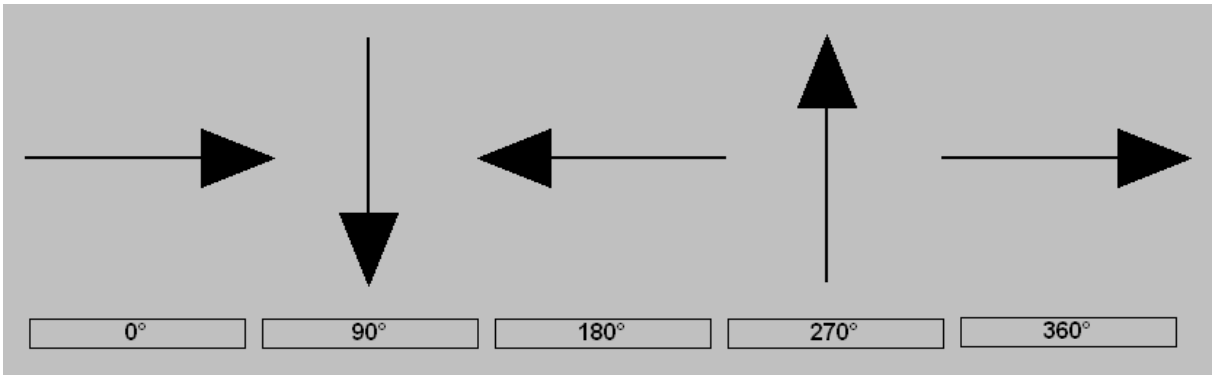
Click on the respective object to select it. To select multiple objects, press the **<SHIFT>** key and click on the objects with the left mouse button or drag a frame around the objects with the left mouse button pressed.

Then press the right mouse button. A selection menu appears. Select the entry **Rotate**.



The objects now rotate 90° clockwise. Selecting the command again triggers another 90° rotation. The rotation can also be accessed with the key combination **<Ctrl> + <R>**.

The following image shows an arrow that was rotated by 90°, 180°, 270° and 360°. Of course, the rotation takes place at the same location. For clarification purposes, the arrow was copied and moved after the rotation.

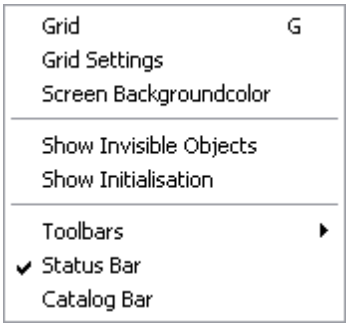


10.3.6.12 Reinitialise

You can select this command using the menu item or, for example, by pressing the **<space bar>** if the object in question is selected.
This function can only be used for a VLO object (template object). VLO objects can be reinitialised with this menu item.

10.3.7 View menu

Functions that support the creation of process diagrams are found in this menu.



10.3.7.1 Gridlines <g>

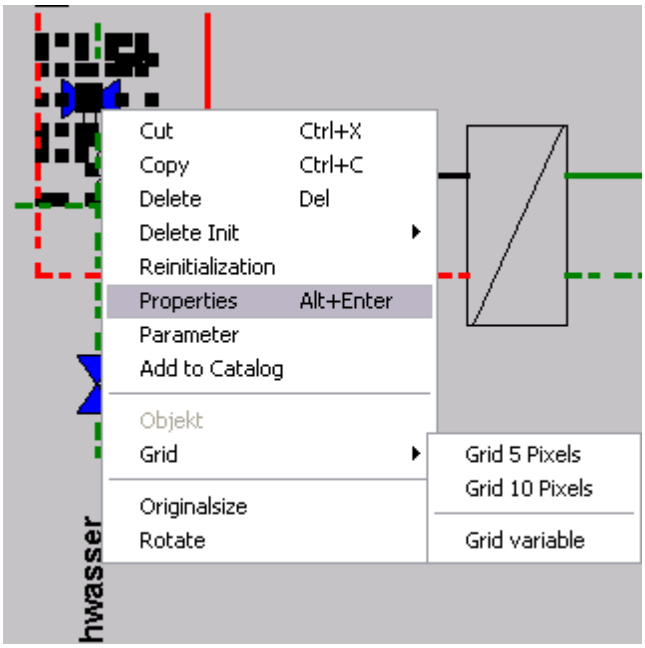
<g> or in the menu "**View > Gridlines**"

With this menu item, guide lines (so-called gridlines) which significantly simplify the alignment of graphical objects while drawing, are shown on the process diagram.

With the grid switched on, elements can only be drawn and placed on the displayed gridlines.

When switching to runtime mode, the gridlines are no longer displayed.

*The gridlines can also be switched on by marking any element, clicking the right mouse button and selecting the **Grid** command from the list that appears.*



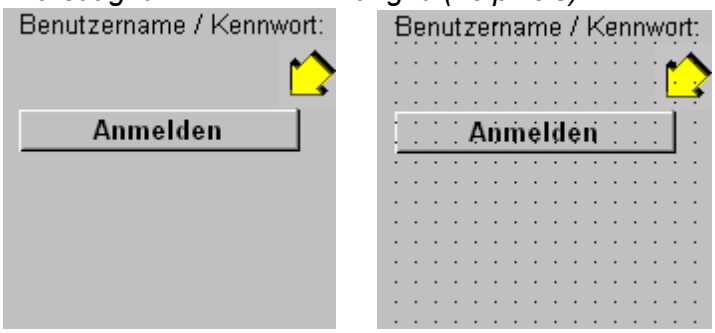
An additional choice appears:

Grid 5 pixels	Gridlines spaced at 5 pixels
Grid 10 pixels	Gridlines spaced at 10 pixels

Variable grid Spacing is freely selectable

If the gridlines are switched on for all windows (via the "Grid Settings" command), the <g> key no longer has any effect.

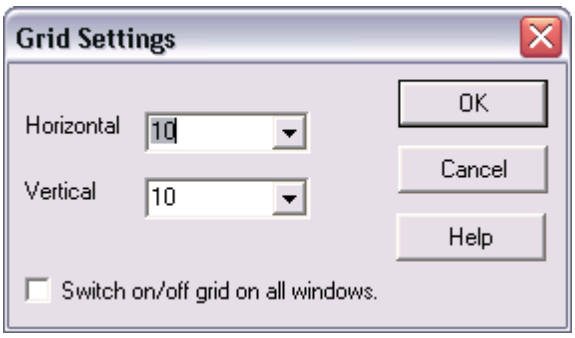
Without grid: *With grid (10 pixels):*



(Excerpt from a defined system diagram)

10.3.7.2 Grid Settings

This function enables the pixel spacing of the gridlines to be changed.

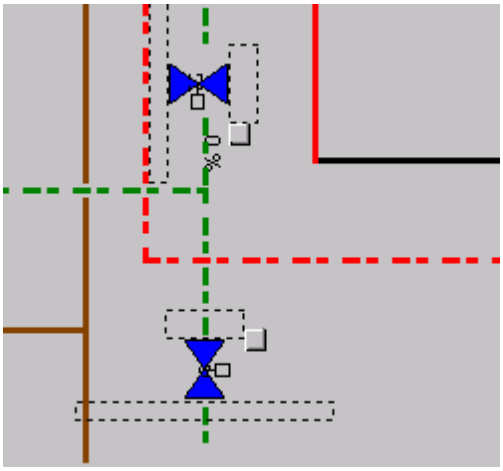


By marking the checkbox **"Switch Grid On/Off for All Windows"** the guide points are switched on for all process diagrams. In so doing, the <g> key loses its function.

We recommend switching the grid on for all windows. A graphical object can, however, be moved pixel-by-pixel (with the arrow keys).

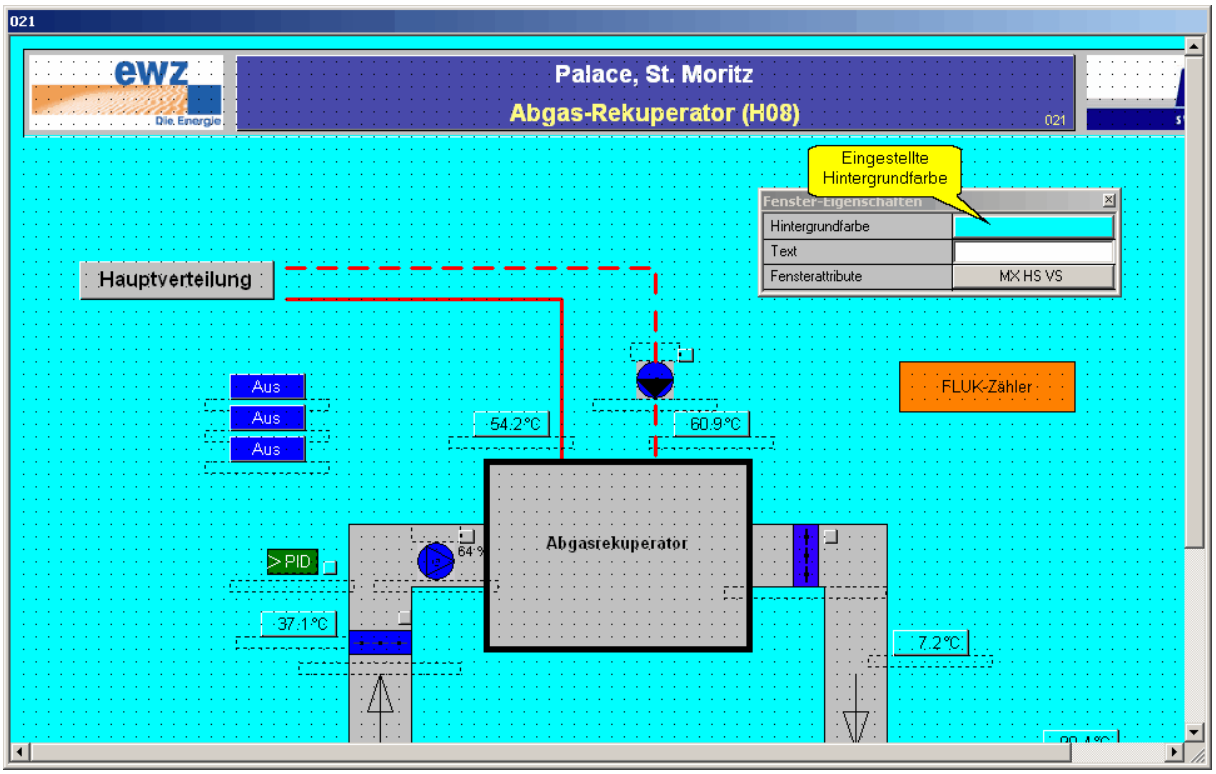
10.3.7.3 Show Hidden Objects

Shows the objects, buttons and images that you have defined as "hidden":



10.3.7.4 Diagram Background Colour

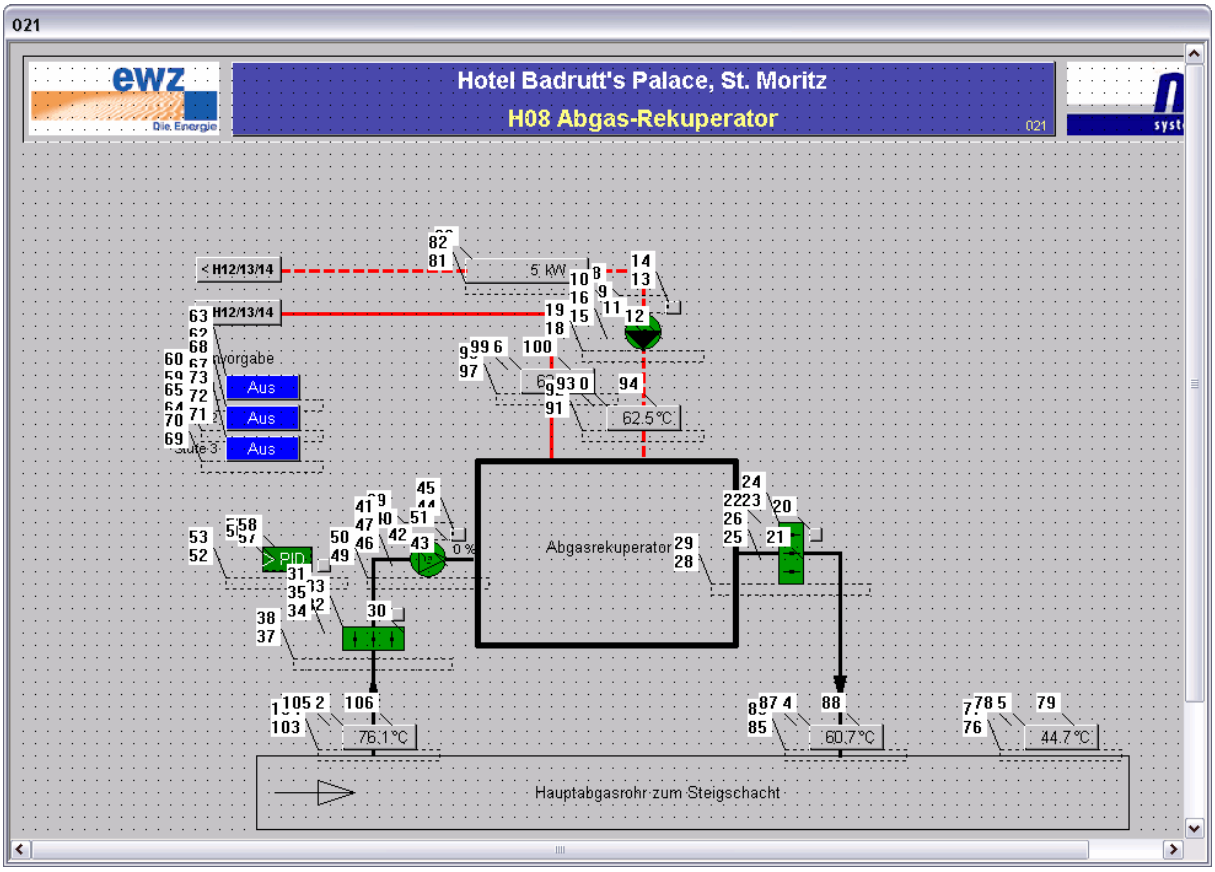
This function enables the background colour of a process diagram to be changed. A different background colour can be assigned to each process diagram:



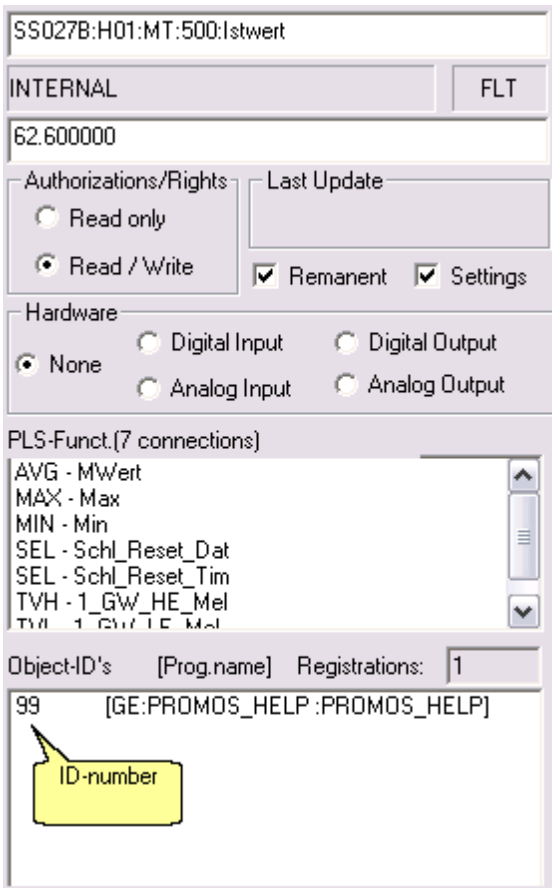
We recommend working with uniform background colours.

10.3.7.5 Display Initialisations

The command "**Display initialisations**" shows the objects registered with the DMS (VLO objects, etc.) in the form of a serial number within the process diagram:



The displayed numbers are also shown in the DMS:



10.3.7.6 Toolbars/Tool Lists

The toolbars/tool lists can be switched on/off here.

Main display:

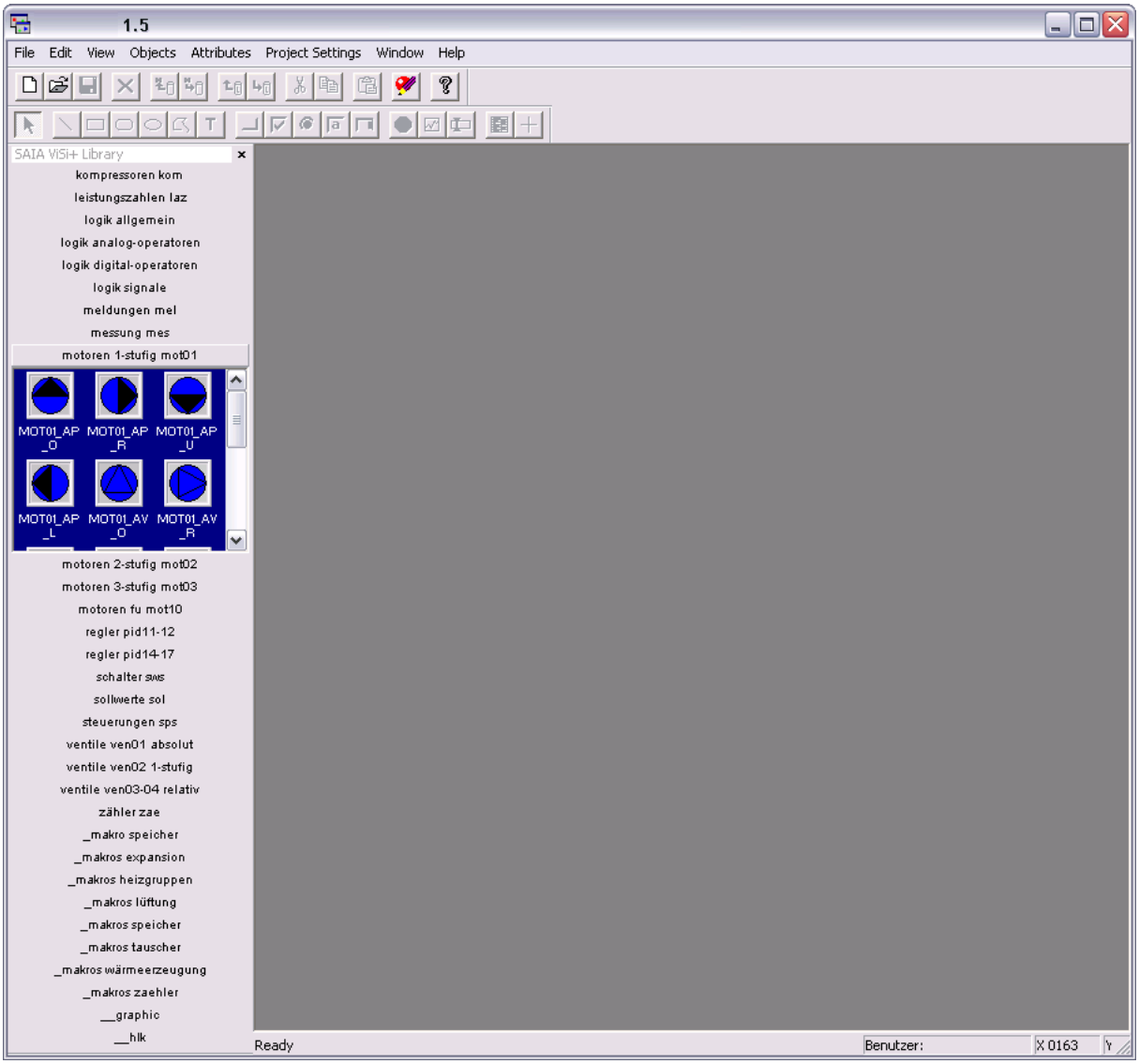
Drawing Tools

10.3.7.7 Status Bar

Enables the status bar at the bottom edge of the screen to be switched on/off.

10.3.7.8 Catalogue Bar

Opens or closes the catalogue bar/library at the left edge of the screen.



The following objects can be stored in a catalogue (library):

- Any graphical object
- Template objects
- Macros

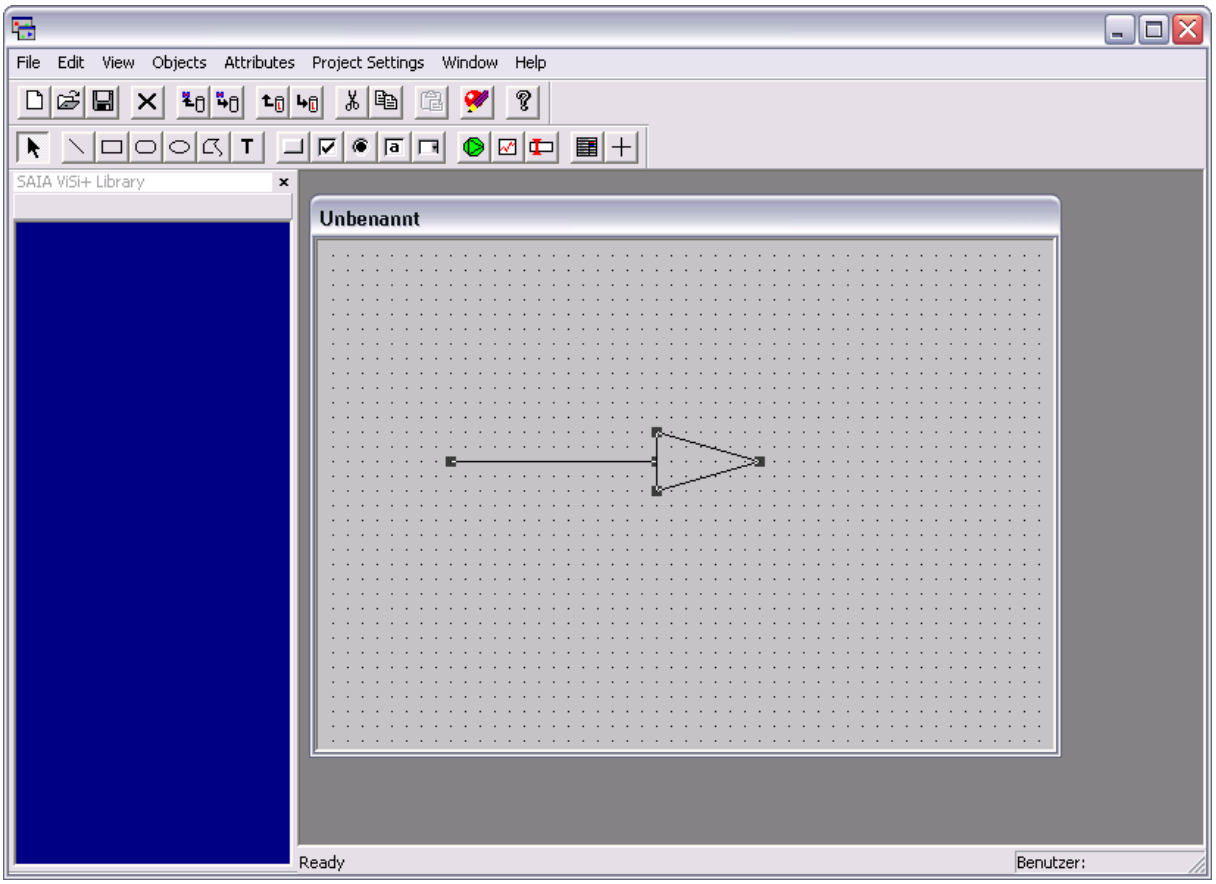
Using the catalogue, you can work can be done much more efficiently, because objects used frequently can be used, managed, modified and expanded with little effort. It is possible to set up and manage several catalogues. You can compile your own object library. In the above example, single stage motors (template object) or heating groups (macro) as well as often-used graphical objects such as an arrow can be stored in separate catalogues.

The catalogue objects can be pulled onto the process diagram with the mouse by means of drag and drop. If a catalogue object is a template object or a macro, the objects can be reinitialised.

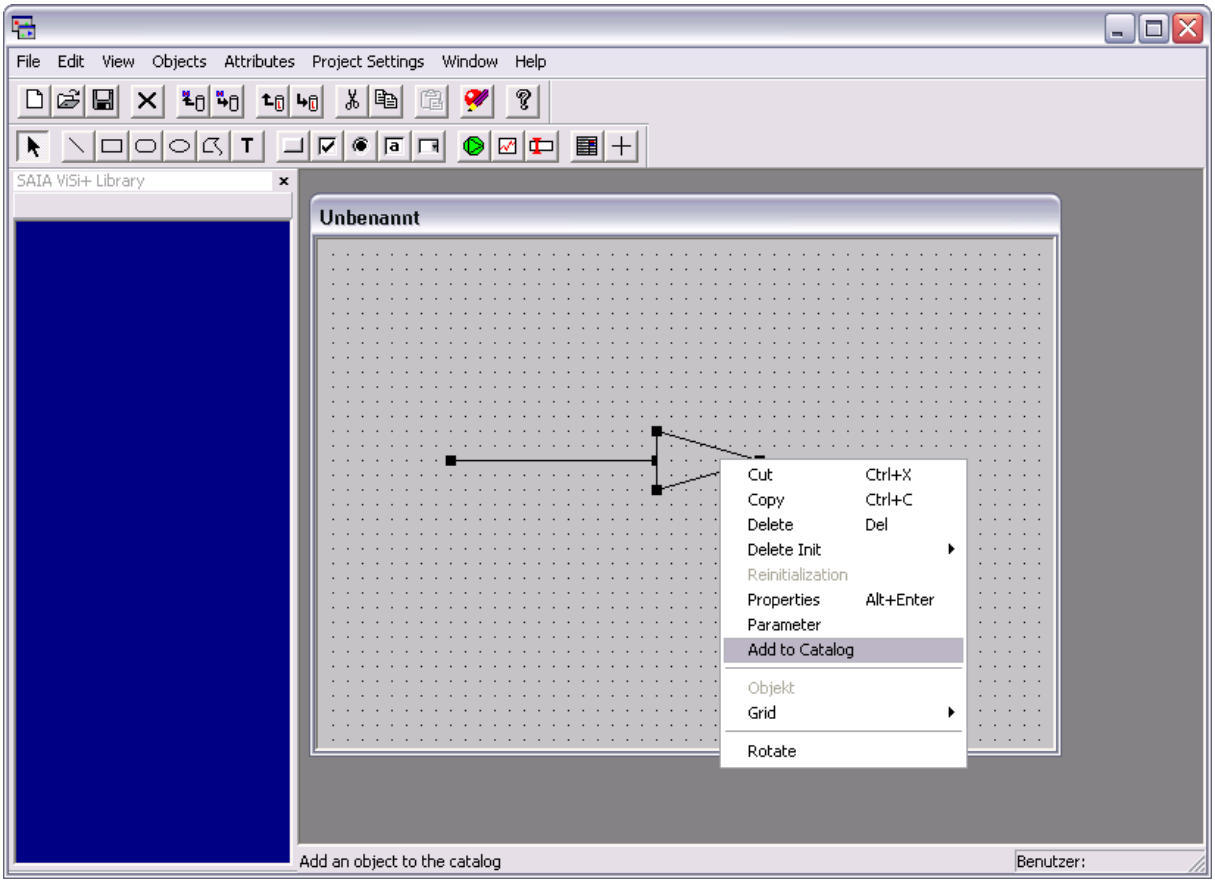
The creation of such a library is explained in the following:

Example 1: Any graphical object

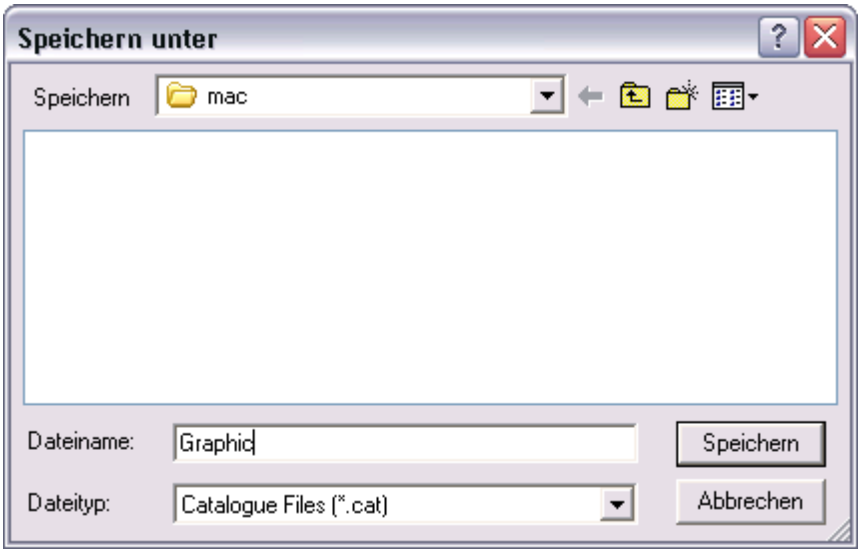
Draw an arrow by using a line and a polyline as an arrowhead:



Combine the two drawn objects by drawing a frame around the two objects with the mouse and then pressing the **<z>** key. Then press the right mouse button:



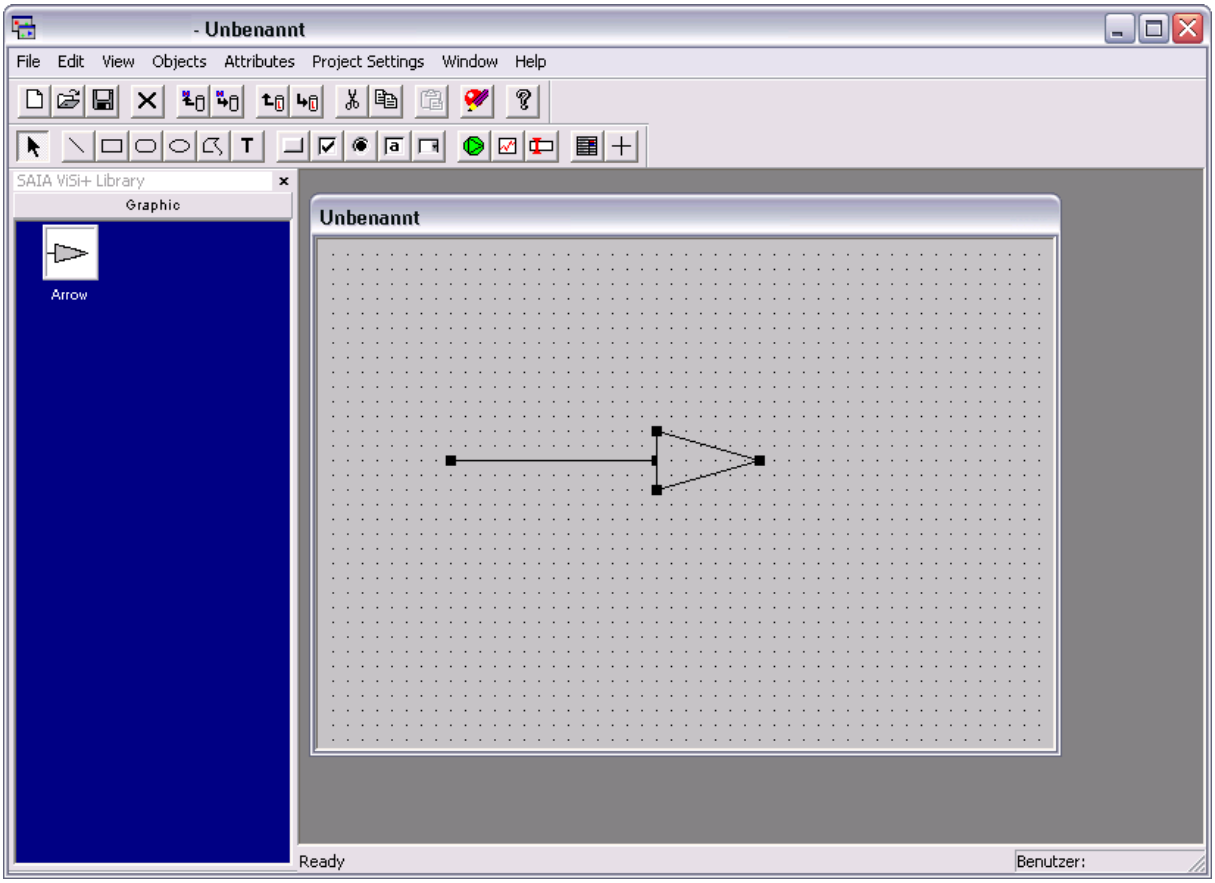
In the context menu, which then opens, select the menu item "**Add Object to Catalogue**". If there is no catalogue present in your project, the following dialog appears:



Enter the name for the catalogue file here and then click on **Save**.

*By default, the catalogues can be saved in the project folder under ...\mac. You can change this directory at any time. Then you should select the menu item **File > Save Catalogues** in order to save the data path for the catalogues under **...\cfg\cat.cfg**. The GE reads this file on start-up and automatically loads the catalogues located in this file.*

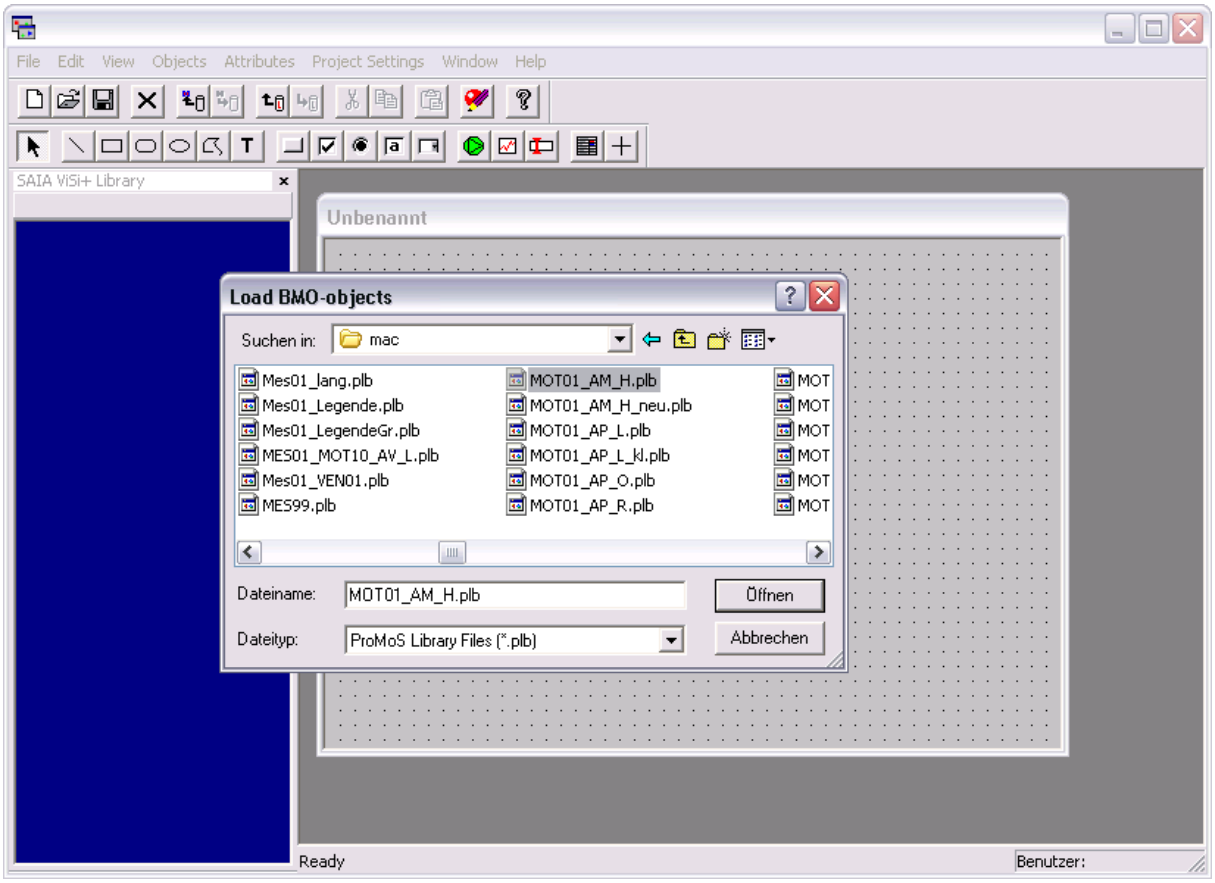
The arrow is now stored in the newly created catalogue. Then you can assign an appropriate name for the catalogue object by clicking on **New** with the mouse and overwriting the entry:



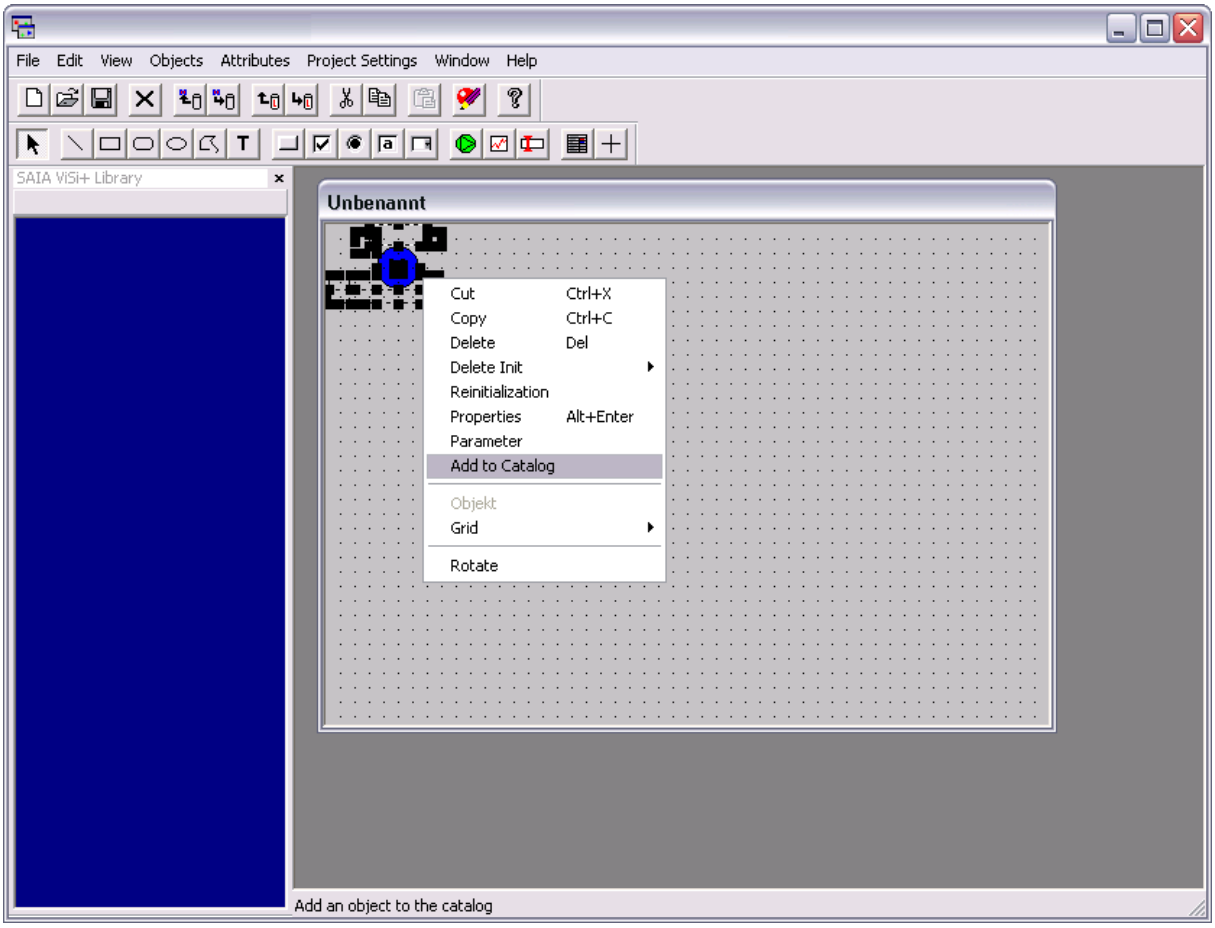
If you want to use a catalogue object in your process diagram, simply pull it to the appropriate location on your process diagram with the mouse by means of drag and drop.

Example 2: Template object

First select the template object that you would like to place on the catalogue bar, as described in the chapter **Load Template Object**:



Mark the template object. Now press the right mouse button and select the menu item "**Add Object to Catalogue**":

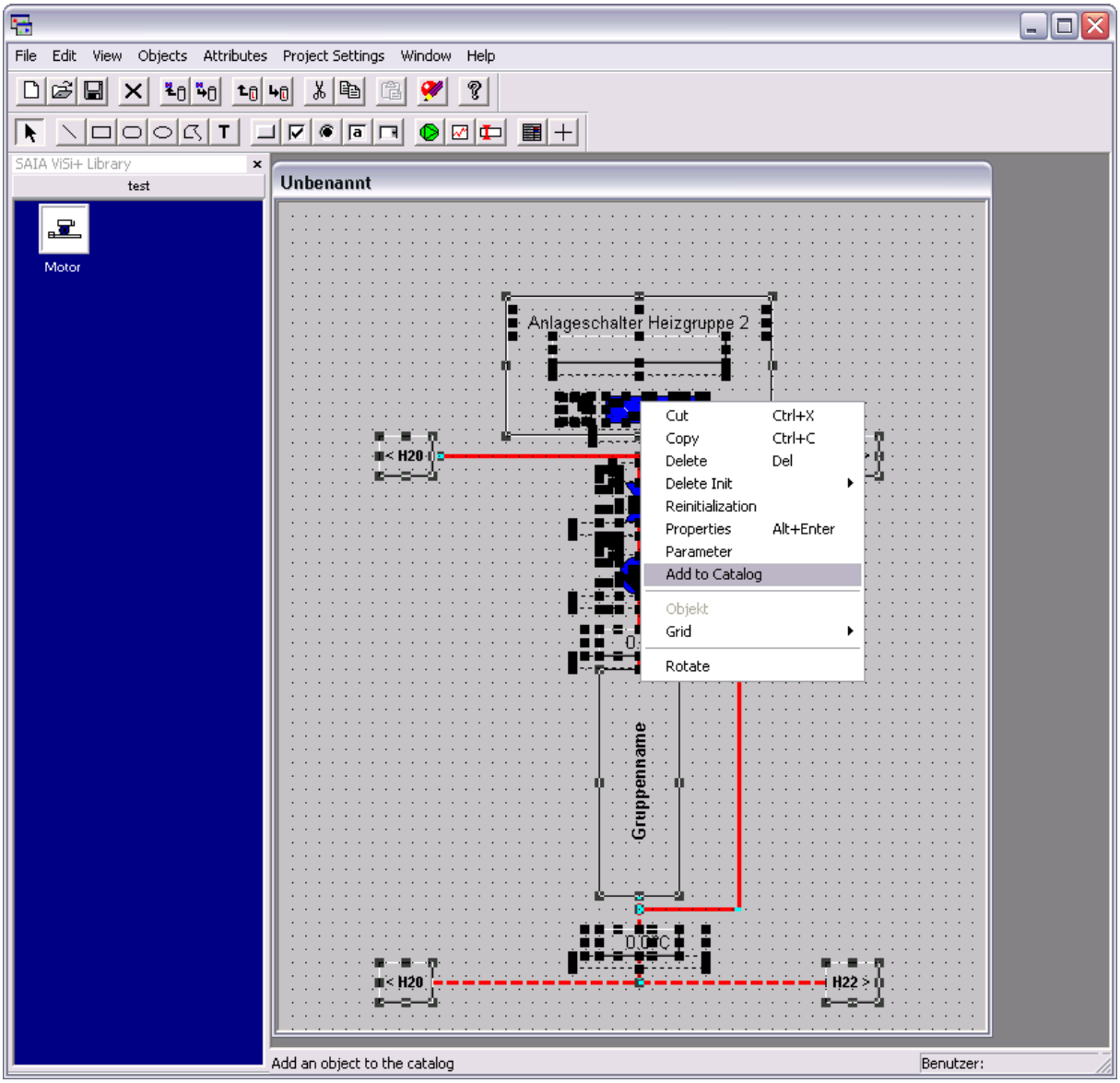


The template object is now stored in the active catalogue. If you pull the object onto the process diagram by means of drag and drop, you are automatically asked whether you would like to reinitialise the object.

When you create a template object and add it directly to the catalogue, it is not recognised as such when you drag it onto the process diagram. You must first save your object as a template object and then reload it. Only then can you add it to the catalogue as a template object.

Example 3: Macro

Select the template objects that you would like to use in the macro from the **Template Object Library** (see Example 2). If you have already save template objects in the catalogue, you can drag them onto the process diagram from the catalogue. Then reinitialise the template objects and add additional graphical objects, such as lines, buttons, etc., to complete your diagram. Then mark all objects that you would like to save in the catalogue as a macro. Press the right mouse button and select the menu item "***Add Object to Catalogue***".



The macro is not saved in the active catalogue. If you drag the object onto the process diagram by means of drag and drop, you are automatically asked whether you would like to reinitialise the macro.

Functions of the catalogue bar

The functions of the catalogue bar are called up by right-clicking on the heading or the blue area of the catalogue bar.

If you click on the heading, the following context menu appears:

Clicking on the blue area opens the following context menu:

New
Creates a new catalogue.

Open
Opens a saved catalogue.

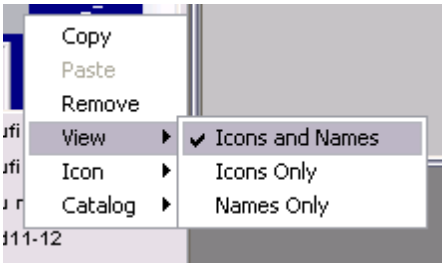
Close
Closes the active catalogue.

Copy
Copies a selected catalogue object to the clipboard. The copied object can be pasted onto the process diagram using <Ctrl>+<V>.

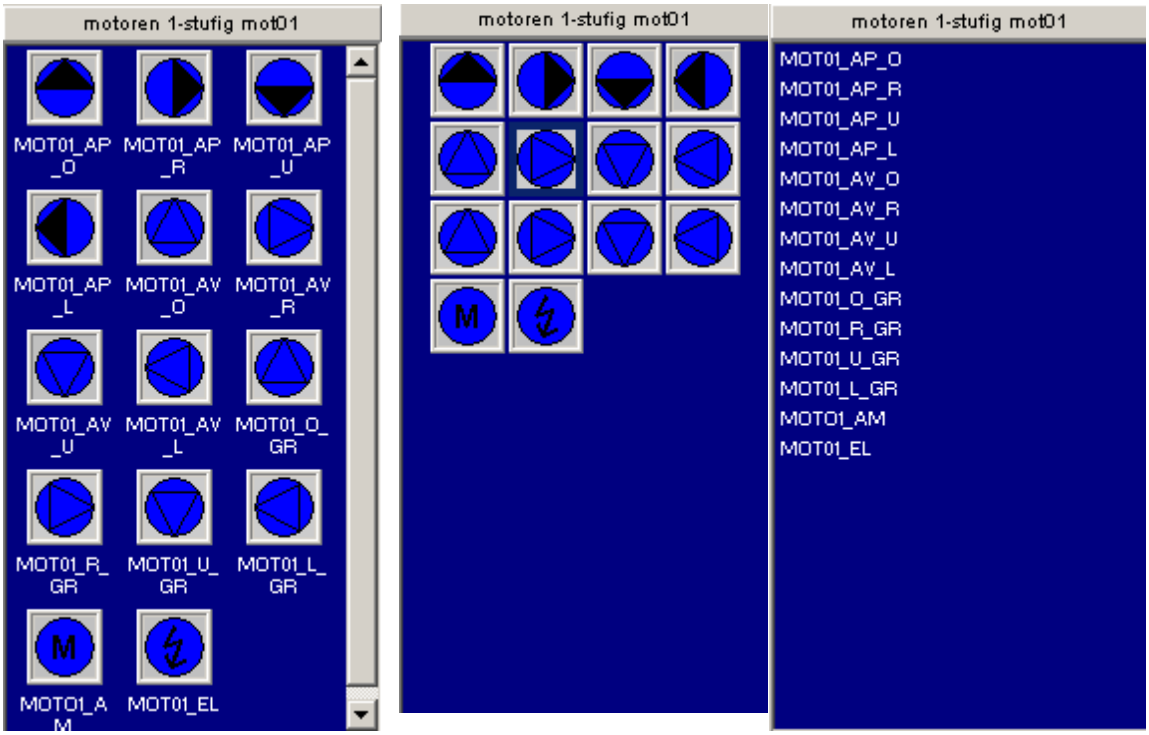
Paste
Pastes a catalogue object into the active catalogue from the clipboard.

Delete
Deletes a selected catalogue object.

View
Different views can be called up from here:

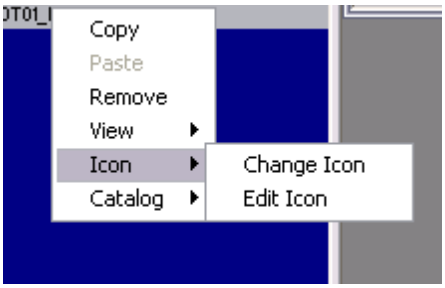


Icon and Names View **Icon View** **Names View**

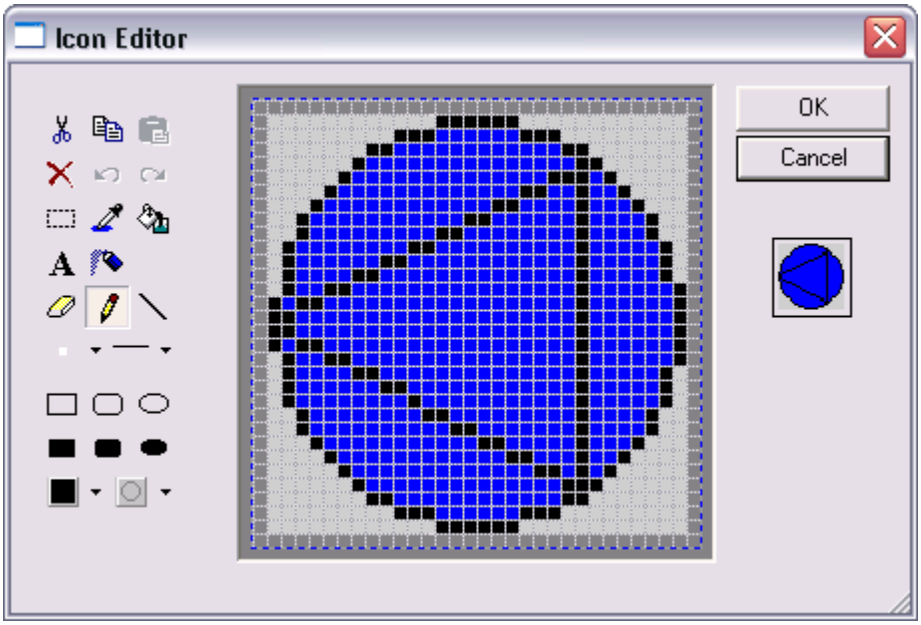


Icons and names are displayed | Only the icon is displayed | Only the name is displayed

Icon > Select Icon
Here a bitmap for the icon can be loaded.

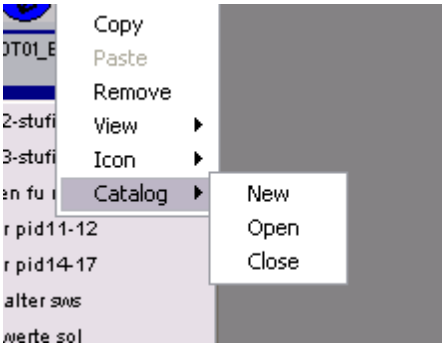


Icon > Edit Icon
The icons can be edited with an icon editor. The icon editor is an easily operated image editor. All commands are self-explanatory and typical for Windows.



.

Catalogue > New
Creates a new catalogue.



Catalogue > Open
Opens a saved catalogue.

*If you are not opening a catalogue from the default directory ...\mac, you should select the menu item **File > Save Catalogue**, in order to save the path data for the catalogue under **...\cfg\cat.cfg**. The GE reads this file on start-up and automatically loads the catalogues located in this file.*

Catalogue > Close
Closes the active catalogue.

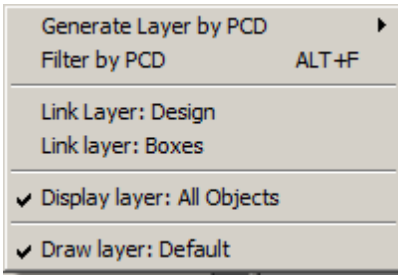
Changing background and text colours
If you would like to change the background and text colours, you must first close the GE. Then you must load the file **cat. cfg** in the directory **...\cfg** in a text editor, change the colour values, save the file and restart the GE.

```
[Window Colour]
Red=0
Green=0
Blue=128
[Text Colour]
Red=255
Green=255
Blue=255
```

The section under [Window Colour] contains the proportions of red, green and blue for the background, whereas the section under [Text Colour] contains the proportions of red, green and blue for the text. By default, the background colour is set to blue and the text is set to black.

10.3.8 Layer menu

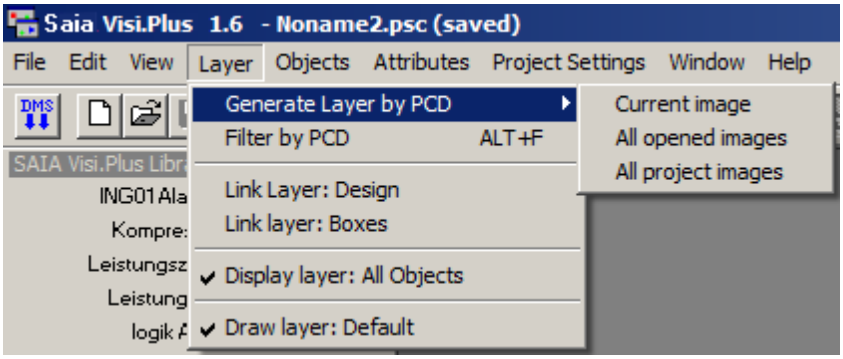
Functions that support the creation and editing of layers are found in this menu.



Working with layers allows complex diagrams to be divided up and the these contents can be made visible or invisible. When working with Mulit-DMS, the layers are required so that the GE knows on which DMS the individual VLOs are located! More information about Multi DMS use can be found in the chapter Multi DMS.

10.3.8.1 Layer generation according to PCD

The display layers are divided up automatically according to the first level in the DMS name.

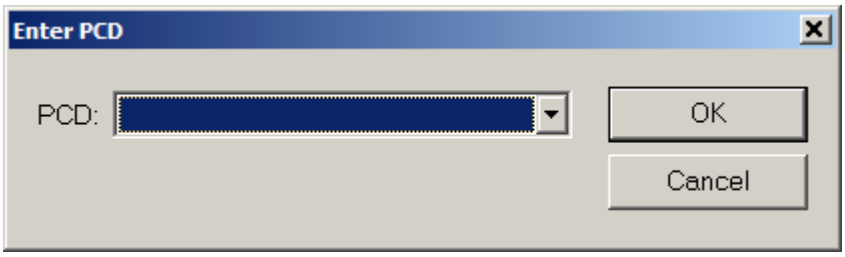


After this, it is possible to display or hide the objects PCD dependently.

*If **<All images of the project>** is actuated, all control screens of the VLOs are generated as well. Depending on the number of images in the scr folder of the project, this could take a few minutes!*

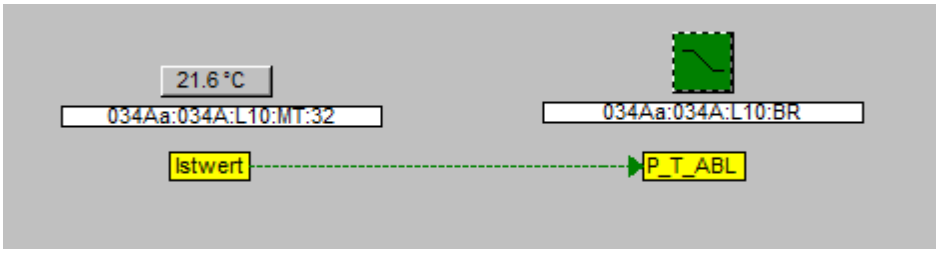
10.3.8.2 Filter by PCD

This function makes it possible to display only those objects which were created on the selected PCD.



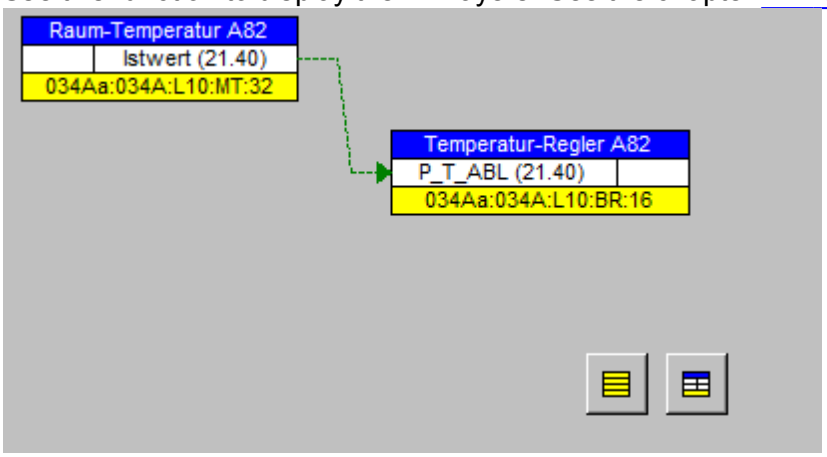
10.3.8.3 Linklayer:Design

Use this function to display the link layers. See the chapter Link objects



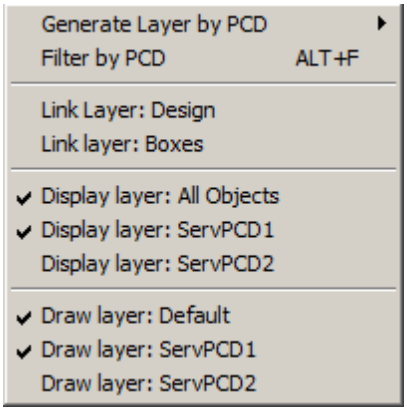
10.3.8.4 LinkLayer:Boxes

Use this function to display the link layers. See the chapter Link objects



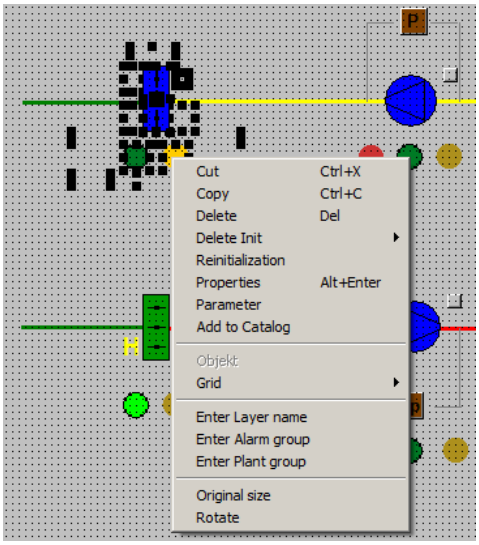
10.3.8.5 Display layer / Drawing layer

This function makes it possible to select which objects are displayed and at what layer drawing should currently take place.

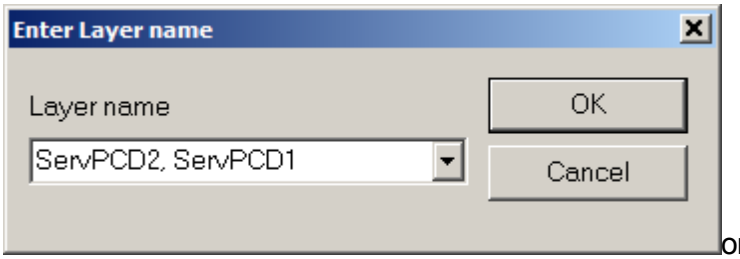


With this setting, a new object is assigned to the layer ServPCD1. In addition, the layers ServPCD1 and PCD2 are visible.

It is also possible to assign the objects to a layer later. To do so, select the objects in the screen and use the **"Specify layer name"** item in the context menu.

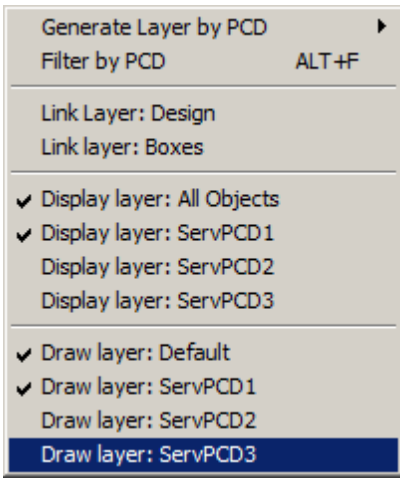


The previously created layers are selected in the mask. It is also possible to create a new layer by entering a new name in the drop-down list.



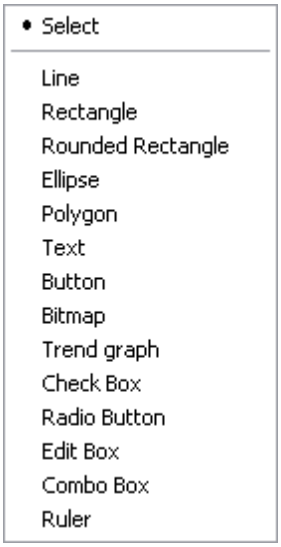
or

after this, the new layer is available in the menu:



10.3.9 Objects menu

Graphical objects for creating process diagrams are available under the **"Objects"** menu item. They are the same objects that are also found on the **"Drawing Tools"** toolbar.



10.3.9.1 General information about drawing graphical objects

As soon as one of the graphical objects has been selected, the **mouse pointer changes into a crosshair**. Small black squares can be seen while drawing. These identify the **handles** of the graphical object.

So that additional objects may be drawn, the current **graphical object is maintained** until a different graphical object or the Select icon  is selected or the background of the process diagram is briefly clicked on.
In order to **correct** or **change** the **attributes**, **dimension** and **position** of a graphical object, the graphical object must be selected.

The graphical object can be selected by first clicking on the Select icon  and then clicking on the graphical object with the mouse.

Selected objects are recognisable from their visible handles.

A graphical object must be selected to carry out the three following types of changes.

Changing the position
Move the mouse pointer over the graphical object, press and hold the left mouse button and drag the object to the desired position.

Changing the dimensions
Move the mouse pointer to one of the handles, press and hold the left mouse button and drag

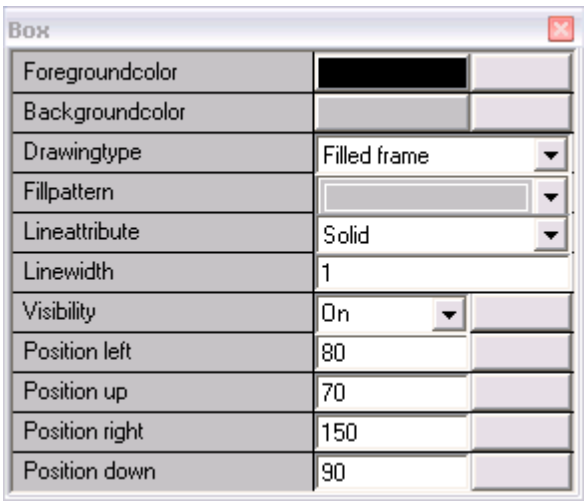
the handle to the desired position.

Changing the attributes
(see the following section, Attributes window)

10.3.9.2 Attributes window

 or **<ALT+ENTER>** or the "**Edit > Properties**" menu or press the right mouse button and select "**Properties**" or hold down the left mouse button for longer than 1 second on an element.

The attributes listed in the displayed dialog window are dependent on the selected graphics object. With this dialog window, the appearance and behaviour of the graphics object is defined.
The **<Tab>, <Shift + Tab>, <B> or <F>** keys or key combinations can be used to switch the properties window between the individual elements!



 The **Name column** describes the respective attribute.

 The**Attribute column** contains the target value for the attributes and shows their actual value.
Example:
Currently, the position of the box is screen position 397. If desired, this value can be adjusted to 400, etc.

 The **Initialisation column** enables the attributes of graphics objects to be defined dependent on a DMS data point.
Example:
The foreground colour (the actual colour of the box) can be made dependent on the status of a flag (DMS name). This process is called initialisation.

 *The Initialisation column always has priority over the Attribute column.*

All attributes and how they are changed or initialised are described in detail under Graphics attributes and Initialisation attributes.

The tables at the end of the graphics objects described below are intended to show which attributes they possess, how they can be initialised and where to find a detailed description of these attributes.

The table columns are divided up as follows.

Attribute
Lists all graphics attributes of the object.

Description
Indicates the page in this manual where the attribute is described in detail.

Initialisation
States whether the attribute can be initialised or not.

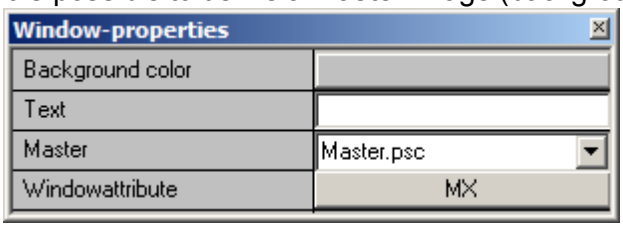
Initialisation description
States which page explains how and for what purpose the attribute can be initialised.

10.3.9.3 Process diagram/window/image attributes

A process diagram (window), like any other object, has attributes that can be changed.

The attributes of a process diagram are adjusted with the help of the "Window Attributes" dialog. To call up the dialog, click on the background of the current process diagram (not on the title bar).
Then the command can be selected in one of three ways:
In the same way as the Attributes window, with  or **<ALT+ENTER>** or in the menu "**Edit > Attributes**".

It is possible to define a master image (background image):



Background Colour
By clicking on the button in the right column the colour can be selected.

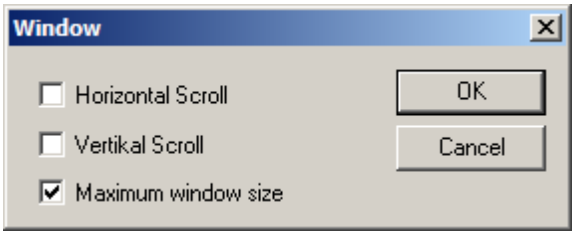
Text
Input field for the title bar of the process diagram.

Master
A master image (background image) can be specified here, which is also loaded automatically when the image is loaded. The master image is created separately. It can only be edited in the original file. If it is integrated into another image, the objects can no longer be edited. Objects like buttons, however, can be operated.

In many cases images consist of a header and footer, which are often identical for many images in a project. They should be created separately in a master image. If you would like to use this template, you only have to specify it in the Attributes window. By doing so, it is no longer necessary to create a new header and footer for each new image.

It is possible to nest master diagrams in Version 1.6 and higher. This means that a master diagram can be inserted in master diagram itself.

Window Attributes
Calls up the dialog window of the same name.



Horizontal Scroll
Shows/hides the image scroll bar at the bottom edge of the process diagram.

Vertical Scroll
Shows/hides the image scroll bar at the right edge of the process diagram.

These two functions are useful if a smaller screen resolution is set on the development computer than on the destination computer.

Maximum Size
Always shows the maximum possible area of the process diagram in runtime mode. If this function is not activated, the process diagram is shown at the saved size.

10.3.9.4 Select

This tool is used to **Select Objects**.
Additional objects can be selected by using the **<SHIFT>** key and the **left mouse button** in combination.

If **multiple objects** are to be selected simultaneously, a **box** can be drawn around all elements to be selected, by simultaneously pressing and dragging with the left mouse button.

Objects which should not be included can then be deselected by pressing the **<SHIFT>** key while clicking on the respective object with the mouse.

10.3.9.5 Line

or in the menu **"Objects > Line"**

The **starting point of the line** is defined by pressing and holding the left mouse button. Then, to define the **end point of the line**, the mouse button must be released.

Descriptions for attributes and initialisations are located:

 **Line**

Attribute	Description	Initialisation	Initialisation Description
Foreground Colour	Line Colour	yes	Colour Change
Line Attribute		no	
Line Width	Width of the line	no	
Visibility		yes	Visibility
Position X1		yes	
Position Y1		yes	
Position X2		yes	
Position Y2		yes	
Delete Initialisation			

10.3.9.6 Box

 or in the menu "**Objects > Rounded box**"

The **starting point of the box** is defined by pressing and holding the left mouse button. Then, to define the **end point of the box**, the mouse button must be released.

Descriptions for attributes and initialisations are located:

 **Rechteck**

Attribute	Description	Initialisation	Initialisation Description
Foreground Colour	Line Colour	yes	Colour Change
Background Colour	Surface Colour	yes	Colour Change
Drawing Type		no	

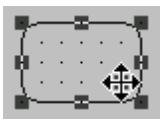
Rechteck

Fill Pattern	no
Line Attribute	no
Line Width	no
Visibility	yes
Left Position	yes
Top Position	yes
Right Position	yes
Bottom Position	yes
Delete Initialisation	

10.3.9.7 Rounded Box

or in the menu "**Objects > Rounded box**"

The **starting point of the rounded box** is defined by pressing and holding the left mouse button. Then, to define the **end point of the rounded box**, the mouse button must be released.



The **rounded corners of the box** can be modified with the additional resizing handles in the lower right corner of the **rounded box**. To do so, place the mouse pointer on the resizing handle (cursor becomes crossed arrows), press and hold the left mouse button and drag the resizing handle accordingly.

Descriptions for attributes and initialisations are located:

Rounded Box

Attribute	Description	Initialisation	Initialisation Description
Foreground Colour	Line Colour	yes	Colour Change
Background Colour	Surface Colour	yes	Colour Change

 **Rounded Box**

Drawing Type	no
Fill Pattern	no
Line Attribute	no
Line Width	no
Visibility	yes
Left Position	yes
Top Position	yes
Right Position	yes
Bottom Position	yes
Delete Initialisation	

10.3.9.8 Ellipse

 or in the menu "**Objects > Ellipse**"

The **starting point of the ellipse** is defined by pressing and holding the left mouse button. Then, to define the **end point of the ellipse**, the mouse button must be released.

Descriptions of attributes and initialisations are located:

 **Ellipse**

Attribute	Description	Initialisation	Initialisation Description
Foreground Colour	Line Colour	yes	Colour Change
Background Colour	Surface Colour	yes	Colour Change
Drawing Type		no	
Fill Pattern		no	
Line Attribute		no	
Line Width		no	
Visibility		yes	

Ellipse

Left Position	yes
Top Position	yes
Right Position	yes
Bottom Position	yes
Delete Initialisation	

10.3.9.9 Polygon/Polyline

or in the menu "**Objects > Polygon**"

The **starting point of the polygon** is defined by pressing and holding the left mouse button, then in order to define the **next corner**, etc. Each of these points is the endpoint of the preceding segment and the starting point of the next segment. The entire object behaves like the line is made out of an elastic band. Therefore, after the polygon is completed, the **points can be moved with the mouse** (for the procedure, see under General information about drawing graphical objects).

A **Freehand Polyline** is obtained by continuously pressing the left mouse button while changing the mouse position.

Double-clicking concludes the setting of points for the Polygon Line.

Additional or unnecessary points can be neither added nor removed. To achieve this, the entire polygon must be deleted and redrawn.

Descriptions of attributes and initialisations are located:

Polygon/Polyline

Attribute	Description	Initialisation	Initialisation Description
Foreground Colour	Line Colour	yes	Colour Change
Background Colour	Surface Colour	yes	Colour Change
Drawing Type		no	
Fill Pattern		no	

Polygon/Polyline

Line Width	no
Action	Yes
Visibility	yes
Delete Initialisation	

10.3.9.10 Text/Text Field

or in the menu "**Objects > Text**"
A Text Field is defined by a square frame.

The **starting point of the Text Box** is defined by pressing and holding the left mouse button. Then, to define the **end point of the Text Box**, the mouse button must be released.

The text itself is entered in the Attributes window under "**Text**". In doing so, additional attributes of the text, such as font, text alignment, colour, etc. can also be initialised.

Descriptions of attributes and initialisations are located:

Text/Textfeld

Attribute	Description	Initialisation	Initialisation Description
Foreground Colour	Line Colour	yes	Colour Change
Background Colour	Surface Colour	yes	Colour Change
Text Colour		yes	Colour Change
Drawing Type		no	
Line Width		no	
Text		yes	
Font	Windows Fonts	no	
Alignment		no	
Visibility		yes	
Left Position		yes	
Top Position		yes	
Right Position		yes	

Text/Textfeld

Bottom Position	yes
Delete Initialisation	

10.3.9.11 Button

or in the menu "**Objects > Button**"

A button graphical object enables other process diagrams to be called up, dialog windows to be opened to make value entries, PLC flags to be controlled and much more.

This graphical object is also defined by a rectangular frame.

The **starting point of the Text Box** is defined by pressing and holding the left mouse button. Then, to define the **end point of the Text Box**, the mouse button must be released.

The appearance, as well as functions and actions are defined with the Attributes window. In the process, additional attributes of the button label, such as font, text alignment, colour, etc. can be adjusted (for more information see Text Colour 88).

Descriptions of attributes and initialisations are located:

Button

Attribute	Description	Initialisation	Initialisation Description
Foreground Colour	Top Line Colour	yes	Colour Change
Background Colour	Surface Colour	yes	Colour Change
Foreground Colour 2	Bottom Line Colour	no	
Text Colour		yes	Colour Change
Drawing Type		no	
Text		yes	
Font	Windows Fonts	no	
Alignment	Left/right, etc.	no	
Action		yes	e.g. Image Change
Visibility		yes	
Left Position		yes	

Button

Top Position	yes
Right Position	yes
Bottom Position	yes
Delete Initialisation	

10.3.9.12 Checkbox

or in the menu "**Objects > Checkbox**"

A checkbox enables the value of a digital or analogue data point to be set from 0 to 1 or vice-versa (toggling), if the **Action** attribute is initialised.

A **checkbox** is defined by a square frame.

The **starting point of the Checkbox** is defined by pressing and holding the left mouse button. Then, to define the **end point of the checkbox**, the mouse button must be released.

The frame of the checkbox contains the checkbox symbol (white box) as well as its name.The **name** itself is entered in the Attributes window under "**Text**". The size of the checkbox symbol cannot be changed.

Descriptions of attributes and initialisations are located:

Kontrollkästchen

Attribute	Description	Initialisation	Initialisation Description
Foreground Colour	Line Colour	yes	Colour Change
Background Colour	Surface Colour	yes	Colour Change
Text Colour		yes	Colour Change
Drawing Type		no	
Line Width		no	
Text		yes	
Font		no	
Alignment	e.g. left/right, etc.	no	
Action		yes	Setting a status

Kontrollkästchen

Visibility	yes
Left Position	yes
Top Position	yes
Right Position	yes
Bottom Position	yes
Delete Initialisation	

10.3.9.13 Radio Button

or in the menu **"Objects > Radio Button"**

A **Radio Button** enables the value of both an analogue and a digital data point to be set if the **Action** attribute is initialised.

A **Radio Button** is defined by a square frame.

The **starting point of the Radio Button** is defined by pressing and holding the left mouse button. Then, to define the **end point of the Radio Button**, the mouse button must be released.

The frame of the **Radio Button** contains the radio button symbol (white cross) and its name. The **name** itself is entered in the Attributes window under "**Text**". The size of the radio button symbol itself cannot be changed.

Descriptions of attributes and initialisations are located:

Radio Button

Attribute	Description	Initialisation	Initialisation Description
Foreground Colour		yes	Colour Change
Background Colour		yes	Colour Change
Text Colour		yes	Colour Change
Drawing Type		no	
Line Width		no	
Text		yes	
Font	Windows Fonts	no	

Radio Button

Alignment	no	
Action	yes	Setting a value
Visibility	yes	
Left Position	yes	
Top Position	yes	
Right Position	yes	
Bottom Position	yes	
Delete Initialisation		

10.3.9.14 Input Field

or in the menu "**Objects > Input Field**"

An **Input Field** enables a data point to be set to a specific value in runtime mode if the **Action** attribute is initialised. By directly inputting a number, a string, or a time or date, both digital and analogue data points as well as the data points of the type STR (string) can be set.

An **Input Field** is defined by a square frame.

The **starting point of the Input Field** is defined by pressing and holding the left mouse button. Then, to define the **end point of the Radio Button**, the mouse button must be released.
Input Fields are shown in the RunTime Mode always on top. It is not possible to hide it behind a other element.

Descriptions of attributes and initialisations are located:

Eingabefeld

Attribute	Description	Initialisation	Initialisation Description
Background Colour	Surface Colour	yes	Colour Change
Text Colour		yes	Colour Change
Text		yes	Text Display/Change
Font	Windows Fonts	no	
Alignment	Left/right, etc.	no	
Action		yes	Entry of a value

Eingabefeld

Visibility	yes
Left Position	yes
Top Position	yes
Right Position	yes
Bottom Position	yes
Delete Initialisation	

10.3.9.15 Combo Box

or in the menu "**Objects > Combo Box**"

A **Combo Box** enables a data point to be set to a specific value in runtime mode. This involves selecting the value from a list. The list is generated with the initialisation of the **Action** attribute. Both digital and analogue data points as well as data points of the type STR (string) can be set.

This graphical object is also defined by a rectangular frame.

The **starting point of the Combo Box** is defined by pressing and holding the left mouse button. Then, to define the **end point of the Radio Button**, the mouse button must be released.

Combo Boxes are only modifiable in their width, not in their height. If the box was made too large during creation, it can be adjusted in height and by changing the font size.
Combo Box are shown in the RunTime Mode always on top. It is not possible to hide a Combo Box behind a other element.

Descriptions of attributes and initialisations are located:

Kombinationsfeld

Attribute	Description	Initialisation	Initialisation Description
Background Colour	Surface Colour	yes	Colour Change
Text Colour		yes	Colour Change
Text		yes	Text Display/Change
Font	Windows Fonts	no	
Action		yes	Selection of a value
Visibility		yes	

Kombinationsfeld

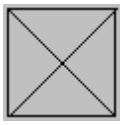
Left Position	yes
Top Position	yes
Right Position	yes
Bottom Position	yes
Delete Initialisation	

10.3.9.16 Bitmap

or in the menu "**Objects > Button**"

A bitmap is defined by a square frame.

The **starting point of the bitmap** is defined by pressing and holding the left mouse button. Then, in order to define the **end point of the bitmap**, the mouse button must be released.



The cross appearing in the box serves as a placeholder for an image file to be defined in the Attributes window under "**Icon Name**".

A description of the Icon Editor or Bitmap Editor can be found under "The Icon Editor".

Descriptions of attributes and initialisations are located:

Bitmap

Attribute	Description	Initialisation	Initialisation Description
Icon Name	Filename	yes	Icon Change
Action	Only Initialise	yes	e.g. Image Change
Visibility		yes	
Left Position		yes	
Top Position		yes	
Right Position		yes	

Bitmap

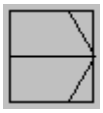
Bottom Position	yes
Delete Initialisation	

10.3.9.17 Trend graph

or **"Objects > Graph" in the menu**

A trend graph is defined using a square frame. We recommend not drawing the box smaller than the size of a credit card at first. After inputting definitions in the attributes window under "**Graphs**", up to 8 trend graphs including auxiliary lines and scales can be shown in this box.

The **starting point of the trend graph frame** is set by pressing and holding the left mouse button. Then, to set **the end point of the trend graph frame**, the mouse button must be released.



The arrow pointing to the right appearing in the box designates this box as a placeholder for a trend graph to be defined (for more information about this, see the **Trend graph examples 1–8**).

All attributes of the "**Trend**" graphics object are to be initialised as already described for the other graphics objects. A trend graph must have been drawn for the initialisation of the attribute "**Graphs**".**The creation of trend graphs is described after the following table on the basis of 8 detailed examples**.

Descriptions for attributes and initialisations are located:

Trendkurve

Attribute	Description	Initialisation	Initialisation Description
Foreground Colour	Line Colour	yes	
Background Colour	Surface Colour	yes	
Foreground Colour 2		no	
Text Colour		yes	
Font		no	
Graph	Only Initialise	yes	Definition of values

Trendkurve

Visibility	yes
Left Position	yes
Top Position	yes
Right Position	yes
Bottom Position	yes
Delete Initialisation	

In order to be able to display trend data in Visi.Plus, the following conditions must be fulfilled:

- *A valid licence must be available for the PDBS and HDA modules!*
- *PDBS (database) and HDA (historical data) must have been opened! It is easiest to open the HDA on project start-up (see the chapter HDA for more information).*
- *The respective data point must have an entry in the PET under the column "**Trend**" (for more information see the following examples and the chapter PET).*

*Trend graph objects do not display their data **immediately** after definition of their attributes. This depends on the **length of the time window** and the set **cycle time**.*

*Further explanations for the **Trend graph** graphics object are provided on the basis of the following examples.*

10.3.9.17.1 Example 1: Inserting a trend graph in the PET

Unfortunately, for reasons attributable to the system it is not possible to explain a trend graph with the simulated data points of **Group1** from the DMS, as with other graphics objects. Therefore the introductory project is used for clarification.

If it has not already been done, start the**Visi.Plus** project, log on and call up the Graphical Editor's (GE) edit mode by pressing the key **<e>**. To do so, proceed as follows:

Procedure:

1. Start up the Visi.Plus Project Manager via Start > **[Programs]** > **[Visi.Plus]** > **[Project Settings]** (or your own shortcut created on the desktop)

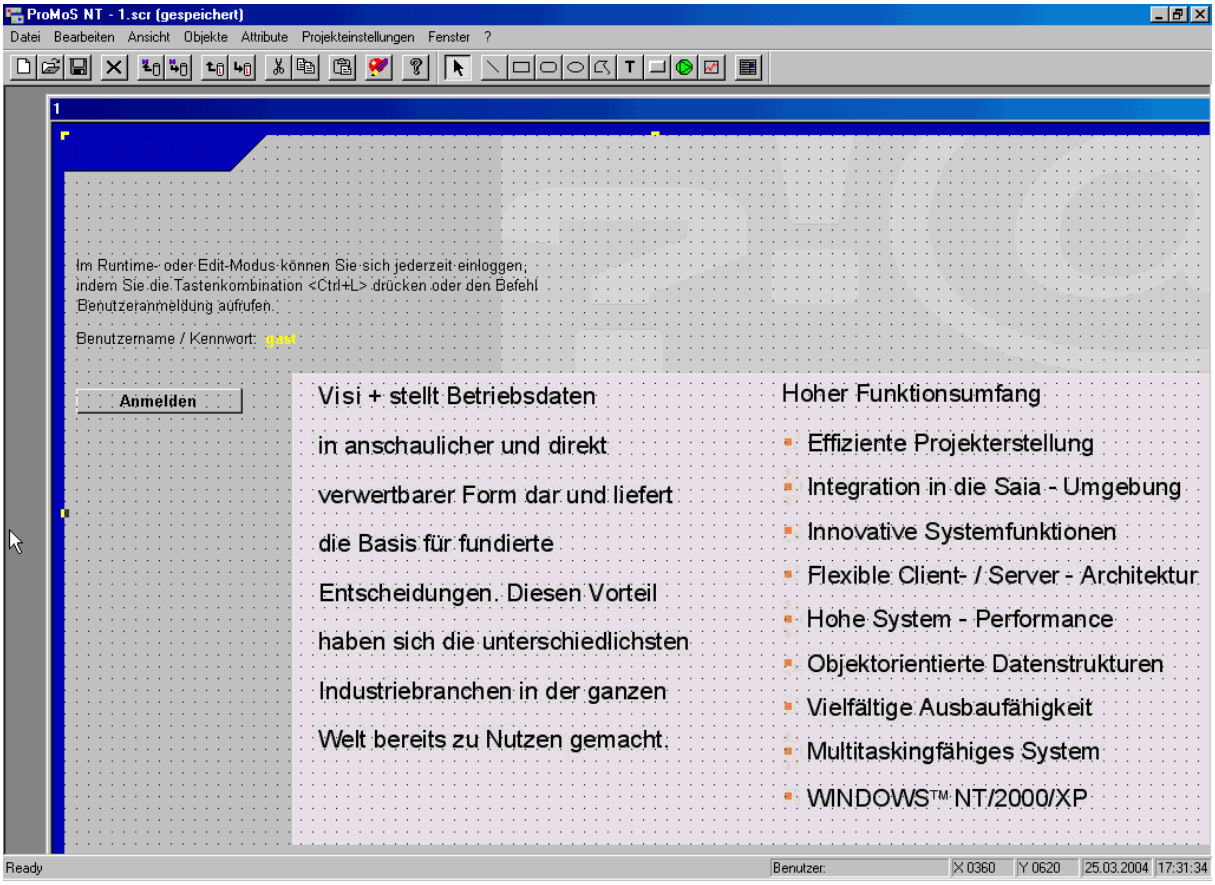
2. **Select Visi.Plus demo project and click <Start>**

3. Log in by clicking on the **<Login>** button or by pressing the keys **<CTRL> + <L>**

4. Enter the word "**guest**" in both the "**User**" and "**Password**" fields, then confirm with **<ENTER>**

5. **Click on the screen with the mouse and press the key <e>**

If everything is correct, the screen will display approximately the following content:



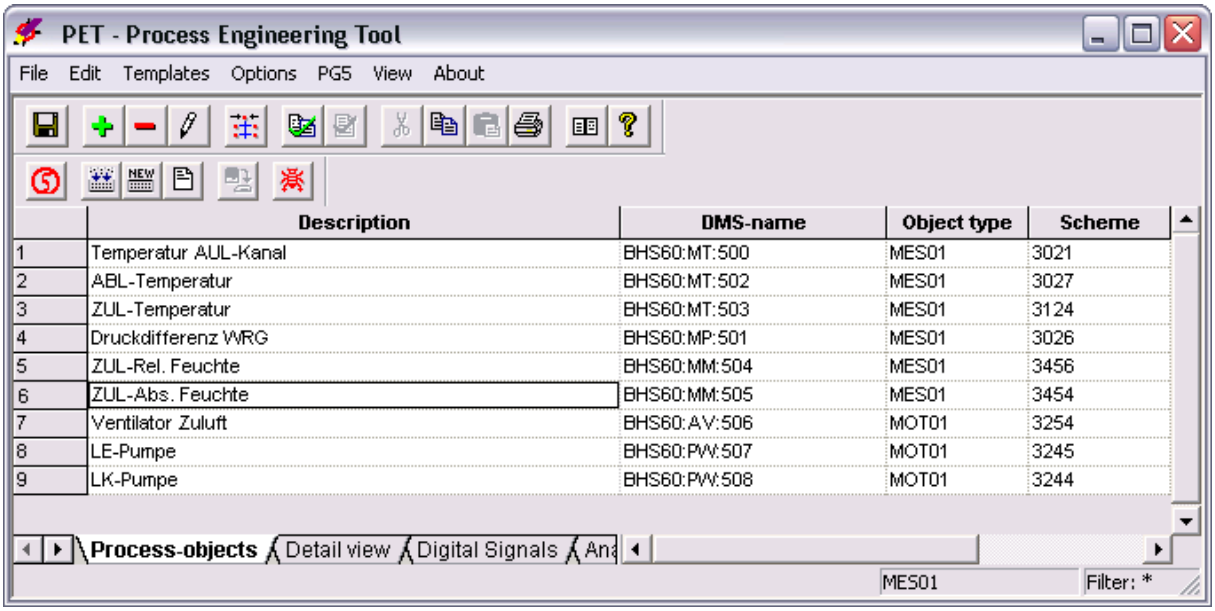
The displayed process diagram **1** can now be closed with  .

Before a new process diagram can be drawn with a trend graphics object and its control buttons, it is advisable to define which data point(s) to register as a trend.

 ***Trend acquisition** also requires the **HDA** (History Data Acquisition) module in addition to the **DMS** and **PDBS** modules. Defining which data points are required for trending in the PET will tell the HDA the **time period** for registering them and at what **intervals** the registration should occur. After entry in the PET the HDA must be **restarted**. This is the only way the HDA can determine which new data points must be acquired.*

Start the **PET** with  in **the GE**.

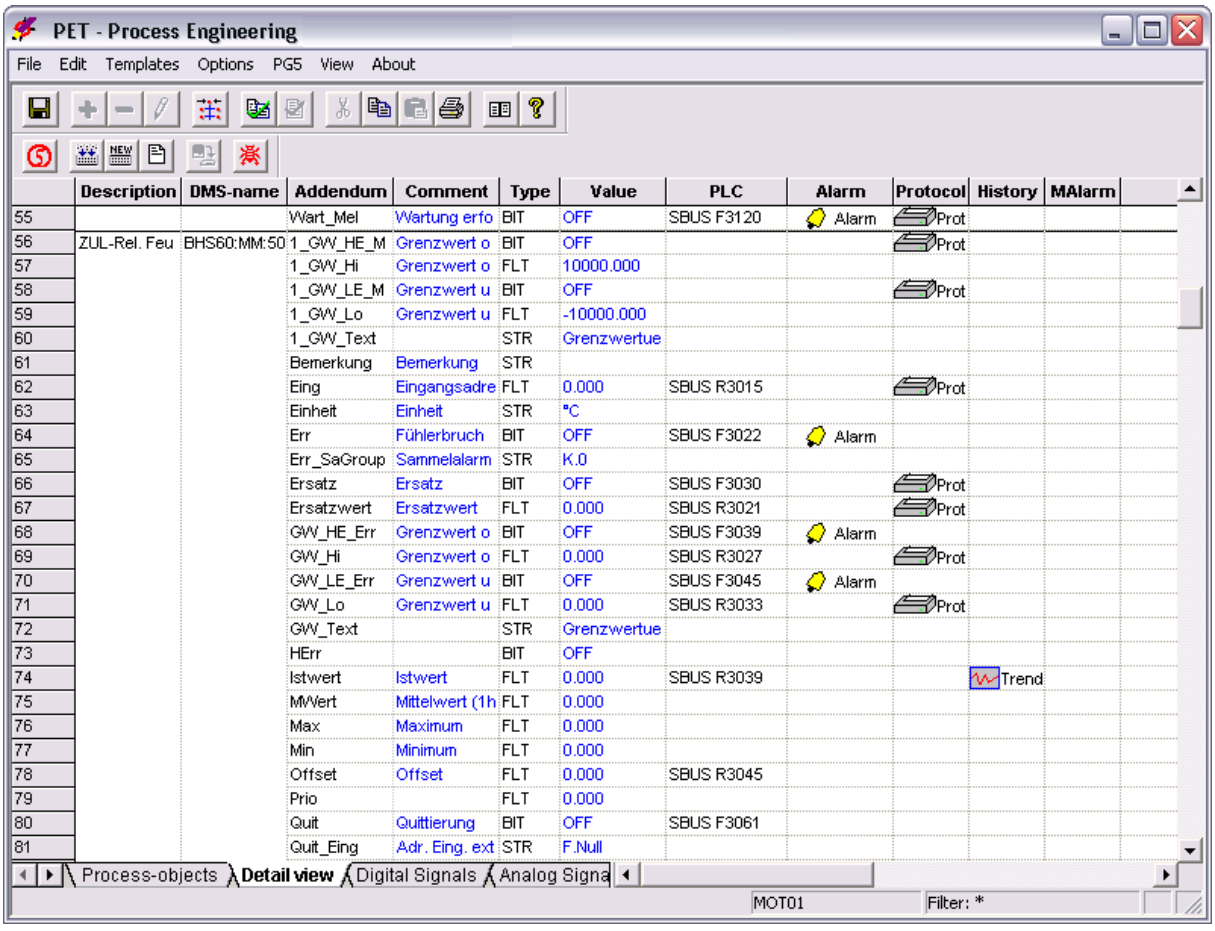
The system objects are listed in the demo project **Visi.Plus**:



Select **Detail View** (thumb index) and scroll to the system object with the **DMS Name** "**BHS60:MM:504**" up to the "**Actual Value**" ("**Attribute**" column).

The 

icon can already be seen in the "**Trend**" column. This icon is displayed when trending has already been activated. Therefore, any empty fields in the same column do not have active trending.



*If the Trend column is not visible, it can be activated via "**View > Trend Column**" in the menu.*

Trend graphs can be inserted for any **Attribute**. This occurs by clicking in a free field in the **Trend** column near the corresponding attribute.

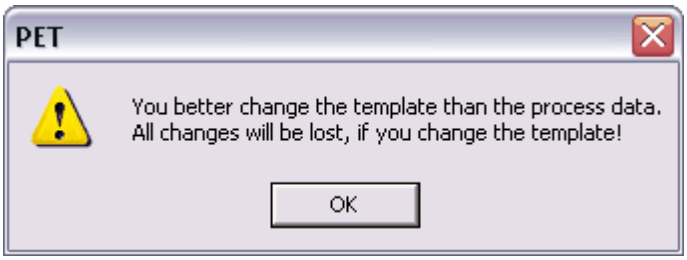
All attributes with a trend graph communicate with the **HDA** program module.

New trend activations will only become active after the HDA has been restarted.

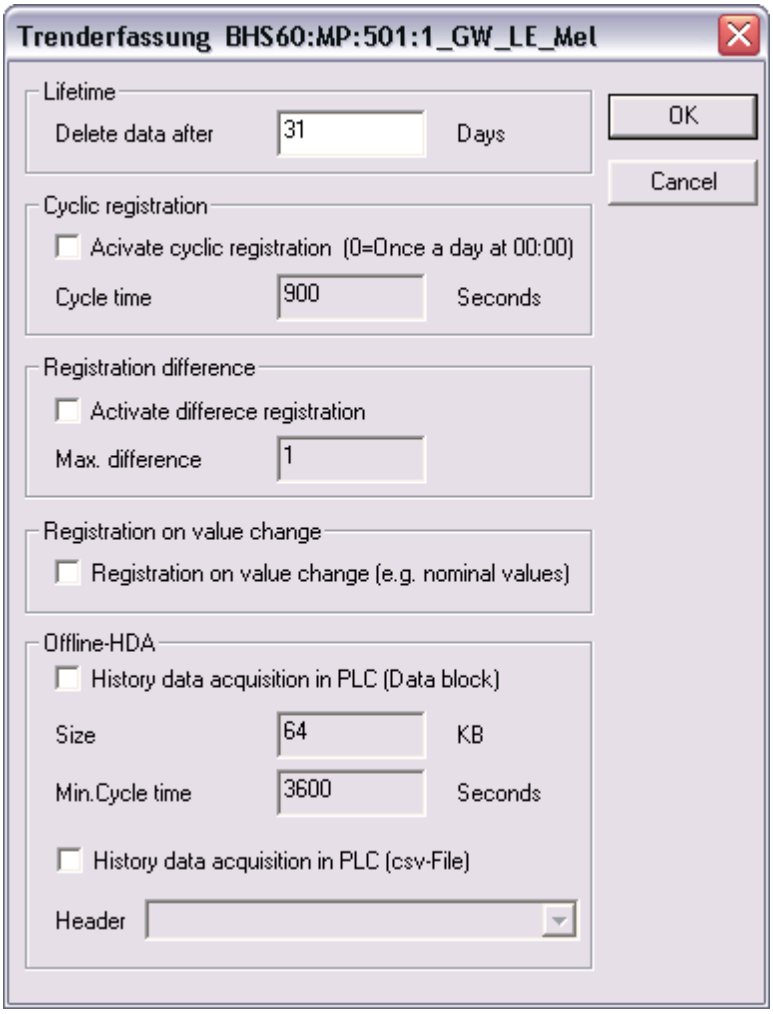
Important: Trends must always be inserted in the Template View, that is, in the template object and not in the system object!
All modifications made to a system object will be lost if the corresponding template object is modified.

By clicking on 

*for the "**Actual Value**" attribute, the following warning is displayed the first time:*



Please take note and confirm. The trend attributes are specified in the following image:



Trend Lifetime
Determines how long the data should be registered before older data is overwritten. The record works like an odometer in a vehicle (..99999, 0, 1, ..)

Cycle Acquisition
Enables data acquisition at defined time intervals

Acquisition on deviation
Saves the value if it deviates from the preceding value by the amount of the value defined here

Acquisition on Change
Saves the value as soon as it changes

The three "**Acquisition**" sections just described can be freely combined.

This finalises the definition in the PET. The PET can now be exited or minimised to the taskbar for later use.

Offline HDA
Acquisition of historical data in PLC (only Saia PCD®)

Size:
Indicates the storage being used on the PLC and accepts the value (only Saia PCD®)

Min. cycle time:
Here the time can be indicated for the minimum intervals in which defined history data should be acquired in the upper part of the window.

More detailed information regarding offline HDA can be obtained from the appropriate chapter.

10.3.9.17.2 Example 2: Displaying a trend graph

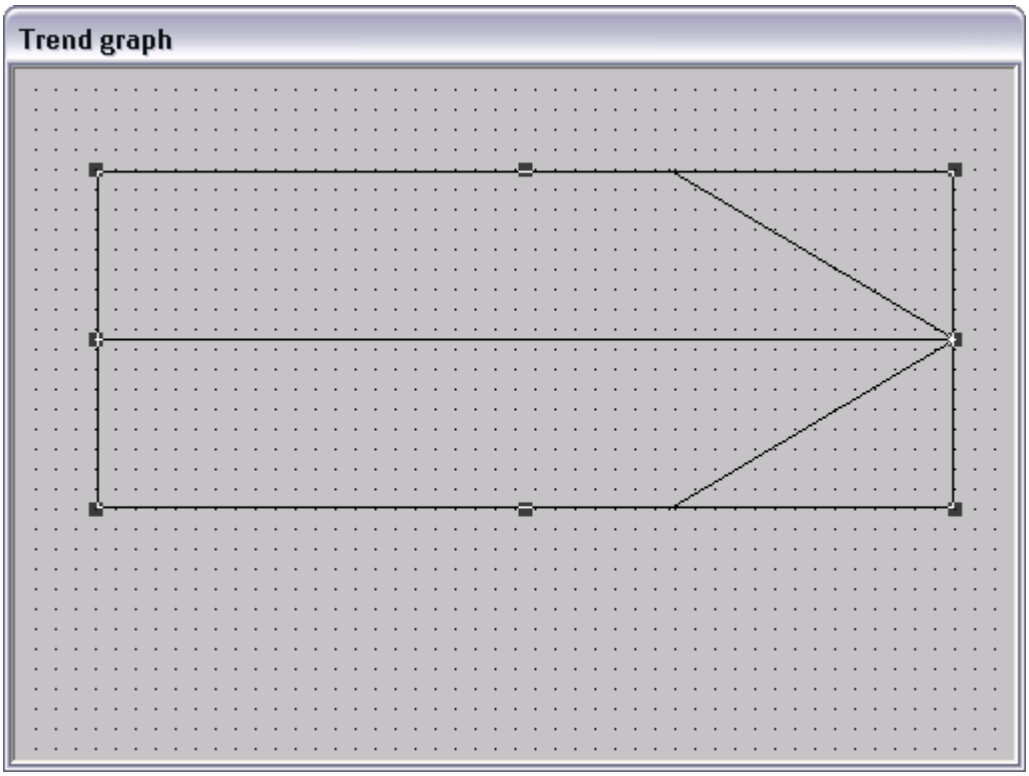
The goal is to use the **GE** (Graphical Editor) to create a new process diagram and the trend graph within it.

First open a new process diagram with .

The new process diagram should have the name **Trend graph**.

To do so, call up **Attributes** with and enter the name **Trend graph** under **Text**.

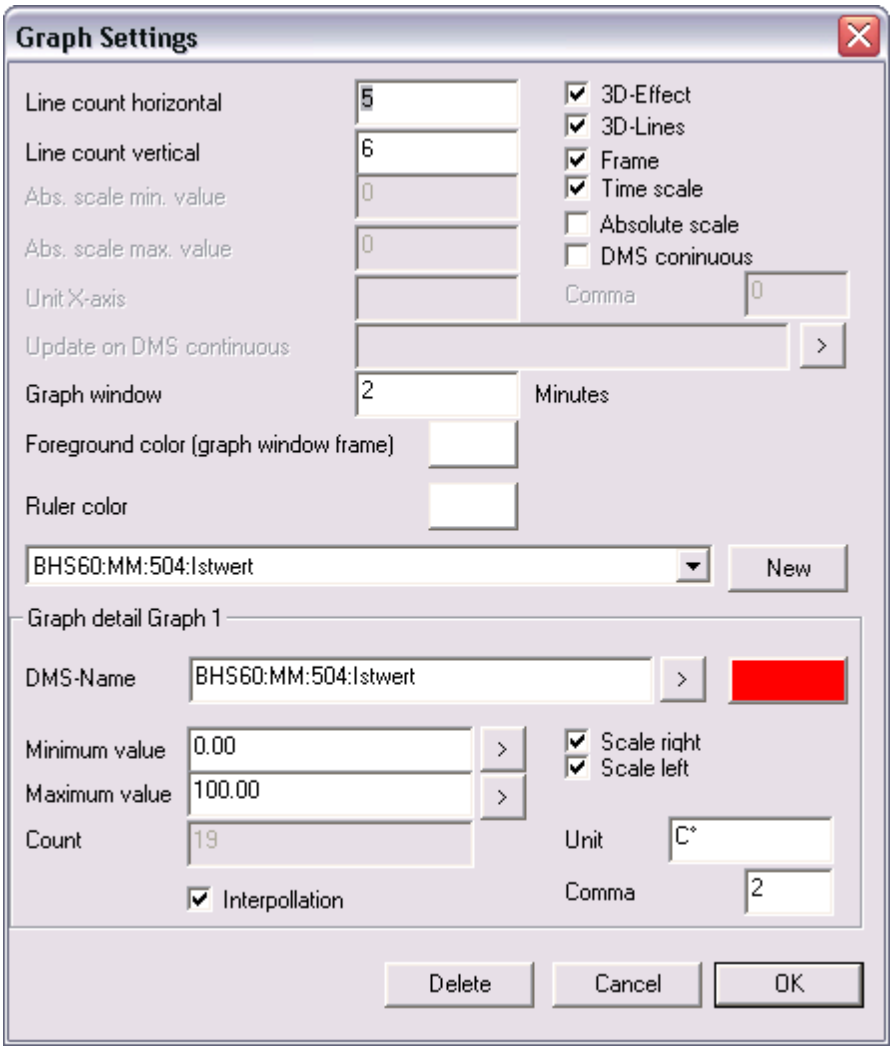
Click on and position the frame for the trend graph in the process diagram as described above, so that the view appears approximately as follows:



In order to access the**Attributes window** for the newly drawn **Trend graph** graphics object, click inside the frame of the **Trend graph**.

Then a graph (first graph of max. 8) of the **Trend graph** graphics object is **initialised** at the data point previously defined in the **PET**.

For this purpose, right-click on the line "**Graph**" in the Attributes window of the trend graph in the **Initialisation column** (right column). The control window **Graph initialisation** opens (see next page).



The **Graph initialisation** control window can be used to initialise up to **8 graphs**

The upper half of the Graph initialisation window pertains to the general settings for the trend graph graphics object.

Horizontal line count defines the number of horizontal auxiliary lines.

Vertical line count defines the number of vertical auxiliary lines.

The following three attribute fields are only active if the checkbox "**Absolute Scaling"** is marked.

Abs. Scaling Minimum Value and **Abs. Scaling Maximum Value** enable the scaled representation of the X-axis (horizontal) between minimum and maximum values.

X-axis unit
Any text for labelling of units in the X-axis

The following attributes field is only active if the checkbox "**Continually from DMS**" is marked.

Update continually from DMS
A DMS name (digital value) is indicated.
With this attribute, consecutive registers can be read from a PLC. The first register address is defined as a DMS data point by the entry field **DMS name** in the section "**Graph Details: Graph ..**". The number of registers that should be read is entered in the field **Total** in the same section.

What function does this attribute have?
Since serial communications between the PLC and computer may be too slow, the PLC should collect data in consecutive registers. These registers are read from the PLC through a high signal of the data point "**Update continually from DMS**" (see Example 4 below).

 *This attribute only works with the driver **SaiaDriver.EXE**, **not with the driver Sdriver.EXE**.*

The following attributes field is only active if the checkbox "**Time Scaling**" is marked.

Trend window
Labels the X-axis with a time bar. The time range, also called time window, is defined by the **Minutes** entry.

 *If a trend window of one hour has been defined, no trend graph will be displayed for the first hour. A trend graph will only be displayed when data is available for the complete time range.*

3D Effect
Activates/deactivates this display option in the trend graph

3D Auxiliary Lines
Activates/deactivates this display option on both horizontal and vertical auxiliary lines

Frame
Enables a frame to be drawn around the trend graph

The lower half, entitled **"Graph Details: Graph .."**, is reserved for the max. 8 individual graphs. For an improved overview, the settings for each graph are shown.

Which graph is currently being displayed can be seen by the number (1–8) after the section name "**Graph Details: Graph ..**" or by the colour of the box next to the **DMS name** input field (in this case, black), provided that colours have already been assigned to the individual graphs.

Currently the default attributes of Graph 1 are displayed, which are not yet defined.

The data point which a graph belongs to is shown in the selection field above it. By clicking the downwards arrow to the right of the field, one of the 8 graphs can be selected (if present).

DMS name
The data point that that should be represented as a graph can also be selected from a list. The list appears after double-clicking in the entry field or by clicking on the button located to the right of it.

The colour of the trend graph can be defined by clicking on the Colour button to the right of the entry field **DMS Name**.

Minimum Value
Text for minimum label on the Y-axis

Maximum Value
Text for maximum label on the Y-axis

Total
Is active if the checkbox **Continually from DMS** is marked. Here the total number of values to be read can be entered. (For a description, see below under "**Update continually from DMS**")

Scale Right
Scale Left
If selected, the scale will be displayed to the left or right of the trend graph.

Unit
Is shown as text above the scale, e.g. **C°**

Decimal places
Input of the number of decimal places to be displayed.

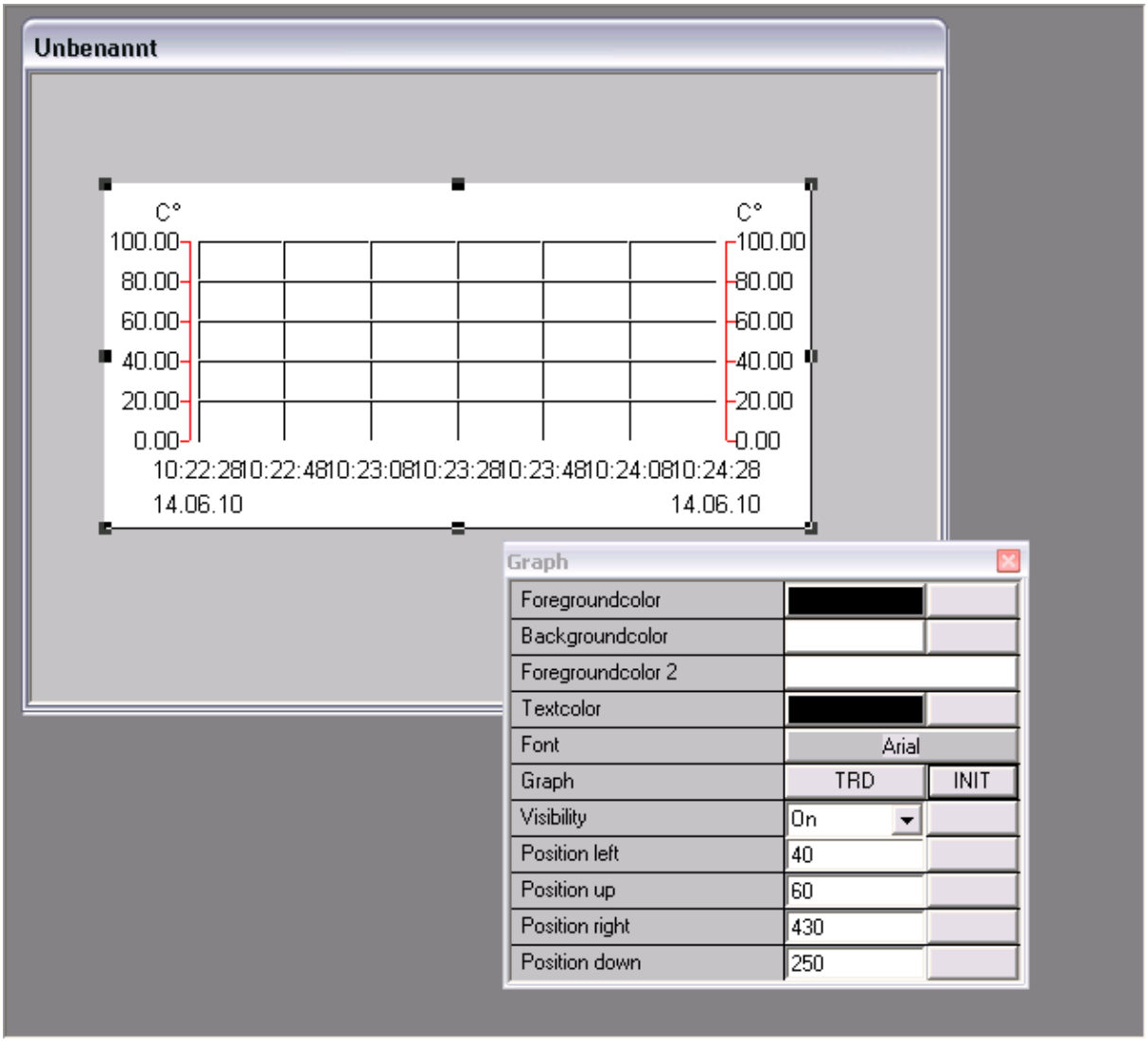
Interpolation of measurements in the trend graph
Smooth transitions from reading point to reading point are created and any intermediate values can be shown as a result. Intermediate values are calculated.

After filling in the control window, confirm with **<OK>**

In the Attributes window of the "**Graph**" graphics object, the following additional changes were made:

Background = white
Colour = white
Foreground = Arial, Bold
colour2
Font

The results should appear approximately as follows (still without a graph):



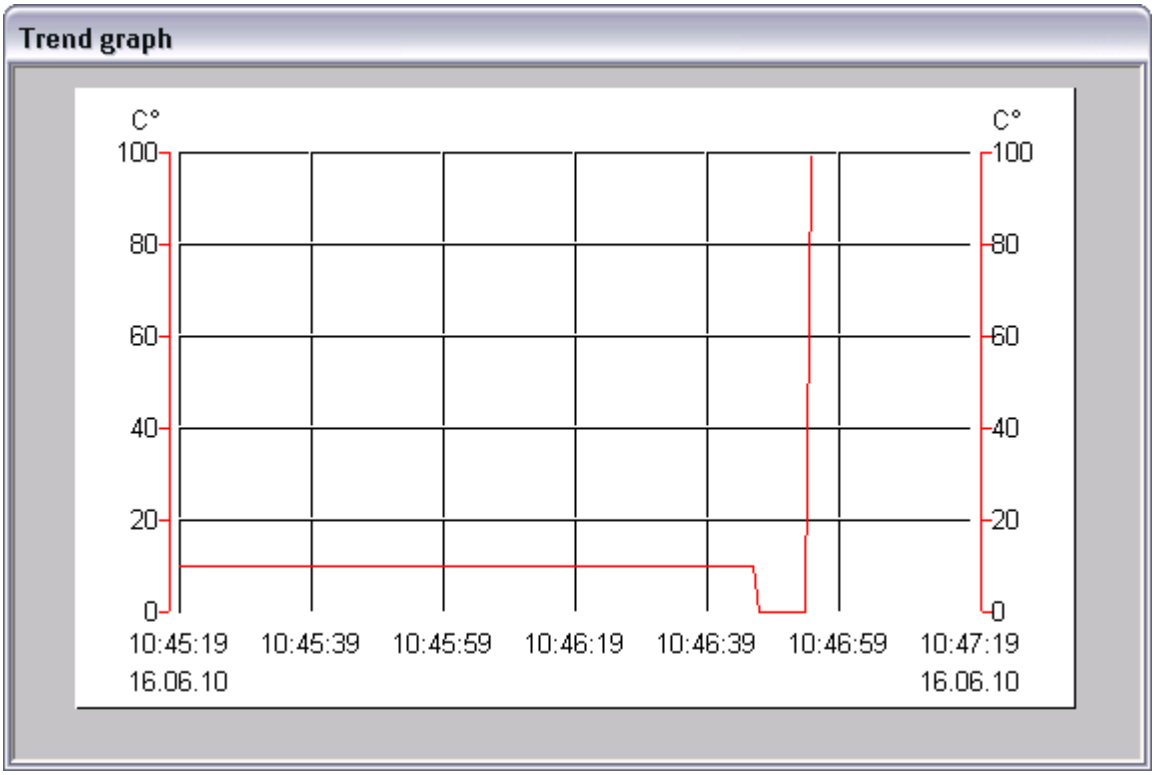
Since the introductory project "Visi.Plus" has no PLC data available, the **PET** is used to enter a few numbers as values.

1. If it is not already running, start up the **HDA** module (runs in the background).

New trend activations will only become active after the HDA has been restarted.

2. Then start up the **PET** or restore it if it has been minimised to the taskbar.
3. In the **Detail View**, find the data point "**BHS60:mm:504**" "**Actual value**".
4. Enter consecutive values in increasing increments of ten from 0 to 100 in the column "**Actual Value**".

After the values have been entered, minimise the PET to the taskbar in order to be able to see the process diagram in the GE. It should look approximately like the following image:



Over time, the graph will shift left and pass out of the Trend window. Just enter more data in the PET.

If no graph is shown, this may be due to the following:

1. *No trend definition has been specified in the **Trend** column in the PET!*
2. *After the Trend Acquisition window has been closed, the time defined under **Trend Window** has not elapsed. The graph is only displayed if this time has elapsed.*
3. *The driver (e.g. "SDriver.exe" or "SaiaDriver.exe") for the PLC was not loaded. More information can be found in the chapter Communications drivers.*

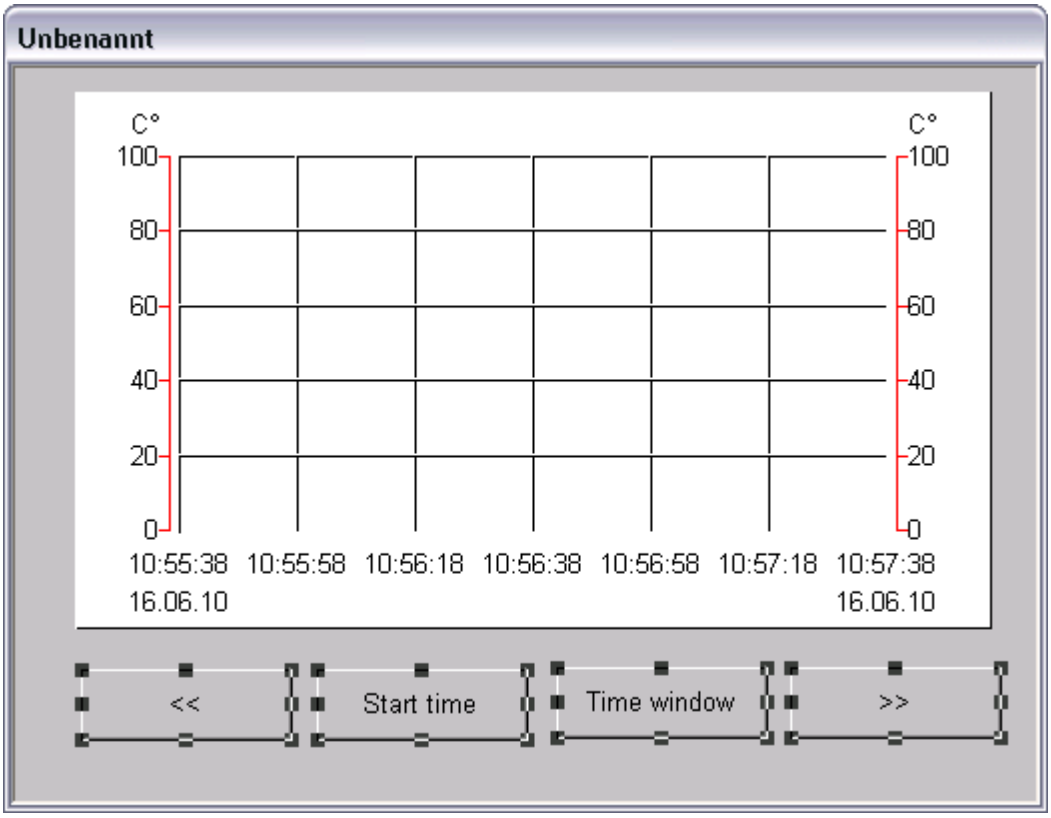
To learn more about trend graphs and their attributes, working through the following examples is recommended!

10.3.9.17.3 Example 3: Creating control buttons for the trend graph

Control buttons for the **trend graph** graphics object can be created quickly and independently.

Although the graphic objects "**Icon**" and "**Polygon**" would be just as suitable for this purpose, the "**Button**" object is used in this case.

The trend graph **process diagram** with the **control buttons** should appear as follows:



First the **Back** button (far left in the image) is created, as described for the button graphics object. The text is set to <<. For the initialisation of the "**Action**" attribute of the button, select "**Graph**" and then "**Scroll back**".

Then it is created as per the procedure just described, for the buttons **Start Time, Time Window** and **Forward** (far right in the image). The buttons are to be initialised to the following actions:
Start time: Action > Graph > Start time
Time window: Action > Graph > Time window
Forward: Action > Graph > Scroll forward
Do not forget to label the buttons.

*If multiple **graphics objects** with the same appearance are to be created, as in the case just described, we recommend duplicating the first object (the Back button in the example) after its attributes have been defined. Duplication is carried out via the key combination **<CTRL+D>** and then positioning the duplicate. Then select **Delete Action** and **Reinitialise**.*

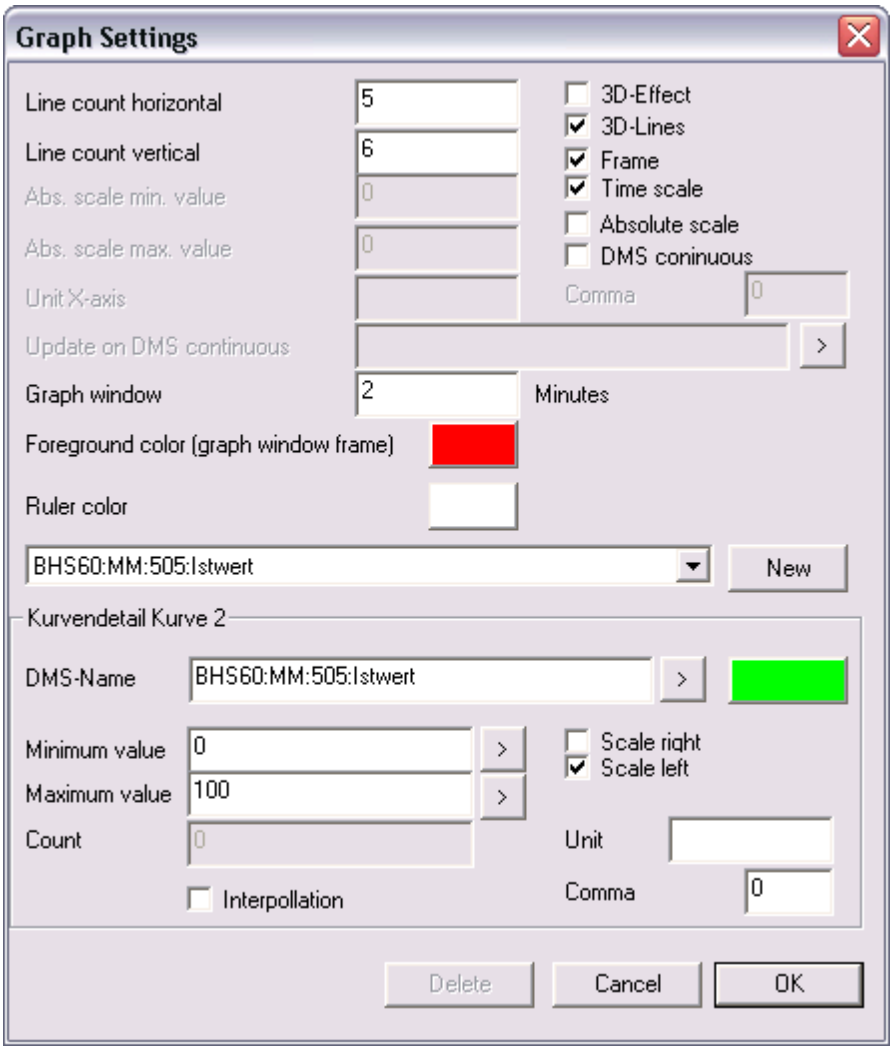
By pressing the **<CTRL>** key and simultaneously clicking the mouse on the desired button in GE edit mode, with the **Back** << and **Forward** >> buttons, the contents of the trend window can now be shifted backwards and forwards along the time axis. If data was registered over a longer period of time, a desired starting point can be defined with the button **Start Time** Startzeit. The time range of the time window can be changed with the help of the **Time window** Zeitfenster button.

10.3.9.17.4 Example 4: Additional trend graphs in the same diagram

In the **Graph Initialisation** window an additional graph can be created by clicking the mouse on the **<NEW>** button.

The settings of the new trend graph in the section "**Graph Details: Graph 2**" are adjusted as already shown in Example 1.

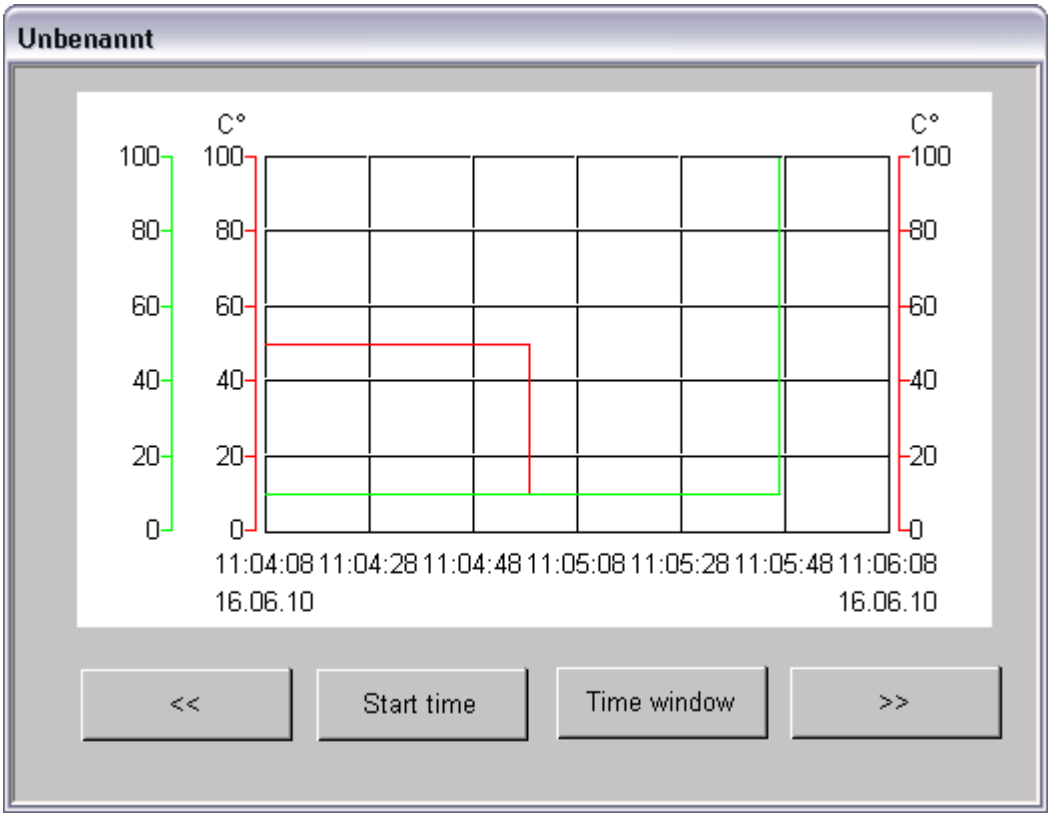
By double-clicking on the input field **DMS Name**, the data point "**BHS60:MM:505:ActualValue**" is displayed. The **Graph Initialisation window** should look like the following image:



Confirm the entries with **<OK>** and close the window.

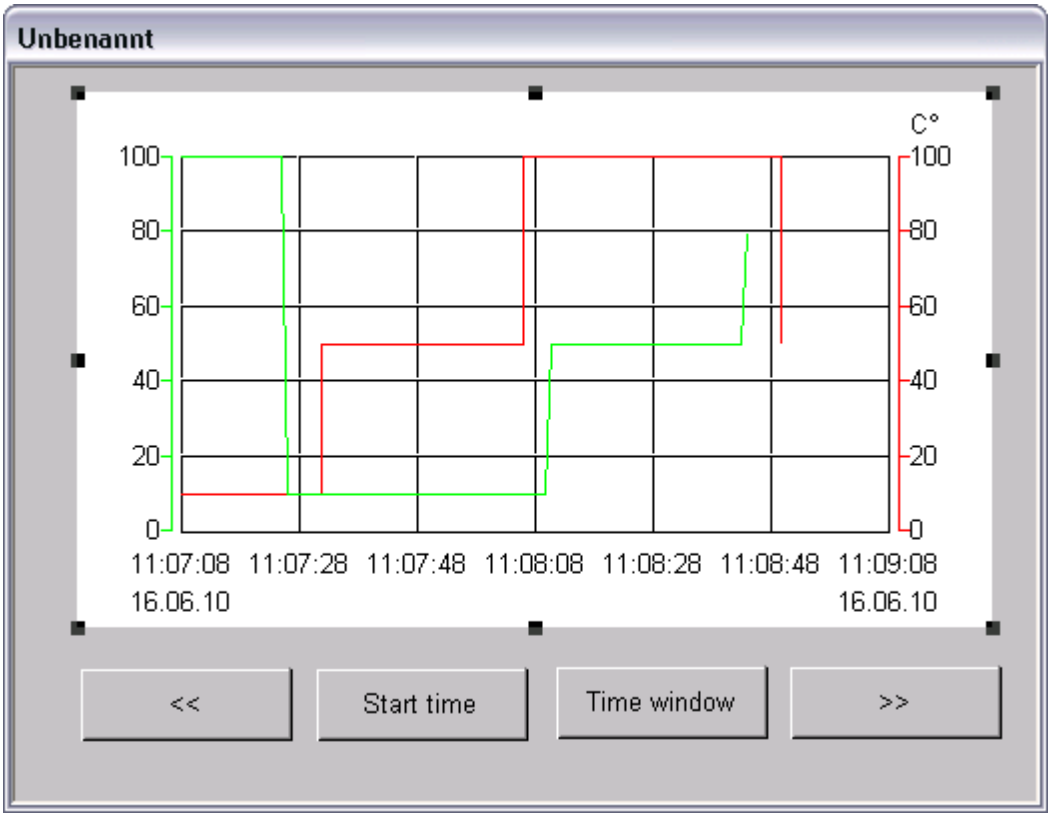
In order to create the two graphs in the trend window with the help of the **PET**, proceed as per Example 2 in this chapter (description at the end of Example 2). First enter the 10 values for the red graph ("BHS60:MM:504" "ActualValue") and then the 10 values for the green graph ("BHS60:MM:505" "ActualValue").

The two graphs should then appear approximately as follows:



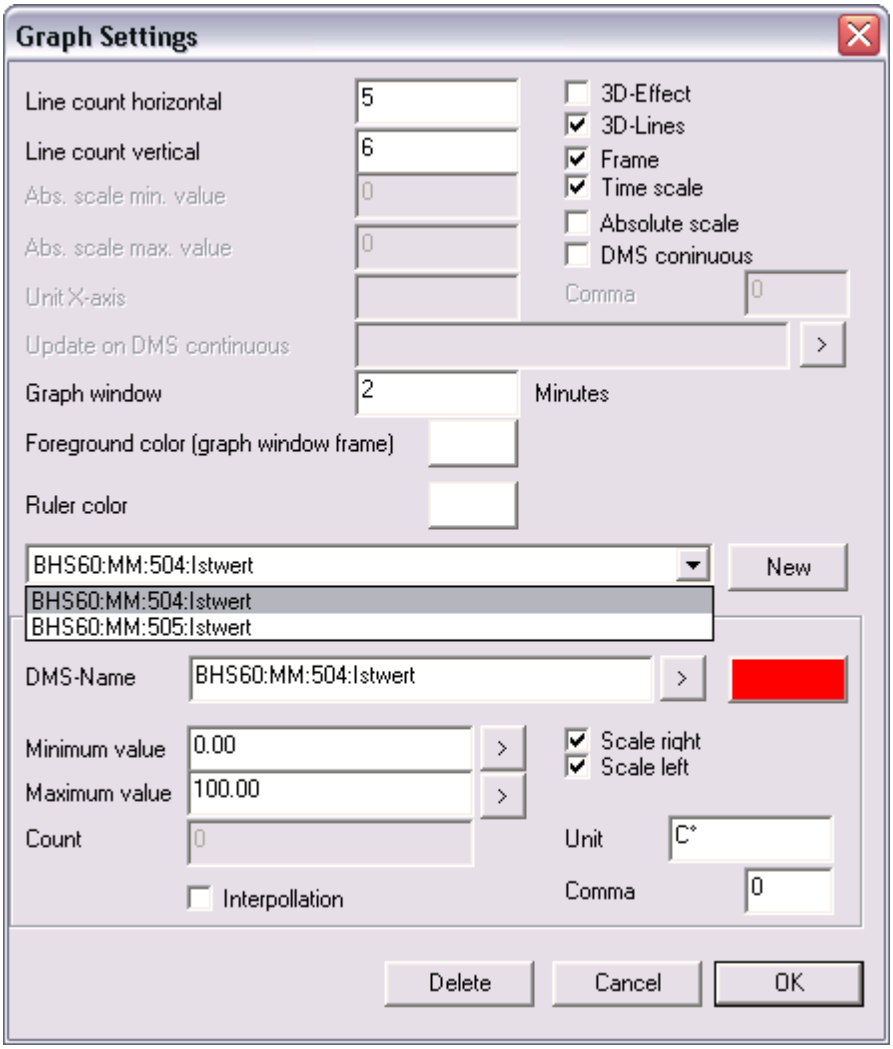
The checkbox **Interpolate** by the green trend graph was marked, therefore smooth transitions from value to value can be seen in the green graph. In addition, only **Scale Left** was switched on.

We now recommend experimenting with the attributes of the trend graph object and its graphs.



An example of a modification would be switching off **Scale Left** for the red graph.

By clicking on the dropdown menu of the selection field above the graph details in the **Graph Initialisation window**, the individual graph attributes of the respective selected trend graph can be modified.



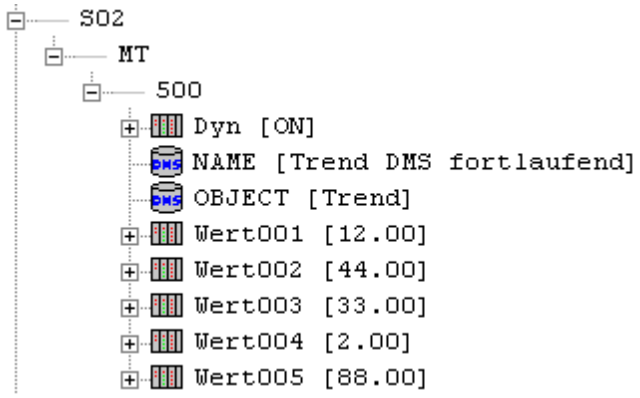
10.3.9.17.5 Example 5: Trend graph with command continually from DMS

If data must be read quickly, the serial communication between the PLC and Visi.Plus may be too slow.For example, this problem can emerge through data storage to consecutive addresses of registers in the PLC.

With a digital signal coming from Visi.Plus through a switch object or from a PLC signal (in this example "**S02:MT:500:Dyn**"), the data can be registered in Visi.Plus in one step (Value**001** to Value**005**).

To use the function "**Continually from DMS**", values whose names end in three-digit numbers (Value **001**, Value **002**, etc.) must be present in the **DMS**. Of course the easiest way to create these data points is with the **PET**.

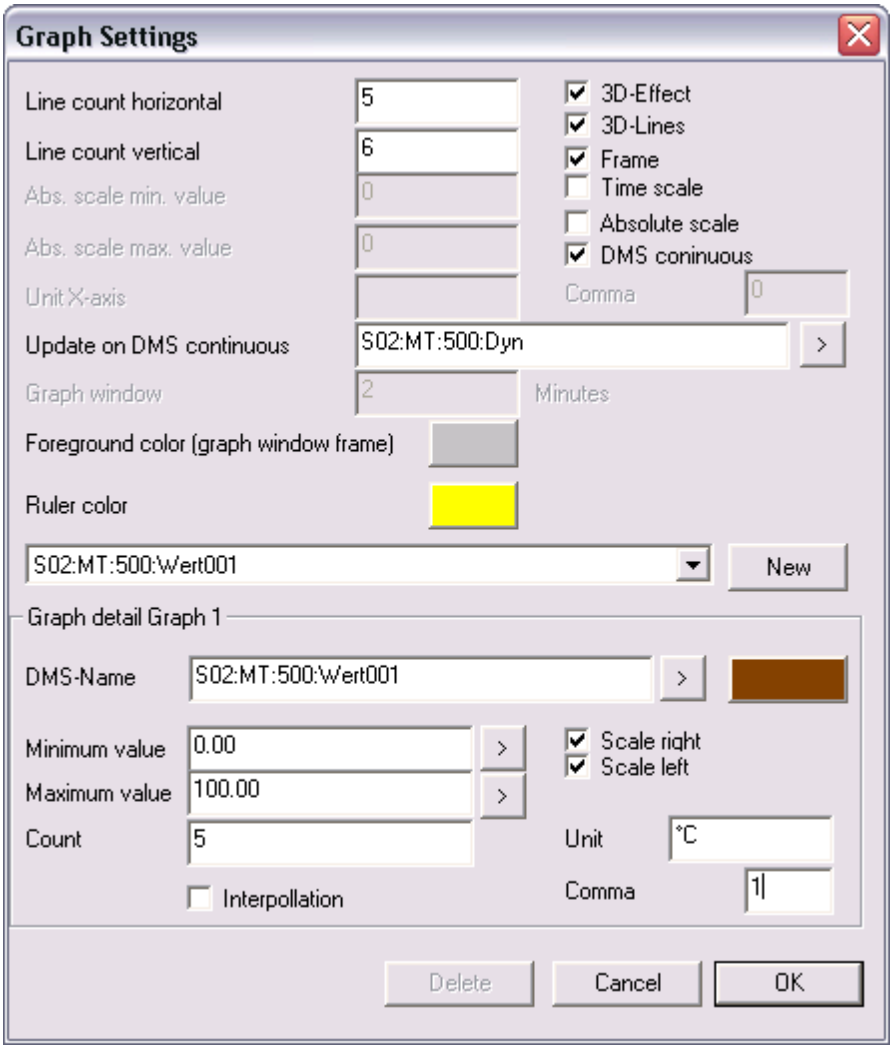
In the following example, five data points of the of floating-point type have been created and numbered from 001 to 005.



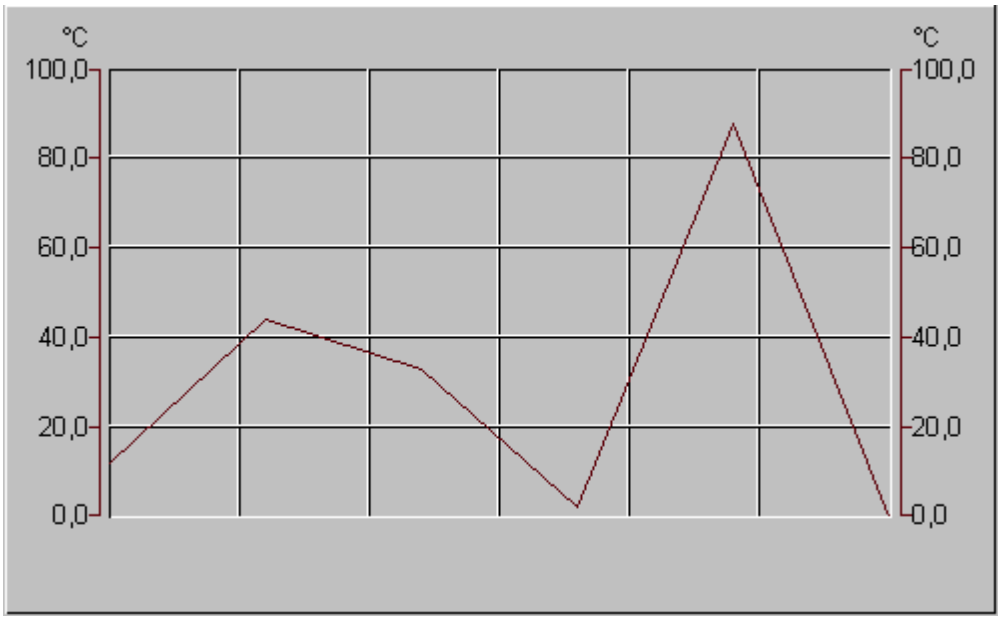
The following values were entered in the Graph Initialisation window:

Continually from DMS = on
Update continually from DMS = S02:MT:500:Dyn

DMS name = S02:MT:500:Value001
Total = 5

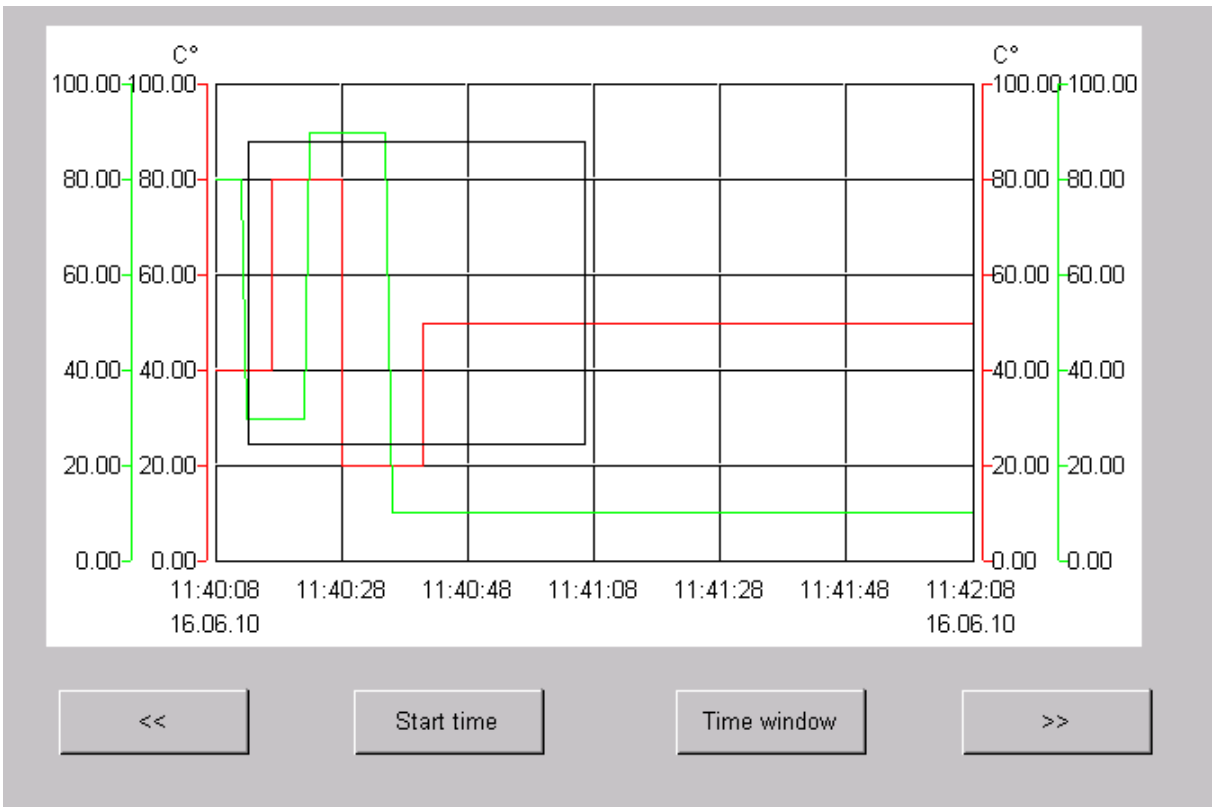


The values read in this manner yield the following trend graph:

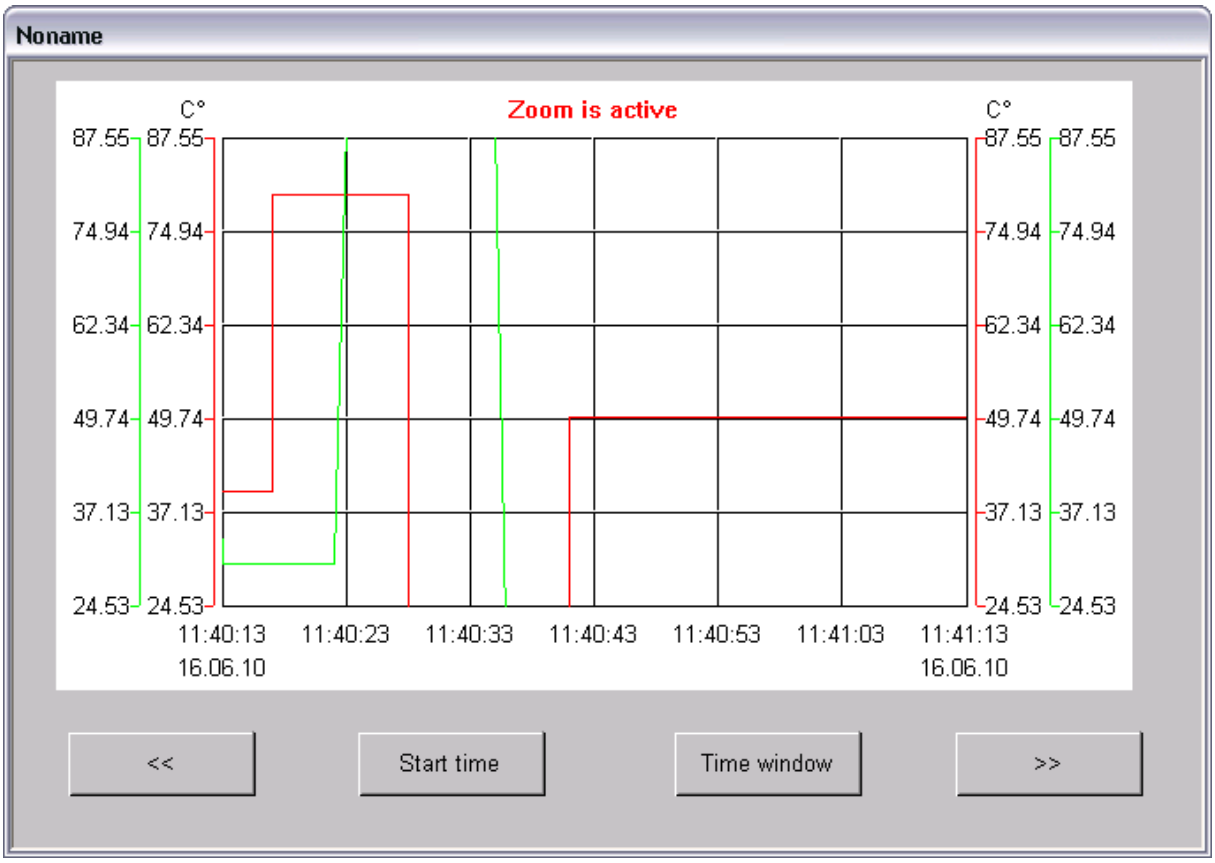


10.3.9.17.6 Example 6: Zooming a trend graph

You can zoom in on trend graphs in runtime mode. To do so, click in the diagram area with the left mouse button and move the mouse to the desired end position while holding the mouse button. The zoom area is identified by a rectangle.



After the mouse button is released, the zoomed area of the graph appears. The message "Zoom switched on" appears above the diagram.



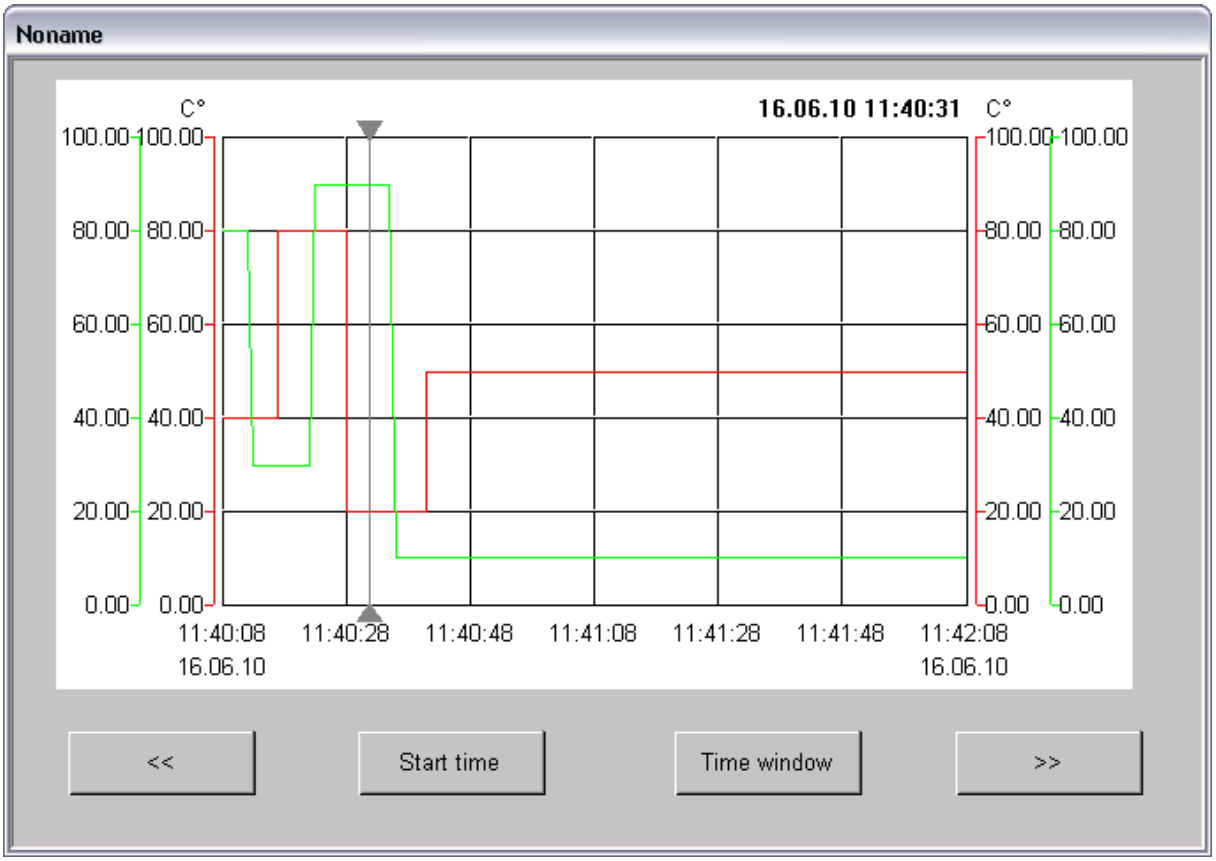
The zoom remains active while scrolling forwards and backwards. It can also be switched off again with a right-click of the mouse in the diagram area. When switching to edit mode zoom is automatically switched off.

10.3.9.17.7 Example 7: Calling up the graph ruler

In Version 1.5 or higher a graph ruler can be shown in runtime mode for trends with a timescale.

The graph ruler function is not available for trends whose values are not ***read directly from the PLC*** *(Graph Initialisation > Continually from DMS).*

The graph ruler can be shown by clicking the left mouse button in the diagram area, and ensuring that the mouse is kept still (otherwise it will zoom). The graph ruler (yellow line with two triangles) is now shown at the X-position where the mouse was clicked.



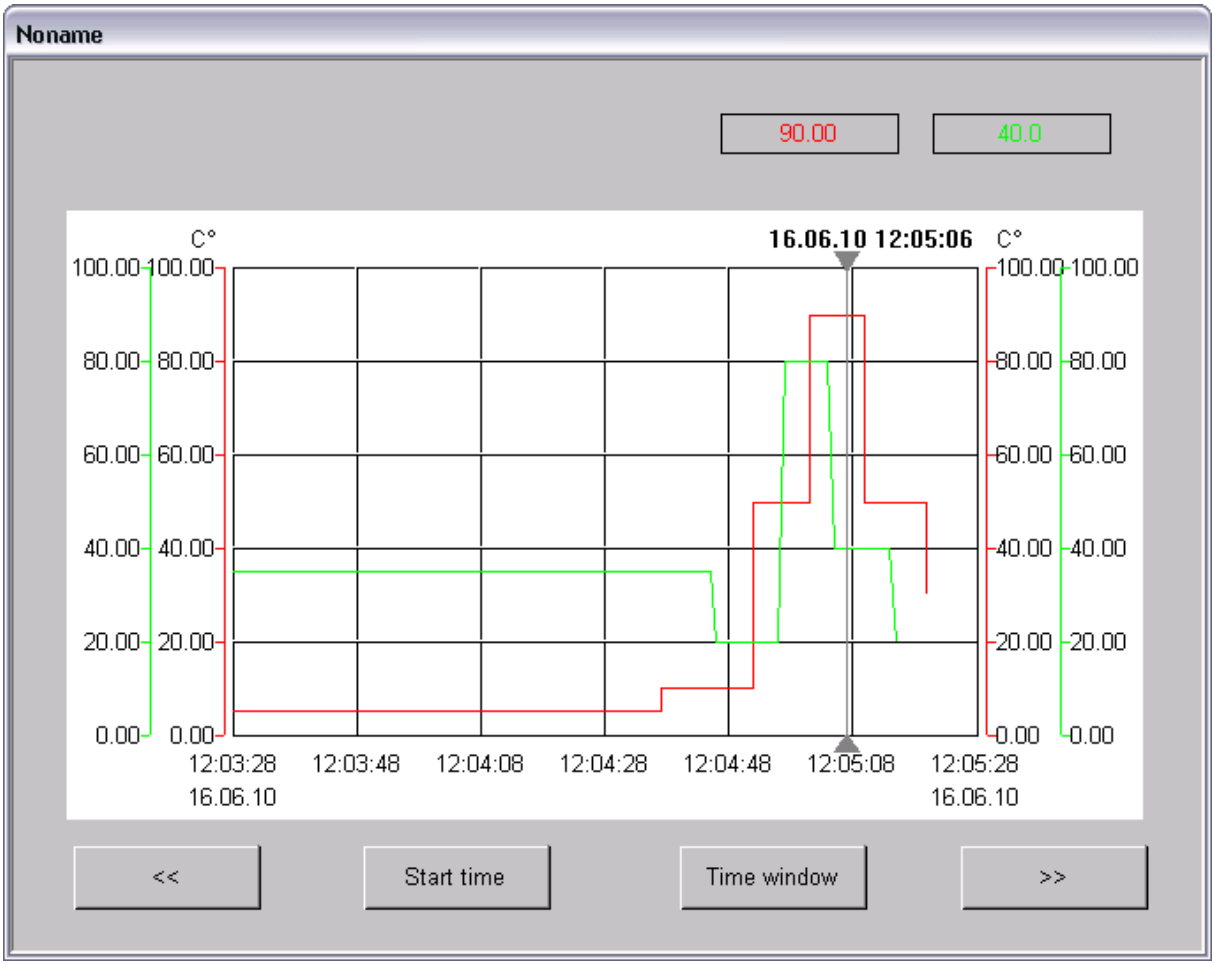
The time that the graph ruler is positioned on is shown above the diagram to the right. Two additional text fields are needed to output the temperature.

The process diagram should appear as follows:

The text fields are created as follows:

- Switch to edit mode (<e> key)
- Draw a text field as described under Text/Text Field
- Call up the Attributes window.
- Click on the button in the Initialisation column by the attribute "Text" and select the command "Value Display".
- Enter the DMS name "BHS60:MM:504:ActualValue" in the Initialisation dialog or select it from the list. Set the Output Format to %3.2f.
- Choose red for the text colour in the Attributes window
- Copy the created text field
- Call up the Attributes window.
- Click on the button in the Initialisation column by the attribute "Text" and select the command "Value Display".
- Enter the DMS name "BHS60:MM:505:ActualValue" in the Initialisation dialog or select it from the list
- Choose green for the text colour in the Attributes window
- Position both text fields above the diagram to the right

Now save the image and switch to runtime mode. Call up the graph ruler (left click in the diagram area). Both text fields now display the value that the graph ruler is at:

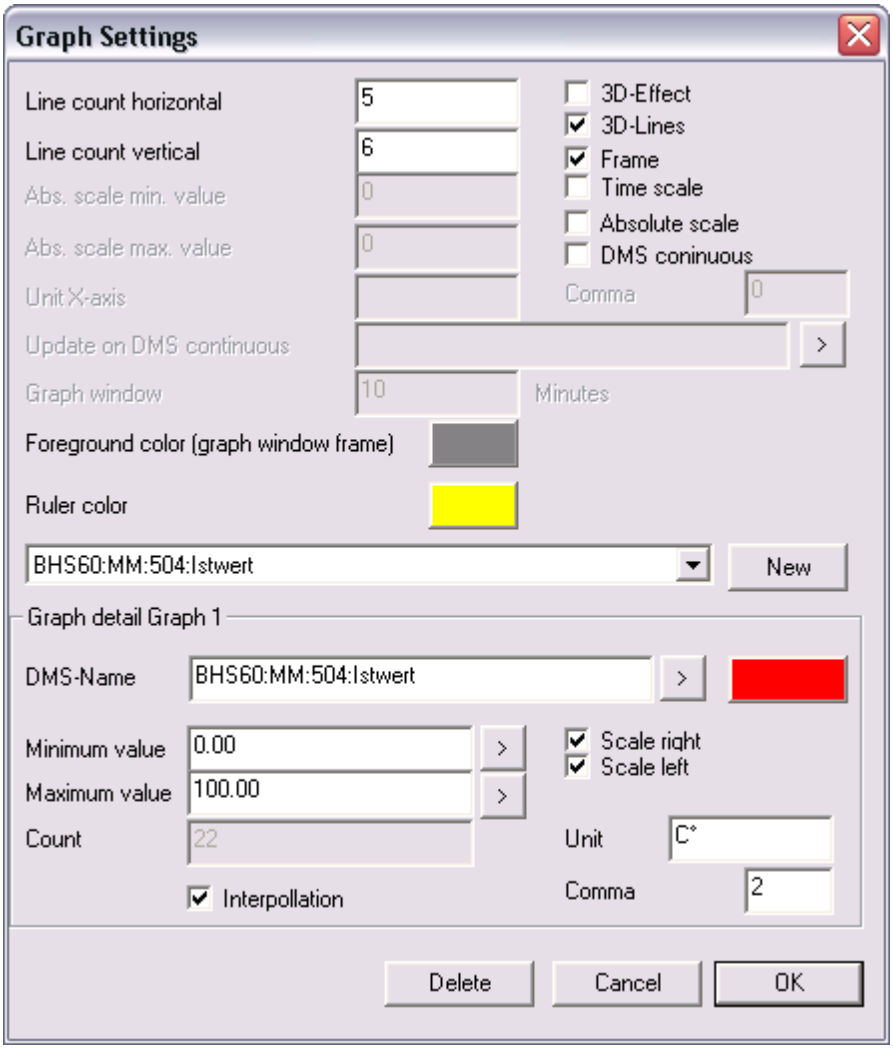


The graph ruler can now be moved back and forth with the left and right arrow keys. In doing so, the time and the displayed values should change. If the graph ruler reaches the right or left edge of the diagram range, the contents of the trend windows are automatically shifted along the time axis.

The graph ruler can also be moved with the mouse. To do so, left click on the upper or lower triangle of the graph ruler and move the mouse while holding down the mouse button. The graph ruler now follows the mouse pointer. When the mouse button is released, the graph ruler stops moving.

The graph ruler is switched off by clicking in the diagram area with the right mouse button. In doing so, the current values from the DMS are displayed again. If zoom has been switched on, it is also switched off now.

The graph ruler colour can be changed in the Graph Initialisation window. To do so, click on the Graph Ruler Colour button and select a new colour from the displayed Colour dialog.



In Version 1.5 or higher it is possible to control the graph ruler using two buttons, which must be initialised accordingly. In the process, the graph ruler automatically shifts either forwards or backwards over the range of the diagram (autoplay function). This function is discussed in detail in the following chapter.

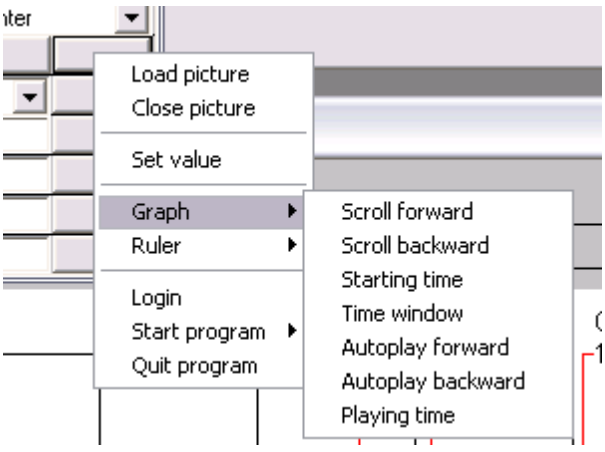
10.3.9.17.8 Example 8: Autoplay function

In Version 1.5 and higher it is possible to control the graph ruler via the so-called autoplay function. This causes the graph ruler to automatically move either forwards or backwards over the range of the diagram. To do so, two buttons must be created with initialisation to **Autoplay Forwards** and **Autoplay Reverse**, as well as a button for entering the **Playback Speed**.

The process diagram should appear as follows:

The buttons are created as follows:

- Copy the button on the far left
- Position the button as per the previous image (bottom left)
- Call up the Attributes window.
- Click on the button in the Initialisation column by the "Action" attribute and delete the initialisation
- Click on the button in the Initialisation column by the "Action" attribute again. The following selection menu opens:



- Click on **Autoplay Reverse**
- In the Reverse Playback initialisation dialog, click OK

- Copy created buttons twice and position
- Delete the initialisation for both buttons
- In the Attributes window, click on the button in the Initialisation column by the attribute "Action"
- Select **Playback Speed** for the 2nd button
- Select **Autoplay Forwards** for the 3rd button
- Then change the texts in the Attributes window according to the image shown above
- Switch to runtime mode and select the graph ruler

The 3 created buttons should now exhibit the following behaviour:

When clicking on 

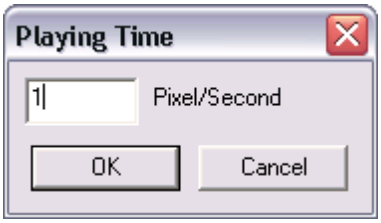
the graph ruler moves to the left. In doing so, the time and the displayed values should change. If the graph ruler reaches the left edge of the diagram range, the contents of the trend window are automatically shifted along the time axis.Clicking again stops the graph ruler.

When clicking on 

the graph ruler moves to the right. In doing so, the time and the displayed values should change. If the graph ruler reaches the right edge of the diagram range, the contents of the trend window are automatically shifted along the time axis.Clicking again stops the graph ruler.

By clicking on 

the following dialog appears:



Here the speed at which the graph ruler travels along the trend graphs can be adjusted. It is preset to 1 pixel/second. The value can be changed via the keyboard.

If the selected playback speed is too high (e.g. if the computer has insufficient memory), the graph ruler will always travel over the range of the diagram at the maximum possible speed.

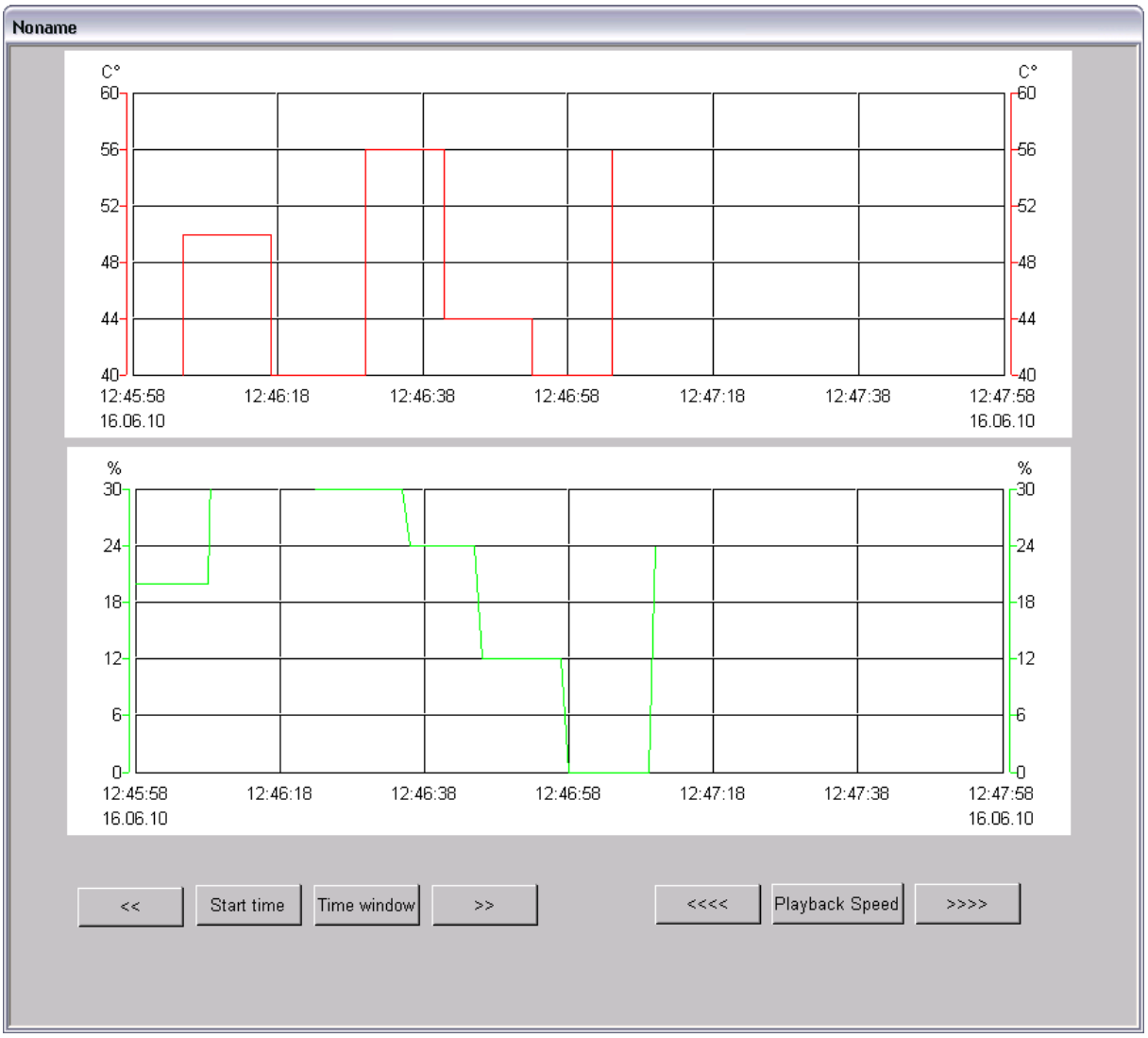
10.3.9.17.9 Example 9: Multiple trend objects in the process diagram

Like the previous examples, this example provides concrete instructions for the creation of a trend image. It is only intended to show how control buttons for multiple trend objects are created. In addition, how the zoom, graph ruler (available in Version 1.5 or later) and autoplay functions (available in Version 1.5 or later) work with multiple trend objects in the same process diagram is explained.

It it sometime useful to be able to create multiple trend objects in the same process diagram because each trend object only permits 8 graph definitions or because the trends need to be grouped (e.g. in order to display the temperature of the supply in the 1st trend object and the

position of the corresponding control valve in the 2nd trend object).

One such image could then appear approximately as follows:



The temperature of the supply is displayed in the first diagram. The position of the corresponding control valve is shown in the second diagram. The diagrams should be the same size and should be aligned with one another. This ensures that the graphs in the two diagrams can be compared more easily.

Control buttons
As the picture shows, only one the toolbar is necessary for operating the two diagrams.

This is advantageous because less buttons must be created and the trend diagrams are temporally synchronised. Therefore, if the Forward or Back button is clicked, for example, both the contents of the 1st diagram and the contents of the 2nd diagram are shifted along the time axis. If the start time or the time window is modified, both diagrams are adjusted.

The buttons for the autoplay function are explained below.

Zoom

If a diagram should be zoomed, the time axis of all other diagrams is automatically adjusted (horizontal zoom). The Y-axis, on the other hand, remains the same. This guarantees that all diagrams have the same time range.

Graph ruler and autoplay function
The graph ruler can be switched on for all diagrams in the process diagram by clicking with the left mouse button on the respective diagram in each case.

Now the graph ruler can be moved with the right and left arrow keys to the most recently created or selected item. If you clicked on the 1st diagram first and then on the 2nd diagram, the arrow keys move the graph ruler in the 2nd diagram. Now click again in the 1st diagram and you can move the graph ruler in the 1st diagram with the arrow keys.

The graph ruler in the 1st diagram is now selected; by clicking in the diagram the graph ruler was drawn at another position. In order to select a graph ruler that is already shown, you must click on the lower triangle of the graph ruler.

If the graph ruler reaches the right or left edge of the diagram range, the contents of all trend windows are automatically shifted along the time axis.

The forward and reverse autoplay functions are also based on the most recently created or selected graph ruler. Therefore it is not possible to move multiple graph rulers simultaneously across the respective diagram.In this case when the right or left edge of the diagram range is reached, the contents of all trend windows are also automatically shifted along the time axis.

If the playback speed is modified via a button (in this example by clicking on "Speed"), the graph ruler speed of all diagrams is adjusted.

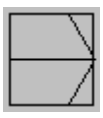
10.3.9.18 Graph ruler object

 or in the menu **"Objects > Graph ruler"**

This function is available in Version 1.5 or higher. As an example, with the help of a graph ruler object the status of a pump or a valve (switched on or off) or the value of an analogue signal (e.g. temperature) can be visualised in a process diagram for a point of time in the past.

A graph ruler object is defined by a square frame. We recommend drawing the frame as an elongated box, preferably across the entire width of the process diagram.

The **starting point of the trend graph frame** is set with the left mouse button pressed and held. Then, to set **the end point of the trend graph frame**, the mouse button must be released.



The arrow pointing to the right appearing in the box designates this box as a placeholder for a trend graph to be defined (for more information about this, see the **Trend graph examples**

1–5).

All attributes of the **"Graph ruler"** graphics object are to be initialised as already described for the other graphics objects. A trend curve must have been drawn for the initialisation of the **"Graph ruler**" attribute.**The creation of trend graphs is described after the following table on the basis of 5 detailed examples:**

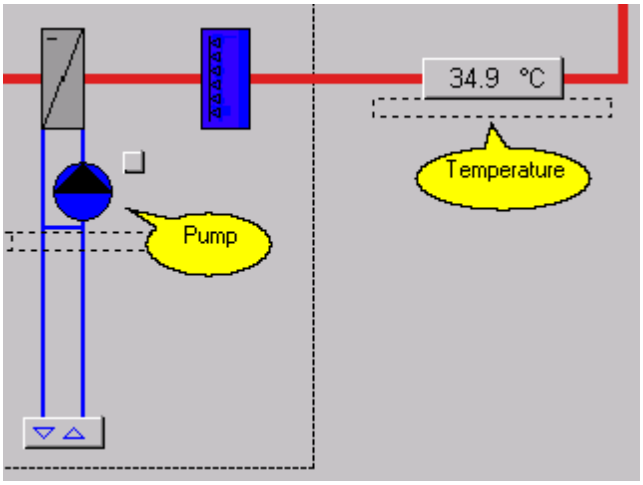
Descriptions for attributes and initialisations are located:

 **Lineal**

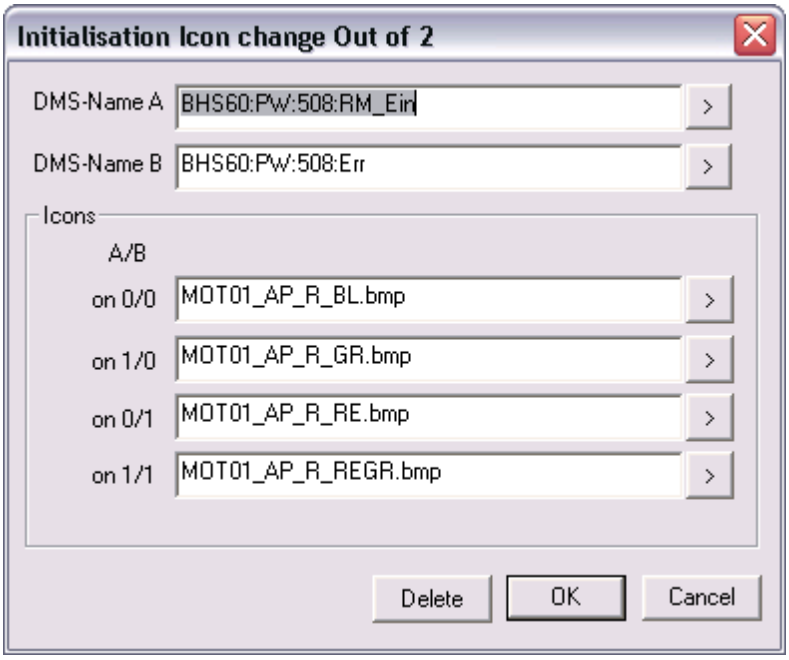
Attribute	Description	Initialisation	Initialisation Description
Foreground Colour	Line Colour	yes	Colour Change
Background Colour	Surface Colour	yes	Colour Change
Foreground Colour 2		no	
Text Colour		yes	Colour Change
Font		no	
Graph ruler	Only Initialise	yes	
Visibility		yes	
Left Position		yes	
Top Position		yes	
Right Position		yes	
Bottom Position		yes	
Delete Initialisation			

10.3.9.18.1 Example 10: Inserting trends in the PET

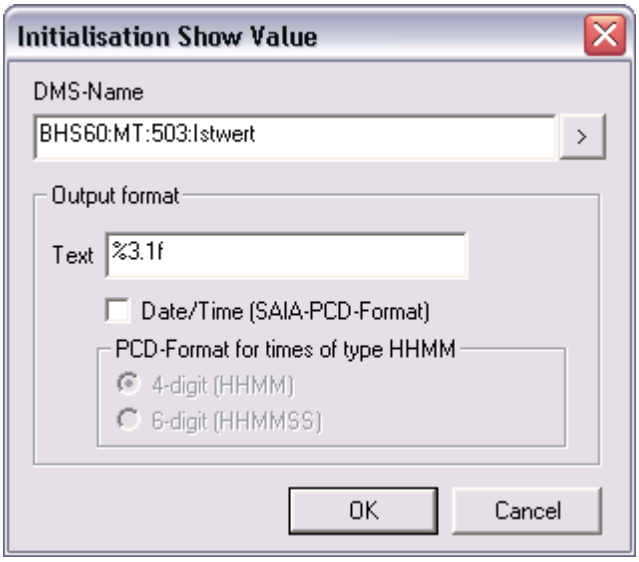
Here the use of a graph ruler is described for the Promo demo project. Start the project as described in Example 1.
After logging in, click further until Image 6 appears. To the bottom right of this image a pump and a button with a temperature indicator are shown.



An icon change for the pump is initialised as follows:

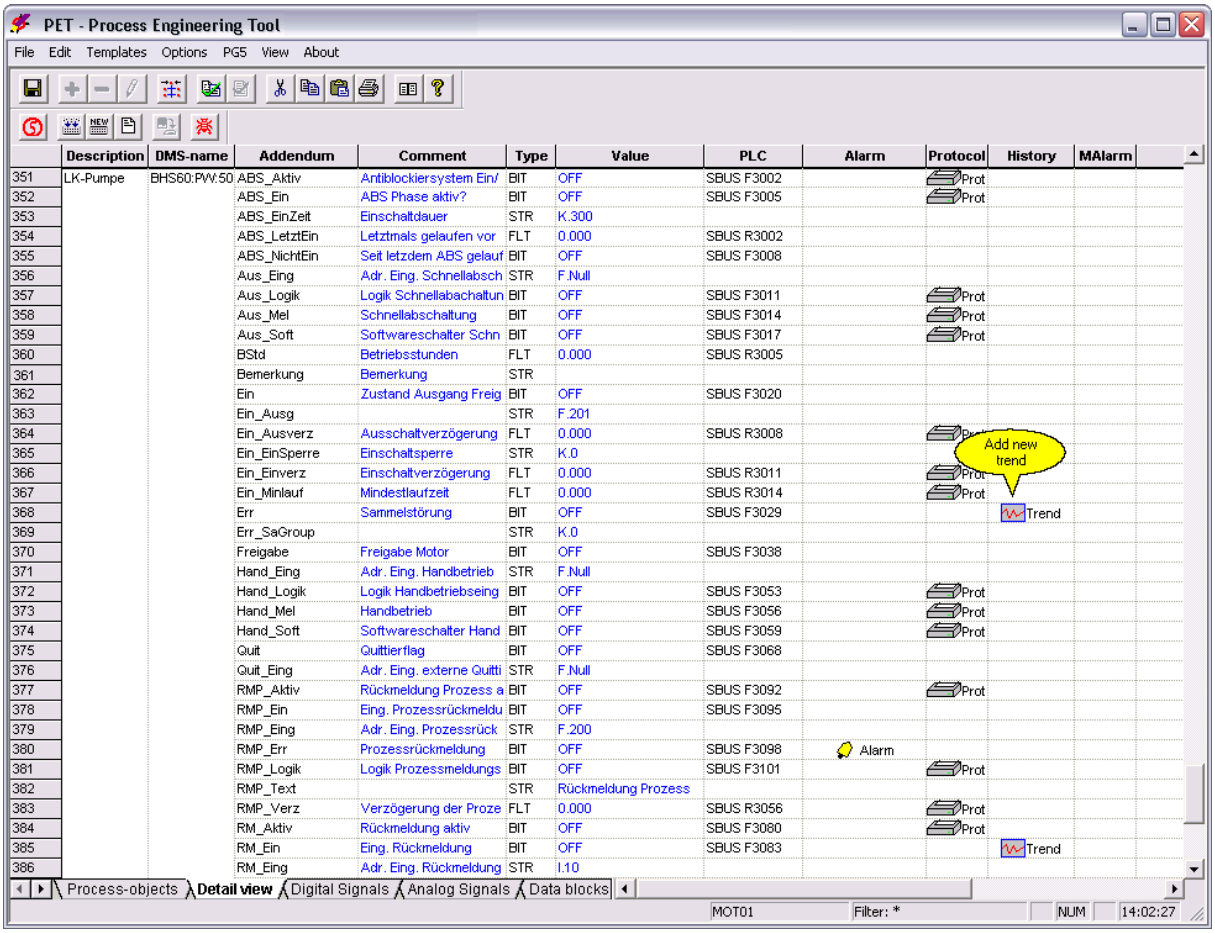


The text on the button is initialised as follows:



The aim is to visualise the status of the valve and the temperature in the process diagram at any point in time. In order to do so, a trend must be defined in the PET for the data points "**BHS60:PW:508:RM_On**", "**BHS60:PW:508:Err**"and "**BH60:MT:503:ActualValue**".

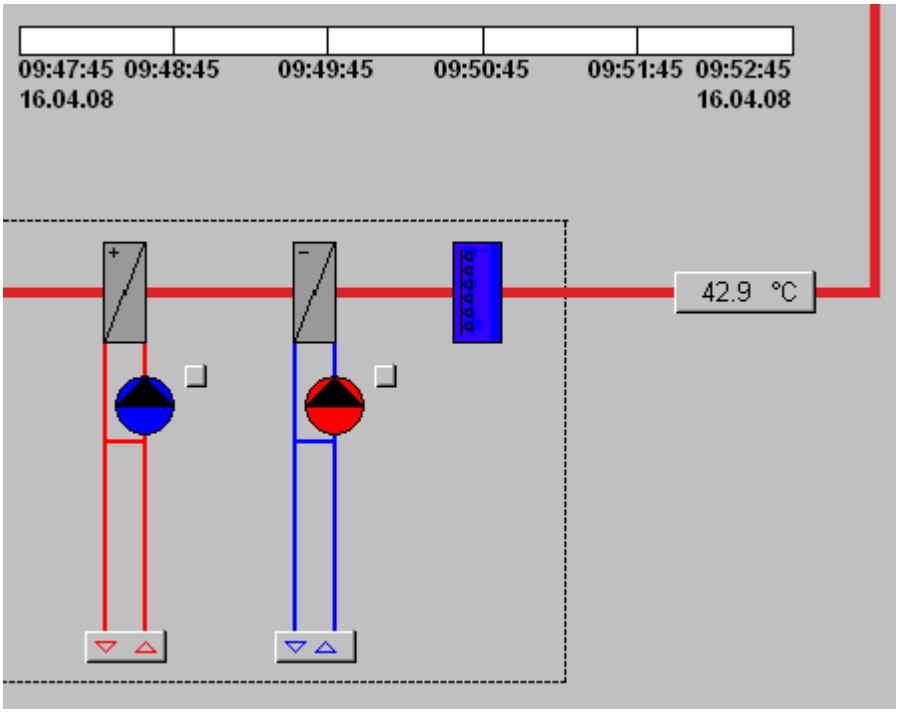
Now switch to edit mode with the **<e> key** and start the **PET** by clicking on  . Switch to the Detail View and define a trend for "BHS60:PW:508:Err" as per the procedure from Example 1. In the process, activate the checkbox **Register when changed** in the Trend Detection dialog. A trend has already been defined for the data points "BHS60:PW:508:RM_On" and "BH60:MT:503:ActualValue". Therefore no more changes are necessary and the PET can be exited.



	Description	DMS-name	Addendum	Comment	Type	Value	PLC	Alarm	Protocol	History	MAlarm
351	LK-Pumpe	BHS60:PW.50	ABS_Aktiv	Antiblockiersystem Ein/	BIT	OFF	SBUS F3002		Prot		
352			ABS_Ein	ABS Phase aktiv?	BIT	OFF	SBUS F3005		Prot		
353			ABS_EinZeit	Einschaltdauer	STR	K.300					
354			ABS_LetztEin	Letztmals gelaufen vor	FLT	0.000	SBUS R3002				
355			ABS_NichtEin	Seit letztem ABS gelauf	BIT	OFF	SBUS F3008				
356			Aus_Eing	Adr. Eing. Schnellabsch	STR	F.Null					
357			Aus_Logik	Logik Schnellabachaltun	BIT	OFF	SBUS F3011		Prot		
358			Aus_Mel	Schnellabschaltung	BIT	OFF	SBUS F3014		Prot		
359			Aus_Soft	Softwareschalter Schn	BIT	OFF	SBUS F3017		Prot		
360			BStd	Betriebsstunden	FLT	0.000	SBUS R3005				
361			Bemerkung	Bemerkung	STR						
362			Ein	Zustand Ausgang Freig	BIT	OFF	SBUS F3020				
363			Ein_Ausg		STR	F.201					
364			Ein_Ausverz	Ausschaltverzögerung	FLT	0.000	SBUS R3008		Prot		
365			Ein_EinSperre	Einschaltsperre	STR	K.0					
366			Ein_Einverz	Einschaltverzögerung	FLT	0.000	SBUS R3011		Prot		
367			Ein_Minlauf	Mindestlaufzeit	FLT	0.000	SBUS R3014		Prot		
368			Err	Sammelstörung	BIT	OFF	SBUS F3029			Trend	
369			Err_SaGroup		STR	K.0					
370			Freigabe	Freigabe Motor	BIT	OFF	SBUS F3038				
371			Hand_Eing	Adr. Eing. Handbetrieb	STR	F.Null					
372			Hand_Logik	Logik Handbetriebseing	BIT	OFF	SBUS F3053		Prot		
373			Hand_Mel	Handbetrieb	BIT	OFF	SBUS F3056		Prot		
374			Hand_Soft	Softwareschalter Hand	BIT	OFF	SBUS F3059		Prot		
375			Quit	Quittierflag	BIT	OFF	SBUS F3068				
376			Quit_Eing	Adr. Eing. externe Quitti	STR	F.Null					
377			RMP_Aktiv	Rückmeldung Prozess a	BIT	OFF	SBUS F3092		Prot		
378			RMP_Ein	Eing. Prozessrückmeldu	BIT	OFF	SBUS F3095				
379			RMP_Eing	Adr. Eing. Prozessrück	STR	F.200					
380			RMP_Err	Prozessrückmeldung	BIT	OFF	SBUS F3098	Alarm			
381			RMP_Logik	Logik Prozessmeldungs	BIT	OFF	SBUS F3101		Prot		
382			RMP_Text		STR	Rückmeldung Prozess					
383			RMP_Verz	Verzögerung der Proze	FLT	0.000	SBUS R3056		Prot		
384			RM_Aktiv	Rückmeldung aktiv	BIT	OFF	SBUS F3080		Prot		
385			RM_Ein	Eing. Rückmeldung	BIT	OFF	SBUS F3083			Trend	
386			RM_Eing	Adr. Eing. Rückmeldung	STR	I.10					

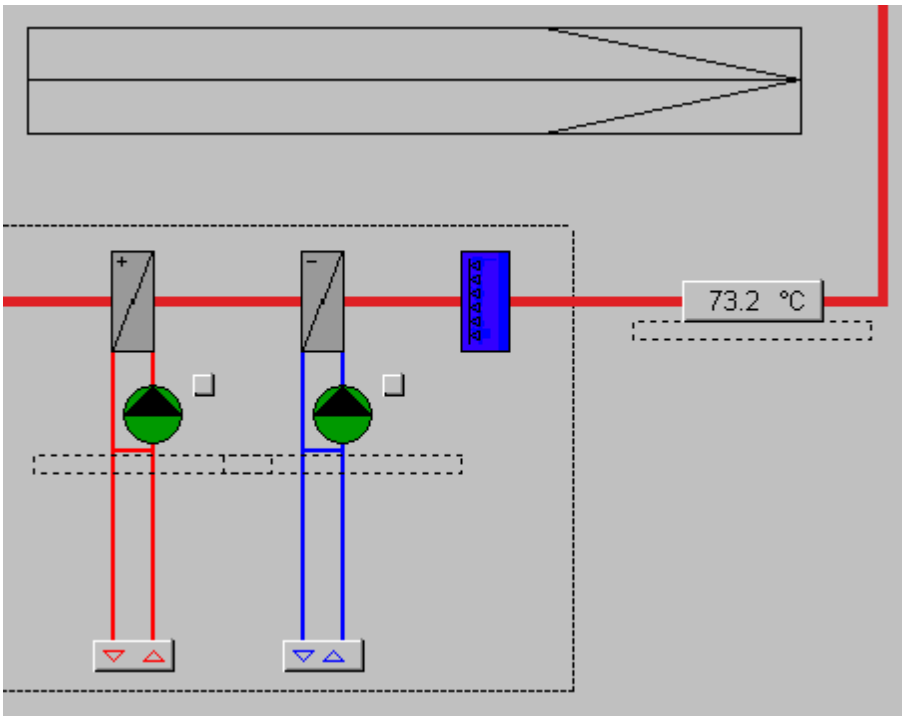
10.3.9.18.2 Example 11: Inserting a graph ruler object

A graph ruler object, which can be placed in any location, should now be inserted into the process diagram. The graph ruler in this example was positioned above the pumps in the lower half because of the subsequent illustration.



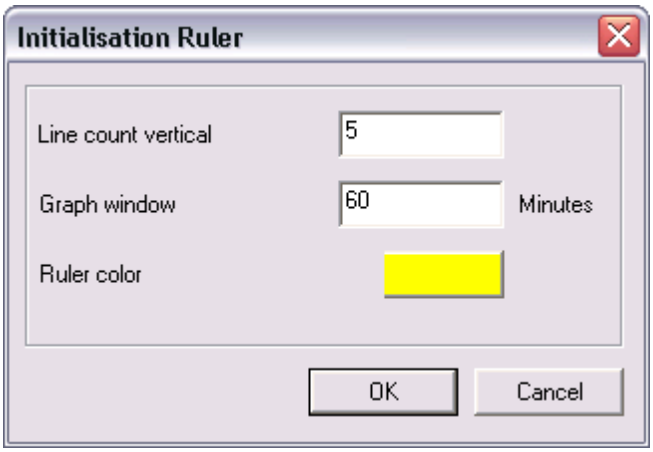
This is how the graph ruler is inserted:

-  **Click** and place the frame for the graph ruler in the process diagram as described above. If there is not sufficient space available for the graph ruler, simply move or reduce the size of the existing objects.



- Click inside the frame of the graph ruler object and call up the Attributes window.

- Click on the button in the Initialisation column near the Graph Ruler attribute. The graph ruler initialisation dialog opens:

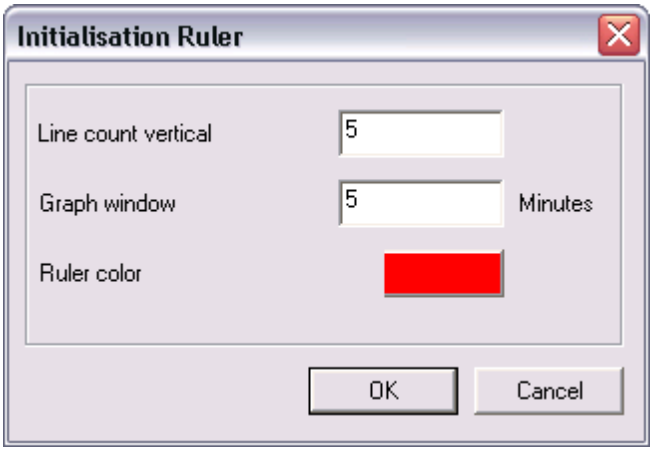


Number of vertical auxiliary lines
Determines the number of vertical lines.

Trend window
Determines the time range of the graph ruler

Graph ruler colour
Determines the colour of the graph ruler within the graph ruler object

- Change the values as follows:



Entries into the edit fields are made using the keyboard. The colour is selected for the graph ruler by clicking on the colour button and choosing the colour in the Colour dialog.

- Confirm entry with **OK**.

- In the aattributes window of the "**Graph ruler**" graphics object, make the following additional changes:

Background = white
Colour = white

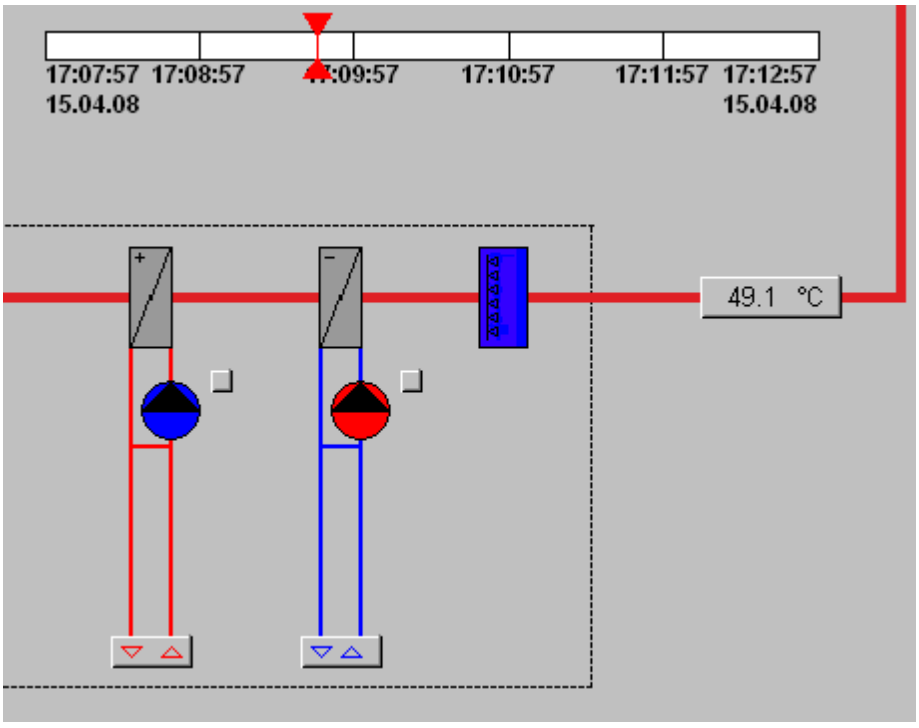
Foreground colour2 = Arial 9pt, Bold
Font

The process diagram should now look like the first image in this chapter.

Since the introductory project "Visi.Plus" has no PLC data available, the **PET** is used to enter a few numbers as values.

1. If it is not already running, start up the **HDA** module (runs in the background).
2. Then start up the **PET** or restore it if it has been minimised to the taskbar.
3. Search for the data point **BHS60:PW:508:RM_On** in the **Detail View**.
4. Enter any value, so long as it is different from the old value.
5. Search for the data point **BHS60:PW:508:Err** in the **Detail View**.
6. Enter any value, so long as it is different from the old value.
7. Search for the data point **BH60:MT:503:ActualValue** in the **Detail View**.
8. Enter any value, so long as it is different from the old value.
9. Repeat steps 3–8 as desired, allowing some time to elapse between the individual entries.

Now switch to runtime mode with the **<e> key**. The graph ruler is only operable in runtime mode. Then click in the frame of the graph ruler with the left mouse button without moving the mouse. A graph ruler (red line with two triangles) is shown at the X-position where the mouse was clicked.



The colour of the right pump and the temperature on the button should change immediately, because both objects (as well as the remaining objects in the process diagram) are disconnected from the DMS and the displayed icons or values now correspond to those of the time at which the graph ruler of the graph ruler object is positioned.
The graph ruler can now be moved back and forth with the left and right arrow keys. The

colour of the right pump and the displayed temperature should change. If the graph ruler reaches the right or left frame boundary, the time range is automatically shifted.

The graph ruler can also be moved with the mouse. To do so, left click on the upper or lower triangle of the graph ruler and move the mouse while holding down the mouse button. The graph ruler now follows the mouse pointer. When the mouse button is released, the graph ruler stops moving.

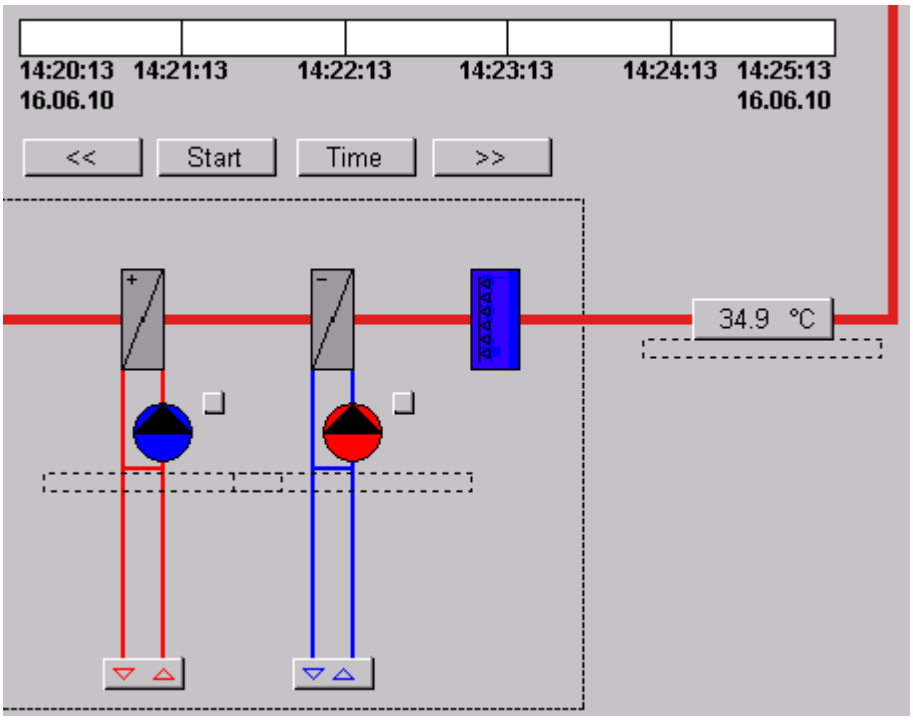
The graph ruler is switched off by clicking in the frame of the graph ruler object with the right mouse button. In doing so, the current statuses or values from the DMS are displayed again.

10.3.9.18.3 Example 12: Inserting control buttons for a graph ruler

Control buttons for the graph ruler **graphics object** can be created quickly and independently.

Although the graphics objects "**Icon**" and "**Polygon**" would be just as suitable for this purpose, the "**Button**" object is used in this case.

The process diagram with the **Control buttons** for the graph ruler should appear as follows:



First the **Back** button (far left in the image) is created, as described for the button graphics object. The text is set to <<. For the initialisation of the "**Action**" attribute of the button, select "**Graph Ruler**" and then "**Scroll Back**".

Then the buttons **Start**, **Time**, and **Forward** (far right in the image) are created as per the procedure just described. The buttons are to be initialised to the following actions:

Start **Action > Graph ruler > Start time**
Time **Action > Graph ruler > Time window**
Forward **Action > Graph ruler > Scroll forward**
Do not forget to label the buttons.

*If multiple **graphics objects** with the same appearance are to be created, as in the case just described, we recommend duplicating the first object (the Back button in the example) after its attributes have been defined. Duplication is carried out via the key combination **<CTRL+D>** and then positioning the duplicate. Then select **Delete Action** and **Reinitialise**.*

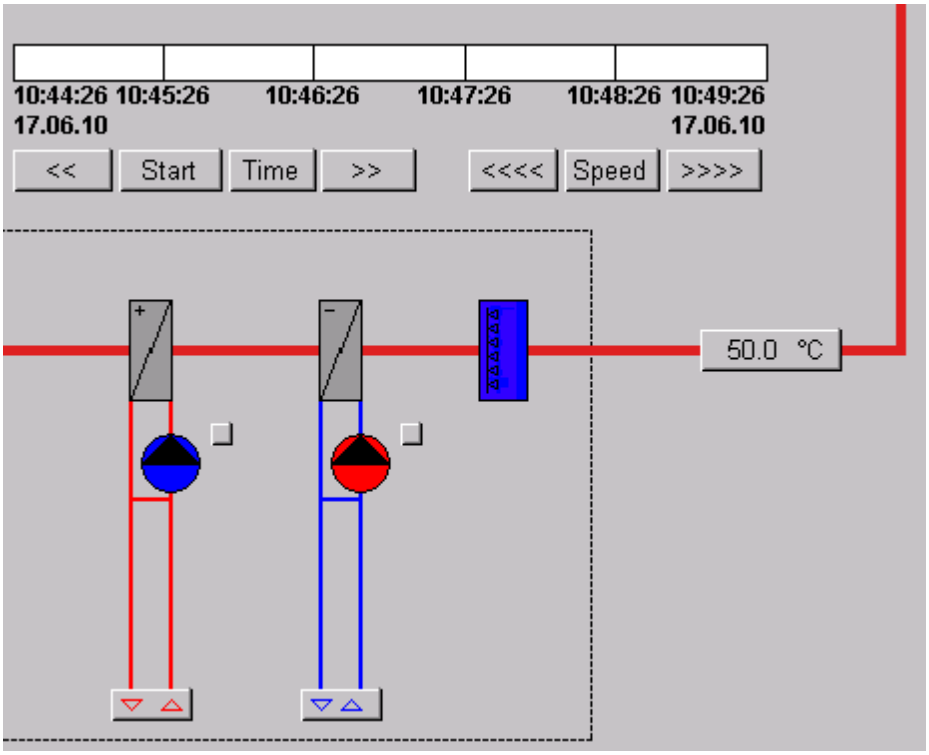
Call up runtime mode by pressing the **<e> key**. Now the time range of the graph ruler object can be move using the buttons **Back** << and **Forward** >>. If data was registered over a longer period of time, a desired starting point can be defined with the **Start** Start button. The time range of the graph ruler object can be changed with the help of the **Time** Zeit button.

The graph ruler (red line with two triangles) within the graph ruler object can be controlled with two buttons, which must be initialised accordingly. In the process, the graph ruler automatically moves either forwards or backwards over the time range of the graph ruler object (autoplay function). This is discussed in detail in the following chapter.

10.3.9.18.4 Example 13: Autoplay function

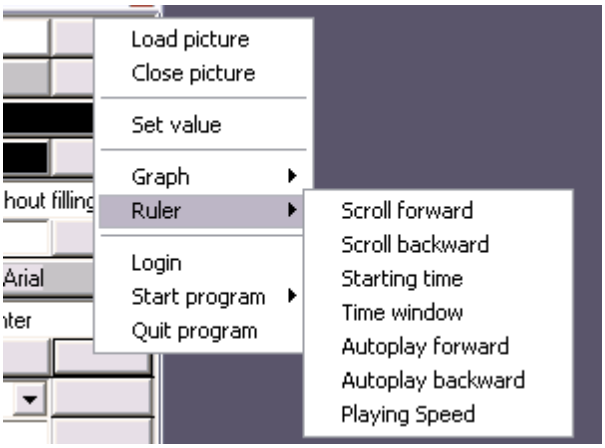
It is possible to control the graph ruler within the graph ruler object through the so-called autoplay function. In the process, the graph ruler automatically shifts either forwards or backwards over the time range. To do so, two buttons must be created with initialisation to **Autoplay Forwards** and **Autoplay Reverse**, as well as a button for entering the **Playback Speed**.

The process diagram should appear as follows:



The buttons are created as follows:

- Copy the button on the far left
- Position the button corresponding to the preceding image
- Call up the Attributes window
- Click on the button in the Initialisation column by the "Action" attribute and delete the initialisation
- Click on the button in the Initialisation column by the "Action" attribute again. The following selection menu opens:



- Click on **Autoplay Reverse**
- Click OK in the "Reverse Playback" initialisation dialog
- Copy created buttons twice and position
- Delete the initialisation for both buttons
- In the Attributes window, click on the button in the Initialisation column by the attribute "Action"
- Select **Playback Speed** for the 2nd button

- Select **Autoplay Forwards** for the 3rd button
- Then change the texts in the Attributes window according to the image shown above
- Switch to runtime mode and select the graph ruler in the graph ruler object

The 3 created buttons should now exhibit the following behaviour:

By clicking on 

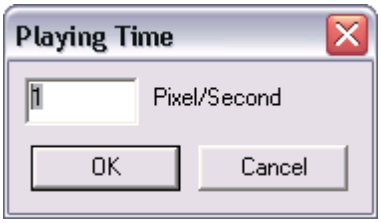
the graph ruler moves to the left within the graph ruler object. The colour of the right pump and the displayed temperature should change. If the graph ruler reaches the left frame boundary, the time range of the graph ruler object is automatically shifted along the time axis.Clicking again stops the graph ruler.

By clicking on 

the graph ruler moves to the right within the graph ruler object. The colour of the right pump and the displayed temperature should change. If the graph ruler reaches the right frame boundary, the time range of the graph ruler object is automatically shifted along the time axis.Clicking again stops the graph ruler.

By clicking on 

the following dialog appears:

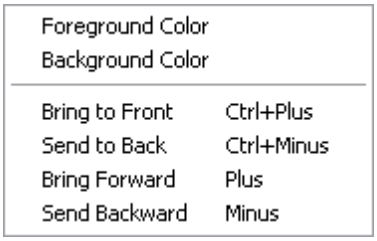


Here the speed at which the graph ruler travels can be adjusted. It is preset to 1 pixel/second. The value can be changed via the keyboard.

If the selected playback speed is too high (e.g. if the computer has insufficient memory), the graph ruler will always travel at the maximum possible speed over the time range.

10.3.10 Attributes menu

In the "**Attributes**" menu there are options for selecting the colour and alignment of objects.



10.3.10.1 Foreground Colour

The colour selection window opens, allowing you to select the desired colour for the foreground of the window or object.

Multiple objects can be marked and the foreground colour of all marked objects can be changed simultaneously!

10.3.10.2 Background Colour

The colour selection window opens, allowing you to select the desired colour for the background of the window or object.

Multiple objects can be marked and the background colour of all marked objects can be changed simultaneously!

10.3.10.3 To the foreground <CTRL> + <Plus>

Places a selected object in the foreground.

10.3.10.4 To the background <CTRL> + <Minus>

Places a selected object in the background.

10.3.10.5 Move a level forward <Plus>

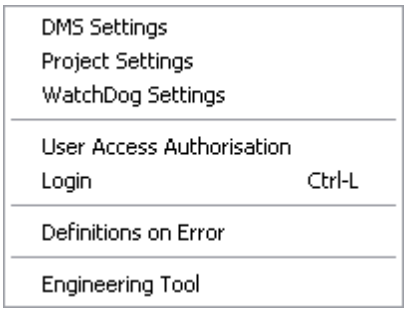
Moves a selected object forward one level towards the foreground.

10.3.10.6 Move a level back <Minus>

Moves a selected object back one level towards the background.

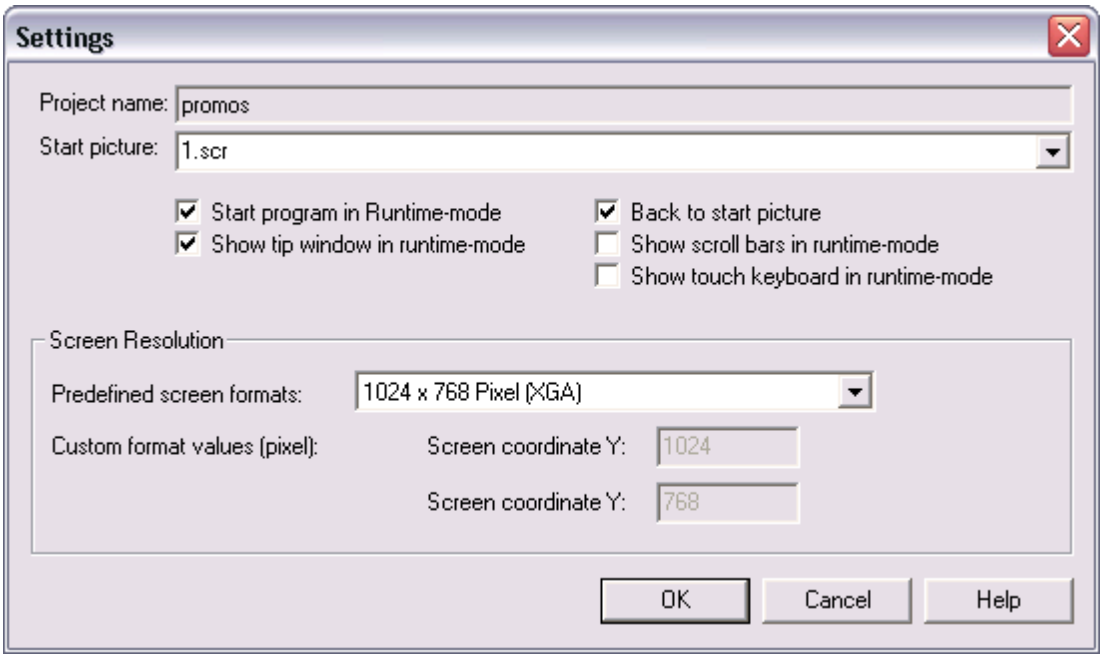
10.3.11 Project Settings menu

Basic settings for projects and settings for the Visi.Plus system can be defined in the "**Project Settings**" menu.



10.3.11.1 Project Settings

The project start screen as well as the resolution of the process diagrams can be set here.



Project Name
Shows the name of the current project (cannot be modified here; only for informational purposes)

Start Screen
Shows the name of the first process diagram on startup of Visi.Plus. A different start screen can be selected by clicking on the drop-down menu.

Program Startup in Runtime Mode
When this checkbox is marked, Visi.Plus starts in runtime mode from then on.

Auto. Jump Back to Start Screen
If no user action is detected for the duration of 5 minutes, Visi.Plus calls up the screen indicated under "**Start Screen**" and jumps back to it.

Show Tips Window in Runtime Mode
When activated, tips and tricks are shown in runtime mode.

Show Scroll Bar in Runtime Mode
When activated, scroll bars are shown in runtime mode as soon as the size of the image exceeds the resolution of the monitor that the GE is currently running on. For example, if the process diagram is generated in a size of 1600x1200, both the vertical and horizontal scroll bars would be shown in runtime mode on a monitor with the screen resolution 1024x768.

Show Touch Keyboard in Runtime Mode
When activated, a touch panel keyboard is displayed in runtime mode for all input fields (including login).

Screen Resolution
The resolution for the process diagrams of the target computer on which the images are to be shown can be adjusted here through the drop-down menu. The process diagrams are stored in the selected resolution as full images.

Logout After xxx Seconds
The time after which a user is automatically logged off with "AutoLogoff". The values are in seconds.

Connect to Mulit DMS
The graphical editor is connected to several DMS and can therefore be used as an overriding operating station. More on this topic can be found in the chapter Multi DMS.

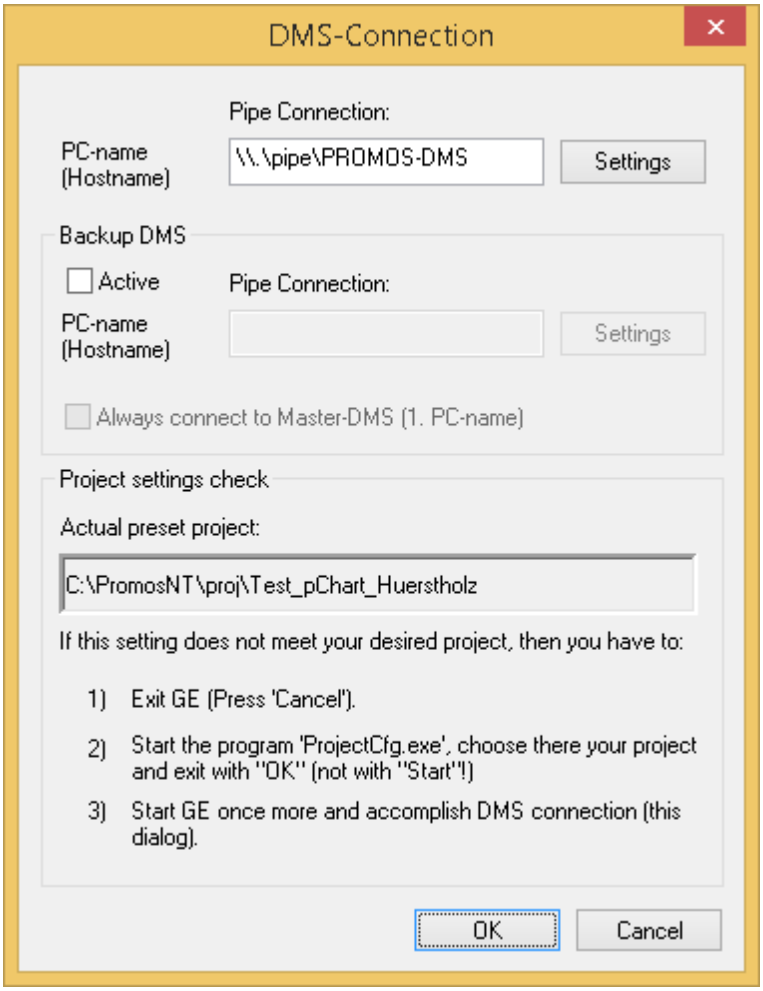
Starting Several GEs
Up to 4 graphical editors can be started on the same PC, either on several monitors or all on one. The GEs start at the same position they were at when they were last closed.
This function requires several licences for the graphical editor.

Delete DMS Links upon Request
This function is still being developed.

Visi.Plus should always be started in runtime mode for systems which are in operation. This prevents the user from being able to edit anything.

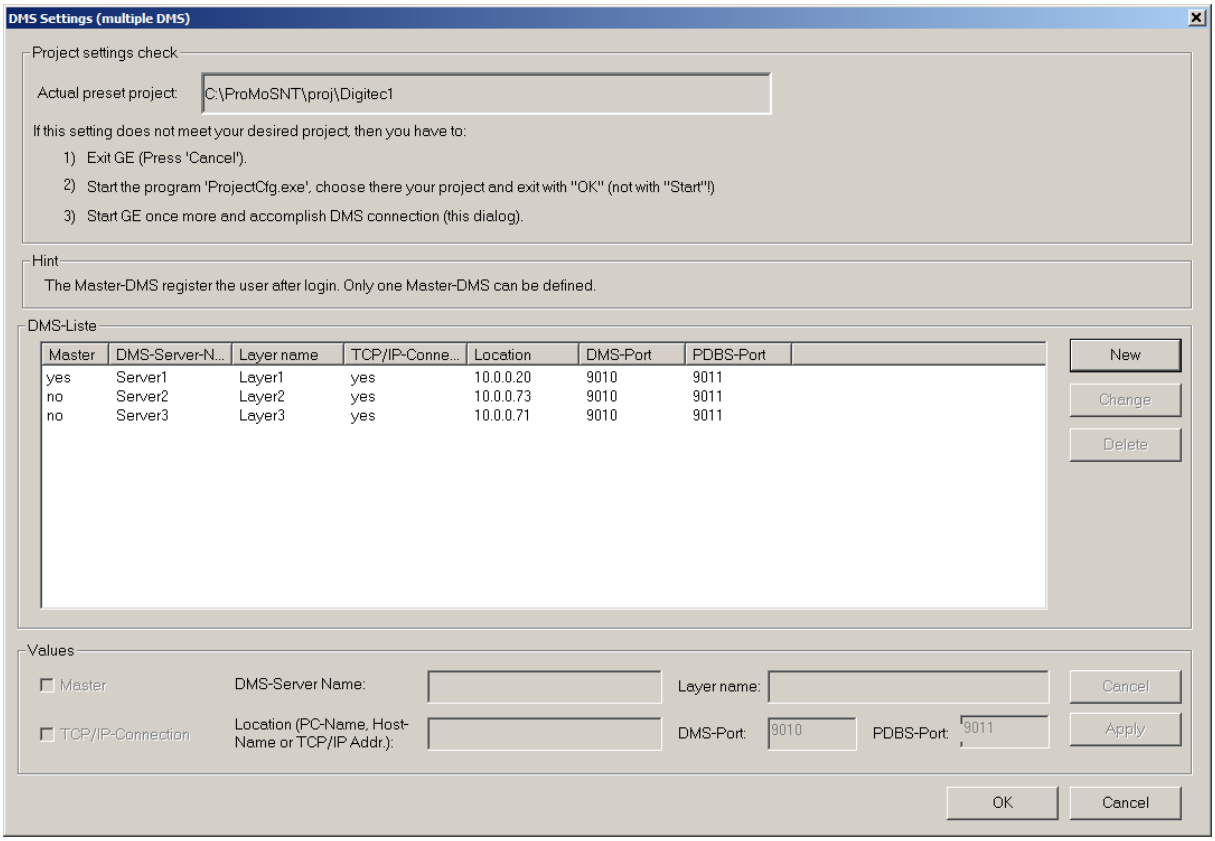
10.3.11.2 DMS Settings

If the checkbox **"Connect to a DMS"** under the menu item **"Project Settings -> Project Settings"** is activated, the following setting window appears under the menu item **"DMS settings"**:



The settings for the connection to the DMS can be made here. A detailed description can be found in the chapter Starting a Remote Graphical Editor.

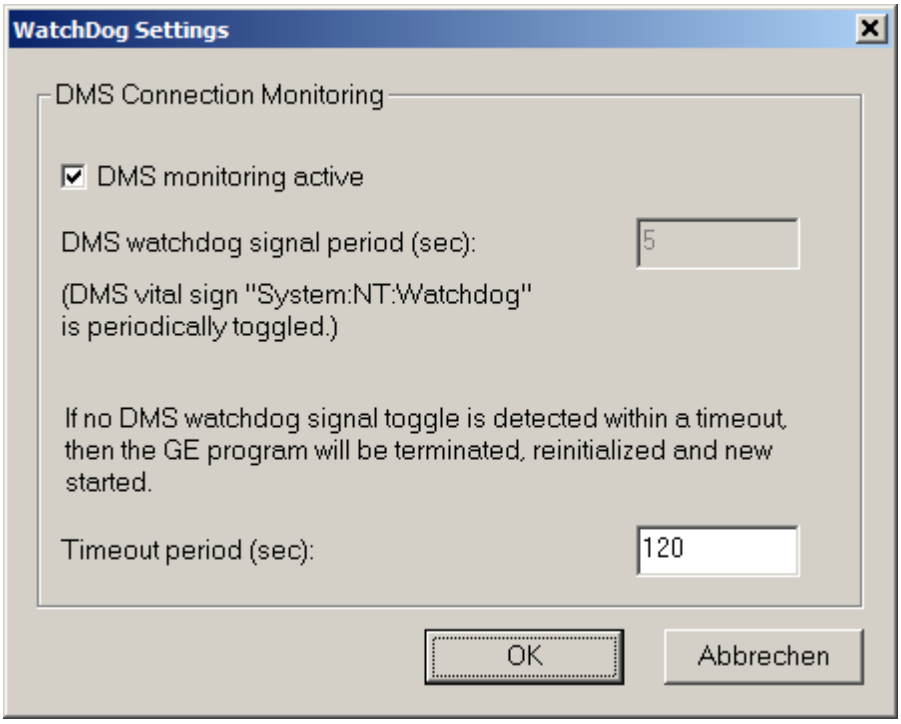
If the checkbox "Mulit-DMS" is activated, the following setting window appears:



A detailed description of a connection to Mulit-DMS can be found in the chapter Multi DMS.

10.3.11.3 WatchDog Settings

The watchdog for the GE can be activated in this window. If the watchdog is activated, the GE is restarted if no data point in the DMS is changed by the GE. When the watchdog is activated, the GE writes to the data point **"System:NT:Watchdog" every 5 seconds.** The Timeout Period text field can be used to set after how many seconds the GE should be restarted if no data point has been written. It is self evident that the timeout must be greater than 5 seconds.



10.3.11.4 Users/Access Rights

In order to assign **Access Rights** for users, they must first be defined. Up to 16 different access rights are available for assigning to the users. The number of users is unlimited.

Selecting the command "**Users/Access Rights**" starts the Visi.Plus User Management module (pUser.exe). The assignment of new users and their rights is explained in detail in the **chapter** User management.

10.3.11.5 User Login

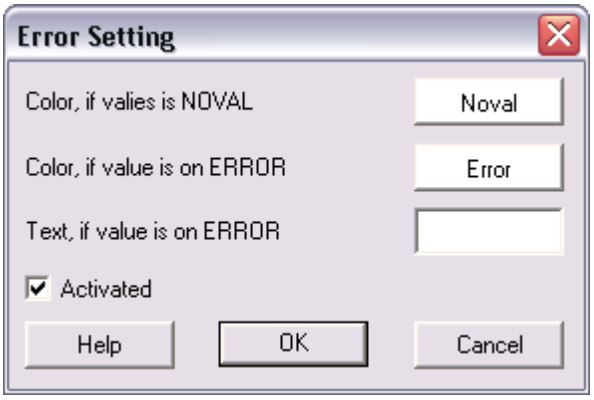
The **Login** dialog window (password input window) is called up with **<CTRL> + <L>** .

As soon as a user and access rights have been generated in the User Management module, the user must log into Visi.Plus via user authentication in order to be able to operate the GE.

Licence
By pressing the "Licence" button, a window with detailed information about the installed system licence opens:

10.3.11.6 Error definitions

Which values (only ones which have been initialised) should be displayed if communications between Visi.Plus and the PLC have been interrupted?
It would make sense to highlight these incorrect values with a colour.
Visi.Plus offers these options via the dialog window **Error Settings**:



<Noval> button
Clicking this button allows you to set the background colour to be shown when there is a communications fault.
Noval stands for "No value".

<Error> button
If there is an error (invalid data), the initialised object is displayed in the colour which can be defined by clicking this button.

ERROR Text
It is also possible to define text to be used in the event of an error. Once the error has been rectified, the initialised values will be shown correctly after the process diagram has been rebuilt.

*A process diagram which does not show **any valid data**, and for which the options in the **"Error Definitions"** dialog window have been configured, should **never be saved**! Otherwise all the elements currently showing an error colour will be saved with this colour, and then the error colours will be used as the standard colour scheme the next time the program is started.*

10.3.11.7 Engineering Tool

Call up the Visi.Plus Project Engineering Tool (PET). The PET can also be opened right from the Windows Start menu.

This tool is not a component of the Graphical Editor. Additional information on the **PET** can be found in the chapter PET.

10.3.12 Window menu

The Graphical Editor offers the option of using multiple windows for the display of the process diagrams on the screen. Each of the windows can contain the same or a different process diagram.

10.3.12.1 New Window

Opens a new window with the same content as the last activated window.

This function can be used, for example, if a process diagram window extends beyond the dimension of the screen currently in use. The contents of the new window can then be scrolled to the right. In the new window, therefore, the right section of the process diagram is displayed, while the left section is shown in the old window. This means constantly scrolling back and forth is no longer necessary. Simply click on the window with the section that should be edited.

10.3.12.2 Cascade Windows

The windows are positioned in an overlapping cascade arrangement.

10.3.12.3 Tiled

Arranges all windows horizontally tiled on the screen.

10.3.12.4 Arrange Icons

If the images have been reduced to icons, the icons are rearranged along the bottom edge.

10.3.13 Help menu

The **Visi.Plus online help** and current information about the Graphical Editor version are found in the "**?**" menu.

10.3.13.1 Help

The Visi.Plus Online Help can be called up through the **"? -> Help"** menu, via the various buttons or by pressing the **<F1>** key.

10.3.13.2 Help Overview

Displays the current version of the Time Switch Program CLKCfg.

Important
The version number must always be indicated in case of support queries.

We recommend our internet forum for support queries. This offers two advantages: First, help is available more quickly; and secondly, other users can benefit from the entries.

10.3.13.3 About ...

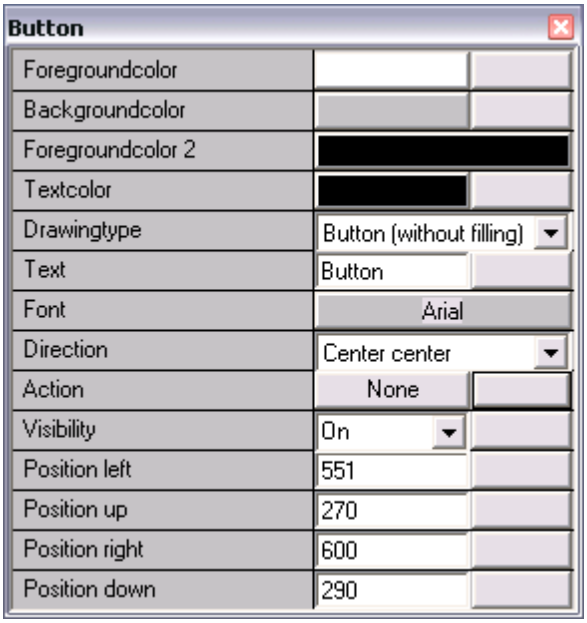
Indicates the current version of the Graphic Editor. The version must be given with any support queries or in the forum (see Appendix for addresses).

With ***ShowVersion.exe***, which is located in the default/**bin** directory, all versions used can be displayed in a simple manner and, for example, can be exported to a text file.

10.3.14 Graphical attributes

The following descriptions of the individual attributes in the Attributes window pertain to the appearance of the objects **independently** of a PLC element, for example.

The settings are made in the middle column of the relevant attribute.



The content of the Attributes window depends on the selected graphical object.

The middle column has a lower priority than the right column, the "Initialisation column". More

information about the Initialisation column can be found in the chapter Initialisation attributes.

10.3.14.1 Frame colour / Frame colour 2

The frame colour for a drawing element can be defined here. It is also possible to define two frame colours for the element button! The frame colours are visible depending on the drawing type.

10.3.14.2 Background Colour

The same procedure as under **Frame colour / Frame colour 2**.

*For the trend graph graphical object, the attribute **Background Colour** defines the surface for the entire trend object.*

*The colour of the graph is defined in the **Graph Initialisation*** dialog window (button in the right column, "graph" line).
*The colour and font of the scale or unit labels are defined by the attributes "**Text Colour**" and "**Font**".*

10.3.14.3 Line Attribute

With the attribute **Line Attribute**, the drawing type of the line is determined (e.g. solid, dotted, etc.). The following drawing types are available:

- Solid
- Dashed
- Dotted
- Dash-Dot
- Dash-Dot-Dot

The line attributes can be modified for each line width.

10.3.14.4 Line Width

With the attribute **Line Attribute**, the drawing type of the line is determined (e.g. solid, dotted, etc.). The following drawing types are available:

- Solid
- Dashed
- Dotted
- Dash-Dot
- Dash-Dot-Dot

The line attributes can be modified for each line width.

10.3.14.5 Visibility

With this dropdown menu the visibility of the object can be switched on or off.

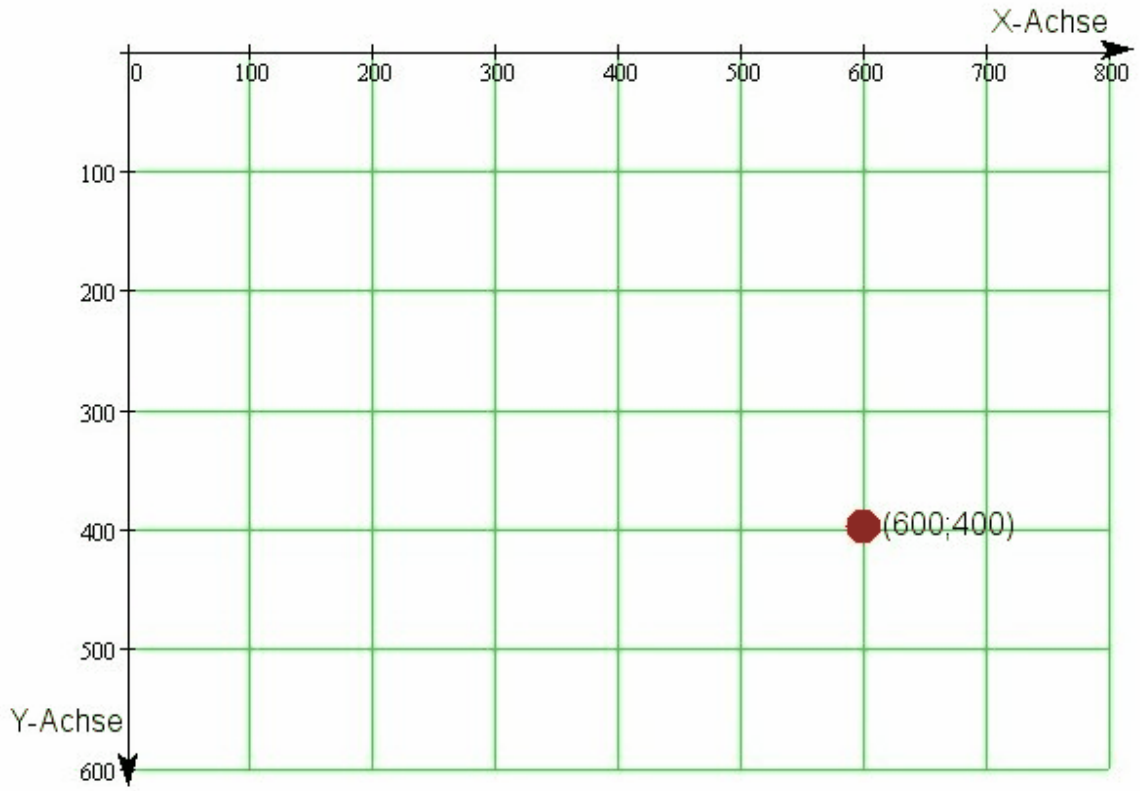
10.3.14.6 Positions start point (X1), (Y1) and end point (X2), (Y2)

Position (X1) and (Y1)
Defines the starting point of the object to the exact pixel

Position (X2) und (Y2)
Defines the end point of the object

All Visi.Plus process diagrams are broken down into XY coordinates. Coordinates are figures that indicate the position of one point in relation to another point. Absolute coordinates, like those used in Visi.Plus, are always based on the origin of the current system of coordinates.

Visi.Plus shows the position in Cartesian XY-coordinate form. XY coordinates show the position as a horizontal and vertical distance from the point of origin. The **point of origin (0x0)** is always at the **top/left for all Visi.Plus process diagrams.** The end point (bottom/ right) is determined by the resolution of the process diagram. With a **screen resolution of 800x600**, the **end point** has the coordinates **799x599**. With a resolution of 1024x768 the end point has the coordinates 1023x 767.



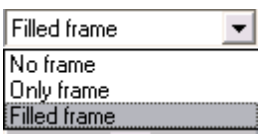
In the Attributes window the coordinates are named as follows:

Left Position stands for **X1**
Top Position stands for **Y1**
Right Position stands for **X2**
Bottom Position stands for **Y2**

When it is created, the resolution of the process diagram must be correctly configured on the basis of the computer on which it is to be used! More information about process diagram resolution can be found in the chapter Project Settings.

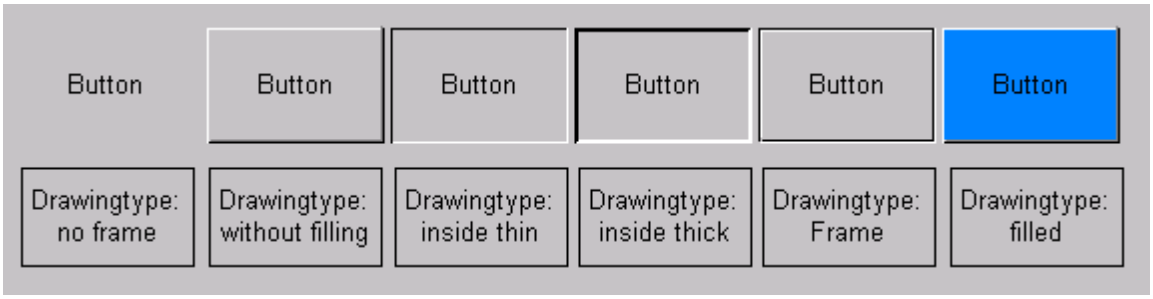
10.3.14.7 Drawing Type

If a graphical object has the attribute "**Drawing Type**", the following selection menu can be opened by clicking on the dropdown menu:



This allows you to select the desired border for the graphical object.

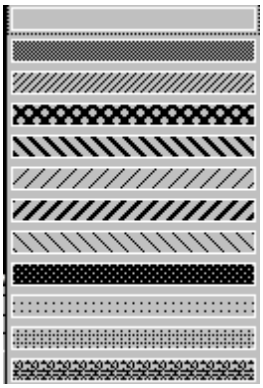
The following drawing types are available for the **Button graphical object**:



*For **Button** graphical objects, the drawing type is set to **Button (without fill)** by default. This means that the **colour of the button** does not change by adjusting the background colour.*
*The drawing type must therefore be set to **"with fill"** if you want to force a colour change on the button.*

10.3.14.8 Fill Pattern

By clicking on the dropdown menu for the "**Fill Pattern**" attribute, the following fill patterns are available:



In addition, the fill pattern can be switched to transparent or non-transparent.

10.3.14.9 Text Colour

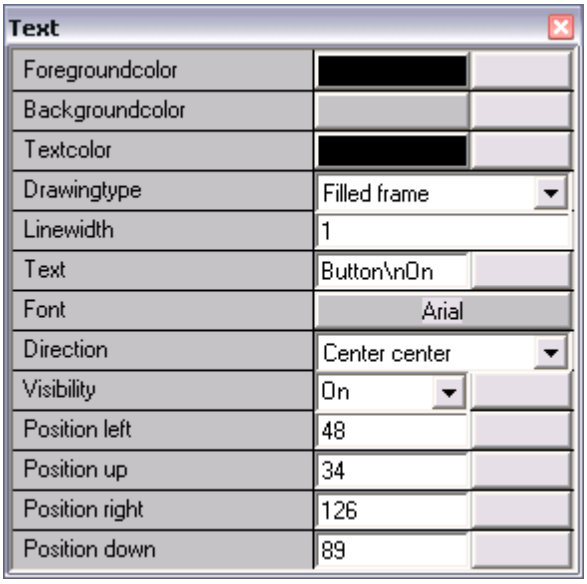
The same procedure applies here as under "Frame colour / Frame colour 2".

10.3.14.1(Text

The text to be displayed can be written directly in the **Text** field.

The text is finalised by pressing the **<ENTER>** or **<TAB>** keys. You can also click somewhere else with the mouse (e.g. on the process diagram).

Texts of two or more lines are also possible (you only need increase the height of the graphical object!).

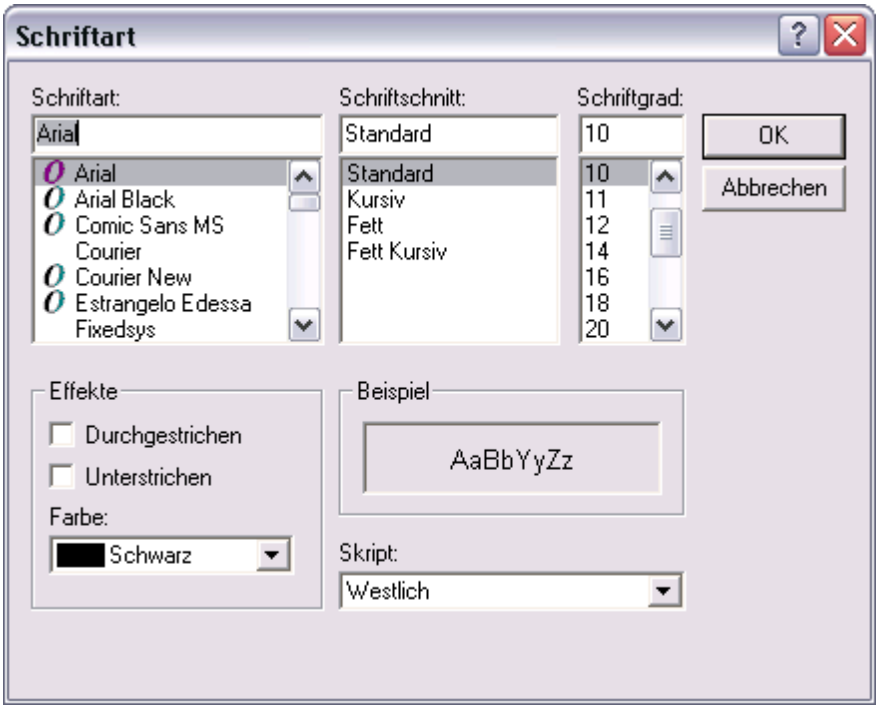


Manual line breaks are defined by the parameter entry **\n**.

10.3.14.1 Font

With the **Font** attribute, you can select the desired font, style, size and additional font properties.

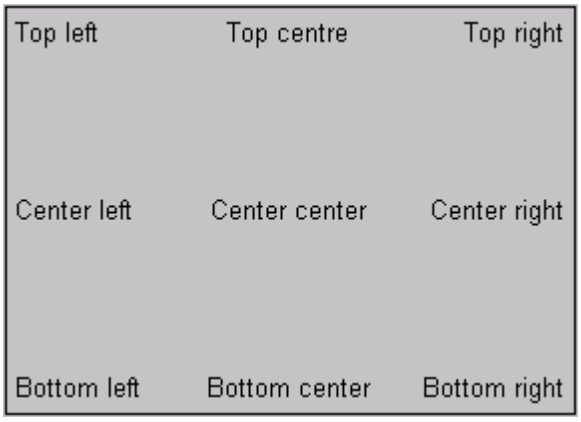
The settings are confirmed and adopted by clicking on the **<OK>** button.



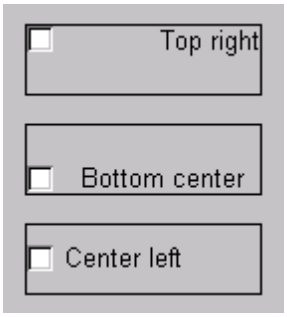
10.3.14.12 Alignment

With the **Alignment** attribute, a text that has been entered can be arranged within a text field. It is possible to rotate the text or place it on one side or in the corner of the text field.

The alignment can be selected by clicking on the dropdown menu.



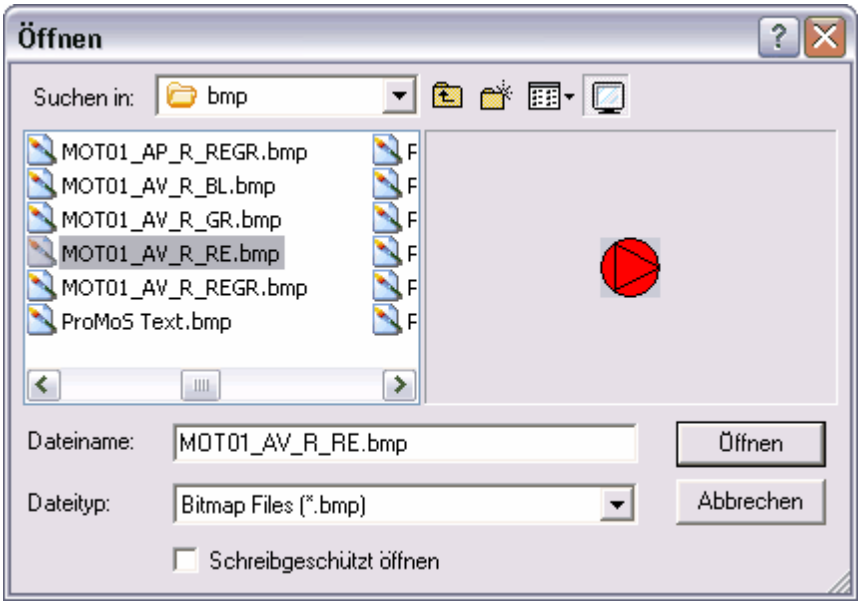
The object is always drawn to the right for checkboxes and radio buttons. Similar to text fields, the text can be aligned within the box. Rotation is not possible.

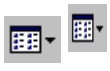


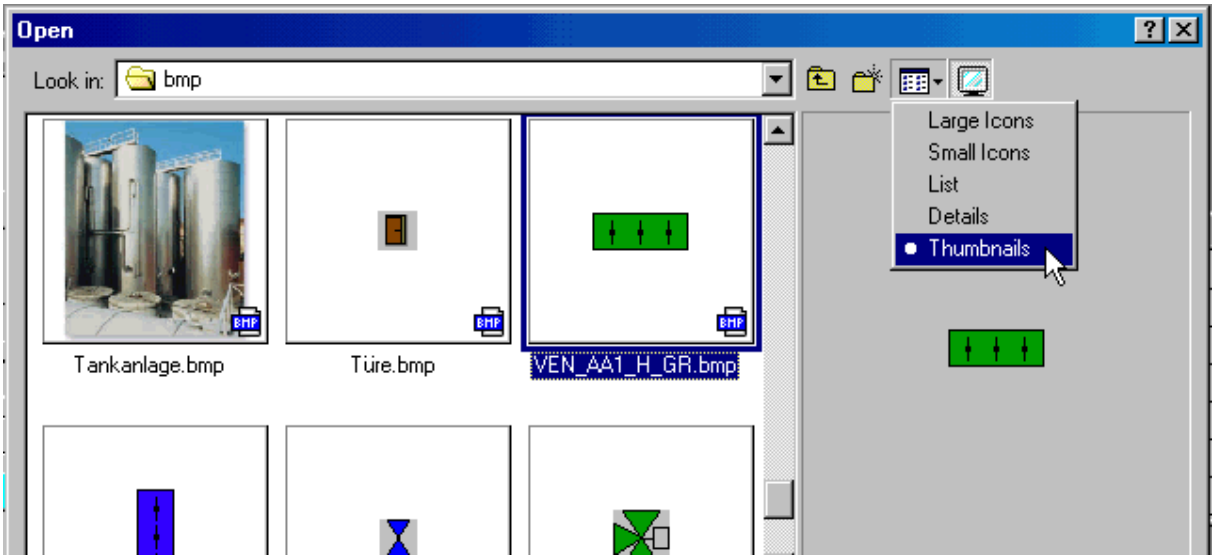
Only left, centre and right are available for input fields. The text cannot be rotated here either.

10.3.14.11 Icon Name

In the input field next to **Icon Name**, enter the icon name (with a **bmp** or **jpg** extension) with the keyboard or select it through a dialog window. The dialog window is opened by double-clicking on the field with the left mouse button:



If multiple icons should be displayed at one time, click on the  icon and select **Thumbnails** in order to see a preview of the icon. It is then possible to select multiple icons in a simple manner.



In order to enlarge the selection window, pull on the **handle**  at the bottom right with the mouse button depressed.

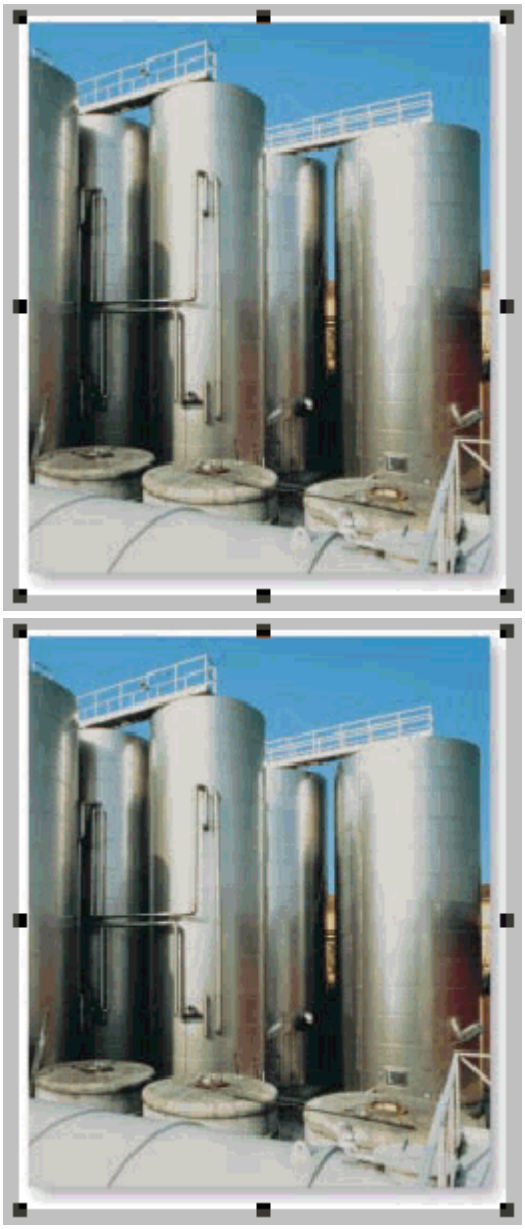
In order to see even more, you can also click on the small screen icon  .

***All** image files used in process diagrams must be saved **in the directory** \bmp **of the respective project directory**.*

In the following example a bitmap was selected from the example project **Visi.Plus** included in the scope of delivery. The bitmap, therefore, is located in the directory "**C:\Visi.Plus\<ProjectName>\bmp**".

The selected bitmap appears in the bitmap frame drawn in advance.

The **original size** of the bitmap can be restored after selection with the right mouse button and subsequently selecting the command "**Original Size**".

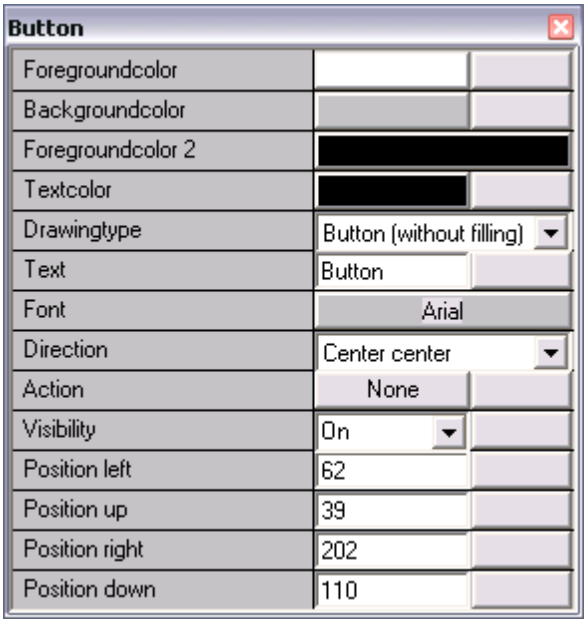


10.3.15 Initialisation attributes

The following descriptions of the graphical object attributes pertain to their **initialisations** to **PLC elements**.

To learn more about the procedure for initialisation, it is mandatory to read the description in the Foreground Colour/Text Colour attribute (following section 10.2.14.1).

Settings for the relevant attribute are made in the right column, called the **Initialisation column**.



The content of the Attributes window depends on the selected graphical object.

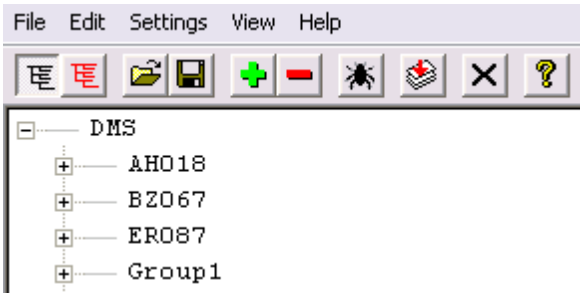
*The middle column, the Attributes column, has a lower priority than the right column, the **Initialisation column**.*

Preparation

Some generally applicable explanations must be made before the individual initialisations are presented:

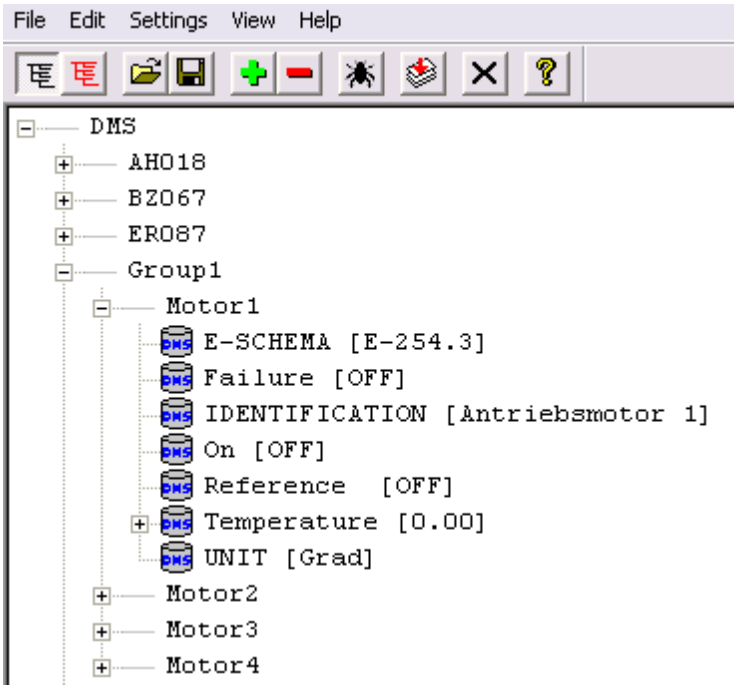
Since a PLC is not always immediately available for testing, the DMS of Visi.Plus provides some simulated data points under "**Group1**" for experimentation purposes.

In order to see these simulated data points, simply double-click on the red dot in the bottom right corner of the screen to call up the DMS window. The top level of the DMS structure becomes visible.



*If "**Group1**" is not visible or should be switched off at a later time, this can be done in the DMS under the menu item "**Settings > Value Simulation (Group1**)".*

After clicking with the left mouse button on the small plus sign in front of the name "**Group1**" and then clicking on "**Motor1**", the structure of "**Motor1**" becomes visible.



*If the switch status of the motor should be shown, the **DMS name** for this is "**Group1:Motor1:On**".*
*More information about the DMS (Data Management System) is found in the **chapter** **Database programs**.*

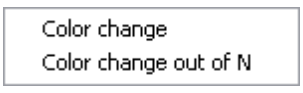
The following initialisations are explained with the help of these simulated data points from "Group1" and some data points from "System".

10.3.15.1 Foreground Colour/Text Colour

A colour change can be performed depending on a signal status with the attribute **Foreground Colour** (also called **Text Colour** for the text, button and graph graphical objects).

*The following instructions for entry of the DMS name, the colour selection and changing the respective initialisation are representative for the remaining **initialisation attributes**.*

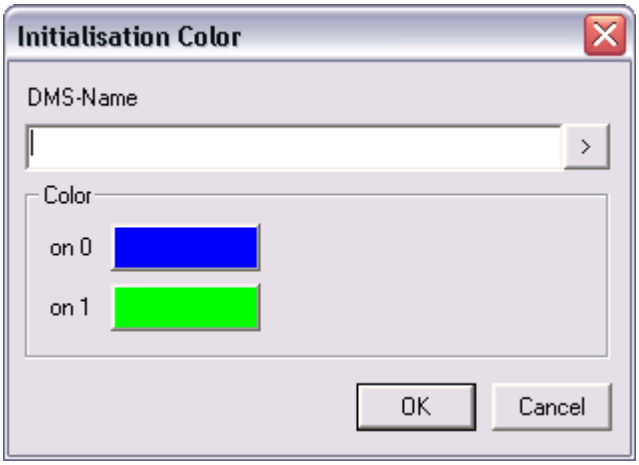
Click on the button **"Foreground Colour** in the Initialisation column. A new selection menu is displayed.



Colour Change
Used to switch colour depending on a PLC flag

Colour Change from N
With this, any number of colours can be displayed depending on a DMS value. An example is the colour designation of a boiler depending on the temperature. This example is presented in the chapter Background Colour.

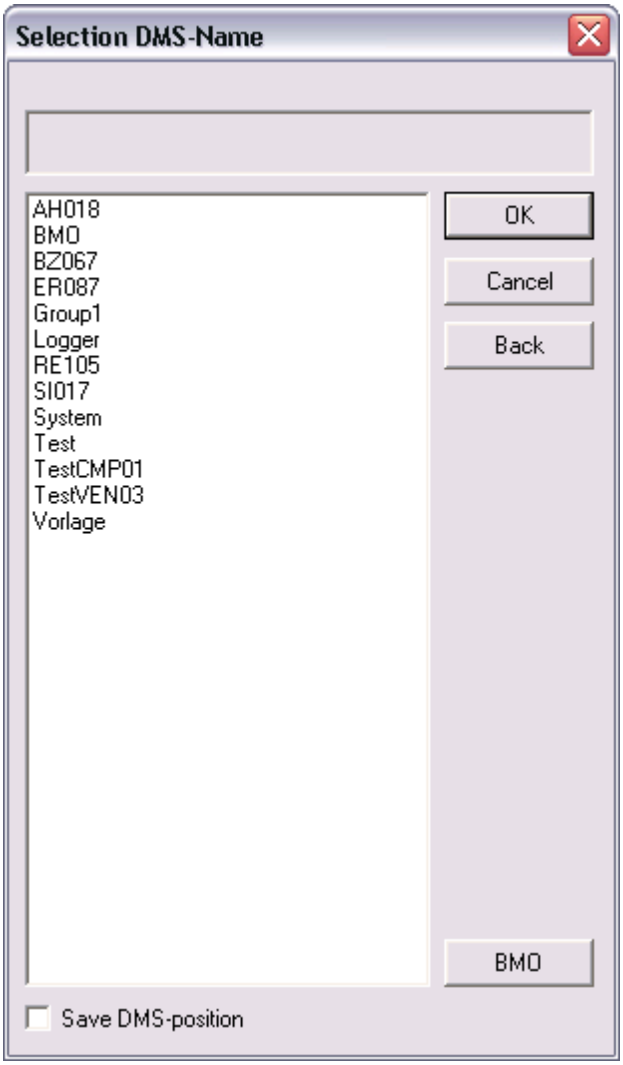
Now click on Colour Change. The following dialog window is displayed:



The blank, white input field awaits the DMS name of the data point, in this case a binary element such as a PLC flag.

*When a DMS name is is explaied in more detail in the **chapter** **AKS system**.*

The DMS name can be typed in (if known) or selected by double-clicking in the input field with the help of the dialog window **Select DMS Name**.



As you can see in the preceding image, the top DMS level is always listed unless **Save DMS Position** was activated in advance (description on the following pages).

In the example shown, only the data points available from Visi.Plus in the DMS by default are currently shown for selection (the same ones as after opening a new project).

The entries BMO, Group1 and System shown in the preceding image are expanded bit by bit (automatically and invisibly in the background) with entries in the PET on entries by the project engineer/programmer.

The description for entering data points in Visi.Plus is found in this manual in the chapter PET.

By selecting the desired entry, the next level is prepared for selection, etc. This should be continued until the actual data point is reached (see the following **example**).

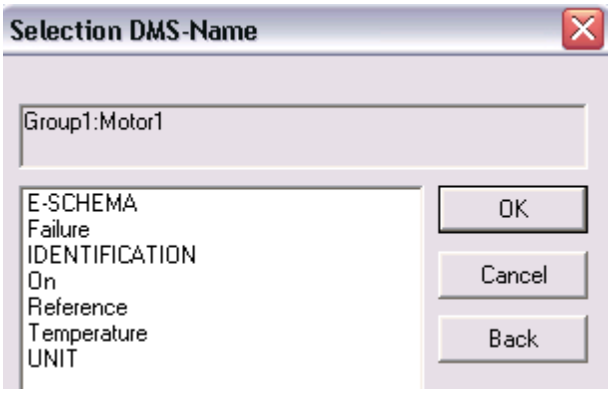
Example task:
Change foreground colour of the graphical object depending on the **fault** status of the object "**Motor1**".

No fault = green
Fault = red

Solution:
Initialisation to "Group1:Motor1:Fault"
The image shown on top is always displayed on the screen!

Select in turn:
Group1 > Motor1 > Fault

During compilation of the DMS name, the complete DMS name "**Group1:Motor1:Fault**" gradually appears in the following dialog window in the field with a grey background.



Description of the controls:

<Cancel>
Cancels the selection and returns to the "Colour Change Initialisation" dialog window

<Back>
Jumps back to the preceding selection

<Templates>
Displays all available templates (VLO)

Save DMS position
Saves the current position (level)
Example:

- Current position = "Group1:Motor1:Fault"
- Activate **Save DMS Position**
- The position "Group1:Motor1:" is already displayed on the next call-up angezeigt

<OK>
Adopts the DMS name

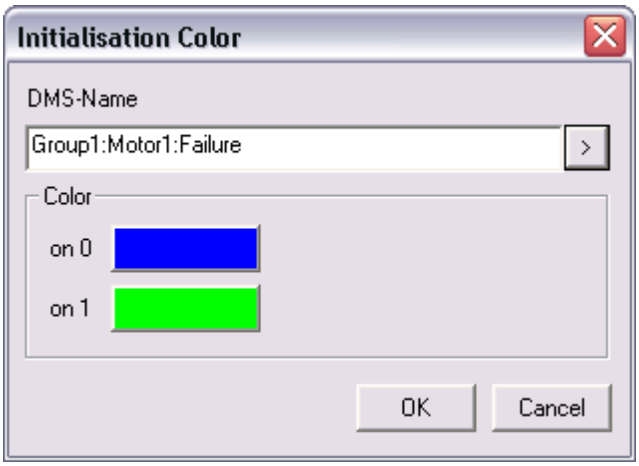
The selected DMS name is now adopted by Visi.Plus into the Colour Change Initialisation dialog window.

Now the colours red and green must still be selected.

The two current colours are displayed in the field "**Colours**".

at 0:
The selected colour is displayed if the selected data point has the value logic 0.
at 1:
accordingly for logic value 1

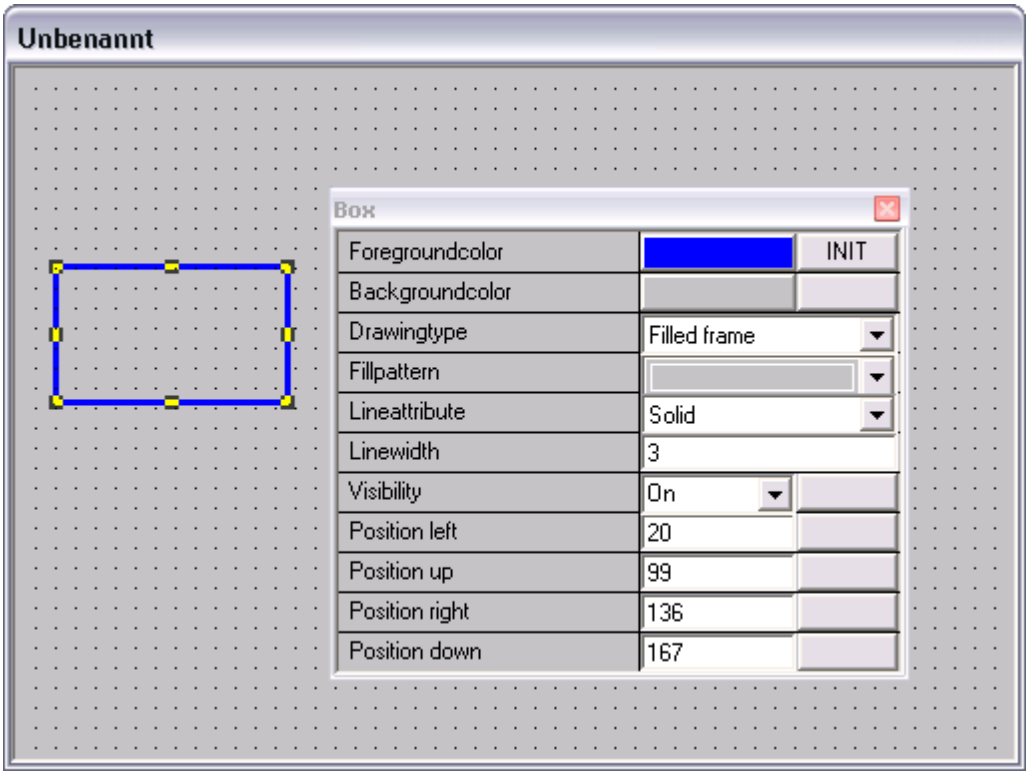
The colours can be selected by clicking on the respective button in the "Colours" field.



<OK>
Adopts and confirms the settings. The current status of the data point on the graphical object becomes immediately visible on the process diagram.

<Cancel>
Cancels the entire initialisation of the colour change.

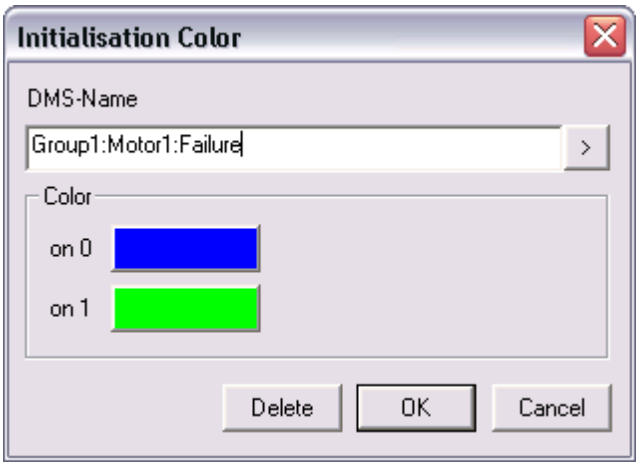
You can now see in the Attributes window that the **foreground colour** has been initialised to a data point (**INIT**) and therefore has priority over the Attribute column (middle column).



Change/Delete Initialisation

In order to **change an initialisation**, click on the **<INIT>** button and carry out the change, as already described for first initialisation.

In the case of an already initialised attribute, the dialog window "Colour Change Initialisation" displays the additional button **<Delete>**, which can be used to remove this initialisation.



10.3.15.2 Background Colour

Colour Change from 1

For the **background colour**(mostly fill surface), proceed exactly the same as with the foreground colour.

Example task:

A suitable graphical object should draw attention with a **flashing background**.

Solution:
Click on the Initialisation column in the Attributes window under **Background Colour** and enter or select the following DMS name (recommended):

System:Blinker:Blink1.0

Select colour, finished.

As a result, the graphical object will flash at the selected frequency.

Blink1.0 *corresponds to one second and is generated by Visi.Plus. Therefore this has nothing to do with the PLC blinker.*
Under the DMS name ***System:Blinker****, there are various Visi.Plus system blinkers available. Since this function places a heavy burden on the system, it should be used sparingly.*

The initialised data point ***System:Blinker:Blink1.0*** *can also be observed in the DMS.*

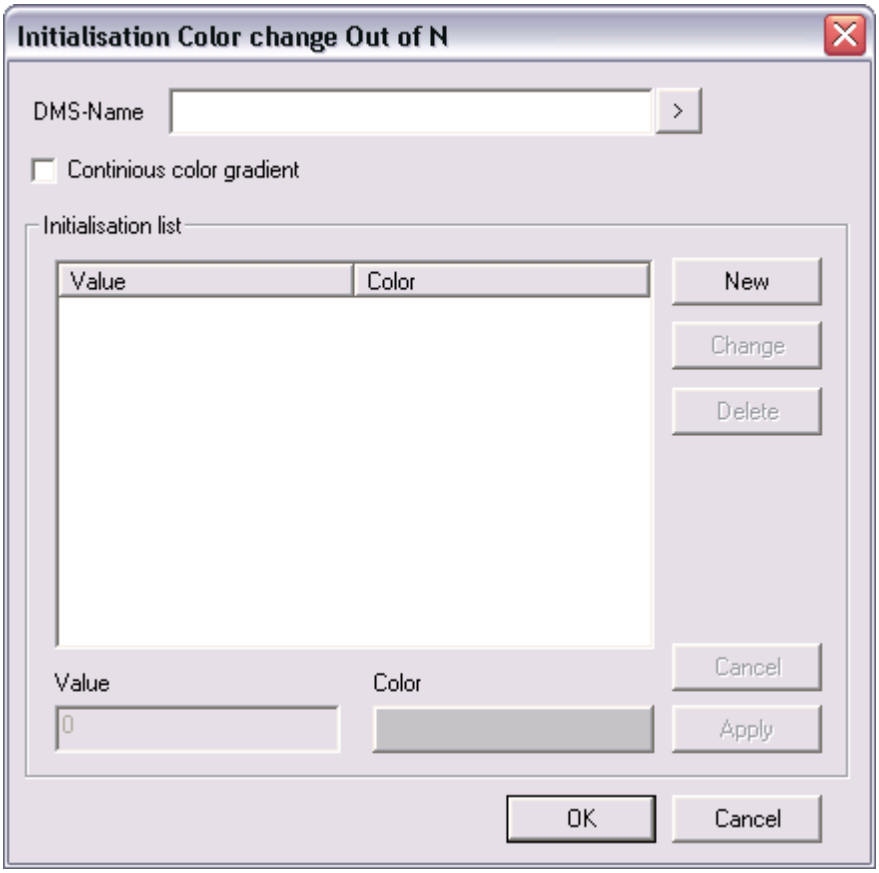
- *To do so, just double-click on the red dot in the Windows taskbar to the bottom right.*
- *The DMS display window becomes visible.*
- *Select System > Blinker > Blink1.0.*
- *The current blinker status is shown in square brackets. The value is only updated every 5 seconds in order to avoid unnecessary load on the system.*

The value update time can be adjusted in the menu by selecting:
"***Settings > General > Update Interval(s)***".

Colour Change from N

Along with the Colour Change from 1 option, Colour Change from N is available for foreground, background and text colours. With this option colour changes can be realised depending on the value of a digital or analogue, e.g. in order to provide a colour visualisation for the temperature of a boiler or a a motor (see examples).

In order to initialise the background colour to a Colour Change from N, click with the left mouse button on the Background row in the Initialisation column (right column) in the Attributes window and select the entry Colour Change from N from the selection menu. The dialog window **Colour Change from N Initialisation** appears.



In the top input field, enter the DMS name that the colour change should apply to or select it from a list. The list appears after double-clicking in the entry field or by clicking on the button to the right of it.

Then the value at which the colour change should take place, as well as the colour that should be shown when the value is exceeded, are indicated. To do so, click on the **New** button. The fields Value and Colour can now be edited. The value is entered with the keyboard. The colour is selected by clicking on the Colour button in the Colour dialog which opens.

By clicking on the **Adopt** button, the new values are adopted into the initialisation list, where the colour is entered as a number with its red, green and blue portions, e.g. "255 0 0" for red. The values in the list can now be changed or deleted. The actions **Change** and **Delete** always refer to the active line (highlighted in blue). In order to select a line, click with the left mouse button on the corresponding line in the Initialisation list or scroll through the list with the arrow keys if a line is already selected.

New button
Allows you to input a value with the corresponding colour

Change button
An already entered value and its colour can be changed.

Delete button
Deletes the active line

Cancel button

A newly created entry is not adopted into the list.

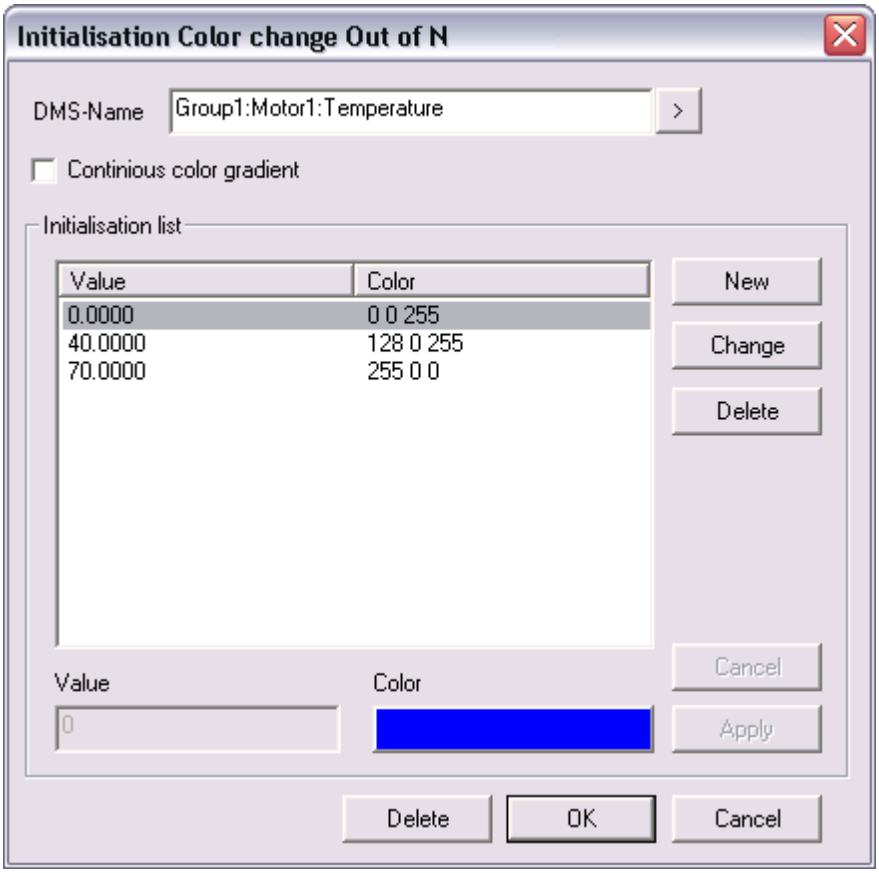
Adopt button
A newly created entry is adopted into the list.

Continuous Colour Gradient
If the checkbox is marked, the colours are mixed (see example 2).

Example objective 1
The temperature of a motor should be displayed by colour. The colour change should take place at discrete values. In addition, the temperature value should be displayed.

Example solution 1
- Draw a text field as described under Text/Text Field
- Call up the Attributes window.
- Click on the button in the Initialisation column by the attribute "Text" and select the command "Value Display".
- In the Initialisation dialog (for more information see the chapter Text), enter the DMS name Group1:Motor1:Temperature or select it from a list. Set the Output Format to %3.2f. The data point **Group1:Motor1:Temperature** leads to a simulated, continuously changing value.
- Drawing a circle for the motor
- Call up the Attributes window.
- Click on the button in the Initialisation column by the attribute "Background Colour" and select the command "Colour Change from N".
- The dialog window Text Change from N Initialisation appears.
- Enter the values as shown in the image:

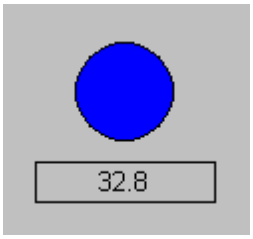
Colour 0 0 255 = blue
Colour 128 0 255 = violet
Colour 255 0 0 = red



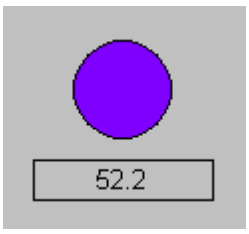
Does the generated circle exhibit the following behaviour?

The colour of the circle changes immediately after closing the dialog.

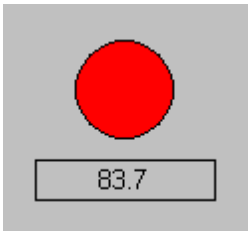
When we join the simulation, the temperature is less than 32.8°C. The circle, whose fill colour indicates the temperature of the motor, therefore is blue:



With a temperature greater than or equal to 40□ a colour change to violet takes place:



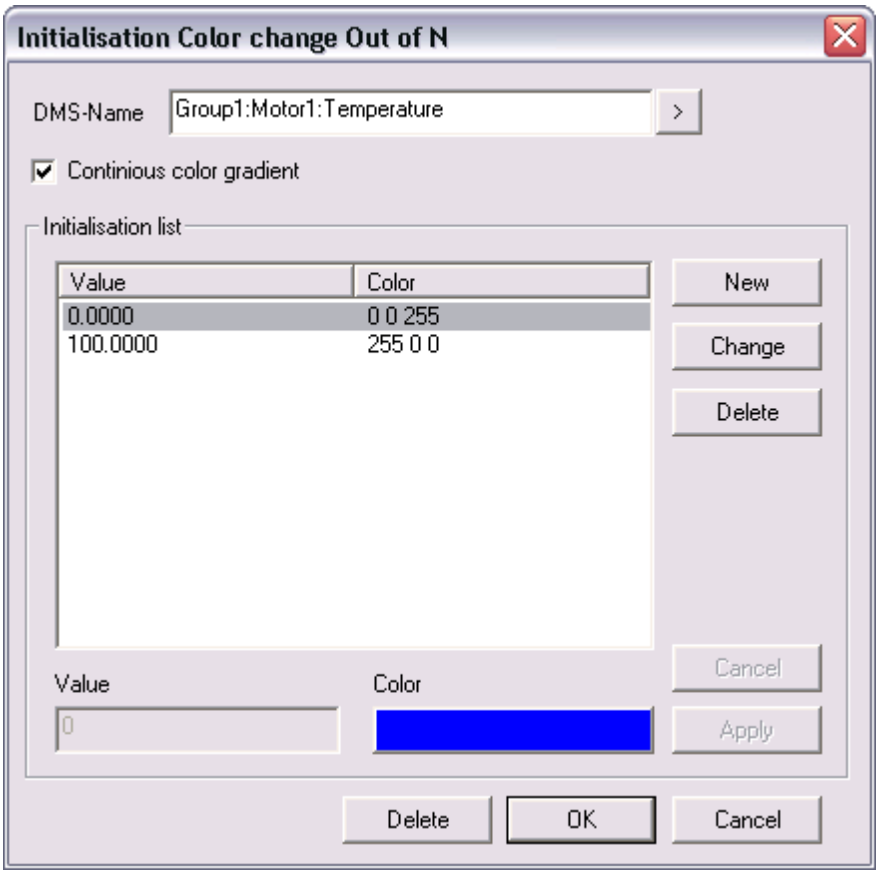
If the temperature exceeds 70□, a colour change to red takes place.



Example objective 2
As in example 1, the temperature of a motor should be displayed by colour. However, in this example the colour change should progress continuously, e.g. from red to violet to blue.

Example solution 2
- Copy the circle and text field generated in example 1.
- Select circle
- Call up the Attributes window.
- Click on the button in the Initialisation column by the attribute "Background Colour" and select the command "Colour Change from N"
- The Initialisation list varies as follows:

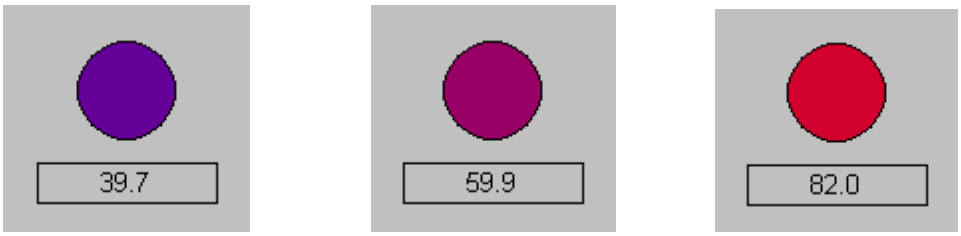
Colour 0 0 255 = blue
Colour 255 0 0 = red



- Mark the checkbox "Continuous Colour Gradient"

Does the generated circle exhibit the following behaviour?

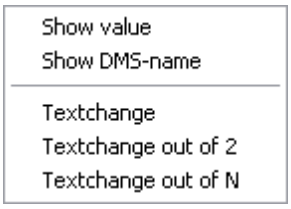
A mixed colour between red and blue is shown, according to how far the temperature is from the threshold, which means a mixture of 60% blue and 40% red is shown at a temperature of 40□ in this example.



10.3.15.3 Text

By initialising a text, different texts and values can be displayed and text changes can be applied.

Click the switch in the initialisation column under the "**Text**" property to open a new selection menu:



Value display
Displays the content of a DMS data point.

Display DMS name
Displays the DMS name in the text field. For more details see chapter Display DMS Name.

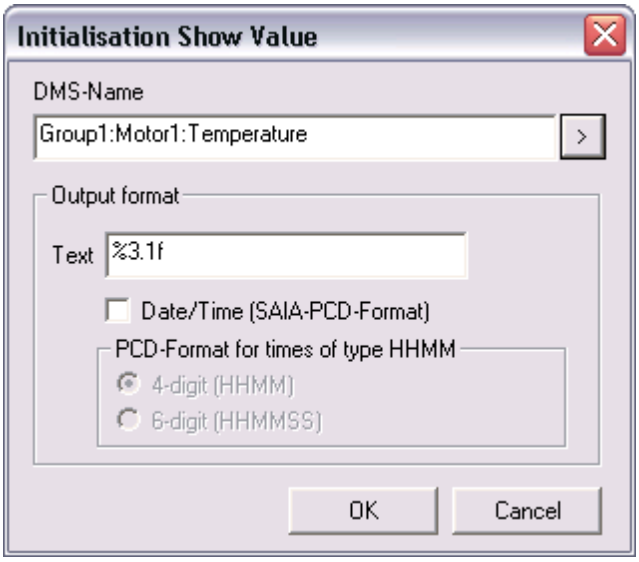
Text change from 1
Used for the text changeover depending on a data point. For more details see chapter Text Change from 1.

Text change from 2
Provides the option to define texts depending on 2 signal states. For more details see chapter Text Change from 2.

Text change from N
Any number of different texts can be displayed here. For more details see the Chapter Text Change from N.

Value display

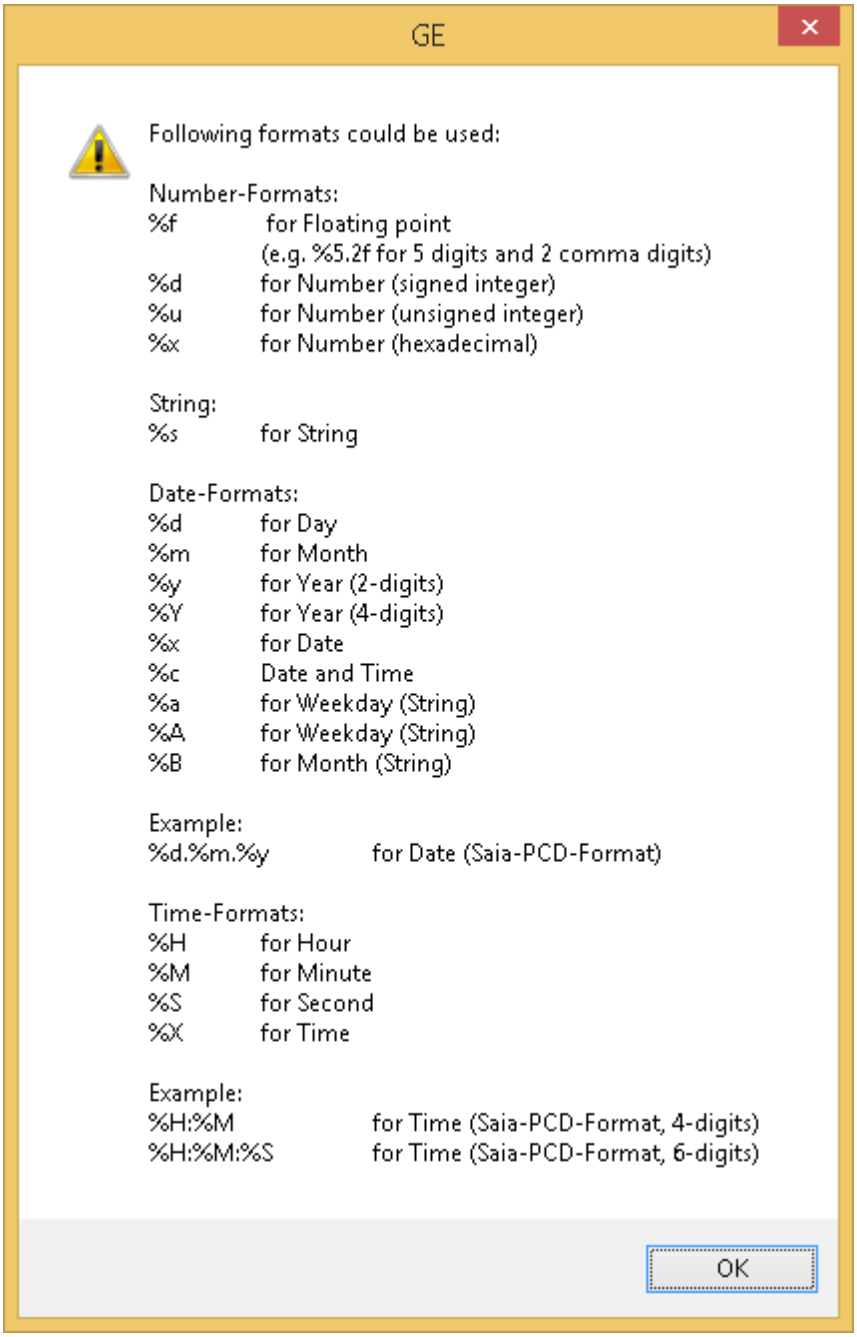
By calling up the **Value display** command, a value (e.g. temperature) can be displayed on a process screen.



A DMS name has already been specified in the input field **(Group1:Motor1:Temperature)**. This data point performs a simulated constantly changing value.

In the **Output format** section **the** value output can be freely defined based on the syntax of the “Programming language C” via the text input field.

The permitted formats can be displayed by clicking the **Formats** switch:



Some examples of this are:

Format
%3.2f
The numeral **3** defines the **number of digits to be output** (including the decimal point!).

The numeral **2** defines the **number of decimal places**

f denotes the **floating point number**.

The numerals can be changed as required, e.g. %6.4f (6 digits to be output (incl. decimal point), 4 of them are decimal places, 1 is a digit before the decimal point).

%3.1f

3 digits of the floating point value with 1 decimal point are displayed (max. 80 characters in the input field).

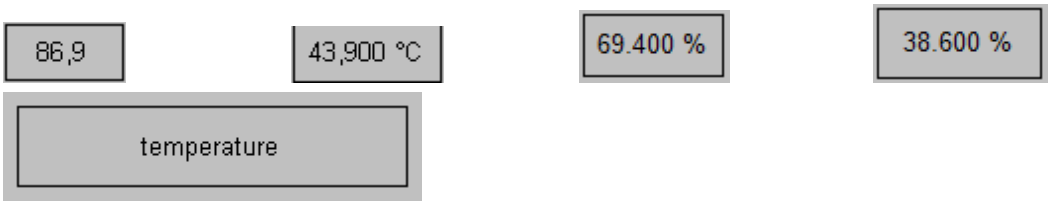
%3.0f °C
3 digits of the floating point value are displayed without decimal points, **including unit** and any text is also output (max. 80 characters in input field).

%6.3f %%
6 digits of the floating point value with 3 decimal points are displayed, **including percent (max. 80 characters in the input field). The percent character must be doubled ("% %").**

%d
Integer value (integer).

%s
String output (character string). The length of the character string may not exceed **80 characters** (including ALL characters).

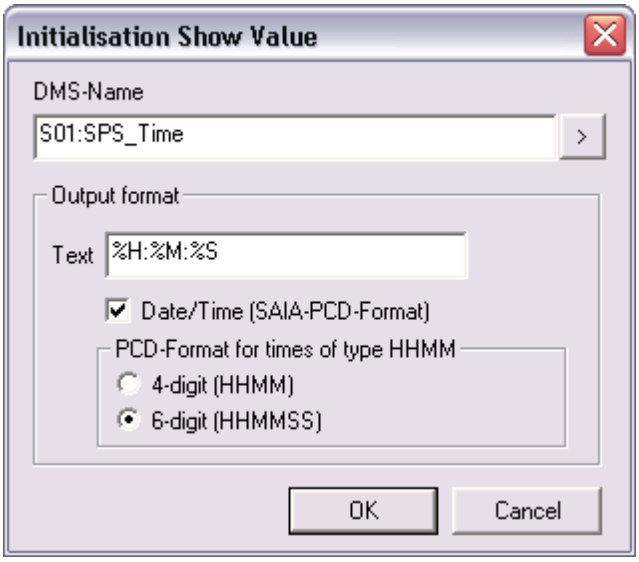
Examples



Format: %3.1f Format: %3.3f °C Format: %6.3f %% Format: %d
Format: %s

A value is never truncated. Therefore if %3.2f is specified as the output format, 2 decimal places must be displayed. As a character is also used for the representation of the decimal point, no digit is displayed before the decimal point (e.g. .34). However Visi.Plus *will display the value correctly (e.g. 23.34).*

The **time format** of Saia PCD® is converted to a floating number by the Visi.Plus communication driver.



For the PLC time to be displayed in the general format, the **Time** checkbox must be activated. The **Date** checkbox must be selected to display the PLC date.

The Saia PCD format used (4-digit or 6-digit), must also be specified for the time. The output format is preset when the checkbox is activated.

Output format time 4-digit: %H:%M
Output format time 6-digit: %H:%M:%S
Output format date: %d.%m.%y

Format

%H:%M
Hours:Minutes (23:15)

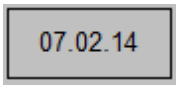
%H:%M:%S
Hours:Minutes:Seconds (23:15:04)

%d.%m.%y
Day.Month.Year (24.12.03)

The predefined format can be overwritten at any time, e.g. the date with the year written as text:

%d.%m.%Y
Day.Month.Year (24.12.2003)

Here, the date is saved in the DMS as a 6-digit number (Saia format), e.g. 140207 for the date 07.02.2014. If 20140207 is saved in the data point, a line is displayed (invalid date). An error message occurs when writing the data point in the Saia format using a button.



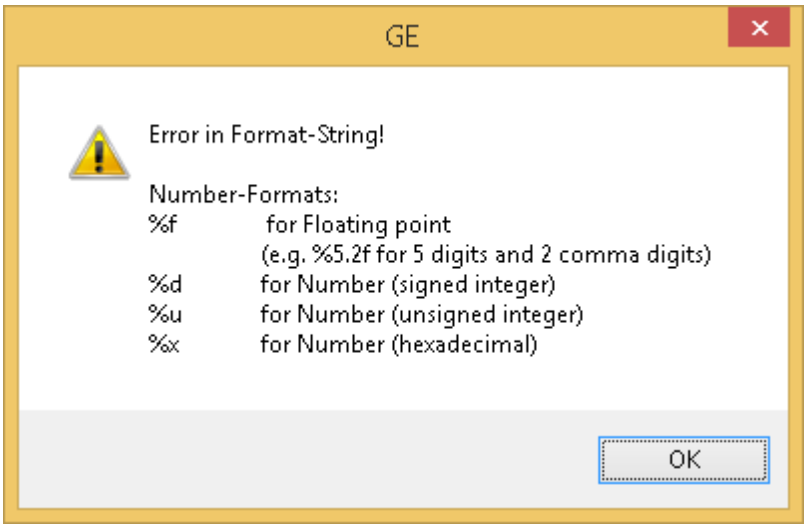
Date

The time is either saved in the DMS as a 4-digit number or 6-digit number (Saia format).
4-digit: 2315 for the time 23:15 (Hours:Minutes)
6-digit: 231504 for the time 23:15:04 (Hours:Minutes:Seconds)

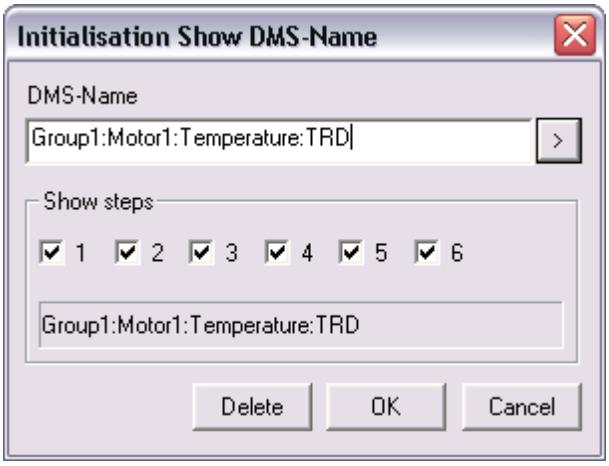
4-digit time *6-digit time*

A line is displayed for an invalid time or date value, as for the DMS date value 140230 (30.02.14 invalid), for example.

Click **OK** to check if the output format entered is permitted. If this is not the case, the formats that can be selected are displayed. The formats displayed are based on the type of data point selected. For example, if it is a FLT data point and it was selected as output format %s, the following error message is displayed:



Display DMS Name
With this, a **DMS name** can be output in the text field:



A DMS name can be selected by double-clicking on it in the input field. In the example, the data point

Group1:Motor1:Temperature:TRD was selected.

In the field **"Display Levels"** the checkboxes 1, 2, ..., 6 are based on the respective levels of the DMS name.

Checkbox 2: relates to Motor1
Checkbox 4: relates to TRD.

By unmarking the respective checkbox, the respective DMS name level is removed from the display.
If all checkboxes except for checkboxes 1 and 2 are switched off, only the text "Group1:Motor1" is output.

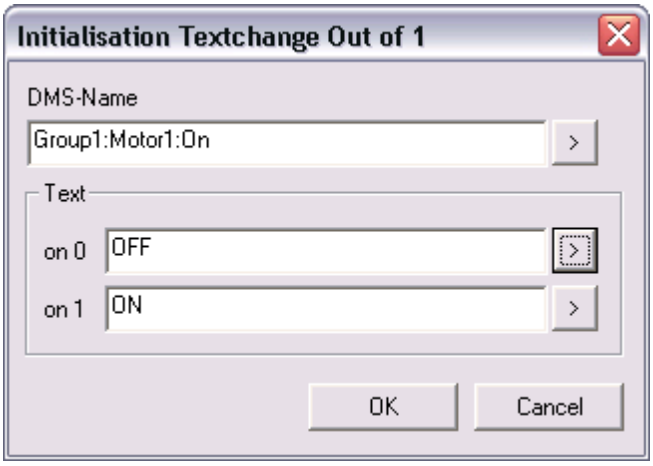
In the field with a grey background below the checkboxes, a preview of the text to be output is shown.

Text Change from 1
The command "Text Change from 1" is used to change text depending on a data point.

Example objective:
The texts Switched off and Switched on should alternate every second!

Example solution:

- Draw a text field as described under Text/Text Field
- Call up the Attributes window.
- Click on the button in the Initialisation column by the attribute "Text" and select the command "Text Change from 1"
- The dialog window Text Change from 1 Initialisation appears:



The text change should take place depending on a data point, which is selected in the top input field.

Entries in the Texts field:

At 0:
Text that is displayed if the selected data point has the value logic 0.

At 1:
Text that is displayed if the selected data point has the value logic 1.

Therefore the dialog window should appear as shown above.

Confirm and adopt entries with **<OK>**.

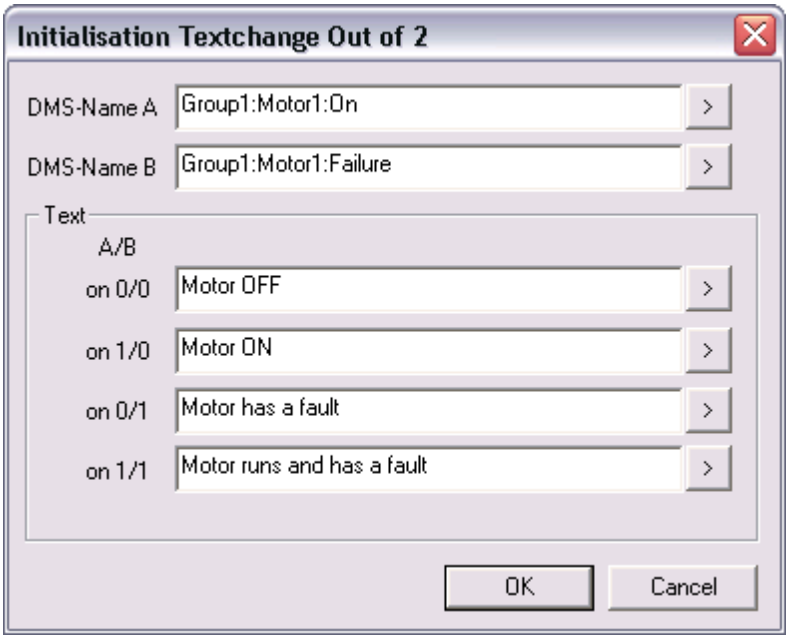
The texts entered may not exceed 80 characters in length! For Text no colon are accepted. GE and as well pWA interpret the character colon as a DMS variable.

The DMS can be used accordingly for monitoring of the data points, as already described with "Group1" under Initialisation attributes.

Text Change from 2
The command "Text Change from 2" offers the option of defining texts depending on 2 signal statuses.

Example objective:
Different texts should be output on the basis of two signals.

Example solution:
- Draw a text field as described under Text/Text Field
- Call up the Attributes window.
- Click on the button in the Initialisation column by the attribute "Text" and select the command "Text Change from 2"
- The dialog window "Text Change from 2 Initialisation" appears:



Input Field A
DMS name for data point A

Input Field B
DMS name for data point B

4 input fields can be seen in the Texts field. The respective text becomes visible:

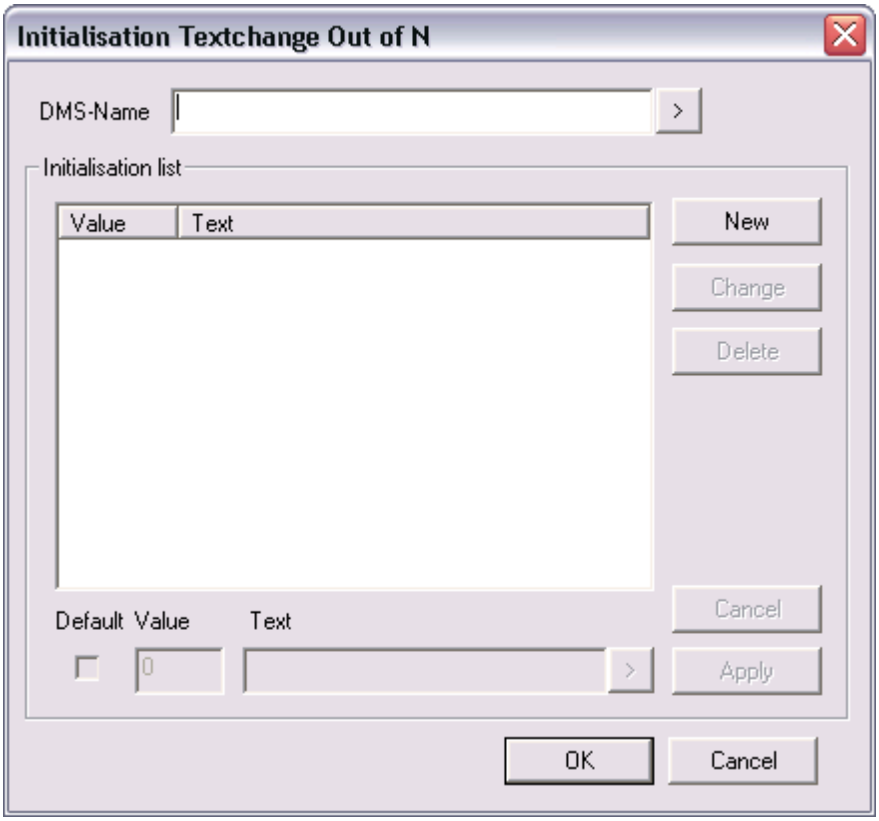
at 0/0	Data point A=0 and B=0	-> Motor switched off
at 1/0	Data point A=1 and B=0	-> Motor switched on
at 0/1	Data point A=0 and B=1	-> Motor has a fault
at 1/1	Data point A=1 and B=1	-> Motor is running and has a fault

Therefore the dialog window should appear as shown in the preceding image.
The settings are confirmed and adopted by clicking on the **<OK>** button.

The texts entered may not exceed 80 characters in length! For Text no colon are accepted. GE and as well pWA interpret the character colon as a DMS variable.
The DMS can be used for monitoring of the data points, according to the same logic as already described under "Initialisation attributes" with "Group1".

Text Change from N
With this option, any number of text changes depending on a data point value (content 0..15) can be displayed.

The Initialisation dialog looks like this:



In the top input field, enter the DMS names that the text change should apply to or select them from a list. The list appears after double-clicking in the entry field or by clicking on the button to the right of it.

Then the value for which the text change should take place, as well as the text to be output, should be entered. To do so, click on the **New** button. The fields Value and Text can now be edited. A string or a DMS name can be entered for the text. The entry of a DMS name can take place directly through the keyboard or by selecting from a list. The list appears after double-clicking in the entry field or by clicking on the button to the right of it. If a DMS name is entered, the data point value is displayed when the list value for the text is reached. The data point, therefore, must be of type STR (string).

By clicking on the **Adopt** button, the new values are adopted into the Initialisation list. The values in the list can now be changed or deleted. The actions "Change" and "Delete" always refer to the active line (marked blue). In order to select a line, click with the left mouse button on the corresponding line in the Initialisation list or scroll through the list with the arrow keys if a line is already selected.

New button
Allows you to input a value with the corresponding text

Change button
An already entered value and its text can be changed

Delete button
Deletes the active line

Cancel button

A newly created entry is not adopted into the list.

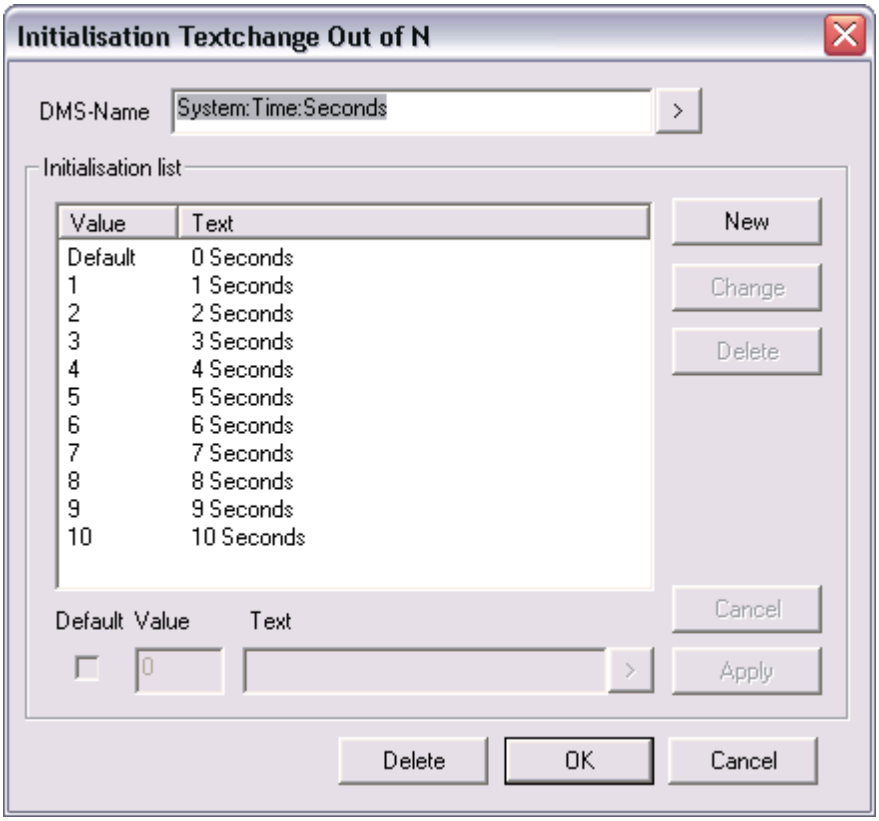
Adopt button
A newly created entry is adopted into the list.

Default checkbox
When the checkbox is clicked on, a default value can be defined. This is shown as the default text when the data point value does not match any of the values from the Initialisation column (see example).

The texts entered may not exceed 80 characters in length! For Text no colon are accepted. GE and as well pWA interpret the character colon as a DMS variable.

Example objective:
The system time seconds should be displayed in plain text from 0–10. The text "0 seconds" should be output for values from 11–60.

Example solution:
- Draw a text field as described under Text/Text Field
- Call up the Attributes window.
- Click on the button in the Initialisation column by the attribute "Text" and select the command "Text Change from N"
- The dialog window Text Change from N Initialisation appears.
- Enter the values as shown in the image:



The corresponding text becomes visible when the data point includes the respective value in the range 0 to 10. If the value is greater than 10, the text "0 seconds" (default value) is displayed. If you delete the default value, "10 seconds" is displayed for values greater than 10.

Are the correct texts displayed during the seconds 0 to 10? Is the text "0 seconds" displayed for the range from 10 to 60?

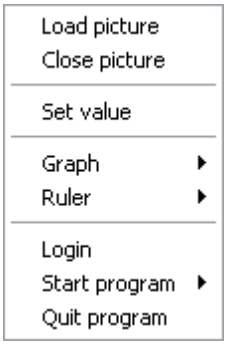
10.3.15.4 Action

The graphical objects can be divided into two groups on the basis of the actions that are executed if you click on the respective graphic object:

1. **Buttons, polygons and bitmaps**
   A number of different actions can be executed for these graphical objects, e.g. an image can be loaded or closed, or the value of a data point can be set.

2. **Checkboxes, radio buttons, input fields and combo boxes**
   For these objects, on the other hand, only the value of a data point is set.

The actions of **buttons, polygons and bitmaps** are defined by clicking on the button in the Initialisation column in the action column. A context menu appears in which the desired initialisation can be selected.



Load Diagram
Opens a process diagram, chapter Action: Load Diagram

Close Diagram
Closes a process diagram, chapter Action: Close Diagram

Set Value
Enables the setting of digital signals and entry of values, chapter Action: Set Value (sub-chapter Buttons, polygons and bitmaps)

Graph
Control elements for trend graphs, chapter Action: Graph

System Login
Provides a user the option of logging in with the press of a button, chapter Action: System Login

Program Start
Starts any desired program, chapter Action: Start Program

Exit Program
Exits the Graphical Editor, chapter Action: Exit Program

If a specific initialisation is selected, the dialog window of the selected type of initialisation is displayed.

The initialisation takes place in a similar manner for checkboxes, radio buttons, input fields and combo boxes. Here, you must also click on the button in the Initialisation column in the Action column. However, a selection menu is not opened, rather the dialog window **Initialisation Value Entry** is opened directly and contains different registers and input fields, depending on the selected object (see the chapter Action: Set Value, sub-chapter Checkboxes, Input fields, etc.).

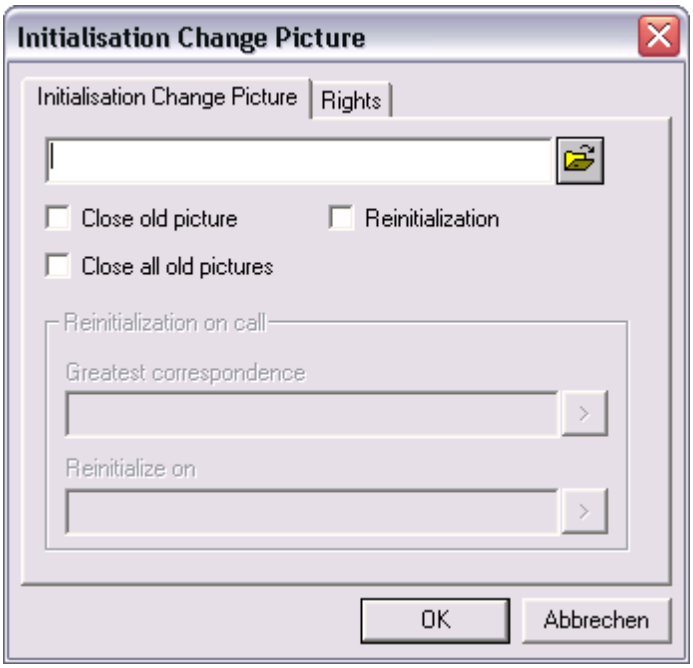
In order to test the initialised actions according to their definition, you must switch from edit mode to runtime mode. This can be accomplished by pressing the ***<e>*** *key or selecting the command "Runtime/Edit Switchover" in the "File" menu.*
By pressing the <CTRL> key and simultaneously clicking the mouse button on the graphical object, an initialised action can be executed in edit mode. The input fields and combo boxes are only operable in runtime mode.

10.3.15.5 Action: Load Diagram

By clicking on the graphical object with the right mouse button, an existing process diagram is loaded.

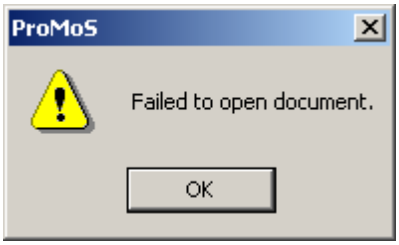
The idea behind this is as follows:
If a screen page provides too little space to display a process diagram, additional process diagrams should be displayed as easily as possible.
With user-defined **Control buttons** you can, for example, switch from one page to another or call up specific pages through assignment of multiple buttons. See also the Visi.Plus demo project.
Whether screen-filling images or only smaller windows are opened in the foreground depends on their definition.

After the action **"Load Diagram"** has been selected, the dialog window "Diagram Change Initialisation" appears.



White input field
Here the filename can be entered for the process diagram along with its extension, .scr (binary format) or .psc (ASCII format), or the diagram to be opened can be selected in the window which opens when the **folder icon** is clicked on.

*The diagram to be opened must already exist in the **scr** directory of the current project before the selection (e.g. **C:\Visi.Plus\ProjectName\src\Diag1**).*
If a diagram name is entered that is not present in the project directory, the following error message appears when you click on the initialised graphical object:



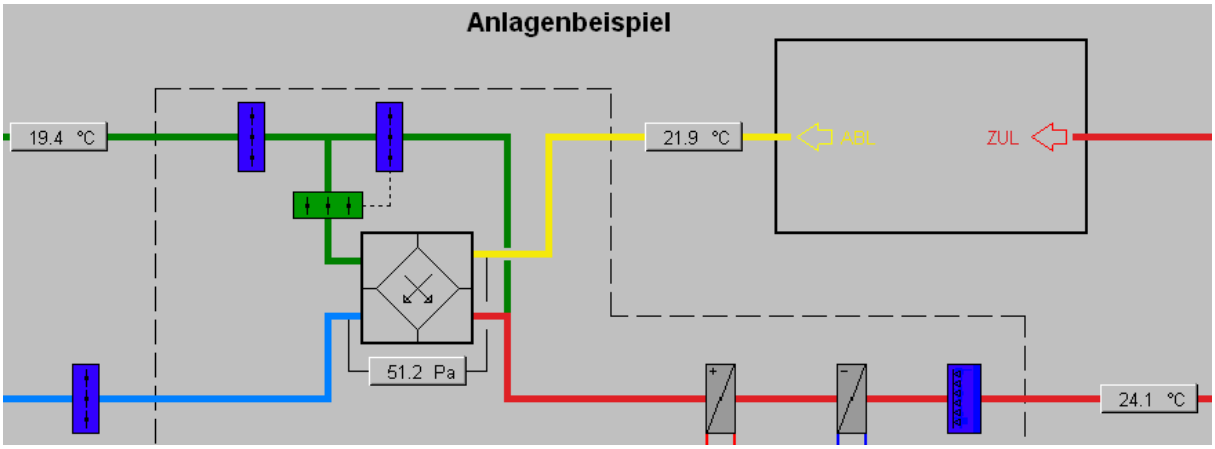
*An open, and thus displayed, process diagram (window) remains in the computer's memory until the button initialised with the action "**Close**" in the same window is clicked on, or until another diagram is loaded by selecting the option "**Delete Old Diagram**" in the **Change Image** initialisation dialog.*

Delete Old Diagram
If this checkbox is marked, the process diagram (diagram with the initialised graphical object) is closed when the diagram is changed.
This function is only logical if the newly called-up diagram is a complete diagram. If only a detail diagram is opened, the diagram with the initialised graphical object is retained.

Reinitialise
If this checkbox is marked, the process diagram is automatically reinitialised during loading and the "Reinitialise on call-up" field is activated.

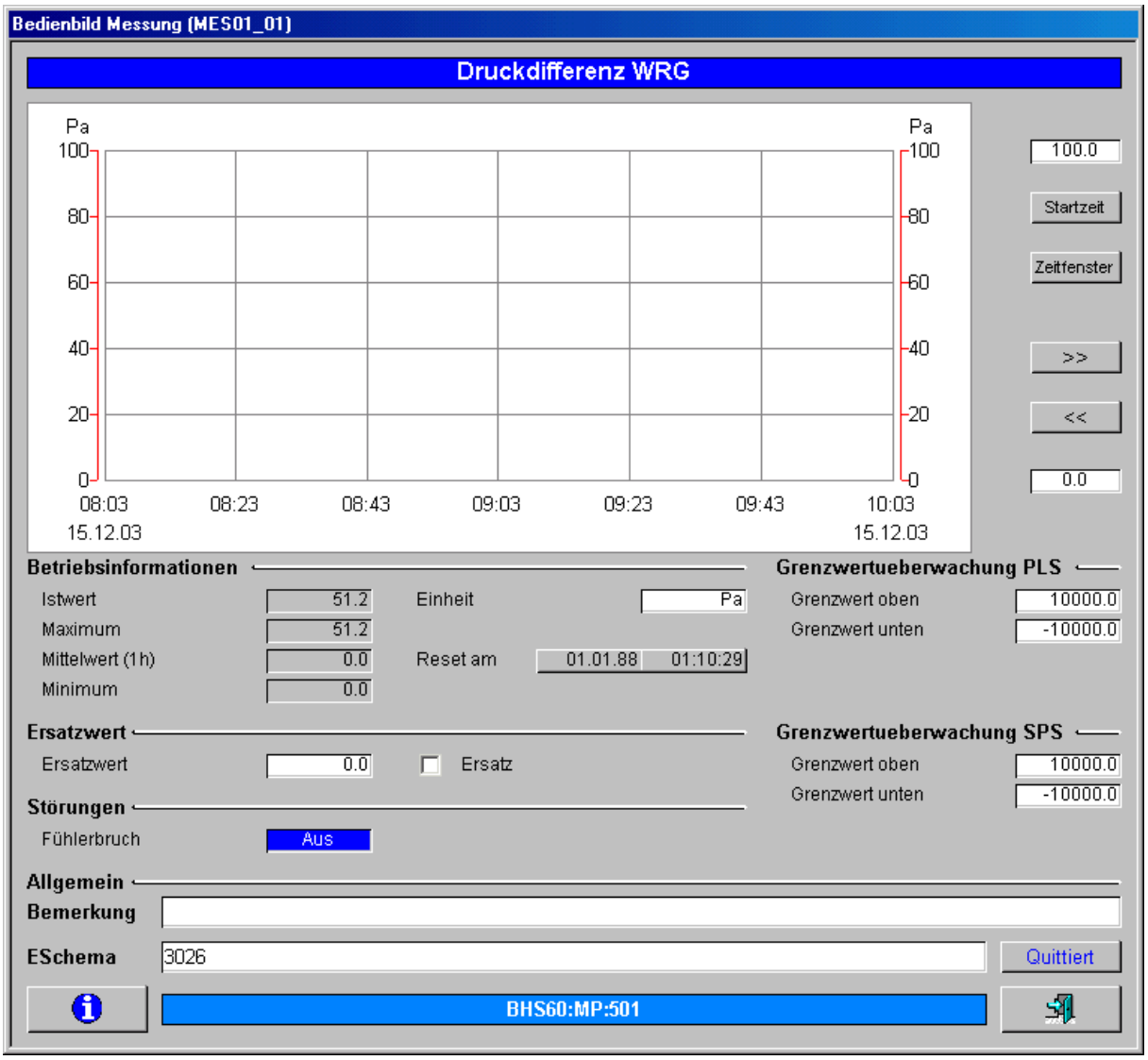
In order to explain this function, it must be examined in greater detail (see the following pages).

Why reinitialise?
An example project has multiple measurement points.
On the demo example page "System example" of the **Visi.Plus** demo project, four buttons can be seen:



A page in which past measurement values are displayed for a period of time (trend) should be displayed for each button (measurement point) after they are clicked on with the mouse. In addition, max/min values and possible sensor breaks should also be displayed. It should also be possible to enter a default value, the unit and diagram number, as well as a comment on

the measurement point. The following image shows an example:



Then one such process diagram should be generated for each measurement point. It would be better if only one would be generated and it could be informed which measurement point (data points) are involved.

Visi.Plus supports this function, which is called **Reinitialisation**.

The corresponding values for the called-up process diagram are included with the reinitialisation.

Advantage:
If changes to an diagram are made, they also affect all similar measurement points.

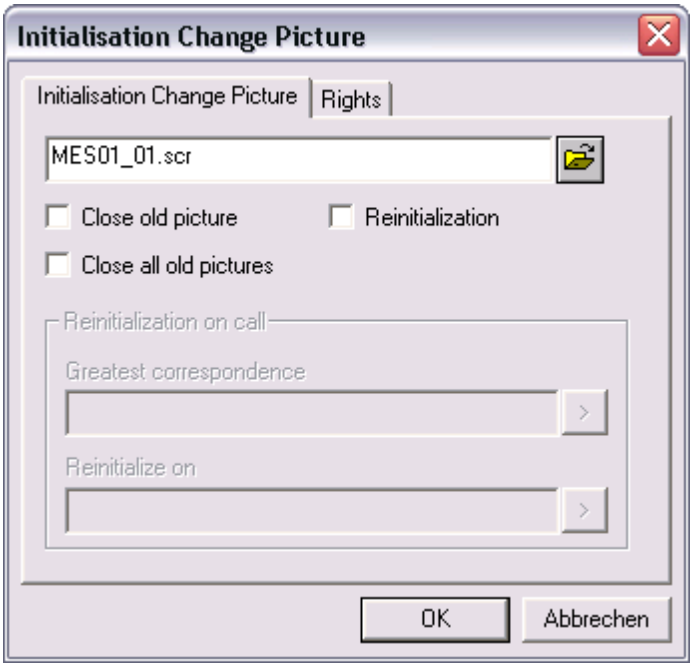
How is reinitialisation executed?

First, a new button object  is created on the demo example page "**System example**" of the **Visi.Plus** demo project. Then click on the newly drawn object again and open the

Attributes window  .

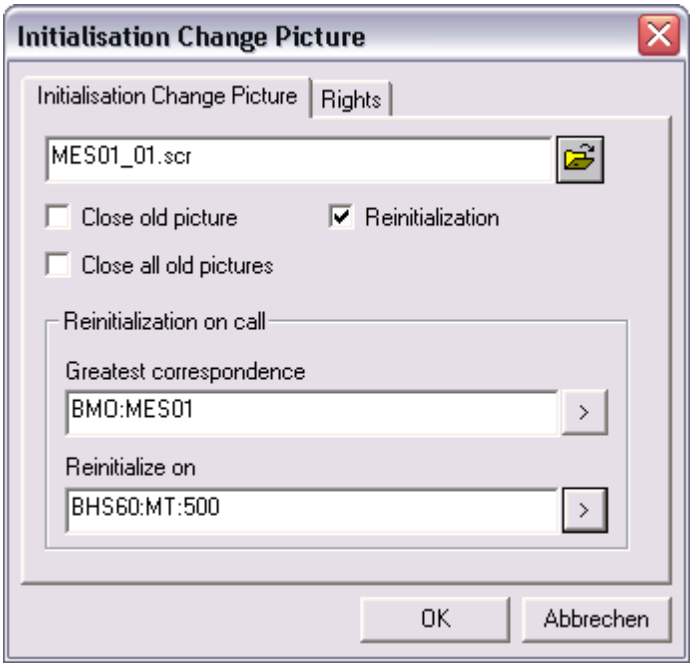
Then the diagram **MES01_01** with the data of **BHS60:MT:500** should be loaded. For this purpose, click on the right button in the Attributes window under **Action** and then select **Load Diagram**.

The window **Image Change Initialisation** is displayed.
Now the file **MES01_01** can be specified or selected via the button with the folder icon.



In order for reinitialisation to be specified, the checkbox **Reinitialise** must be marked. Then the DMS name can be entered in the field **Reinitialisation on call-up** or it can be selected from a list. The list appears after double-clicking in the entry field or by clicking on the **" >"** button to the right of it.

Smallest Common Name



The DMS name of the VLO (template object), here **MES01** beginning with **BMO**, is reinitialised to the DMS name of the data point that contains the data, i.e. **BHS60:MT:500**.

With this option, any button object can be informed that all data of the respective data point (DMS name) should be adopted when the diagram is loaded.

Confirm and accept the entry with the **<OK>** button.

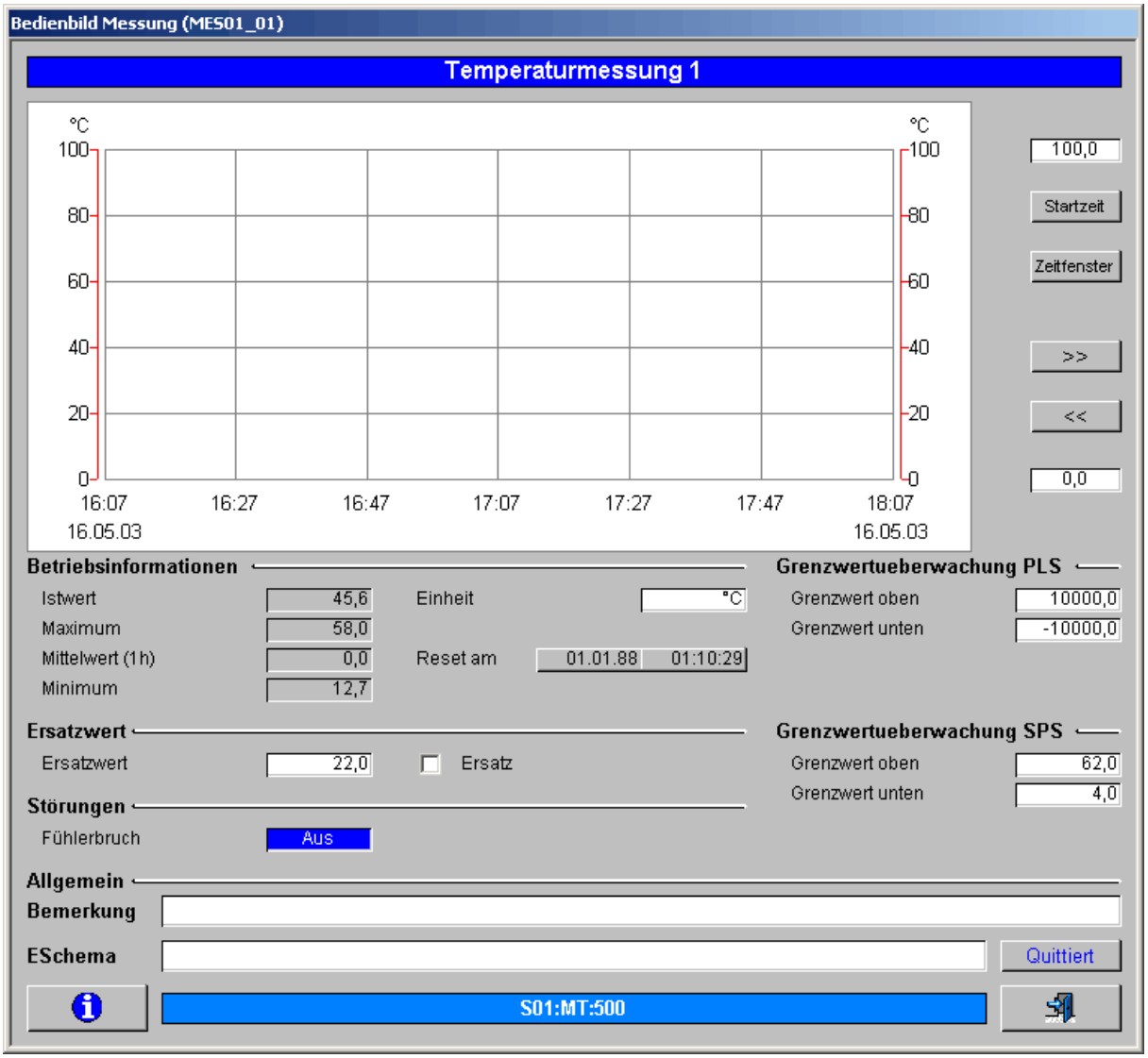
There are two ways to test the button just created in edit mode:

- *Switching to runtime mode with the* **<e>** *key and then clicking on the button with the left mouse button*

or

- *simultaneously pressing the* **<CTRL>** *key and clicking the left mouse button on the button that was just initialised*

In both cases the detail control diagram of the object MES01 is loaded and closed again with  .



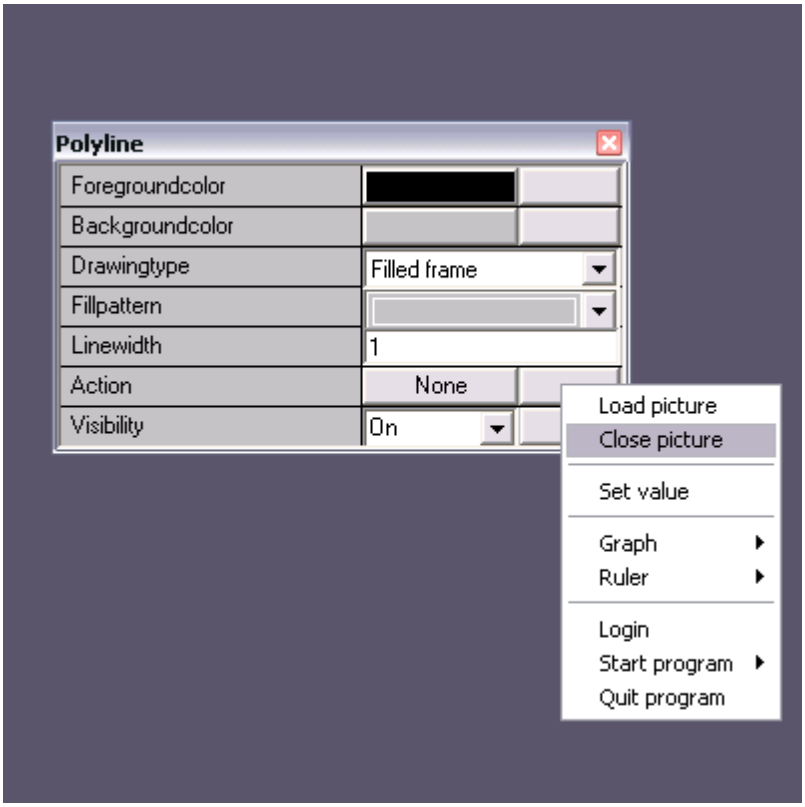
As you can see in the preceding image, all initialisations in the control diagram of the template object "**MES01**" are directly **initialised** to the template object and therefore display all values of the system object.

This procedure was applied to all four buttons in the demo example. These buttons were configured with just a few clicks of the mouse.

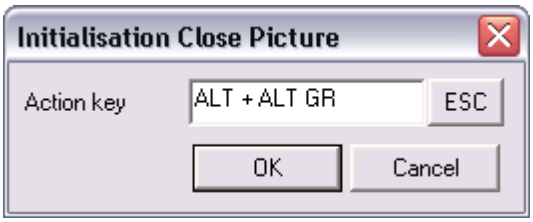
Now select the first button and copy it as many times as needed. Then you only have to reinitialise each individual button to the respective system object and you are already finished.

10.3.15.6 Action: Close Diagram

In order to close a process diagram with the assistance of a **button** graphical object, the **action** "**Close Diagram**" is to be used.



When this function is selected, the dialog window **Close Diagram Initialisation** appears.



Here it is also possible to close the process diagram with a key combination.

To define the key combination, click in the input field "**Action Key**" with the mouse. Then press desired key on the keyboard or key combination and confirm with **<OK>**.

(Almost) any key combination can be selected. Single key combinations beginning with the key **<ALT>** cannot be used. The **<ALT>** key is reserved for calling up Visi.Plus menu items.

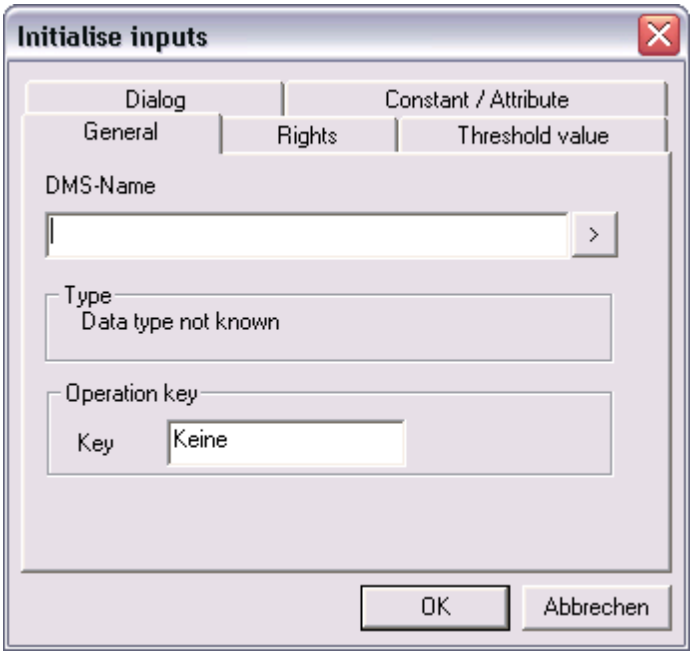
*STRG (Steuerung) stands for **<CTRL>** key (Control), depending on the keyboard labelling and country of the computer.*
*Operating system keyboard shortcuts, such as **<CTRL+C>** or **<STRG+C>** for the function "Copy marked items to the clipboard" and **<CTRL+V>** ("Insert the contents of the clipboard at the cursor position"), should be not be used for the assignment of key combinations in Visi.Plus out of consideration for the user.*

10.3.15.7 Action: Set Value

10.3.15.7.1 Buttons, polylines, bitmaps

In order to enable a system or machine operator to enter digital and analogue values when clicking on a polygon, button or bitmap, the "**Set value**" action is available.

If the function is selected, the **Value Entry Initialisation** dialog window appears.

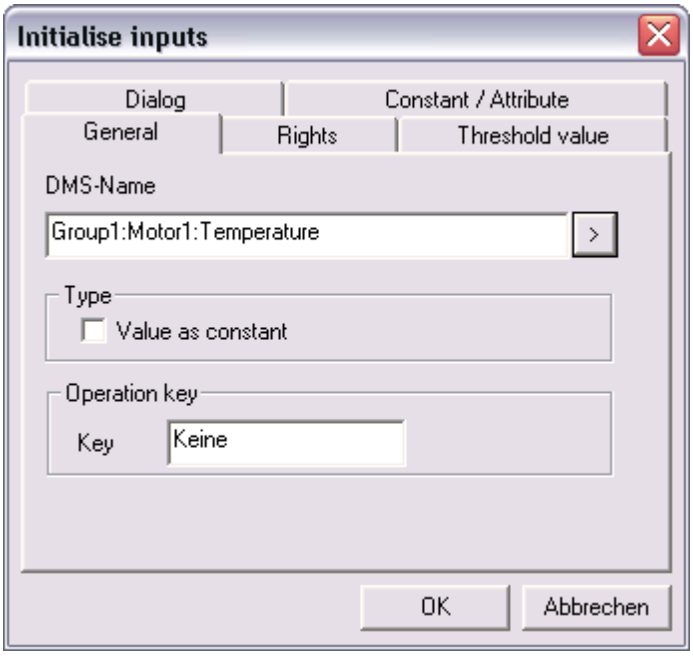


The registers shown, "**General**", "**Access Rights**", etc. are dependent on one another. Therefore the values in the tab "**Constants/Attributes**" are only effective if the set type is appropriately selected in the "**General**" tab. We therefore recommend checking the settings of the individual registers.

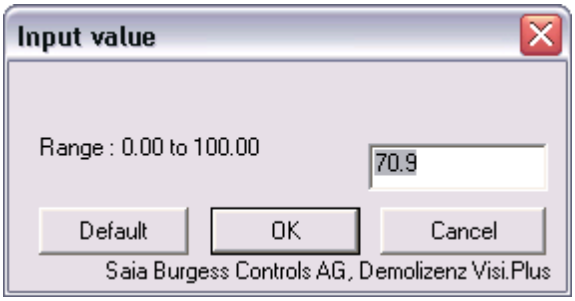
General tab
The DMS name of the value to be set can be entered in the first input field or selected from a list.
The list appears after double-clicking in the entry field or by clicking on the button located to the right of it.

Set Type
Depending on the type of DMS value, either analogue or digital, it is possible to determine how it should be set.

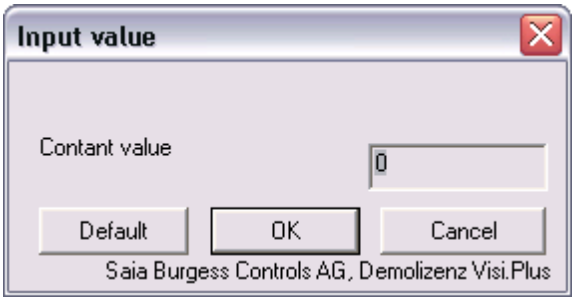
Set Type: Analogue DMS value



If the "**Set type**" checkbox is **not marked**, the following window with input field opens:



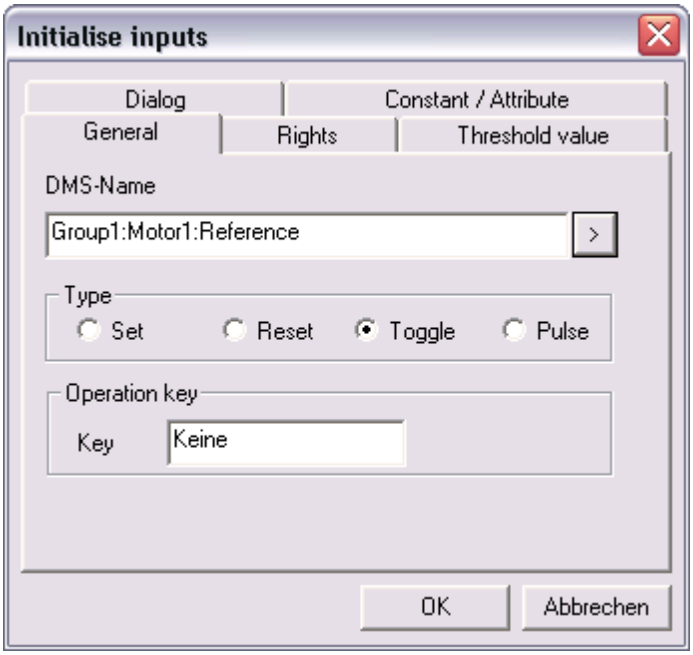
If the "**Set type**" checkbox is **not marked**, the following window for indicating the constant value opens:



By pressing the **<OK>** button, the value set in the "**Constants/Attributes**" register is written directly to the data point. By pressing **<Cancel>**, this is prevented and the dialog window is closed.

*The display of the window contents is influenced by the "**Dialog**" tab. If the window with the constant value should be disabled and the value of the constants be written directly to the selected data point, the checkbox "**No Dialog for Constants**" should be marked in the "**Dialog**" tab.*

Set Type: Digital DMS value



Four switching variants for changing the digital data point are provided under "**Set type**".

They are:
Set , switch on from logic 0 to logic 1
Reset, switch off from logic 1 to logic 0
Toggle , switch from 0 to 1 or 1 to 0
Impulse logic 1 as long as the mouse button is depressed

The following also applies here:
*If the checkbox "**No Dialog for Constants**" is not marked in the "**Dialog**" tab, a confirmation prompt is displayed. With the checkbox marked, switching is executed immediately without the query.*

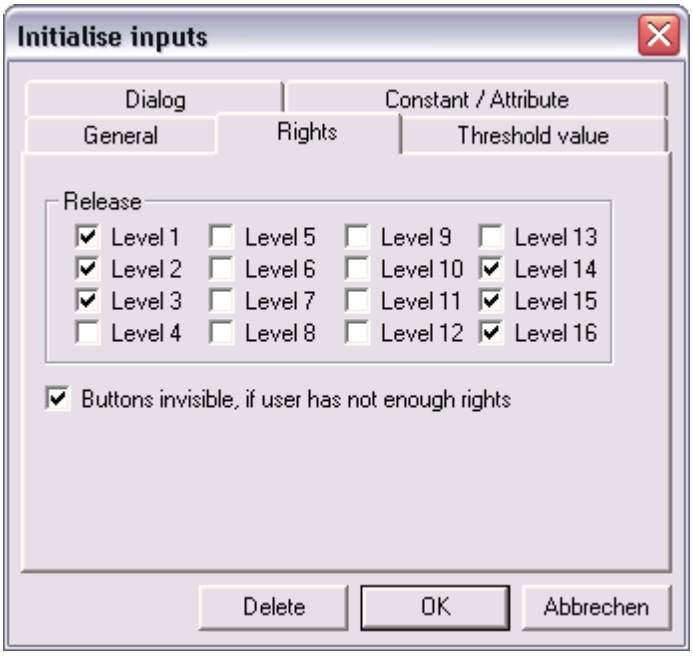
Control key

Offers the option of setting the value via a PC key combination.
To do so, click on the input field in the "**Control Key**" section and press the desired key or key combination on the keyboard.
While in runtime mode, the initialised value is transmitted by pressing the control key/s that were just defined.

"**None" means** no keys are assigned.

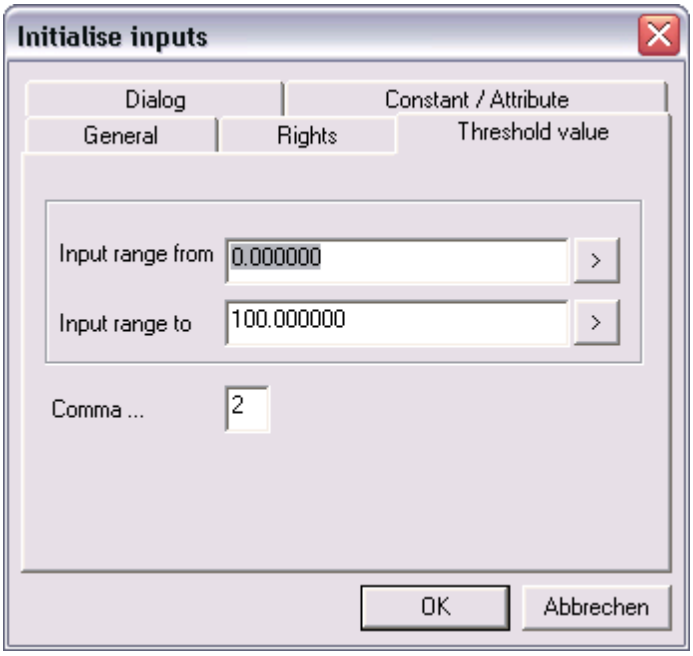
Access Rights tab
Which level has the required access rights?

Release
Enables the access rights defined in the User Management module (see the chapter **User management**) to be assigned. If a user does not have the necessary user rights, the initialised action cannot be executed. If no user rights are assigned, all users can execute the action.



Control button not visible without access rights
The polygon, button or bitmap graphic objects can be made invisible, and therefore inoperable, for users without assigned rights.

Threshold Values tab
Enables the definition of an input range.



Minimum entry
For the smallest possible value to be entered

Maximum entry
For the largest possible value to be entered

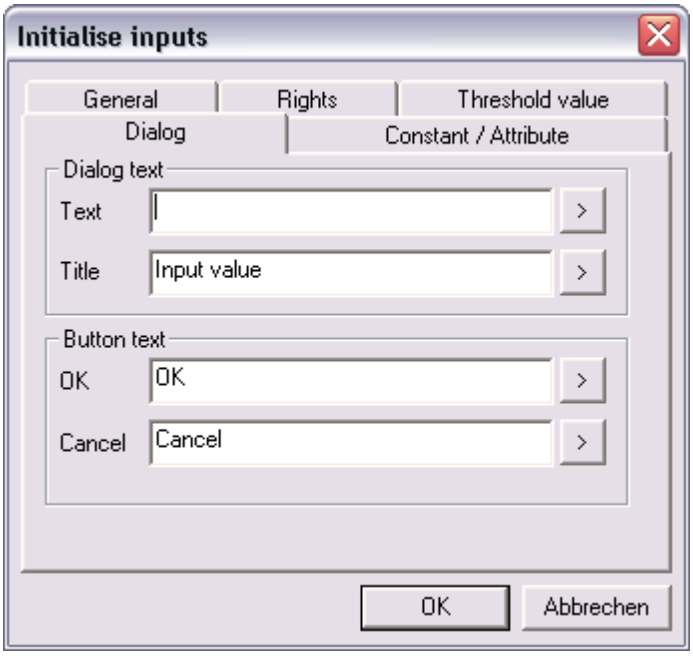
Then it is no longer possible for a user to enter a value outside of the defined input limits.

The two **threshold values** can also be **initialised**. By double-clicking on the respective input field with the mouse or by clicking on the button to the right of it, the desired DMS name can be selected as a limit value.

Decimal Places
Defines the number of decimal places for the two input fields.

Depending on the country in which you are located, the display of numerical values can vary somewhat. In Switzerland numerical values are separated by a dot, whereas they are separated by a comma in Sweden. If you do not enter the value here according to the Windows language and country settings on the computer, it will lead to very particular problems.

Dialog tab
Defines the appearance of the Value Input window in runtime mode

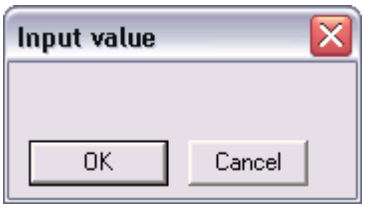


Text Dialog Text
Defines the display text in the window.

Title Dialog Text
Defines the dialog window title.

Button Text
Defines the texts in the two buttons of the dialog window, **<OK>** and **<Cancel>**.
Texts deviating from the default texts can be defined, e.g. **<Yes>** and **<No>**. This has the advantage that buttons can also be defined in a different language. The button text for **OK** refers to the left button and the button text for **Cancel** refers to the right button.

If, when in runtime mode, you then click on the polygon, button or bitmap that was just defined, the following dialog window appears.



Using Dialog tab initialisations

All four input fields in the **Dialog tab** can be initialised to a DMS name. This option makes it possible to produce extremely flexible dialog windows.

For example, the designation of a motor can be displayed as text in the dialog window.

The initialisation of a field is carried out by double-clicking on the respective input field or by clicking on the button to the right of it. Then a selection dialog window appears where a DMS name can be selected.

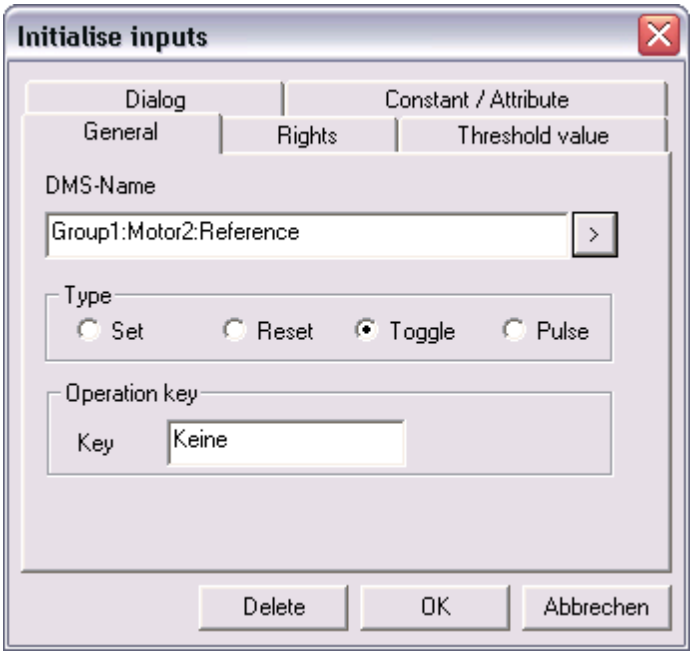
The content of the selected DMS name is used, depending on its assignment, as text in the

dialog window or as a label for the two buttons.

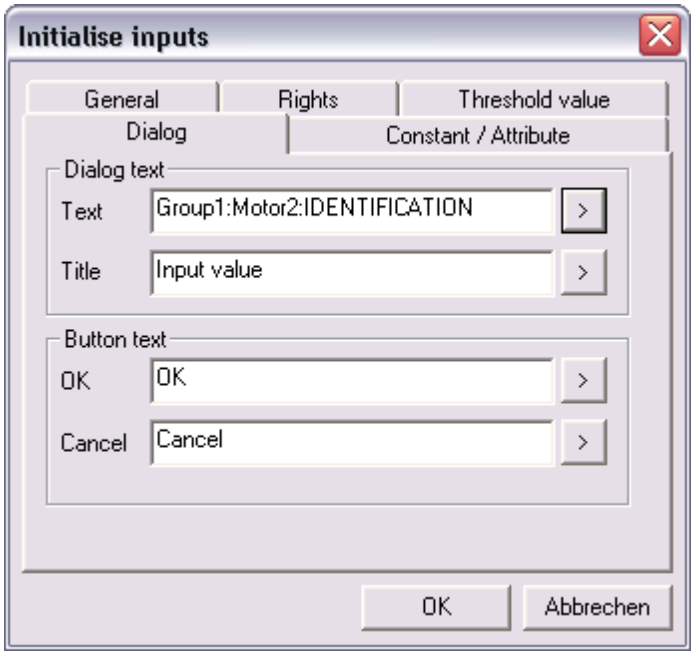
The following example provides further information in this regard:

Example: Initialisation of a dialog text

- Draw a "**Button**" graphical object
- Click on the drawn button **again**
- Open the context menu with the right mouse button and choose **Attributes**
- Choose the "**Set Value**" action
- Double-click in the first input field in the "**General**" tab
- Select the DMS name "**Group1:Motor2:Target**" .
- Set the "**Set Type**" to **toggle**. As a result, the indicated data point status is switched in runtime mode by clicking on the button.

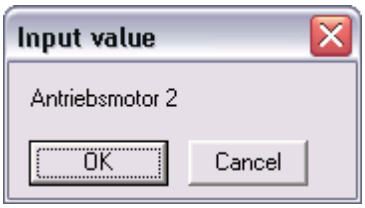


- Initialise the text to the DMS name "**Group1:Motor2:NAME**" in the **Dialog tab in the section Dialog Text at Text**:



- Save Process Diagram
- Switch to runtime mode
- Click on the button

The results should appear as follows:



The text **Drive motor 2** was adopted directly from the DMS.

More experiments with the newly created button

The data point/s can be observed and changed directly from the DMS.

The DMS can be conveniently opened by double-clicking on the red dot in the taskbar (bottom right).

By clicking on the button on the far left in the toolbar, the system object view is displayed.
In order to receive a list of the data points for the system object "Motor2", click on the small "+" signs in succession.

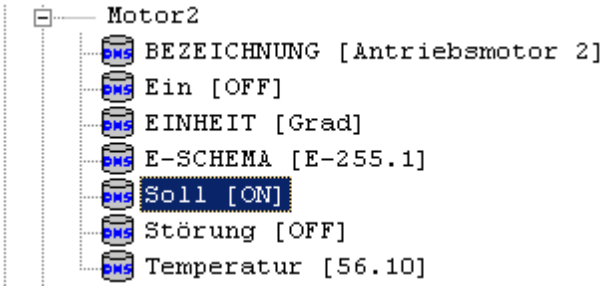
"Group1" -> "Motor2" (it should appear as per the following image).
The content of the DMS data point "NAME" is actually "**Drive motor 2**" and the data point "Target" indicates its status.

Returning to the Graphical Editor: Press the key combination <ALT+TAB> until the GE is selected and then release the keys.
This function for switching from application to application is a Windows operating system function!

The dialog window in the preceding image remains displayed.

If you now click on the **<OK>** button in the dialog window, the status of the digital data point "**Group1:Motor2:Target**" (blue background, see next image) is switched from the status [OFF] to the status [ON], or vice-versa.

In order to check this, switch to the DMS again (with the <ALT+TAB> keys as described above).



Personal self test:
Further expansion of the "**Button**" graphical object.

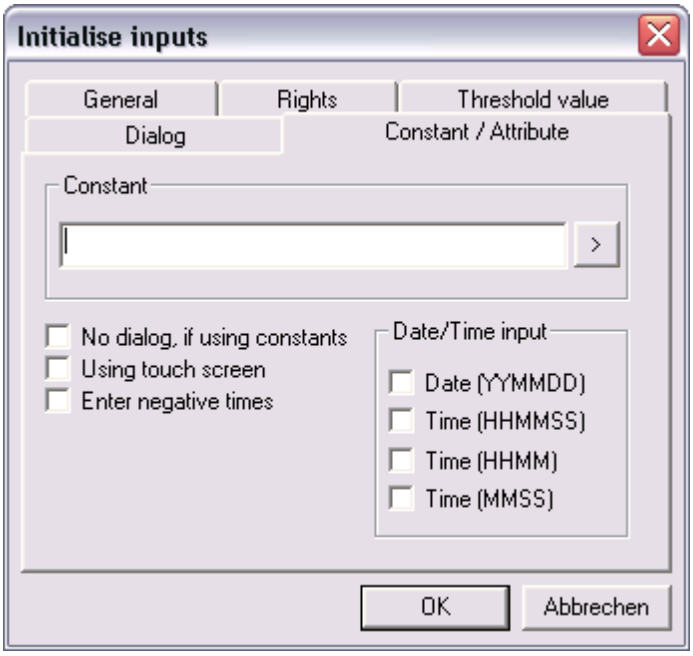
Initialise the button label to the DMS name "**Group1:Motor2:NAME**" and then initialise the background of the button to the DMS data point "**Group1:Motor1:Target**" with colour change. In doing so, the switching status of the motor is visible for the user as a colour change of the button.

Do not forget!
*Set the attribute "**Drawing Type**" to "**Button (filled)**" so that the colour change becomes visible! More information about this is found under Drawing Type.*

Constants/Attributes tab
Provides the option of transmitting a constant value to a DMS data point.

*The "**Constants**" tab is only active if an analogue signal is initialised in the "**General**" tab.*

Constants
The input field assumes the desired value. A number or a DMS name can be entered for the constants. The entry of a DMS name can take place directly through the keyboard or by selecting from a list. The list appears after double-clicking in the entry field or by clicking on the button to the right of it. The content of the data point is taken for the constants on the initialisation to the data point.

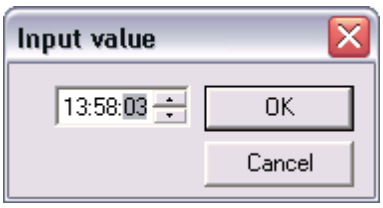


No Dialog for Constants
If this checkbox is marked, display of the dialog window is disabled in runtime mode and the constant value is set directly. All settings in the "**Dialog**" tab become invalid.

Input via Touchscreen
Enables the entry of values through a touchscreen. The user, therefore, does not have to rely on a PC keyboard.

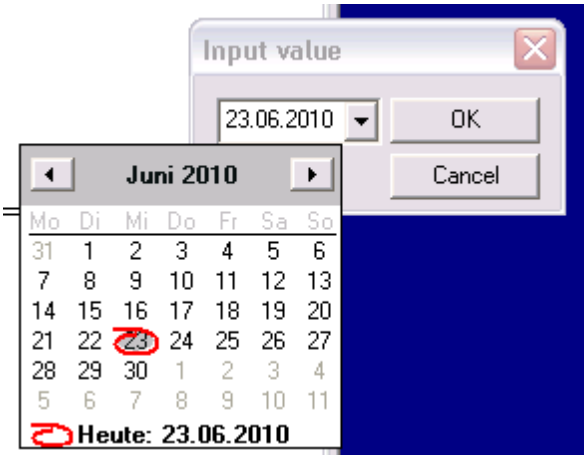
Date/Time
This section contains two checkboxes for entering the date or time. If both checkboxes are marked, the date entry has priority.

Time (HHMMSS)
The checkbox activates the dialog box for entering the time.



The entered time value is, as shown in the example, applied to the DMS data point as 135803 (Saia PCD® time format).

Date (YYMMDD)
The checkbox activates the dialog box for entering a date. If the arrow symbol pointing down at the right side of the input field is clicked on, a calendar opens up for convenient selection of the desired day.

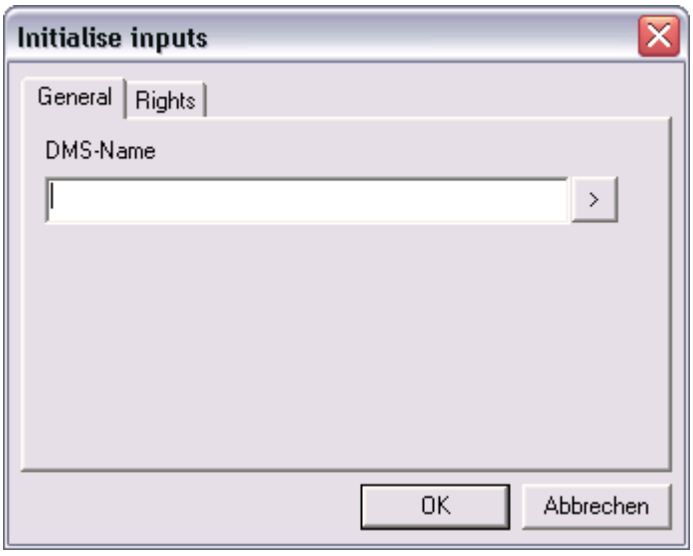


The entered date is, as shown in the example, applied to the DMS data point as 030516 (Saia PCD® date format).

10.3.15.7.2 Checkbox

With the help of checkboxes, a digital or analogue value can be set from 0 to 1 or the opposite (toggling). To do so, click on the checkbox (see the example).

In order to initialise a checkbox, click on the button in the Initialisation column in the Action column. The **Value Entry Initialisation** dialog window appears:



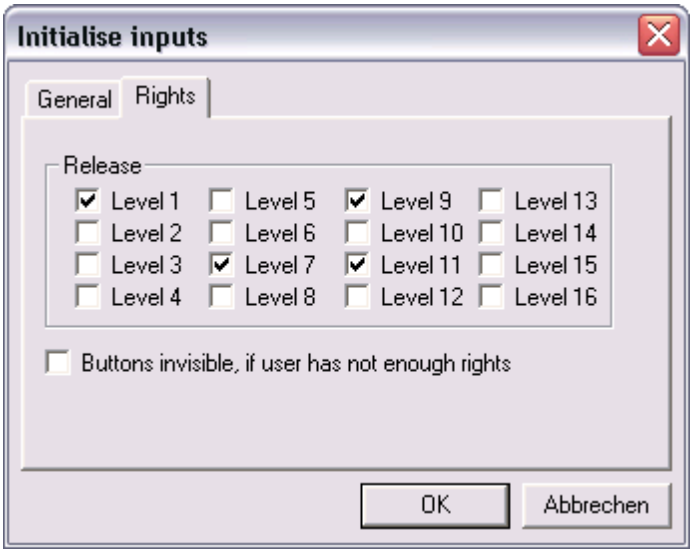
The window consists of the two registers "**General**" and "**Access rights**".

General tab
The DMS name of the value to be set can be entered in the input field or selected from a list. The list appears after double-clicking in the entry field or by clicking on the button to the right of it.

Access Rights tab

Which level has the required access rights?

Release
Enables the access rights defined in the User Management module (see the chapter **User management**) to be assigned. If a user does not have the necessary user rights, the initialised action cannot be executed. If no user rights are assigned, all users can execute the action.



Control button not visible without access rights
The checkbox can be made invisible, and therefore inoperable, for users without assigned rights.

For the following example, you must start the Visi.Plus project, which is included in the scope of delivery.

Example objective
The digital data point BHS60:AV:506:On should be toggled. The current value of the data point should be displayed as text in the checkbox.

Example solution
- Draw a checkbox
- Call up the Attributes window.
- Click on the button in the Initialisation column by the attribute "Text" and select the command "Value Display".
- Enter the DMS name "BHS60:MM:506:On" in the Initialisation dialog or select it from the list. Set the Output Format to %3.2f.
- Click on the button in the Initialisation column by the "Action" attribute.
- The Value Entry Initialisation dialog window appears
- In the General tab, enter the DMS name "BHS60:AV:506:On".

The created checkbox should then appear as follows:

Does the created checkbox exhibit the following behaviour?

The value of "BHS60:AV:506:On" equals 0. Therefore the checkmark is not set. The checkbox becomes operable on switching to runtime mode. By clicking on the checkbox, a checkmark is set. The value of "BHS60:AV:506:On" is now set to 1.

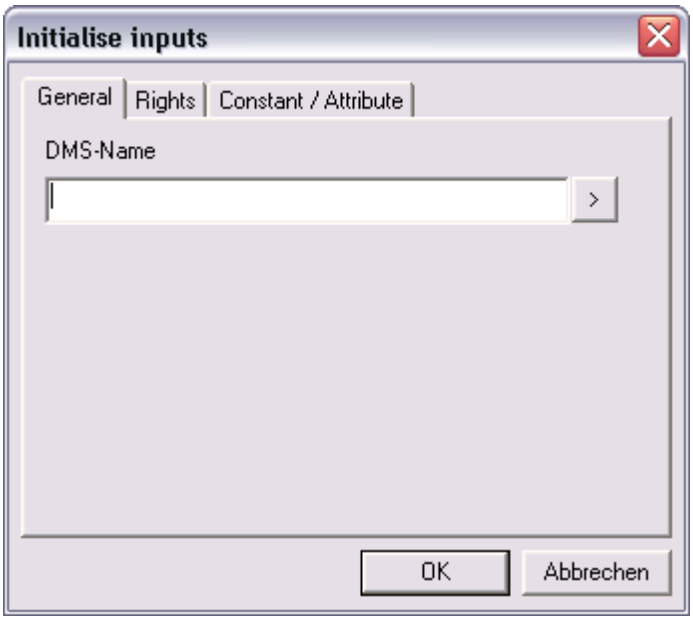
Clicking again removes the checkmark and the value of "BHS60:AV:506:On" is set to 0.

The data point (in the example "BHS60:AV:506:On") is logged in the DMS. This means that the checkbox is updated if the value changes.

10.3.15.7.3 Radio Button

With the help of radio buttons, both analogue and digital data points can be set. To do so, click on the corresponding radio button. The use of radio buttons is only logical if multiple radio buttons are created and they are initialised to the same data point (see examples).

In order to initialise a radio button, click on the button in the Initialisation column in the Action column. The **Value Entry Initialisation** dialog window appears:

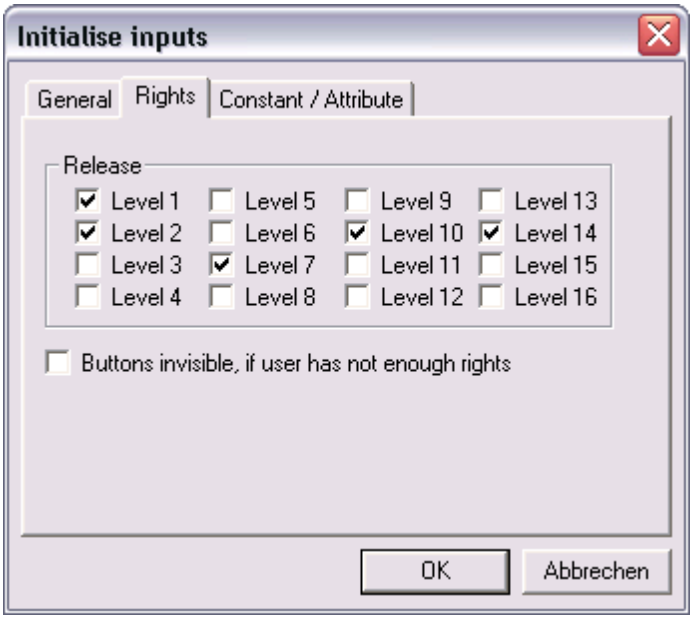


The window consists of the three registers "**General**", "**Access Rights**" and "**Constants/ Attributes**"

General tab
The DMS name of the value to be set can be entered in the input field. It can also be selected from a list. The list appears after double-clicking in the entry field or by clicking on the button to the right of it.

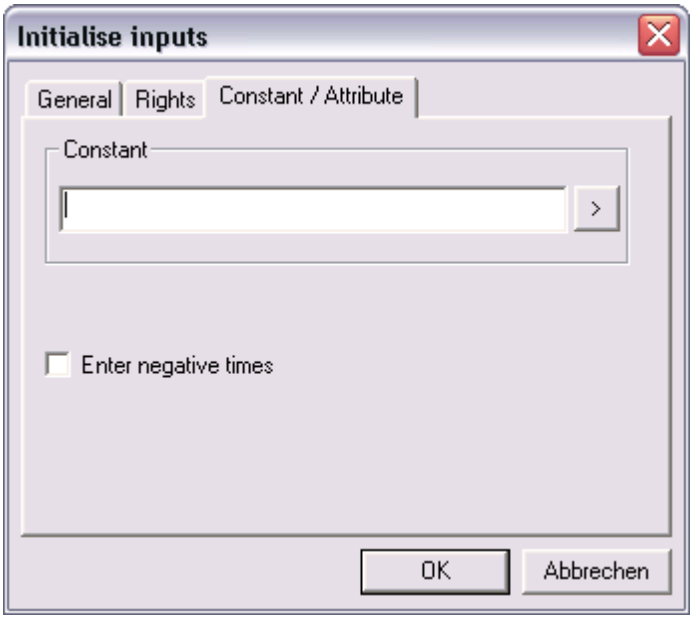
Access Rights tab
Which level has the required access rights?

Release
Enables the access rights defined in the User Management module (see the chapter **User management**) to be assigned. If a user does not have the necessary user rights, the initialised action cannot be executed. If no user rights are assigned, all users can execute the action.



Control button not visible without access rights
The radio button can be made invisible, and therefore inoperable, for users without assigned rights.

Constants tab



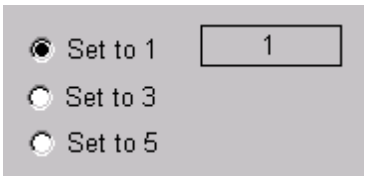
The input field registers the desired value (entry via keyboard). A number or a DMS name can be entered here. The entry of a DMS name can take place directly through the keyboard or by selecting from a list.The list appears after double-clicking in the entry field or by clicking on the button to the right of it. With the initialisation to a data point, the content of the data point is adopted by clicking on the radio button (see Example 2).

Example objective 1
For the following example, you must start the Visi.Plus project, which is included in the scope of delivery.
The data point "BHS60:MM:504:ActualValue" should be set to 1, 3 and 5. The value of the data point should be displayed.

Example solution 1
- Draw a radio button
- Call up the Attributes window.
- For the "Text" attribute, enter "Set to 1" in the middle column.
- Click on the button in the Initialisation column by the "Action" attribute.
- In the Initialisation dialog in the General tab, enter the DMS name "BHS60:AV:506:On". Enter the value 1 in the Constants/Attributes tab.
- Duplicate or copy the radio button twice and then position
- Call up the Value Display Initialisation dialog and enter the value 3 for the second radio button and the value 5 for the third radio button in the Constants/Attributes tab.
- Enter "Set to 3" for the second radio button in the text field of the Attributes window and "Set to 5" for the third radio button
- Draw a text field as described under Text/Text Field
- Call up the Attributes window
- Click on the button in the Initialisation column by the attribute "Text" and select the command "Value Display".
- Enter the DMS name "BHS60:MM:504:ActualValue" in the Initialisation dialog or select from the list. Set the Output Format to %d.

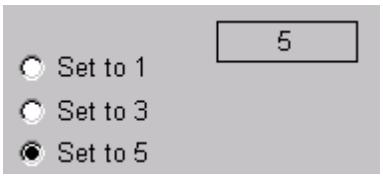
The created image should then appear as follows:

.

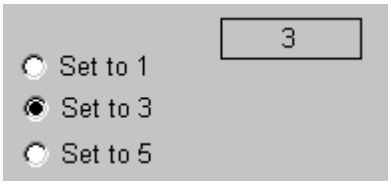


Do the radio buttons exhibit the following behaviour?

The current value of "BHS60:MM:504:ActualValue", which is displayed in the text field, equals 1 (you may have a different value here). Therefore the radio button "Set to 1" is active. The radio buttons become operable on switching to runtime mode. When clicking on the radio button "Set to 5", a dot is placed in this button and the original dot is removed from the radio button "Set to 1". Now the value of "BHS60:MM:504:ActualValue" is 5.



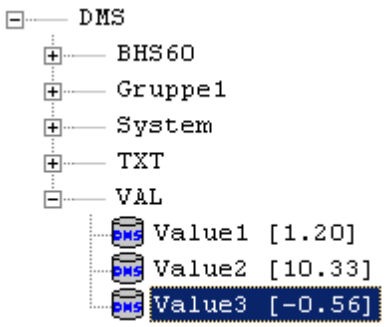
Clicking on "Set to 3" leads to the following image:



Example objective 2

First for this example, three new data points must be created in the DMS:

- VAL:Value1 (type FTL, value 1.2)
- VAL:Value2 (type FLT, value 10.33)
- VAL:Value3 (type FLT, value -0.56)

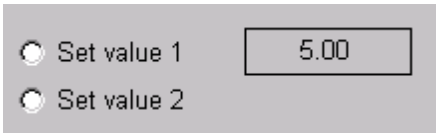


The data point "BHS60:MM:504:ActualValue" should be able to be set to the values of the data points "VAL:Value1" and "VAL:Value2". The value of the data point should be displayed.

Example solution 2

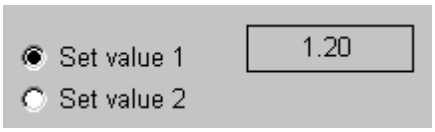
- Copy two of the three radio buttons from Example 1 as well as the text field.

- Call up the Attributes window
- Enter "Set value 1" (1st radio button) or "Set value 2" (2nd radio button) for the "Text" attribute in the middle column.
- Click on the button in the Initialisation column by the "Action" attribute.
- Enter the DMS name VAL:Value1 (1st radio button) or VAL:Value2 (2nd radio button) in the Constant Attributes tab in the Initialisation dialog.
- Set the Output Format of the text field to %3.2f.
- The created image should then appear as follows:

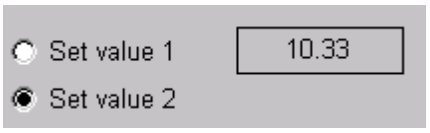


Do the radio buttons exhibit the following behaviour?

The current value of "BHS60:MM:504:ActualValue", which is displayed in the text field, equals 5 (you may have a different value here).Since this value does not match any of the initialised constants, no radio buttons are active.The radio buttons become operable on switching to runtime mode. By clicking on the radio button "Set value 1", the value of "BHS60:MM:504:ActualValue" is set to 1.20.



By clicking on the radio button "Set value 2", the value of "BHS60:MM:504:ActualValue" is set to 10.33.

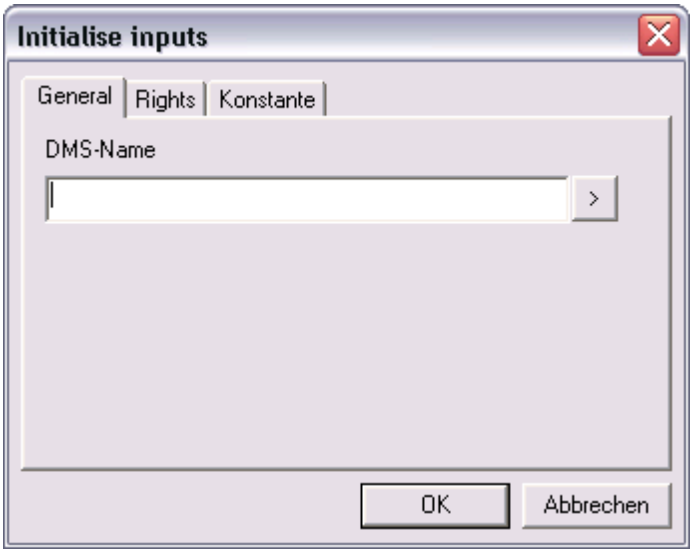


The data point (in the example "BHS60:MM:504:ActualValue") is logged in the DMS. This means that the radio buttons are updated when the value changes. If the value in the DMS does not match any of the initialised constants, no radio button is active.

10.3.15.7.4 Input Field

With the help of input fields, analogue or digital data points as well as data points of the type STR (strings) can be modified directly from the keyboard (see the example):

In order to initialise an input field, click on the button in the Initialisation column in the Action column. The **Value Entry Initialisation** dialog window appears:

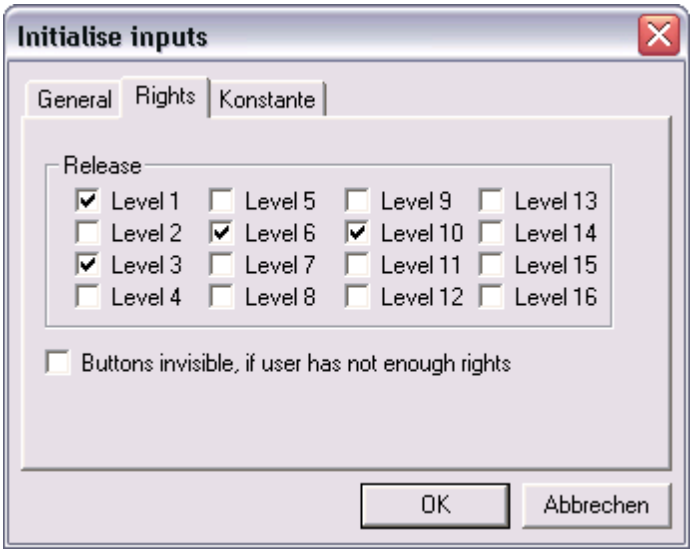


The window consists of the three registers "**General**", "**Access Rights**" and "**Constants**"

General tab
The DMS name of the value to be set can be entered in the input field. It can also be selected from a list. The list appears after double-clicking in the entry field or by clicking on the button to the right of it.

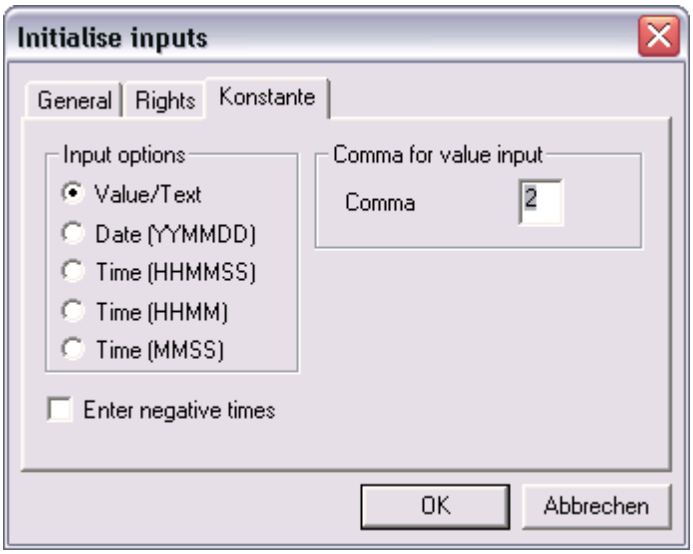
Access Rights tab
Which level has the required access rights?

Release
Enables the access rights defined in the User Management module (see the chapter **User management**) to be assigned. If a user does not have the necessary user rights, the initialised action cannot be executed. If no user rights are assigned, all users can execute the action.



Control button not visible without access rights
The input field can be made invisible, and therefore inoperable, for users without assigned rights.

Constants tab



With the help of input fields, values or texts as well as the date or time can be entered. Please select the desired option here. The number of decimal places that should be displayed can be entered for values.

The entered time value is, as shown in the example, applied to the DMS data point as 135803 (Saia PCD® time format).
The entered date is, as shown in the example, applied to the DMS data point as 16052003 (Saia PCD® date format).

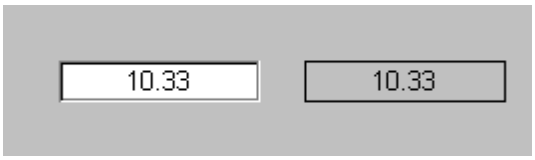
Example objective

For the following example, you must start the Promos project, which is included in the scope of delivery.
The data point "BHS60:MM:504:ActualValue" should be set to 200.5555. The value of the data point should be displayed.

Example solution

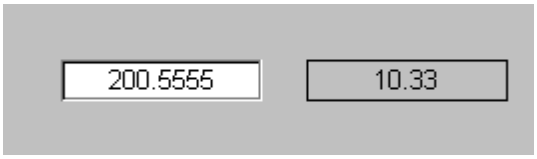
- Drawing an input field
- Click on the button in the Initialisation column by the "Action" attribute.
- In the Initialisation dialog in the General tab, enter the DMS name "BHS60:AV:506:On" or click on the option value/text from a list in the Constants tab and set the decimal places setting to 2
- Draw a text field as described under Text/Text Field
- Call up the Attributes window
- Click on the button in the Initialisation column by the attribute "Text" and select the command "Value Display".
- Enter the DMS name "BHS60:MM:504:ActualValue" in the Initialisation dialog or select it from the list. Set the Output Format to %3.2f.

The image should then appear as follows:

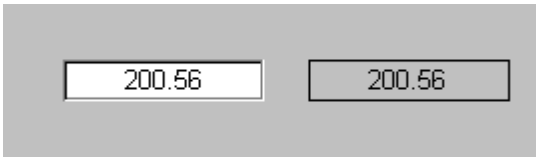


Does the created edit field exhibit the following behaviour?

The current value of "BHS60:MM:504:ActualValue" is 10.33 (you may have a different value here). This is displayed both in the edit field and in the text field. Now you must switch to runtime mode. Only the entry of a new value is possible here. Then click on the edit field with the mouse and enter the value 200.5555:



Then press the return key. Only now is the value of "BHS60:MM:504:ActualValue" changed. The value in the edit field is rounded to two decimal places.



You must switch to runtime mode to make the entry. The entry must be completed with the enter key.
The data point (in the example "BHS60:MM:504:ActualValue") is logged in the DMS. That means that the value in the edit field is updated when the value changes.

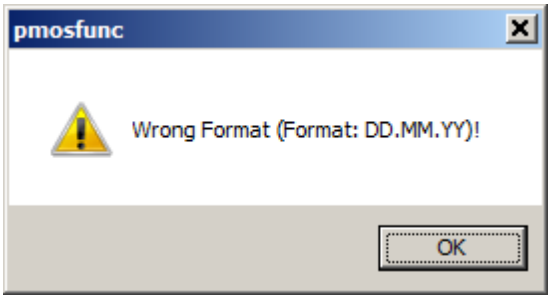
*The entry can be cancelled by clicking on the background or pressing the **<ESC>** key. The old value is restored in the process. If multiple edit fields are present in the process diagram, they can be selected with the tab key*

The entry of texts, data and times is done in the same way.

A date must be entered in the following format:**DD.MM.YYYY**
The usual limits for day, month and year may not be exceeded or fallen short of.

The entered date is applied to the DMS data point in Saia PCD® date format, meaning the entry of 10.03.2008 becomes 080310.

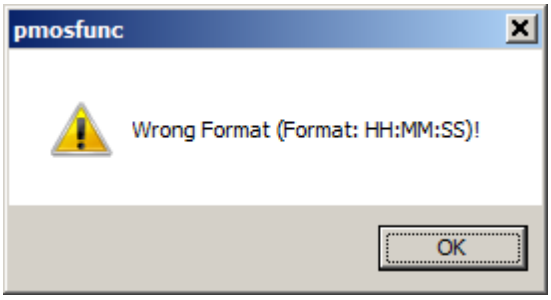
If the date is not valid, such as with the entry of 31.13.2003, a warning is issued!



A time must be entered in the following format:**HH:MM:SS**
The normal limits for hours, minutes and seconds may not be exceeded.

The entered time is applied to the DMS data point in Saia PCD® time format, meaning the entry of 12:53:48 becomes 125348.

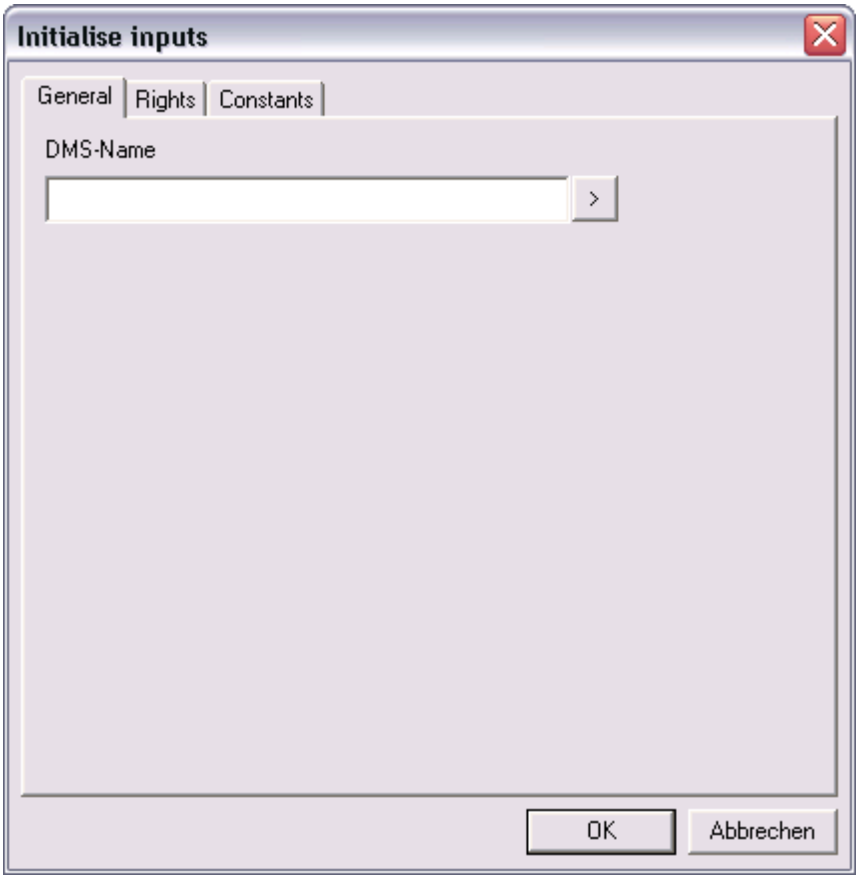
If the entry is incorrect, e.g. 30:60:16, a warning is issued!



10.3.15.7.5 Combo Box

A **combo box** enables a data point to be set to a specific value in runtime mode. This involves selecting the value from a list. Both digital and analogue data points as well as data points of the type STR (string) can be set.

In order to initialise a **combo box**, click on the button in the Initialisation column in the **Action** column. The **Value Entry Initialisation** dialog window appears:

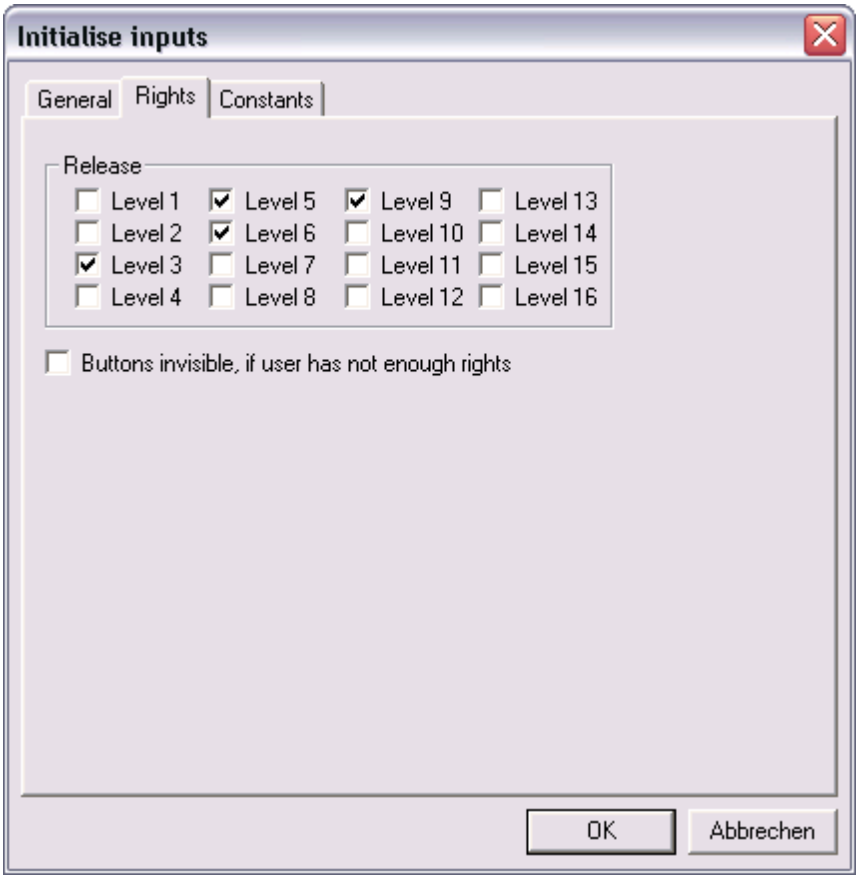


The window consists of the three registers "**General**", "**Access Rights**" and "**Constants**"

General tab
The DMS name of the value to be set can be entered in the input field. It can also be selected from a list. The list appears after double-clicking in the entry field or by clicking on the button to the right of it.

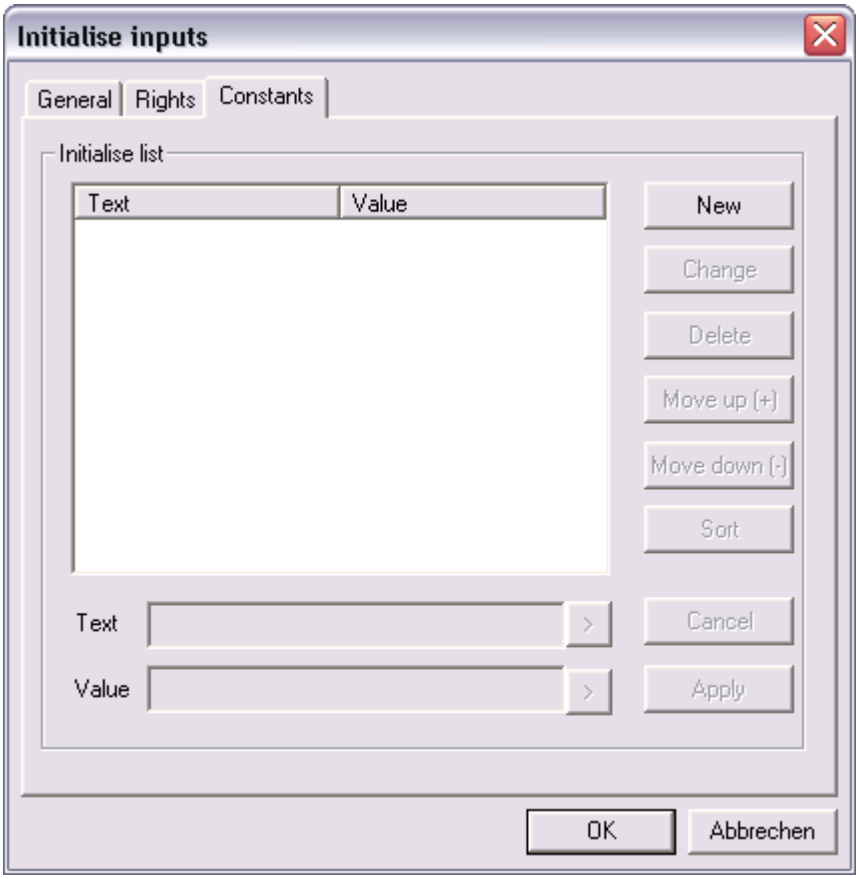
Access Rights tab
Which level has the required access rights?

Release
Enables the access rights defined in the User Management module (see the chapter **User management**) to be assigned. If a user does not have the necessary user rights, the initialised action cannot be executed. If no user rights are assigned, all users can execute the action.



Control button not visible without access rights
The input field can be made invisible, and therefore inoperable, for users without assigned rights.

Constants tab



Here you can edit the list that appears by clicking on the dropdown menu in the combo box. The text is displayed in the list field. One value belongs to each text. By selecting a list field, the initialised DMS data point is set to this value.

In order to generate the list, click on **New**. The text and value can now be edited.A string or a DMS name can be entered for the text. The entry of a DMS name can take place directly through the keyboard or by selecting from a list.The list appears after double-clicking in the entry field or by clicking on the button to the right of it. With initialisation to a data point, the content of the data point is displayed in the combo box (see Example 2). The data point, therefore, must be of type STR (string).

A number, a text or a DMS name and be entered for the value. Selection of the DMS name takes place in the same way as the procedure described for texts. With the initialisation to a data point, the content of the data point is taken (see Example 2).

Clicking on the **Adopt** button completes the entry. The new values are adopted to the initialisation list. They can now be changed, deleted or sorted. The actions Change, Delete and Sort always refer to the active line (marked blue). In order to select a line, click with the left mouse button on the corresponding line in the initialisation list or scroll through the list with the arrow keys if a line is already selected.

New button
Allows you to input text with the corresponding value.

Change button
An already entered text and its value can be changed.

Delete button
Deletes the active line.

Up button (+) or the <+> key
Moves the selected line up by one position.

Down button (-) or the <-> key
Moves the active line down by one position.

Sort button
The list entries are sorted alphabetically in ascending order according to the text.

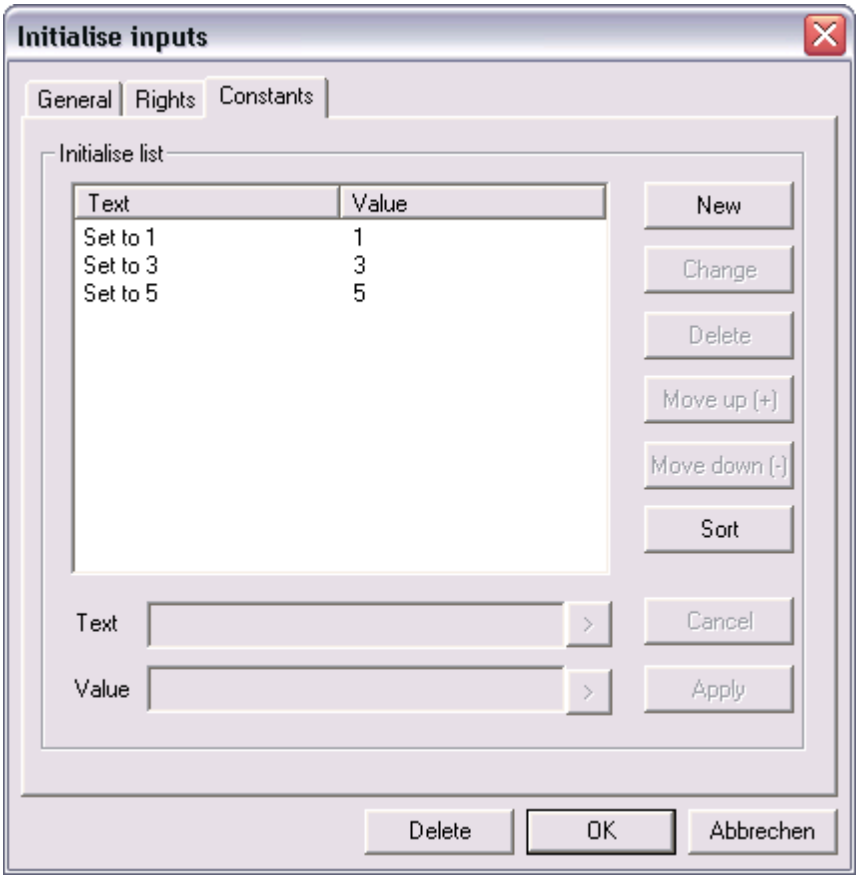
Cancel button
A newly created entry is not adopted into the list.

Adopt button
A newly created entry is adopted into the list.

Example objective 1
For the following example, you must start the Visi.Plus project, which is included in the scope of delivery.
The data point "BHS60:MM:504:ActualValue" should be able to be set to 1, 3 and 5. The value of the data point should be displayed.

Example solution 1

- Draw a text field as described under Text/Text Field
- Call up the Attributes window.
- Click on the button in the Initialisation column by the attribute "Text" and select the command "Value Display".
- Enter the DMS name "BHS60:MM:504:ActualValue" in the Initialisation dialog or select it from the list. Set the Output Format to %3.2f.
- Drawing an input field
- Click on the button in the Initialisation column by the "Action" attribute.
- Enter the DMS name "BHS60:AV:506:ActualValue" in the Initialisation dialog in the General tab or select from the list. The following list is created in the Constants tab:

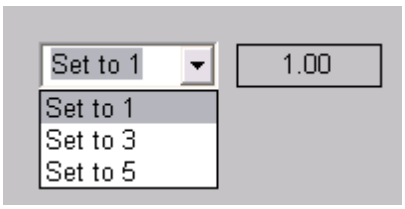


The created image should then appear as follows:

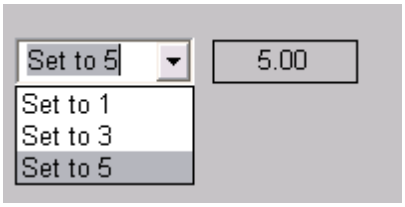
Does the combo box exhibit the following behaviour?

The current value of "BHS60:MM:504:ActualValue", which is displayed in the text field, equals 47.13 (you may have a different value here).Since this value does not match a value from the Initialisation column, the combo box is blank.Now you must switch to runtime mode. Only then does the dropdown menu of the combo box become operable.

Click on the dropdown menu and select Set to 1. Now the value of "BHS60:MM:504:ActualValue" is set to 1.

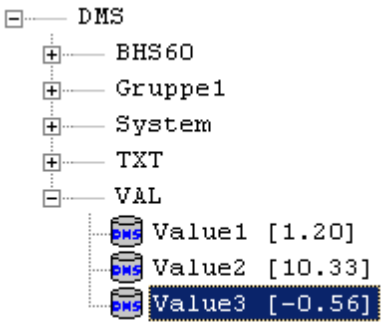


Click on the dropdown menu again and select Set to 5. Now the value of "BHS60:MM:504:ActualValue" is set to 5.

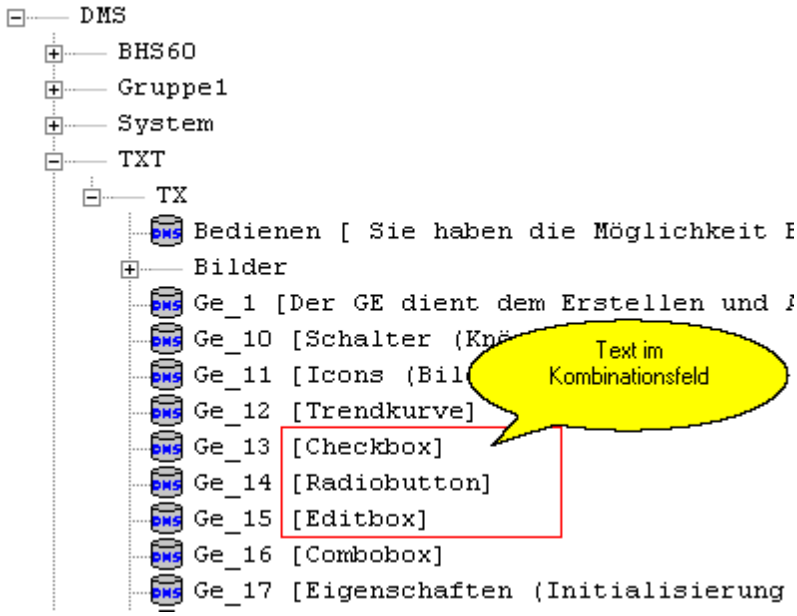


Example 2:

Example objective 2
First, three new data points, VAL:Value1 (type FTL, value 1.2), VAL:Value2 (type FLT, value 10.33) and
VAL:Value3 (type FLT, value -0.56) must be created in the DMS.

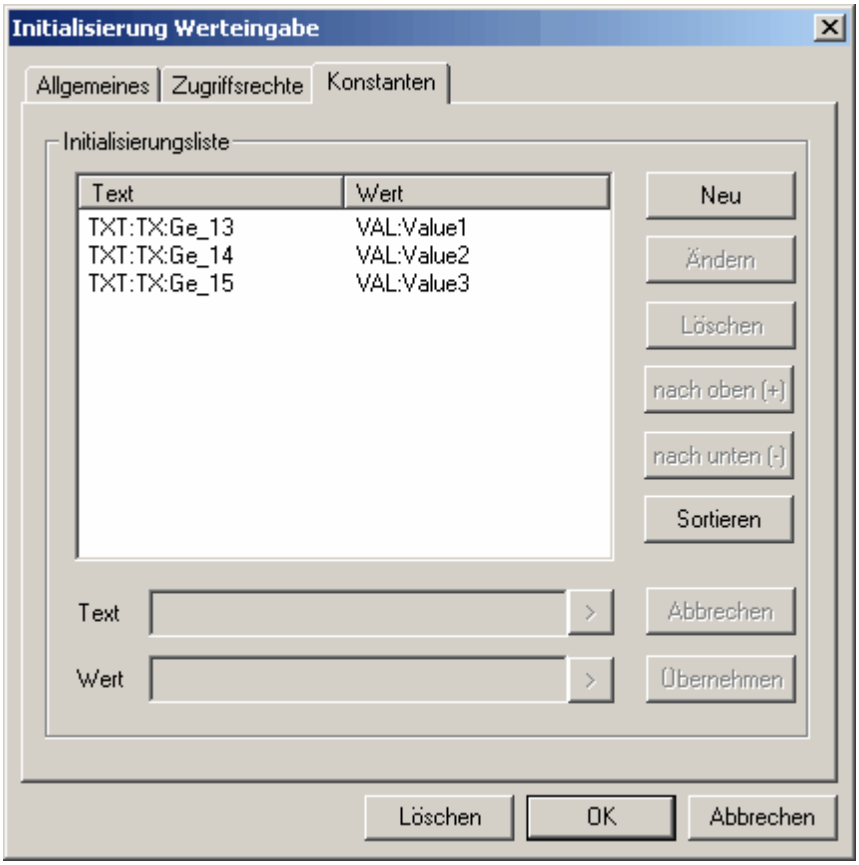


The data point "BHS60:MM:504:ActualValue" should be able to be set to the values of the data points VAL:Value1, VAL:Value2 and VAL:Value3. The texts in the combo box should be taken from the DMS.



Example solution 2
- Copy the combo box from Example 1

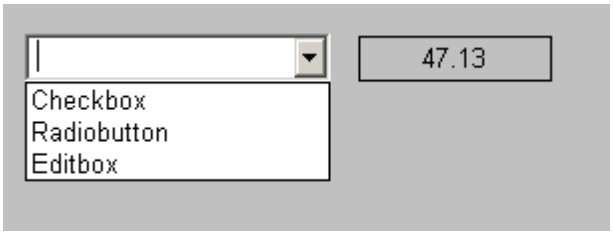
- Call up the Attributes window.
- Click on the button in the Initialisation column by the "Action" attribute.
- Enter the DMS name "BHS60:AV:506:ActualValue" in the Initialisation dialog in the General tab or select from the list. The following list is created in the Constants tab:



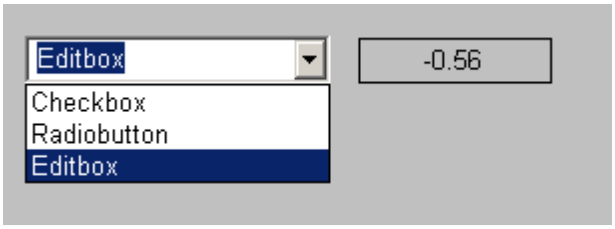
The image should then appear as follows:

Does the created combo box exhibit the following behaviour?

The dropdown menu of the combo box can only be operated in runtime mode, so switch to runtime mode. The list field of the combo box now appears as follows:



If you select the entry "Editbox" in the combo box, the value of "BHS60:MM:504:ActualValue" is set to the value of VAL:Value3, meaning to -0.56.



A combo box can only be operated in runtime mode. The data point (in the example "BHS60:MM:504:ActualValue") is logged in the DMS. This means that the selection in the combo box is changed when the value changes. If the value in the DMS does not match any of the initialised constants, the combo box remains blank. If the texts were initialised to a data point, the list in the combo box is also updated when the values change.

10.3.15.8 Action: Graph

The **action "Graph"** is used for initialisation of a polygon, button or bitmap graphical object for **trend graph** operation. The time-dependent trend graph objects themselves have no control buttons.

The position of the control button, colour, dimensions, label, etc. can be defined independently with this method. Great flexibility is yielded as a result.

The chapter Trend Graph describes how **trend graphs** are used.

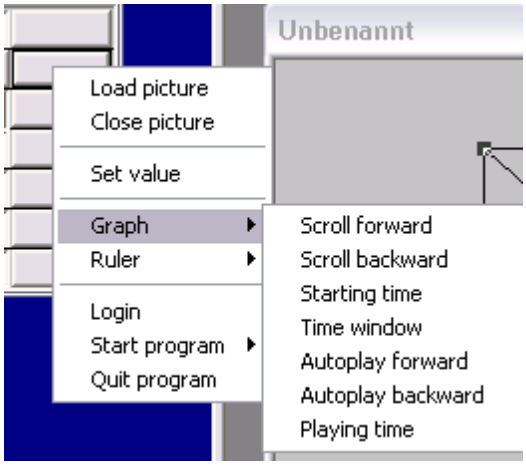
The contents of the **trend curve objects** can assume large dimensions. Such amounts of data cannot be represented on screen at a reasonable resolution. Therefore it makes sense to use a so-called **time window** which can represent a range of any size.

In order to determine the size of the time window, its starting position and the its forward and backward movement on the time axis requires **Control buttons**, as already described above.

From Version 1.5 a graph ruler can be shown in the trend graph diagram (runtime mode), which can be moved using the arrow keys (see chapter Trend graphs - Example 7). The graph ruler can also be controlled with the autoplay function (see chapter Trend graphs - Example 8). This causes the graph ruler to automatically move either forwards or backwards over the range of the diagram. **Control buttons** are also necessary for this.

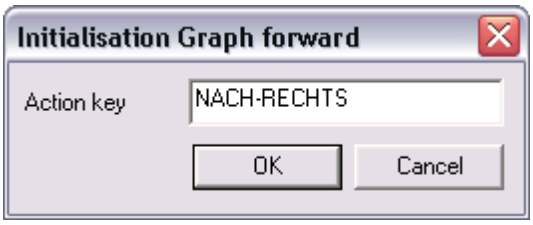
The actions that have just been presented are described in greater detail in the following:

In the **Attributes window** of the polygon, button or bitmap graphical object of the Initialisation button (in the outer right column of the Attributes window), when the attribute **Action** is clicked, the following dropdown menu opens depending on the version:



Graph Forward or Scroll Forward
With this action, by clicking on a polygon, button or bitmap in runtime mode, the contents of the trend graph are shifted **to the right** by the length of the time window, towards more recent data (see "**Time Window**" under **Action**). If the newest trend data is already displayed, the graphical object no longer has any function.

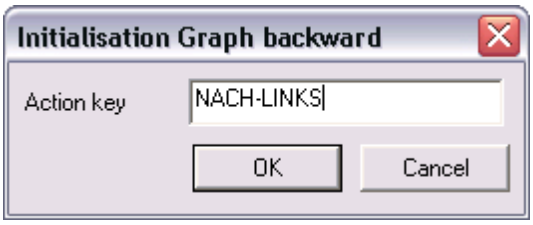
A key combination can be assigned by clicking with the mouse in the input field "**Action key**" and then pressing the desired key or key combination.



Key combinations can be assigned to each of the actions, as described above. This also applies for the following actions.

If a graph ruler is shown in the trend window (possible in Version 1.5 or higher), the right arrow key (action key RIGHT) is no longer valid for shifting the time axis. Instead, the graph ruler is shifted to the right (higher priority). It is therefore recommended to select another key combination, e.g. CTRL + RIGHT, in GE 1.5.

Graph Backwards or Scroll Backwards
With this action, by clicking on a polygon, button or bitmap in runtime mode, the contents of the trend curve are shifted **to the left** by the length of the time window, towards older data (see "**Time Window**" under **Action**).



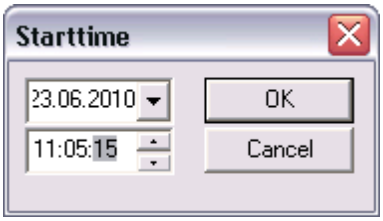
If a graph ruler is shown in the trend window (possible in Version 1.5 or higher), the left arrow key (action key LEFT) is no longer valid for shifting the time axis. Instead, the graph ruler is shifted to the left (higher priority). It is therefore recommended to select another key combination, e.g. CTRL + LEFT, in GE 1.5.

Graph Start Time
The graph action "**Start Time**" allows the user to indicate the starting position of the time window within the trend data in runtime mode.

With the assignment of **Action** > **Graph** > "**Start Time**" in the Attributes window the following dialog window appears:

The **<OK>** button is used to confirm and adopt the **start-time initialisation**.

If the polygon, button or bitmap graphical object is then clicked on in runtime mode and the corresponding trend graph is located in the same process diagram, the following control window is displayed:

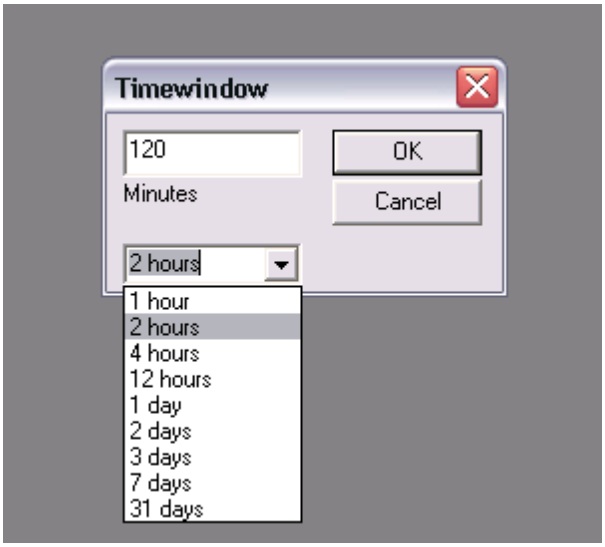


Here the user can determine the **Start Time** of the trend graph.

*The button with the action "**Graph**" – "**Start Time**"* **must** *be located in the same process diagram as the respective trend graph. Otherwise the initialised "**Button**" graphical object has no function.*
The chapter Trend graph *describes how trend graphs are used.*

Graph Time Window
The initialisation of the action **Time Window** is executed in the same manner as the action **Start Time**.

If the polygon, button or bitmap graphical object is clicked on in runtime mode and the corresponding trend graph is located on the same process diagram, the following control window is displayed:



In this control window the user can select the length of the time window with a dropdown menu - **2 hours** in this case.

Graph Autoplay Forwards

With this action, by clicking on a polygon, button or bitmap in runtime mode the autoplay function is started forwards. This causes the graph ruler of the trend object to continuously move to the right, in the direction of more recent data. If the graph ruler reaches the right frame boundary, the contents of the trend graph are automatically shifted **to the right** by the length of the time window if the time on the right edge is less than the current time. If this is not the case, meaning the graph ruler has reached the current time, the graph ruler is automatically stopped. The graph ruler can be stopped by clicking on the graphical object again.

Graph Autoplay Backwards

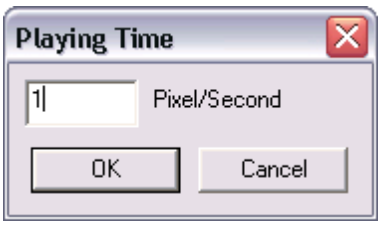
With this action, by clicking on a polygon, button or bitmap in runtime mode the autoplay function is started backwards. This causes the graph ruler of the trend object to continuously move to the right, in the direction of older data. If the graph ruler reaches the left frame boundary, the time axis is automatically shifted **to the left** by the length of the time window. The graph ruler can be stopped by clicking on the graphical object again.

Graph Playback Speed
With this action, by clicking on a polygon, button or bitmap in runtime mode the playback speed is determined, meaning how fast the graph ruler should move over the range of the diagram.

With the assignment of **Action > Graph > Playback Speed** in the Attributes window, the following dialog window appears:

The **<OK>** button is used to confirm and adopt the initialisation.

If the polygon, button or bitmap graphical object is then clicked on in runtime mode and the corresponding trend graph is located in the same process diagram, the following control window is displayed:



It is preset to 1 pixel/second. The value can be changed via the keyboard.

If the selected playback speed is too high, e.g. if the computer has insufficient memory, the graph ruler will always travel over the range of the diagram at the maximum possible speed.

10.3.15.9 Action: Graph Ruler

The **action** "Graph Ruler" is used to initialise a polygon, button or bitmap graphical object for the operation of the graph ruler object, which is available in Version 1.5 or higher.

The time-dependent graph ruler object itself has no control button.

The position of the control button, colour, dimensions, label, etc. can be defined independently with this method. Great flexibility is yielded as a result.

As an example, with the help of a graph ruler object the status of a pump or a valve (switched on or off) or the value of an analogue signal (e.g. temperature) can be visualised in a process

diagram for a point of time in the past.

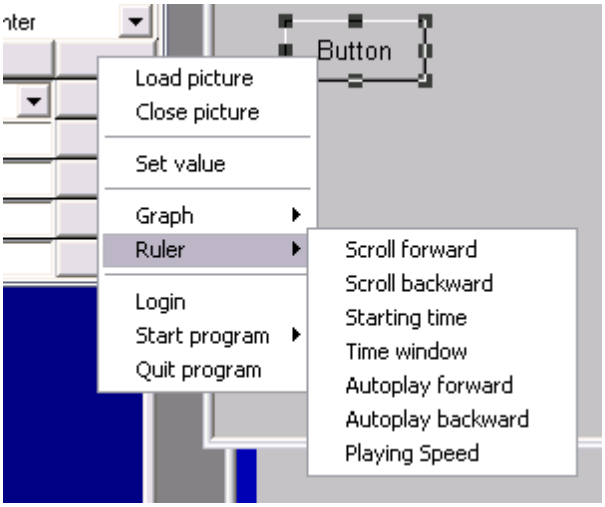
The chapter Graph ruler in combination with 4 examples describes how graph ruler objects are used.

Control buttons for changing the size of the time window, determining the starting position as well as for shifting the time axis forwards or backwards are created in the same way as the control buttons of the trend object.

In runtime mode a graph ruler, which can be moved with the arrow keys (see chapter Graph ruler - Example 10), is shown by clicking on the graph ruler object. The graph ruler can also be controlled with the autoplay function (see chapter Graph ruler - Example 12). This causes the graph ruler to automatically travel either forwards or backwards over the time range within the graph ruler object. **Control buttons** are also necessary for this.

The actions that have just been presented are described in greater detail in the following.

In the **Attributes window** of the polygon, button or bitmap graphical object of the Initialisation button (in the outer right column of the Attributes window), when the attribute **Action** is clicked on, the following dropdown menu is displayed:



Scroll Graph Ruler Forwards
With this action, by clicking on a polygon, button or bitmap in runtime mode, the time axis of the graph ruler object is shifted **to the right** by the length of the time window, towards more recent data (see "**Time Window**" under **Action**). If the right time value is equal to the current time, the graphic object no longer has any function.

A key combination can be assigned by clicking the mouse in the input field "**Action Key**" and then pressing the desired key or key combination.

Key combinations can be assigned to each of the actions, as described above. This also applies for the following actions:

Scroll Graph Backwards
With this action, by clicking on a polygon, button or bitmap graphical object in runtime mode, the time axis of the graph ruler is shifted **to the left** by the length of the time window, towards older data (see "**Time Window**" under **Action**).

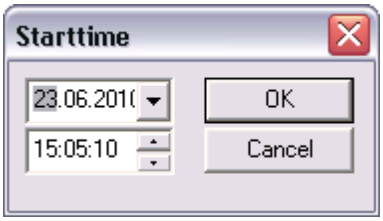
If the graph ruler is shown in the graph ruler object, it can be moved back and forth with the arrow keys. If the left *or* right arrow *action key was selected under "Graphical Objects", which would allow for scrolling the graph ruler object forwards or backwards, no action is executed by these objects. Therefore the selection of a different key combination than RIGHT (right arrow key) or LEFT (left arrow key) is suggested.*

Graph Ruler Start Time
The graph action "**Start Time** allows the user to indicate the starting position of the time window in runtime mode.

With the assignment of **Action > Graph Ruler > "Start Time**" in the Attributes window, the following dialog window appears:

The **<OK>** button is used to confirm and adopt the **start-time initialisation**.

If the polygon, button or bitmap object is now clicked in runtime mode and the corresponding graph ruler object is located on the same process diagram, the following control window is displayed:



Here the user can determine the **Start Time** for the graph ruler object.

*The button with the action "**Graph**" – "**Start Time**" must be located on the same process diagram as the graph ruler object. Otherwise the initialised "**Button**" graphical object has no function.*

Graph Ruler Time Window
The initialisation of the action **Time Window** is executed in the same manner as the action "**Start Time**".

If the polygon, button or bitmap object is clicked on in runtime mode and the corresponding graph ruler object is located on the same process diagram, the following control window is displayed:

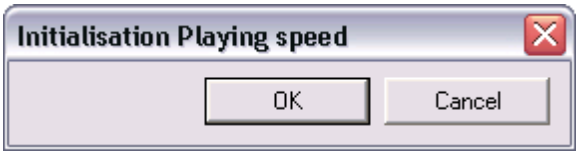
In this control window the user can select the length of the time window with a dropdown menu - **2 hours in this case.**

Graph Ruler Autoplay Forwards
With this action, by clicking on a polygon, button or bitmap in runtime mode the autoplay function is started forwards. This causes the graph ruler of the graph ruler object to continuously move to the right, in the direction of more recent data. If the graph ruler reaches the right frame boundary, the time axis of the graph ruler object is automatically shifted **to the right** by the length of the time window if the time on the right edge is less than the current time. If this is not the case, meaning the graph ruler has reached the current time, the graph ruler is automatically stopped. The graph ruler can be stopped by clicking on the graphical object again.

Graph Ruler Autoplay Backwards
With this action, by clicking on a polygon, button or bitmap in runtime mode the autoplay function is started backwards. This causes the graph ruler of the graph ruler object to continuously move to the left, in the direction of older data. If the graph ruler reaches the left frame boundary, the time axis is automatically shifted **to the left** by the length of the time window. The graph ruler can be stopped by clicking on the graphical object again.

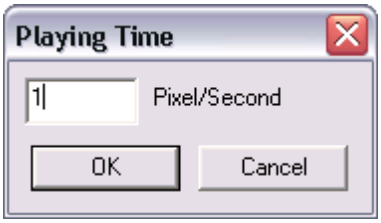
Graph Playback Speed
With this action, by clicking on a polygon, button or bitmap in runtime mode the playback speed is determined, meaning how fast the graph ruler of the graph ruler object should move over the time range of the diagram.

With the assignment of **Action > Graph Ruler > Playback Speed** in the Attributes window, the following dialog window appears:



The **<OK>** button is used to confirm and adopt the initialisation.

If the polygon, button or bitmap object is now clicked in runtime mode and the corresponding graph ruler object is located on the same process diagram, the following control window is displayed:



It is preset to 1 pixel/second. The value can be changed via the keyboard.

If the selected playback speed is too high, e.g. if the computer has insufficient memory, the graph ruler will always travel over the range of the diagram at the maximum possible speed over the time range.

10.3.15.1 Action: System Login

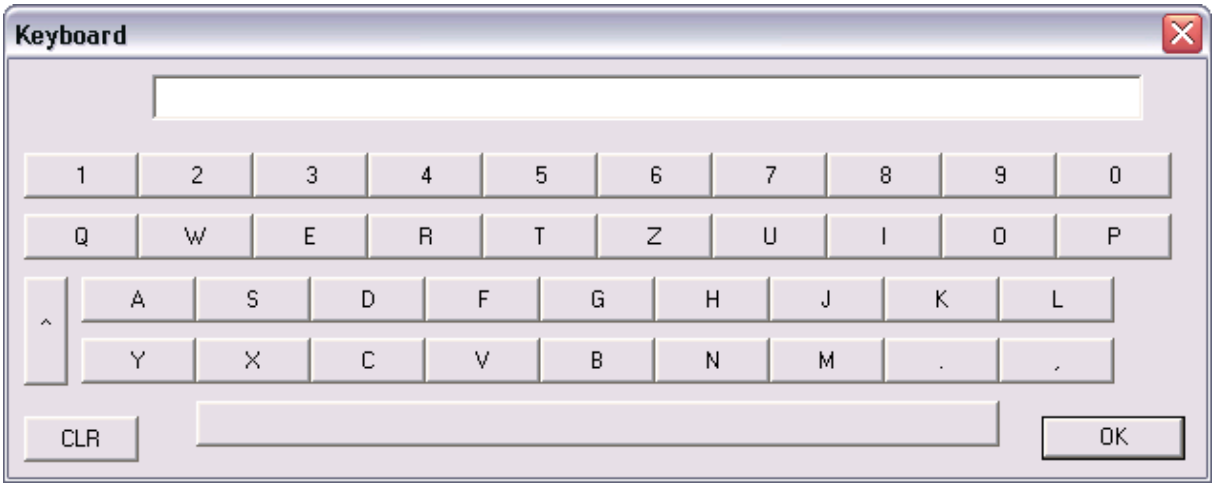
If the action "**System Login**" is assigned to the polygon, button or bitmap, the user can log in to the visualisation system.

During the first initialisation, the dialog window "**Login Initialisation**" is displayed. Acknowledge this with **<OK>**.

By clicking on the graphical object assigned to the action "**System Login**" in runtime mode, the dialog window **Login** is displayed:

If "Enter via Touchscreen" is selected, additional buttons become visible for entry of the user name and password.

If these buttons are selected, the following window opens up, in which the entries can be typed in using a touchscreen:



User
Name of the user.
The user must have been created using the **User Management** module (program) (see chapter) before logging in.

Password
The password assigned to the user in the **User Management** module is entered here for the login.

More information about user management and passwords can be read in the **chapter User management**.

*In runtime or edit mode a user can log in at any time by pressing the key combination **<CTRL+L>** or by selecting the command "**User Login**" in the menu "**Project Settings**" .*

10.3.15.1 Action: Start Program

As the name of the action suggests, Visi.Plus modules (programs) or other programs can be opened in runtime mode with this action using a graphical object (polygon, button or bitmap).

The following three modules can be assigned by default to a graphical object (polygon, button or bitmap):

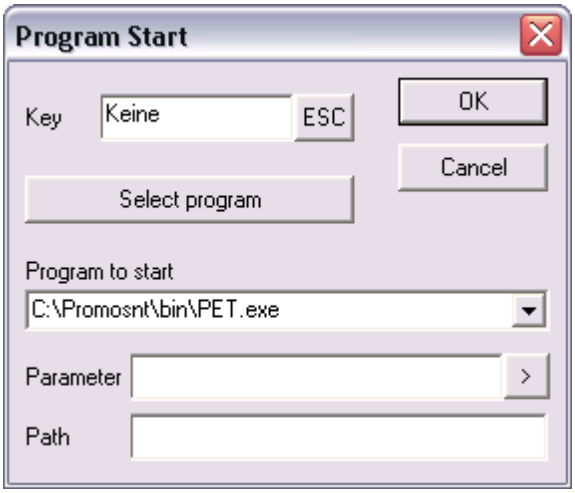
Use these prepared functions for these modules. If, for example, you call up AlmView.exe using you own choice of program, it will eventually lead to problems. For example, AlmView will not start in web functionality.

Alarm Program
(ALMView) Filename **ALMView.EXE**

Logging
(PRTview) Filename **Prtview.EXE**

Time switch program
(CLKcfg) Filename**CLKcfg.EXE**

Arbitrary program
Means any program. Starting such a program requires some information which must be entered in the dialog window **Program Start**.



Key
The input field enables any key or key combination (except for with **<Alt>**) to be defined as an alternative to clicking on the "button" graphical object. If such a program should be started with the **<ESC>** key, the **<ESC>** button to the right of the input field must be clicked on.

Program Selection
Opens a file selection window. The program to be started can be selected via this window. In the example the Process Engineering Tool (PET) of Visi.Plus is selected. In order to adopt the settings, click on **<OK>**. The program **PET.exe** is found in the installation directory **c:\Visi.Plus\bin**.

It is also possible to indicate starting parameters with the program. This should be included on the line with the program name. Example: **c:\windows\notepad.exe info.txt**
(When it starts, the program **Notepad** will open the file info.txt)

Program to be Started
The directory path of the program to be started is shown here.

Parameter
A DMS name can be entered here. If the parameter is of the string type, the content of the DMS is included as a parameter (e.g. a filename). If the parameter DMS name is of another type, the GE converts the parameter to a filename (e.g."**H02:MT:500:Err**" is converted to "**H02_MT_500_Err.txt**"). So, for example, a unique filename can be generated and/or edited for each data point.

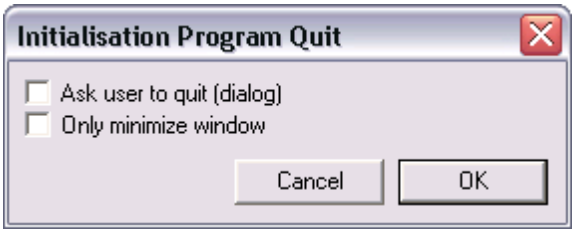
Path
Holds the path where the program to be started should be executed. Alternatively, the system variable %PROJECT% can also be entered.

Example:

```
%PROJECT%\txt
```
switches to the directory c:\Visi.Plus\proj\System1\txt (where System1 is the current project).

10.3.15.12 Action: Exit Program

If the Graphical Editor should be exited via a polygon, button or bitmap, initialise the action "**Program Exit**" to one of the graphical objects.



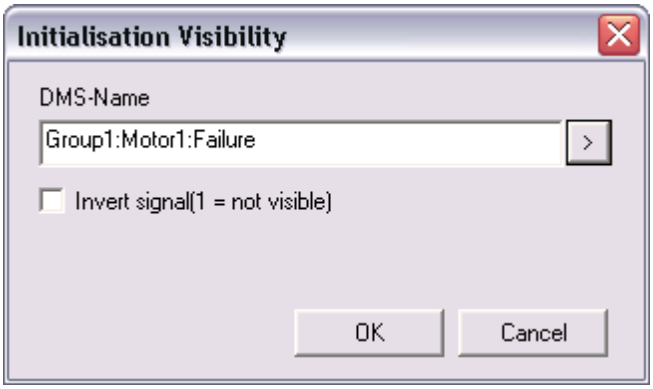
Program Exit with safety prompt
If the checkbox is activated, Visi.Plus asks whether the editor should really be exited before exiting the Graphical Editor.

Only Minimise a Program
Does not exit the Graphical Editor, rather only minimises it to the taskbar.

Initialisations can be deleted very easily in Visi.Plus. To do so, activate the initialised graphical object (polygon, button or bitmap) and right-click on it. Select the command ***"Delete Init"*** *. A submenu appears (all attributes not initialised are shown in grey) with which all or only individual initialisations of the graphical object can be deleted with the click of a mouse.*

10.3.15.13 Action: Visibility

In order to initialise the attribute "**Visibility**" of an object, click on the Initialisation button for "**Visibility**". The window **Visibility Initialisation** appears.



In the input field
... a binary DMS data point must be entered, which controls the visibility of the graphical object.

Invert Signal
Causes an opposite reaction to the status of the DMS data point.

10.3.15.14 Action: Position

Every Visi.Plus process diagram is broken town into Cartesian XY coordinates.

Coordinates are figures that indicate the position of one point in relation to another point. Absolute coordinates, like those used in Visi.Plus, are always based on the origin of the current system of coordinates.

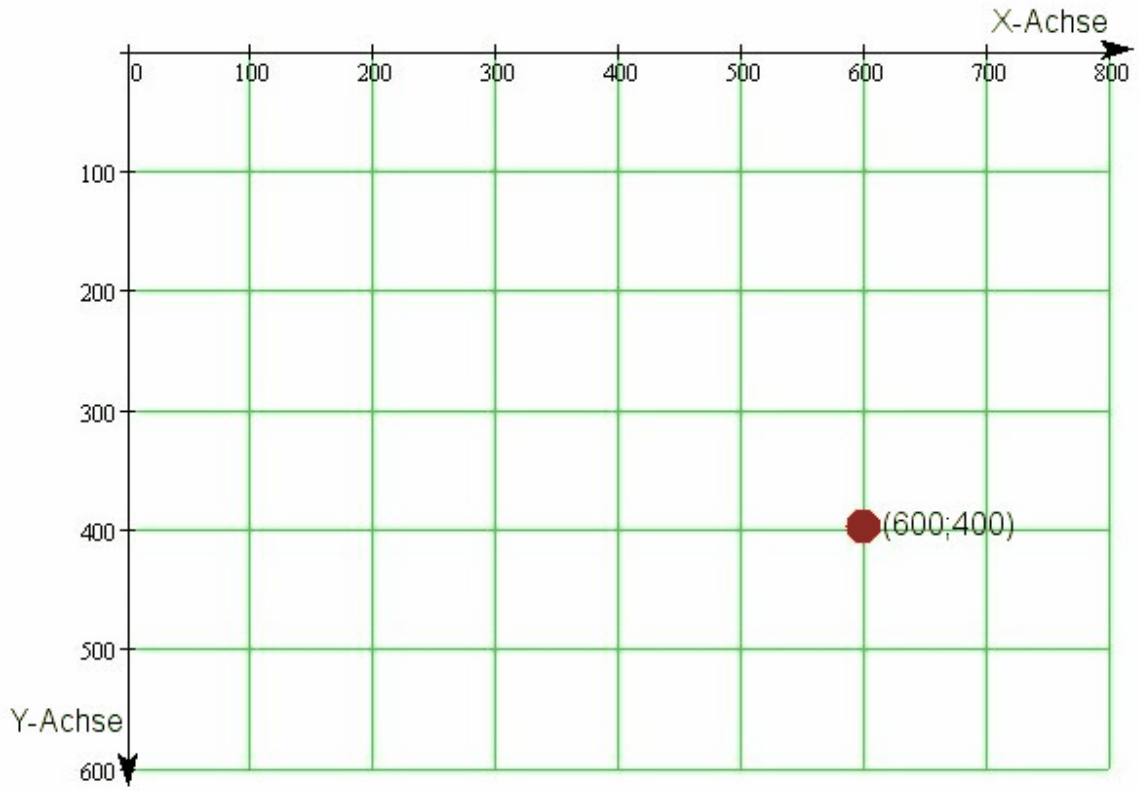
XY coordinates represent the position as a horizontal and vertical distance from the **coordinate zero point (0:0)**, which is found in the **upper left corner** of each process diagram.

The lower right corner of the process diagram, the **end-point coordinate**, is determined by the process diagram resolution. The largest process diagram should lie within the maximum screen resolution.

When it is created, the resolution of the process diagram must be correctly configured on the basis of the computer on which it is to be used! More information about process diagram resolution can be found in the chapter Project Settings.

Examples of screen resolutions (in DPI):
800x600 has the process diagram end-point coordinates 799x599
1024x768 has the process diagram end-point coordinates 1023x767

In the following image an example of an XY coordinate is represented:



*In Visi.Plus the **Dimensions** of graphical objects (except for polygons) can be set using their **upper left** and **lower right corners**. These are listed in the Attributes windows of the graphical objects as follows:*

Left Position	corresponds to **X1**
Top Position	corresponds to **Y1**
Right Position	corresponds to **X2**
Bottom Position	corresponds to **Y2**

For precise positioning of graphical objects on the screen, the values of the four position fields can be entered manually.

*As soon as the value has been entered for X1, jump to the next field "**Position**", etc. with the **<TAB>** key. Do not forget to press the **<ENTER>** key after the final entry, Y2, otherwise the value will not be accepted.*

Initialise Positions to DMS Names
Initialisation of positions only makes sense if they are initialised to an analogue signal. In order to explain initialisation of the positions, the system objects simulated by Visi.Plus from "**Group1**" are used.

Advantage:
No PLC is required to create signals.

The DMS name "**Group1:Motor1:Temperature**" supplies a suitable analogue signal, which constantly counts up and down from 32 to 88.

This can be checked very easily in the DMS (similar procedure to the one described under "More experiments with the newly created button").

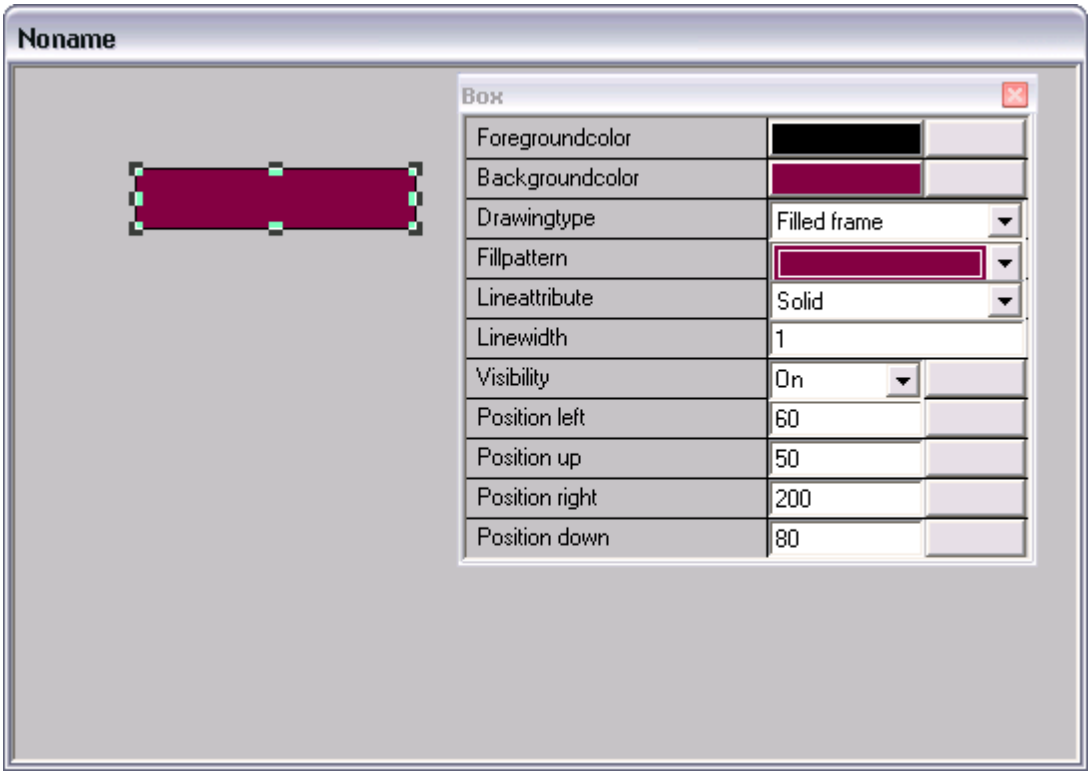
Example: Temperature Indicator

Task:
The width of a rectangular graphical object should be changed depending on the value of the DMS data point "**Group1:Motor1:Temperature**".

Solution:
First position a **rectangular graphical object** anywhere. Then correct its "**Position**" attributes as follows:
left = 60
top = 50
right = 200
bottom = 80

Set the attribute "**Background Colour**" to any colour (not grey)!

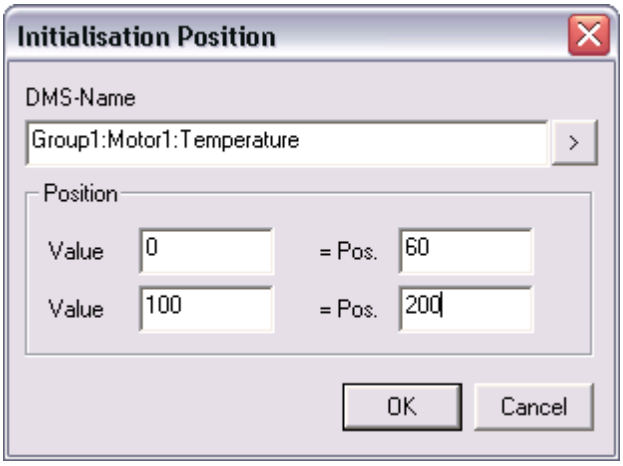
It will then appear roughly as follows:



Before the **Right Position** is initialised, the value **must** be set to the same value as **Left Position**. In this example this is the **reference value for the initialisation,** i.e. **60**. Visi.Plus adds the initialised value to this reference value. As soon as the initialisation (see below) has been carried out, the reference value can no longer be changed. (The only remedy is to delete the initialisation.)

In order to initialise the right side of the rectangular graphical object to the DMS name "**Group1:Motor1:Temperature**", click on the Initialisation button (right column of the Attributes window) of the attribute "**Right Position**".

The dialog window **Initialisation Position** now opens (still without values).



Double click in the first input field and select "**Group1:Motor1:Temperature**".

Position
The values can be transferred/scaled to the positions in this framed area.

Line 1: **Value**
If the content of the DMS data point corresponds to the **value 0**, this corresponds to the **=Pos.**
Position 60 on the process diagram.

Line 2: **Value**
If the content of the DMS data point corresponds to the **value 100**, this corresponds to the **=Pos.**
Position 200 on the process diagram.

The intermediate values are calculated by Visi.Plus. The result is the midpoint between 0 and 100 = 50, which converts to "**Right Position**" 130.

When **<OK>** is clicked, the bar changes to reflect the content of the DMS data point.

Summary:
The range of signal values must be entered in the **Value** input fields. The minimum value is entered in the top field and the maximum value is entered in the bottom field. The respective **coordinates** are entered in the "**=Pos.**" input fields.

The position in the upper input field may not be greater than that of the lower input field.

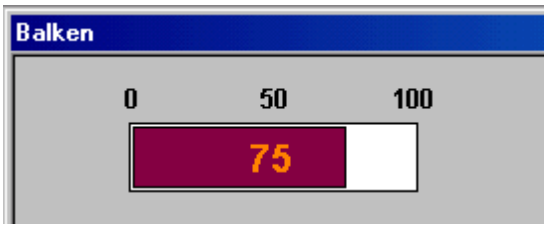
Example Expanded Temperature Indicator

A second **rectangular graphical object** should form a frame around the indicator bar. Enter the **Positions** as follows:
left = 58
top = 52
right = 202
bottom = 82

Set the attributes "**Background Colour**" and "**Foreground Colour**" to any colour (except grey)!

Modify the box graphical object with the **menu item** "**Attributes**" **>** "**In the Background**".

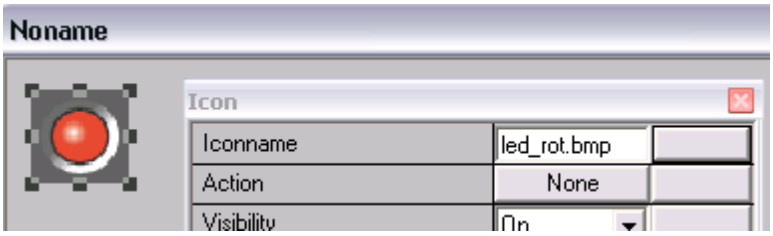
The bar indicator is completed by some **Text** graphical objects.



10.3.15.15 Icon Name

The **Icon/bitmap** graphical object only has the attribute **Icon Name**.

Icon Name
In the input field on the "**Icon Name**" line the name of the bitmap file to be displayed should be entered.

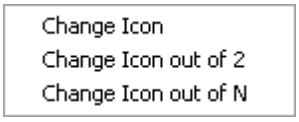


In the example, the file "**led_red.bmp**" is assigned to the "**Icon/bitmap** graphical object. The results can already be seen in the GE's edit mode, to the right of the Attributes window.

*All bitmap files that should be used for the current project **must** already be present in the subdirectory **\bmp** of the current project directory prior to their use.*

10.3.15.16 "Icon Change" initialisation

If **Icons/bitmaps should be changed** depending on the status of binary DMS data points, the commands "**Icon Change**" and "**Icon Change from 2**" are available in the Initialisation column under the attribute "**Icon Name**".



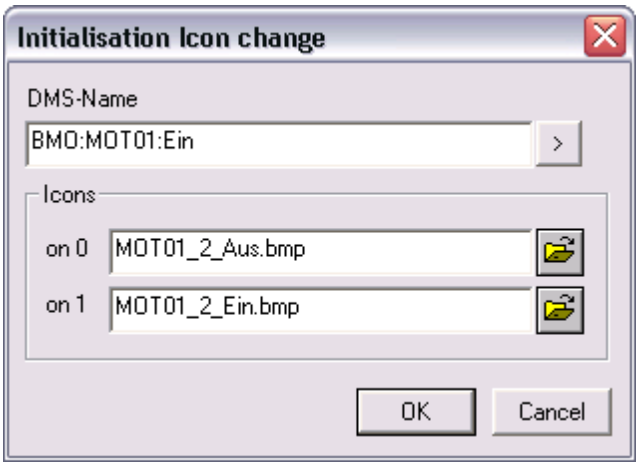
Icon Change
Can show two icons/bitmaps alternating dependent on a DMS data point. (More information can be found in the chapter Icon Change).

Icon Change from 2 enables the icons/bitmaps to be shown alternating dependent on 2 binary signal statuses. (More information can be found in the chapter Icon Change from 2).

Icon Change from N enables the icons/bitmaps to be shown alternating dependent on multiple binary signal statuses. (More information can be found in the chapter Icon Change from N).

The three functions are explained by the following examples.

Icon Change
A binary DMS data point shows the first specified bitmap at logic 0 and the second bitmap is shown at logic 1 of the data point.



Input Field
Here the DMS name of the binary data point is to be shown, which controls both bitmaps.

Icons at 0
Filename of the bitmap. If the data point has the value 0, this bitmap is to be shown.

Icons at 1
Filename of the bitmap. If the data point has the value 1, this bitmap is to be shown.

Icon change example:
The system objects of "Group1" simulated by Visi.Plus are used in the following.

- First, an "**Icon/bitmap**" graphical object is placed in the process diagram.
- Then the Attributes window is called up.
- At the "**Icon Name**" attribute, click on the button in the Initialisation column.
- Choose the "**Icon Change**" command.
- The icon Change Initialisation dialog window appears.

- Fill in the dialog window.

As soon as the entries are finalised with **<OK>**, the bitmap switches back and forth from green to blue in one-second intervals.

MOT01_2_On < switches > **MOT01_2_Off**

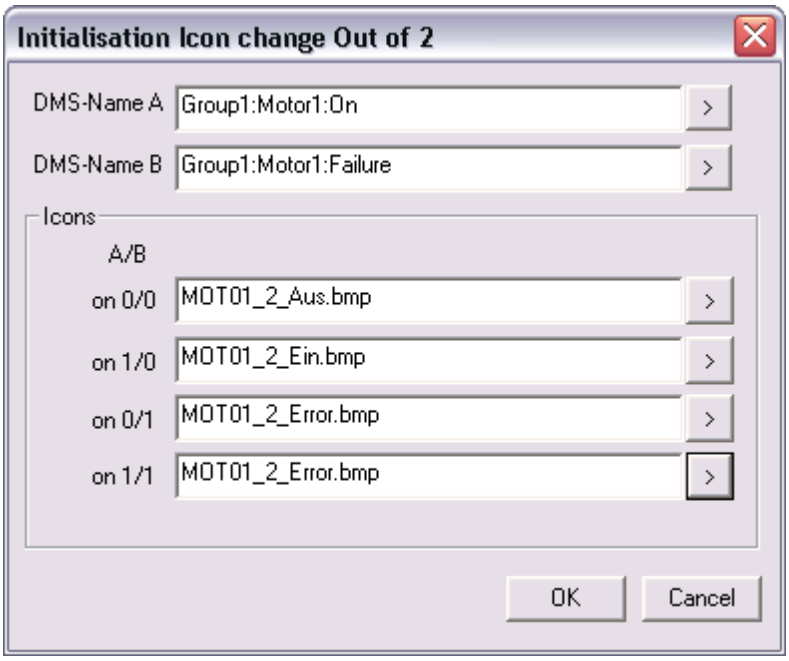
*If desired, this can be checked in the DMS. The procedure for monitoring is similar to the one already described under More experiments with the newly created button. However, the data point would be "**Group1:Motor1:On**". It can be observed in the DMS that the bitmap in the process diagram changes depending on this value.*

*Both bitmap files are located in the Visi.Plus directory in the Visi.Plus project, **c:\Visi.Plus\<project-name>\bmp**.*

TIP:
*If you want bitmaps to be shown in their original size, select the bitmap and then click the right mouse button and select "**Original Size**".*

Icon Change from 2

The command "**Icon Change from 2**" enables a visible icon/image **from 4 different bitmaps** to be shown **depending on two binary DMS data points**.



Input Field A
DMS data point of the first binary value for the icon change.

Input Field B

DMS data point of the second binary value for the icon change.

The respective icon/bitmap becomes visible when:

Icons at 0/0
both data points **A and B are logic 0**.

Icons at 1/0
Data point **A is logic 1** and **B is logic 0**.

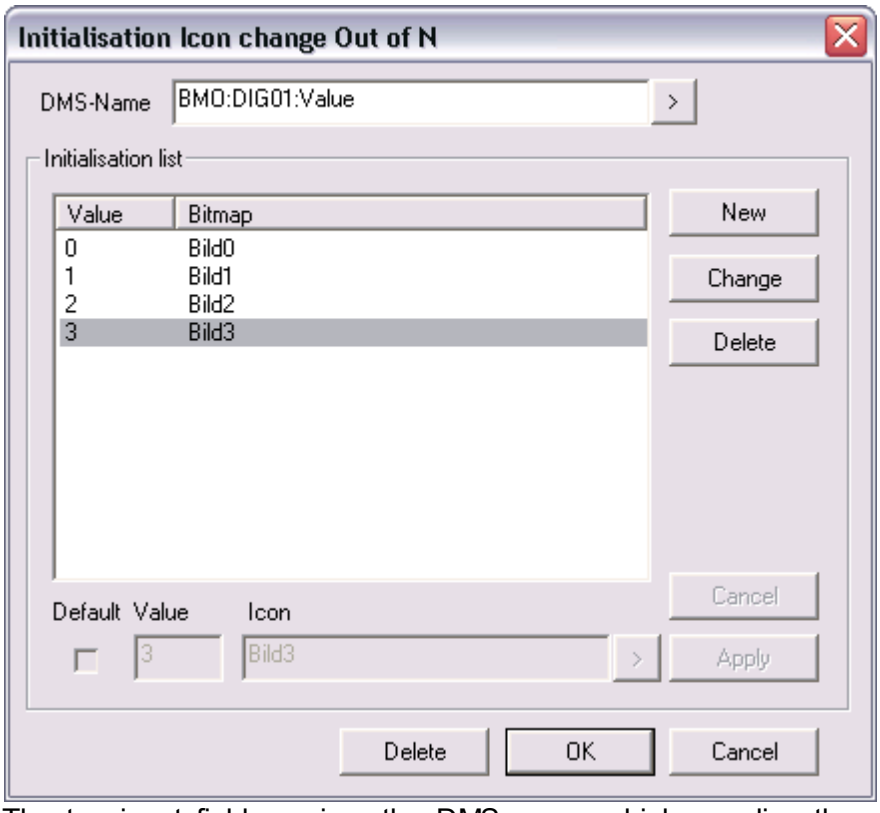
Icons at 0/1
Data point **A is logic 0** and **B is logic 1**.

Icons at 1/1
both data points **A and B are logic 1**.

Icon Change from N
With this option, various icon changes can be shown depending on a data point value.

Example solution:

- Draw an icon field
- Call up the Attributes window.
- **Click on the button in the Initialisation column by the attribute "Icon" and select the command "Change from N".**
- The dialog window **Icon Change from N Initialisation** appears:



The top input field requires the DMS name which supplies the value for selection of the corresponding .

By pressing New, a new value can be defined by specifying the image.

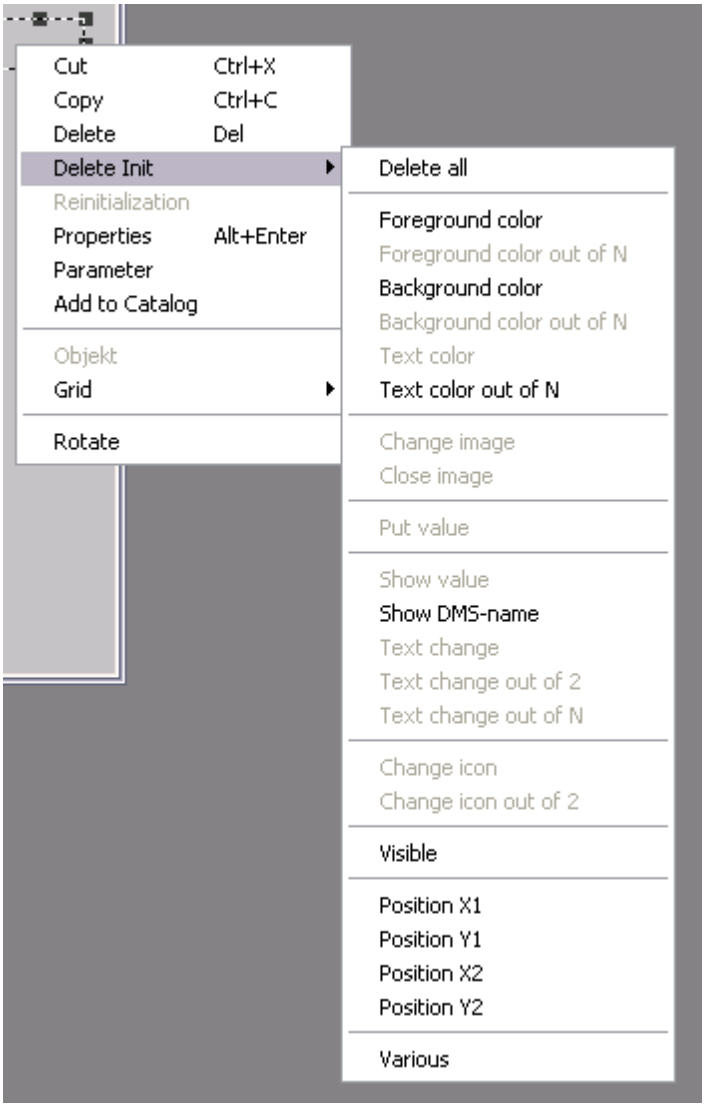
The corresponding image becomes visible when the data point includes the respective value in the range 0 to 3. If the value is greater than 3, image 3 is displayed. With **Adopt**, the defined values are adopted to the list, where it is possible to **Change** or **Delete** them again.

The settings are confirmed and adopted by clicking on the **<OK>** button.

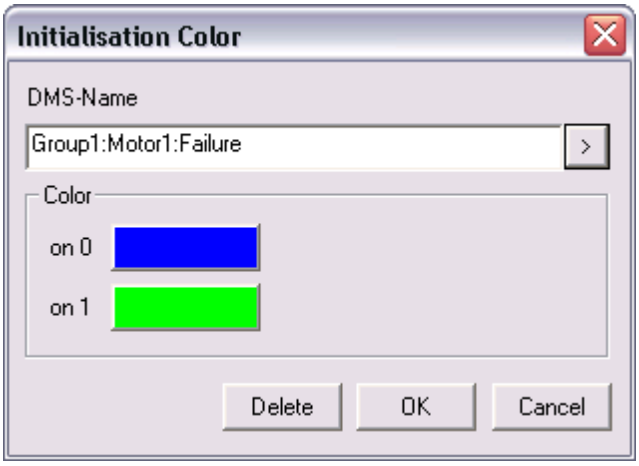
Icon input field
Enter the image to be used here.

10.3.15.17 Delete Initialisation

Initialisations can very easily be deleted in Visi.Plus.
To do so, select the initialised graphical object (polygon, button or bitmap) and click with the right mouse button.
Select the command "**Delete Init**". This causes a new menu to appear. Here you can select to "**Delete All**" or just **particular initialisations** of the respective graphical object, using the mouse.



All attributes that are not initialised are shown in grey.
*There are initialisations which can be removed directly from the control window of the initialisation via the "**<Delete>**" button, e.g. for "Icon Change Initialisation".*

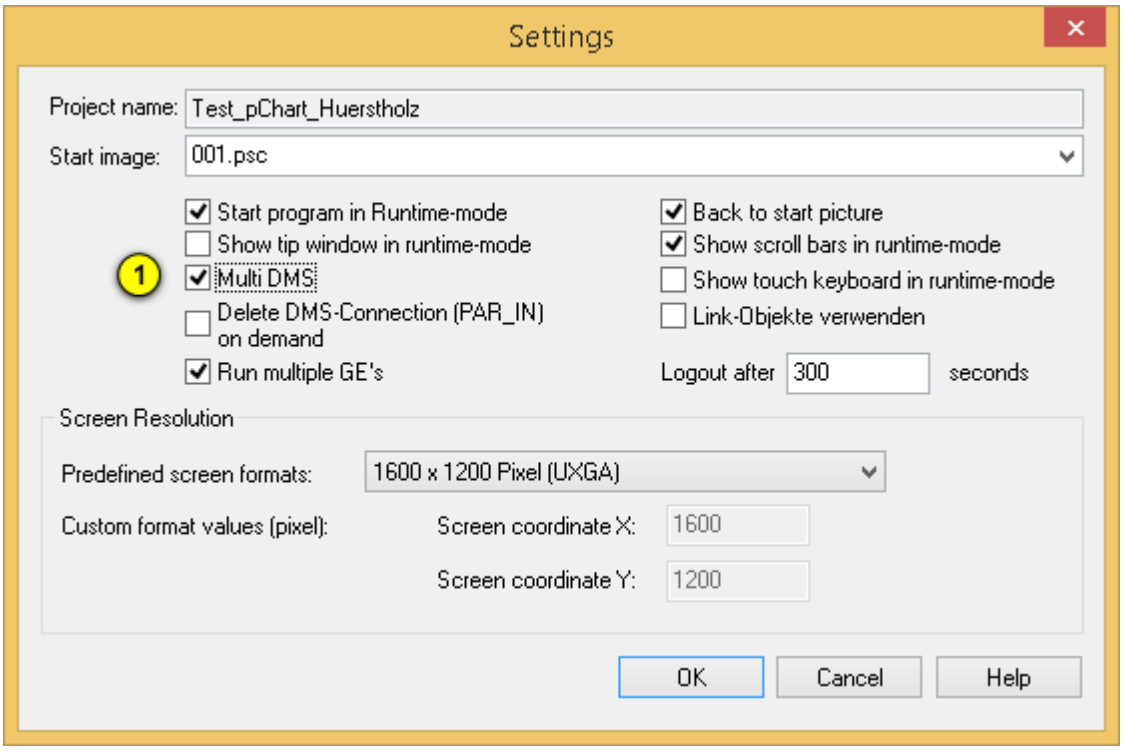


10.3.16 Multi DMS

With Multi DMS it is possible to connect several DMSes with each other and thereby distribute them over different computers. CPU-intensive projects can be distributed over different computers in this manner. Furthermore, each DMS can be started with an SDriver.

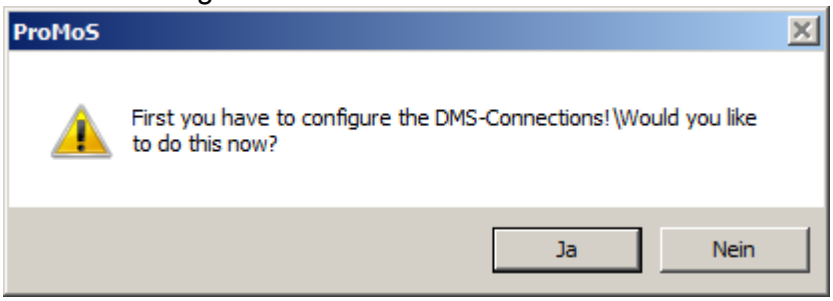
10.3.16.1 Defining Connections

Multi DMS can be activated in the GE under *Project Settings -> Settings* Project:

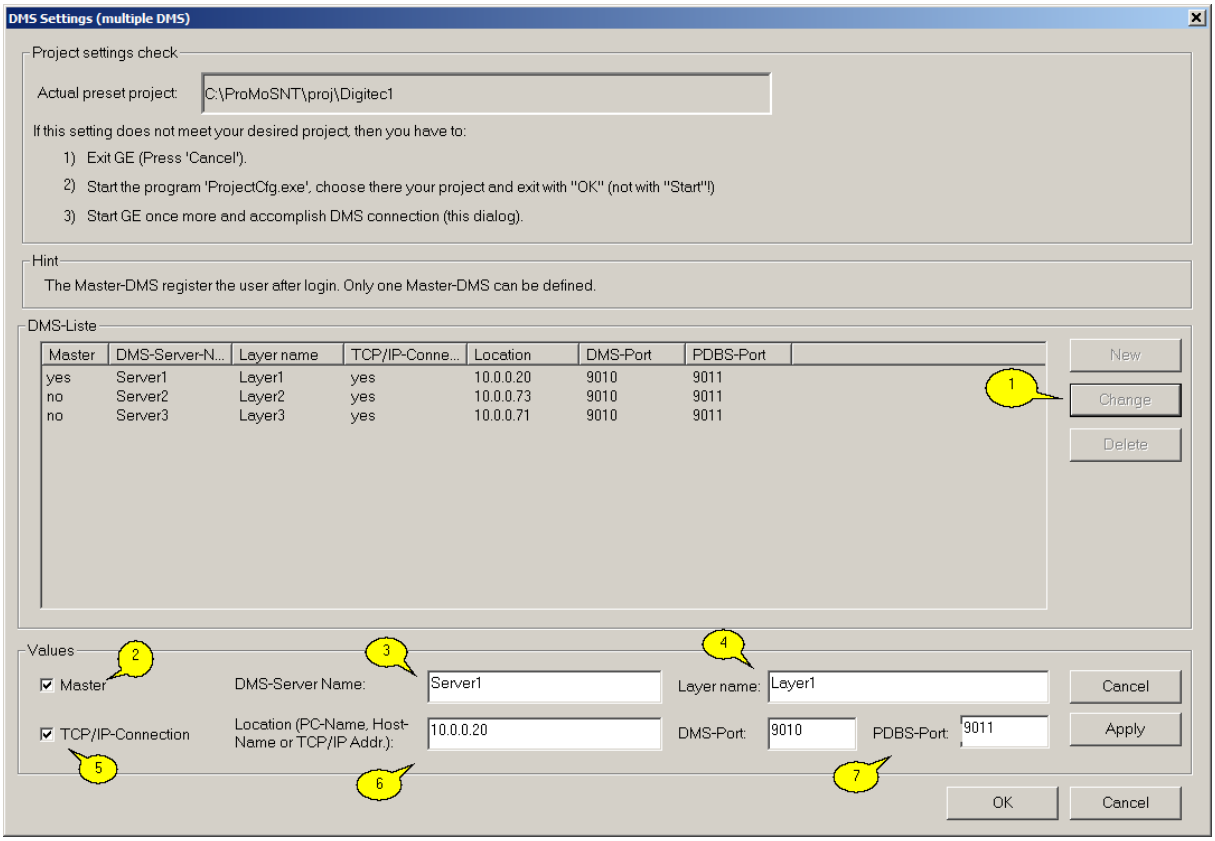


1) Connect to Mulit-DMS:
Mark the checkbox to activate Multi DMS.

When the **<OK>** button is actuated, there will be a prompt whether the DMS connections should be configured.



When the **<Yes>** button is actuated, the configuration window is called up. This window can also be called up in the GE through *Project Settings -> DMS Settings -> Mulit-DMS*.



1) DMS List:
Use <New> to configure a new DMS connection. Use <Change> to modify an existing one and <Delete> to delete a connection.

2) Master:
Activating the master checkbox defines this DMS as the master. In the case of several DMSes, it is mandatory that one DMS take over the master function for managing the data exchange, etc. There can only be 1 DMS in the entire system which functions as the master. All other DMS instances are slaves. If the master DMS drops out during operation, a new master is determined automatically.

3) DMS Server Name:
A DMS server name must be defined here.

4) Layer Name:
A name for the layer must be defined here. The layer name is used for the objects so that the GE knows in which DMS the data points are stored.

5) TCP/IP Connection:
If this checkbox is activated, a TCP/IP connection is established. Otherwise a pipe connection is established.

6) Location:
The IP address of the computer with the corresponding DMS must be entered here.

7) Port Number:
With a TCP/IP connection, the port number must be specified for the DMS and for the PDBS. The default is 9010 for the DMS and for the PDBS, 9011.

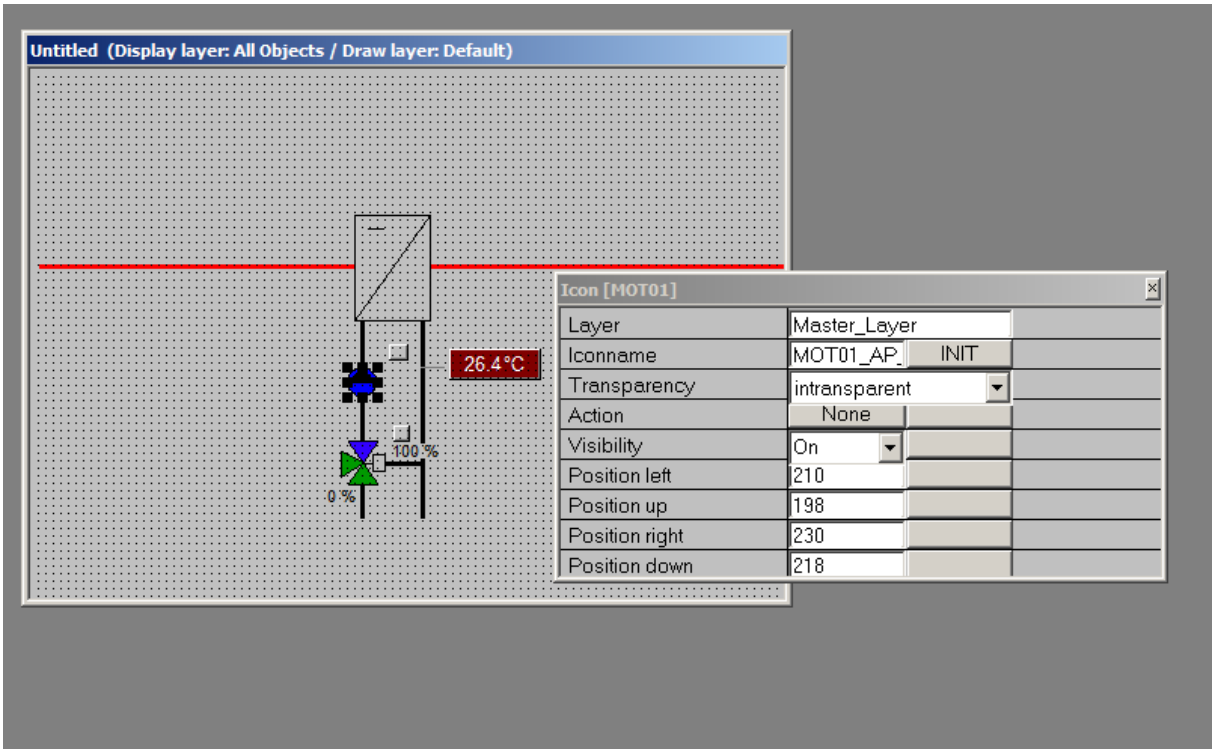
Afterward, the GE must be restarted for the connections to become active.

The dot- sign "." for the connection is only working for Pipe. With TCP/IP it won't work. For local connection under TCP/IP the setting localhost or IP-Address 127.0.0.1 is used.

If the Multi DMS settings were configured in AlmViewer or pChart, the configuration is read from the MultiConn.cfg file at startup. If this was not the case, the GEConn.cfg file, if available, is copied to MultiConn.cfg. Otherwise, the original configuration is copied from the GECon.cfg to the MultiConn.cfg file. In all three cases, the UseCommMultiCfgFile entry in the Settings section is set to 1 in Alm.cfg. GE now uses the configuration from the MultiConn.cfg file. If the MultiConn.cfg file was newly created, it can now be used by AlmViewer and pChart. Therefore, only one configuration file has to be created for all three modules. If an individual configuration is required that is saved in the GEConn.cfg file, the UseCommMultiCfgFile entry must be set to 0.

10.3.16.2 Assign layer

In order for the GE to know in which DMS an object is stored, the layer must be assigned to every object. The layer name is assigned in the configuration menu for several DMSes. In order to assign a layer to an object, the object must be selected and the associated attribute window must be opened.

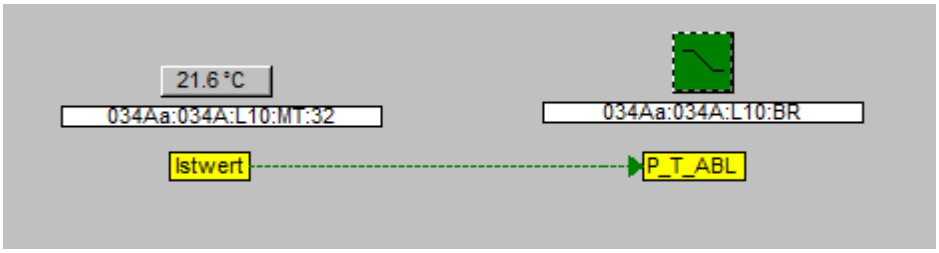


In the example, we have a motor, valve and a temperature measurement. As in the example in the chapter Defining connections, the motor is in the DMS which is located on the local PC. The valve is stored in the DMS on the PC with the IP address 10.0.0.93 and the temperature measurement is in the DMS with the address 10.0.0.92. The name Slave2_Layer must now be entered under Layer in the attribute window of the valve. The name Slave1_Layer must be entered for the temperature measurement. The names are assigned in the configuration menu for several DMSes.

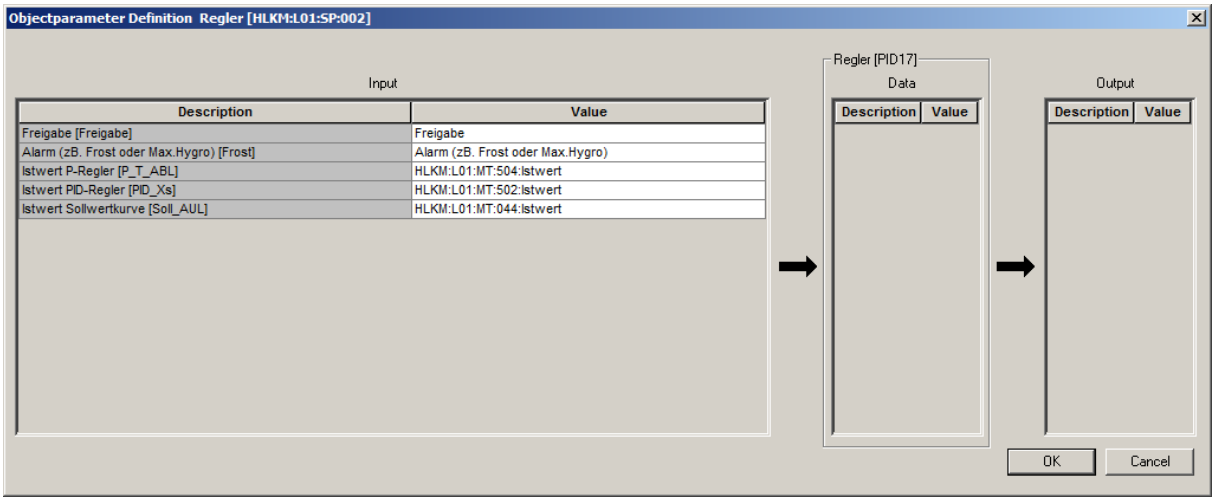
If the error message "Data point does not exist" appears, it may be that the layers are not assigned correctly. It may be possible to correct the problem in the **"Layer -> Generate layer according to PCD"** menu.

10.3.17 Link objects

Link objects visualise the DMS links between the system objects:



Up until now in the GE, these links could only be displayed in tabular form using the object parameter :



The parameters can be edited via a double click on a system object. Using the link tool the system objects can now be linked directly with one another.

10.3.17.1 Views

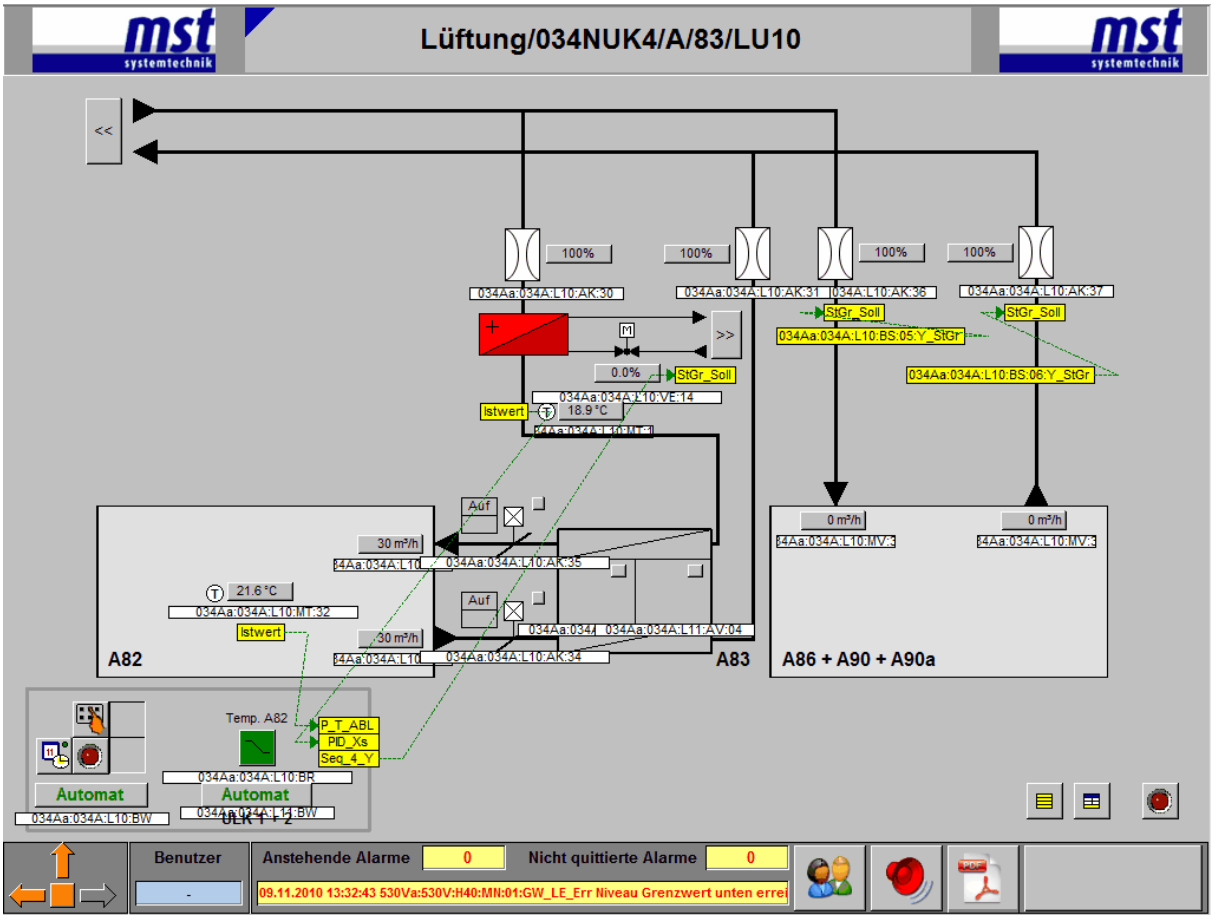
There are two views for link objects. The Design and the Link Boxes views. It is possible to switch between the two views using the toolbar:

Further options for switching views are presented in the chapter Open View.

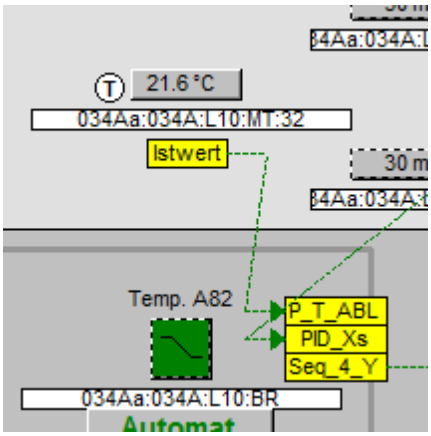
10.3.17.1.1 Design view

The Design view can be opened using the button.

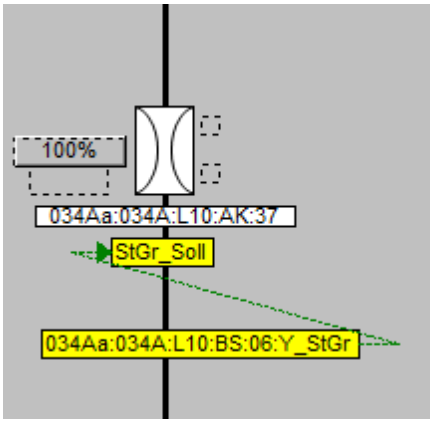
Here, the link objects are shown together with the system objects:



If the system object which the DMS link references is present in the diagram, then, by default, only the attribute is displayed in a yellow rectangle:



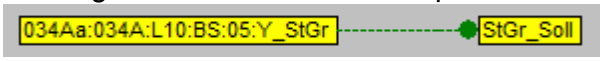
If there is no system object, however, the entire DMS name is shown:



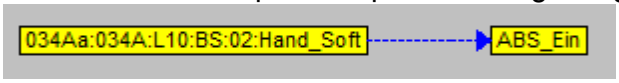
Six signal types can be differentiated based on the standard colours and the arrow tips:

A dark green arrow with a pointed tip stands for an analogue signal:

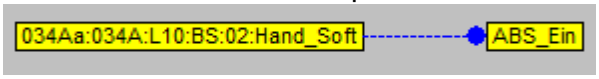
A dark green arrow with a round tip means an inverse analogue signal:



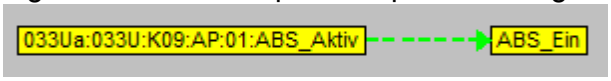
A blue arrow with a pointed tip marks a digital signal with a status of 0:



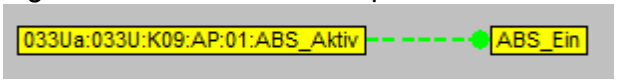
A blue arrow with a round tip is used for an inverse digital signal with a status of 0:



A green arrow with a pointed tip marks a digital signal with a status of 1:



A green arrow with a round tip stands for an inverse digital signal with a status of 1:



The logic can be inverted with a double click on the link object.

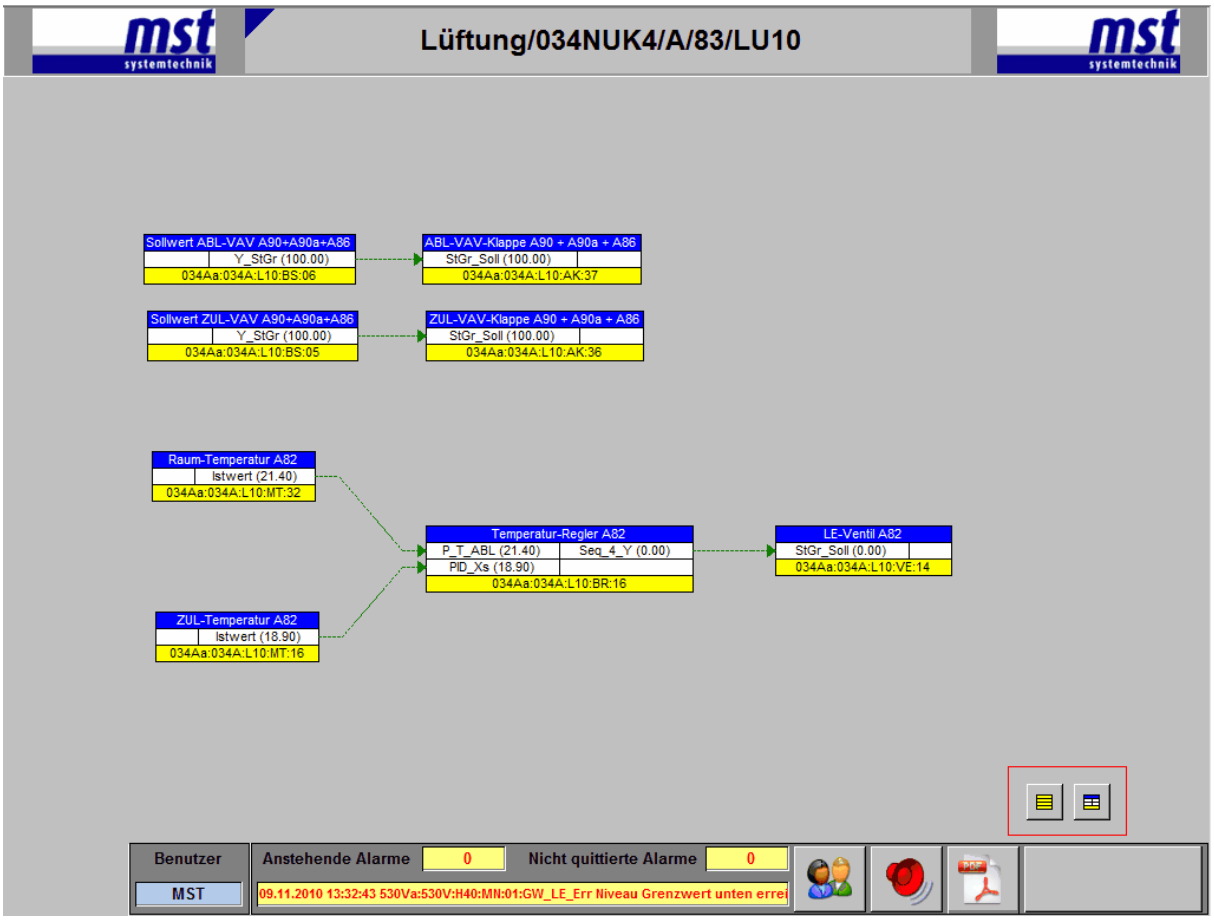
The screen is refreshed if there is a status change of the digital signals to the PLC.

The colours can be changed in the **"Attributes > Specify link object colour"** menu (for more information, see the chapter Changing colours).

10.3.17.1.2 Link Boxes view

The Link Boxes view can be opened using the button.

This view displays only the link boxes together with the objects in the master diagram. The system objects are hidden:



Buttons which are initialised for "System:Node:<PCName>Layer:LINK" or "System:Node:<PCName>Layer:LINKBOXES" are also visible (see the figure to the right below, with a red border). These are required for switching between the two views in runtime mode (see the chapter Open View).

The link boxes consist of the header (blue), the DMS attributes (white) and the footer (yellow).

Header
This shows the designation of the system object. Clicking on the header opens the associated control screen.

DMS attributes
The inputs (PAR_INs) are displayed on the left side. The outputs are shown on the right side. For digital signals, the current value is indicated in parentheses. Inverse signals are marked by an exclamation point (!).

Footer
The DMS name of the system object is displayed in the footer.

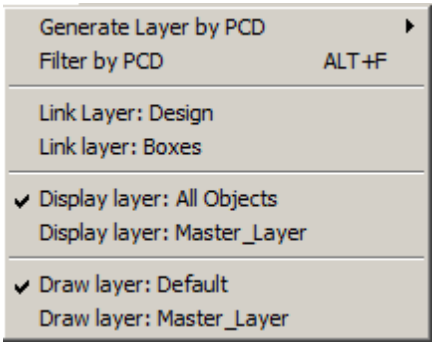
The 6 signal types are displayed using the corresponding arrows already discussed in the Design view.

The screen is refreshed if there is a status change of the digital signals to the PLC or a value change of the analogue signals.

10.3.17.1.3 Open View

1. Edit mode

In the edit mode, a change of the view can be accomplished through the Layer menu:



Layer menu > Link Layer: Design opens the Design View
Layer menu > Link Layer: Link Boxes opens the Link Boxes view

In addition, there is a toolbar available for changing views:

 For the Design view

 For the Link Boxes view

The menu items and the toolbar are only visible when in the **View > Toolbars** menu, the item **Link** is activated.

2. Runtime mode or Emulation mode (Ctrl pressed)

Changing the view can be controlled via buttons or icons when these are initialised accordingly.

The objects of the Design view are stored on the layer with the name LINK. The link boxes are stored on the layer with the name LINKBOXES.
In the DMS there is a data point available for both views:

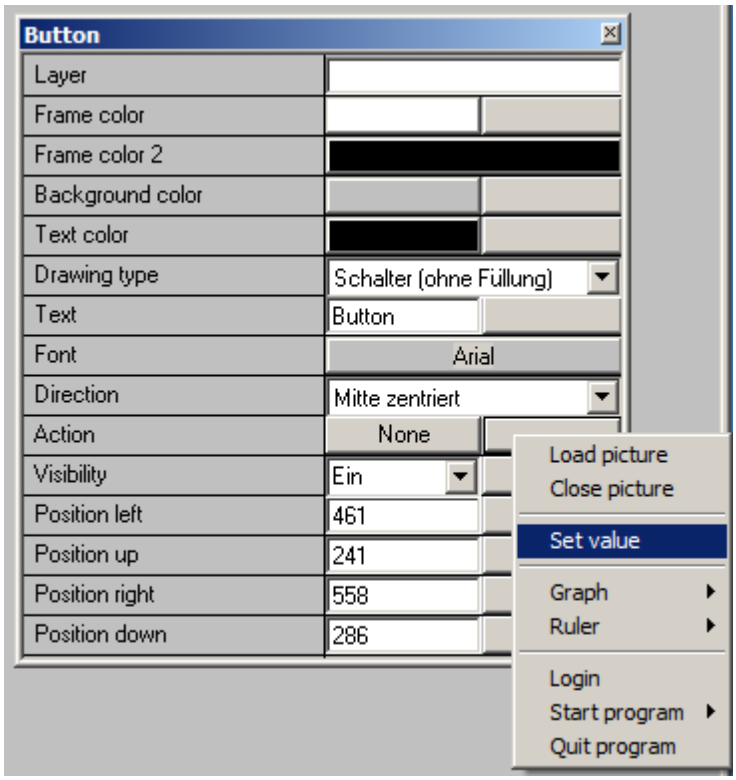
System:Node:<PCName>Layer:LINK	For the Design view
System:Node:<PCName>Layer:LINKBOXES	For the Link Boxes view

PC Name = Name of the stationary PC

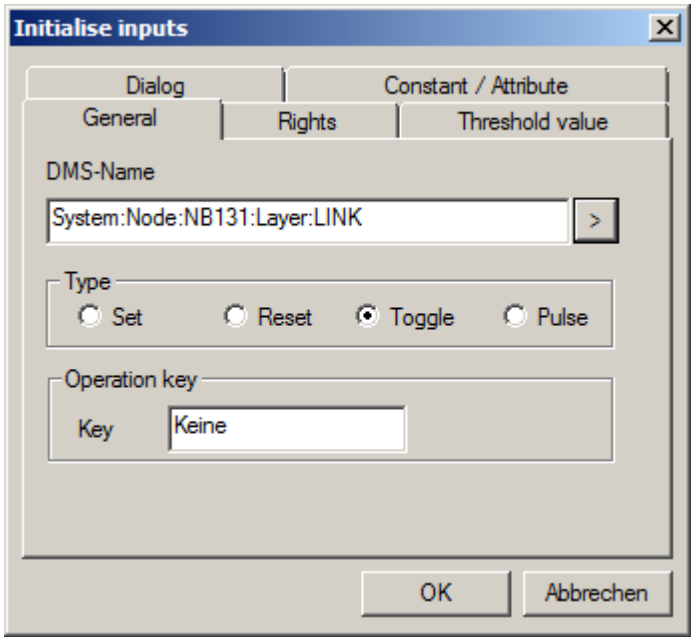
If the layer data point is set in the DMS, the corresponding view is displayed.

The two buttons with a red border in the screen from the chapter Link Boxes view must be initialised as follows:

1. In the Properties dialog, choose the **"Set Value"** action:



2. In the "Value Entry Initialisation" dialog,

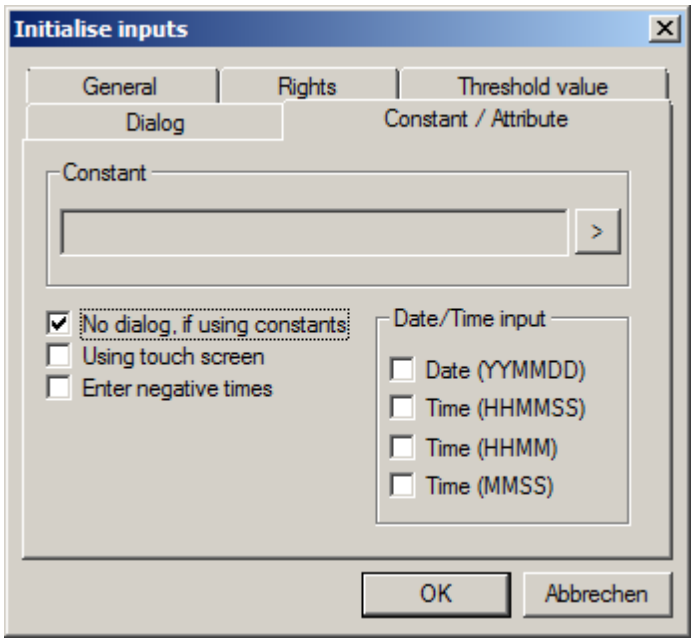


enter the DMS name from the following table and select the Toggle set type.

Button	DMS name	Set Type
	System:Node:<PCName*>Layer: LINK	Toggle
	System:Node:<PCName*>Layer:LINKBOXES:	Toggle

*PCName = Name of the stationary PC

3. On the Constants / Attributes tab, activate the "No Dialog for Constants" option.



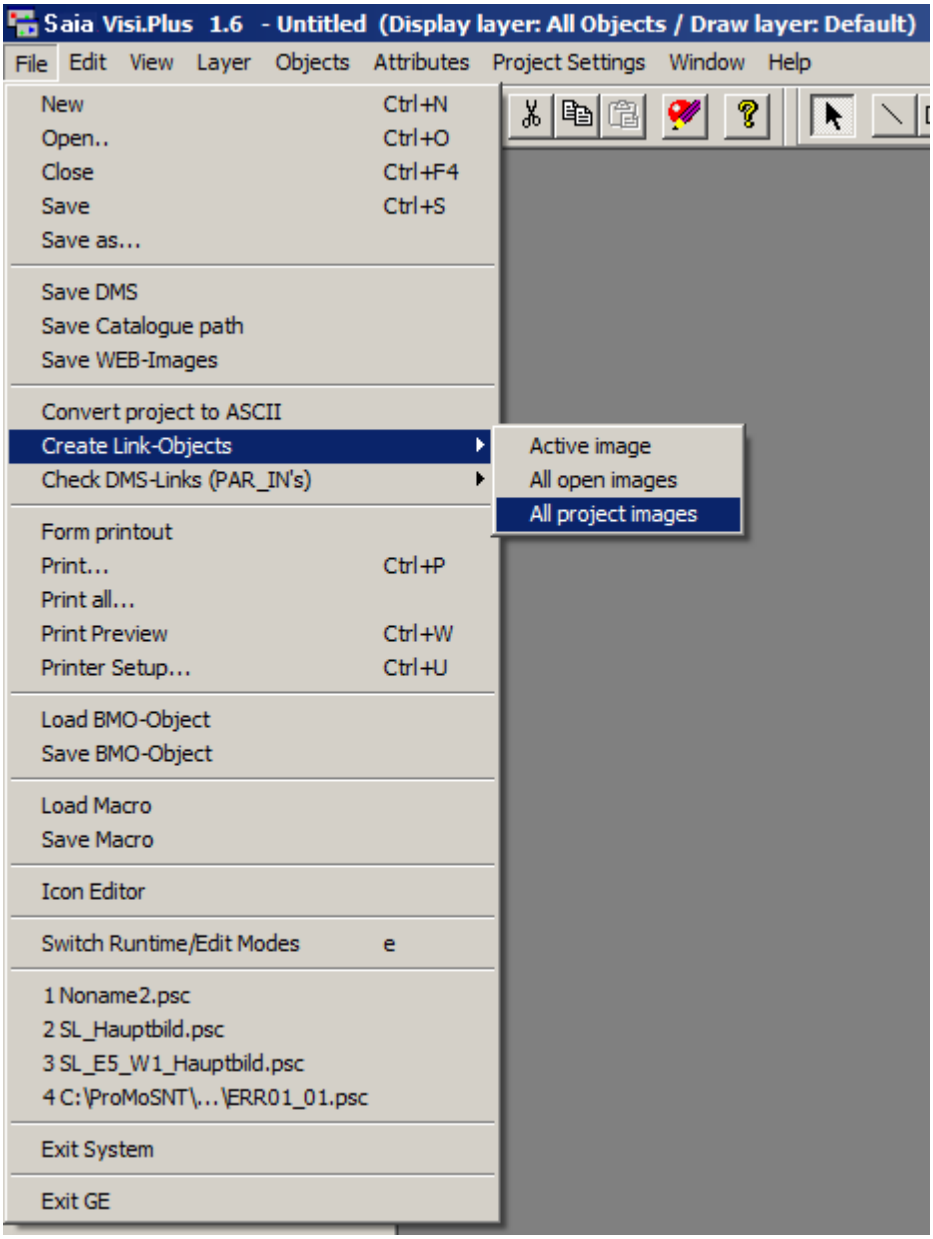
A click on  now opens the Design view. The link boxes can be displayed by clicking on

10.3.17.2 Create link objects

10.3.17.2.1 Automatic creation

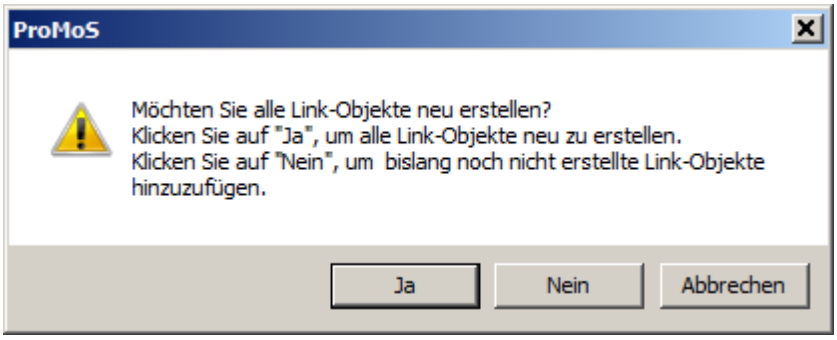
If your project already contains DMS links, these can be created automatically.

To do so, you must select the item "Create link objects" in the File menu:



In the pop-up menu you then have the opportunity to have link objects created for the **Active diagram**, for **All open diagrams** or for **All diagrams of the project.**

Once you have made a selection, the following message window will be shown:



Yes button

If old link objects exist, they will be deleted and the new link objects will be created.

This is recommended:
- If you have already worked with an earlier 1.6 version, i.e. <1.6.1.122, because, unfortunately, the creation of link objects there was faulty.
- If you have switched from version 1.5 to version 1.6.

No button

In this case:
- link objects that have not yet been created are added,
- if DMS links have been deleted, e.g. the link objects in another diagram. The positions of the unaffected link objects are retained.

Cancel button

The action is cancelled. Link objects are not created.

Once you have clicked either the **Yes** or **No** button in the message window, the creation of the link objects begins. In the event that you have chosen to have the link objects created for all project diagrams, you can observe the progress on the progress bar which is shown:

During creation, a check is made whether the data point referenced by the DMS link (PAR_IN) still exists in the DMS. If this is not the case, the error is subsequently displayed in a message window (see the chapter Checking DMS links).

The link objects are then only visible if the **View > Toolbars** menu has the **Link** item activated.

The GE automatically switches to the Design view before creating the link objects.

10.3.17.2.2 Graphical editor

GE offers the option of creating DMS links directly on the screen.

The link tool is available to you for this purpose. It is located on the Drawing Tools toolbar:

The link tool can only be selected if the Design view or the Link Boxes view is activated (see the chapter Open View).

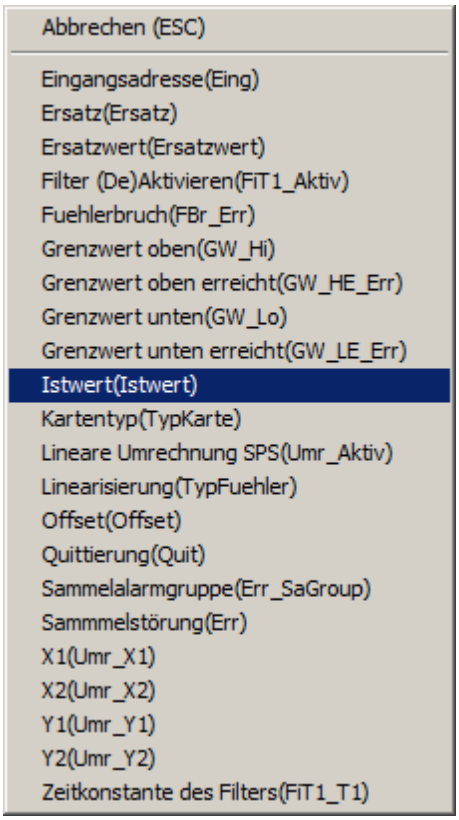
How to use the tool will be explained first. Then, several simple examples follow.

Examples 1 - 4 explain creation in the Design view. Example 5 explains how a DMS link can be realised in the Link Boxes view.

Using the  link tool, you can link the PCD output of a BMO object to the PCD input of another BMO object.

To do so:

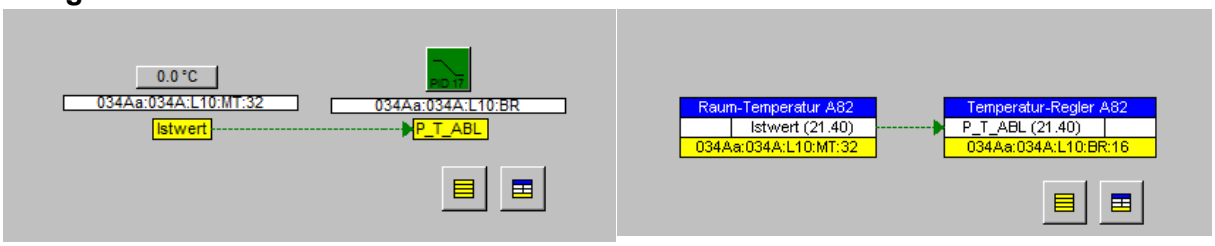
1. In the Design view, select the link tool  from the toolbar.
2. Click on the first BMO object with the left mouse button. A pop-up menu will open from which the output can be selected.
   The following figure shows the pop-up menu of an MES01 object as an example:



3. Click on the second BMO object with the left mouse button. Select the input from the pop-up menu.

This results in the following two views (see example 1):

Design view **Link Boxes view**



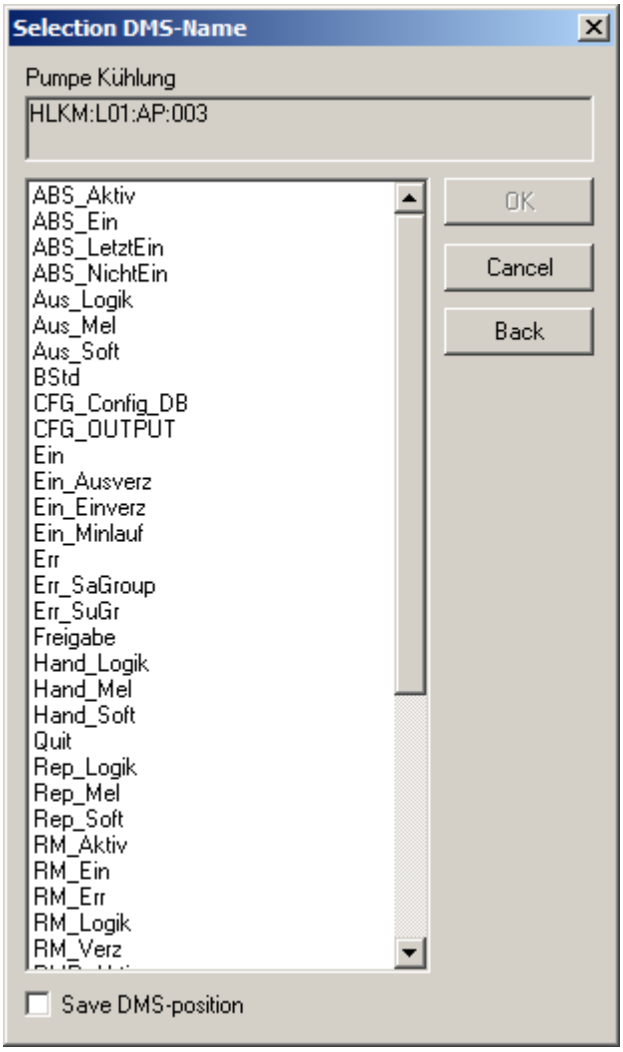
The data point PAR_IN was created in the DMS for the second object (see example 1).

Now you can create further links. This can be done both in the Design and the Link Boxes views.

In the Design view, the outputs and inputs of the same object are automatically grouped together in blocks. These can be moved either individually or together with the object. Further details on this can be found in the chapter Moving link objects.

The pop-up menu is explained in more detail in the later chapter Pop-up menu:

BMO objects can also be linked with each other if they are not on the same screen. To do this, you must click on a free area in the screen with the link tool when creating the link. The dialog box for selecting a DMS name will then open:

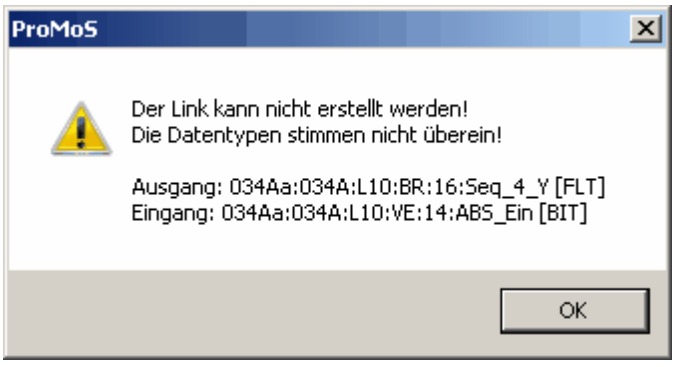


The input or output can then be selected here.

The DMS link is only created if the data type of the output matches that of the input.

This is guaranteed to be the case if the link objects are created by clicking on the respective system object as explained in the previous example. It is not guaranteed, however, if a DMS name can be freely selected as in this case. Here, the data types of the input and output are compared to each other.

If the data types differ, an error message is generated:



The link objects are not created in this case.

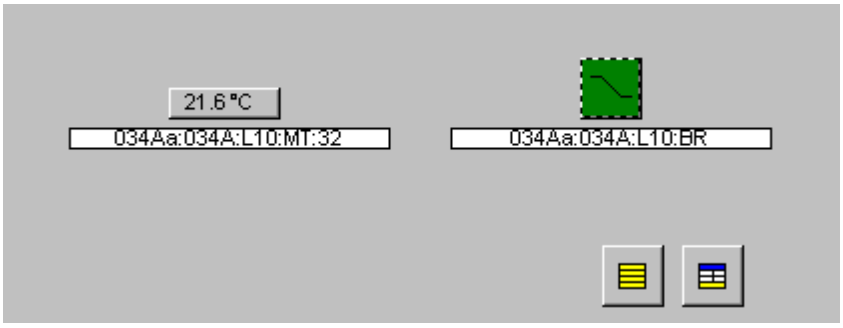
Example 1:

Objective
The actual value of a measurement object, derived from MES01, is to control the actual value P controller input (P_T_ALB) of a PID controller, derived from PID17.

Solution

Step 1: Create the BMO object.
1. Load an MES01 object from the library or the catalog and reinitialise it.
If the measurement object contains DMS links, delete them in the Parameters dialog (Input column).
2. Load a PID17 object from the library or the catalog and reinitialise it.
If the PID17 object contains DMS links, delete them in the Parameters dialog (Input column).
3. Create two buttons which make it possible to switch between the Design and the Link Boxes views (see the chapter Open View).

The screen should look about like the following:

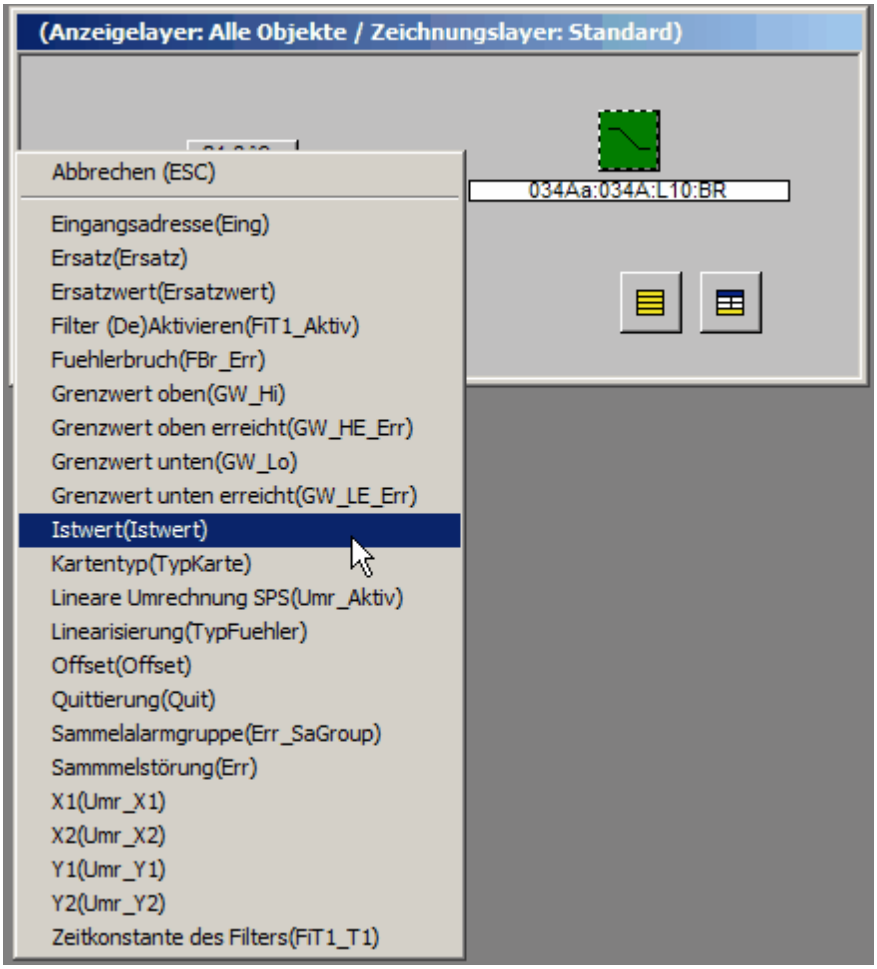


Step 2: Link the BMO object

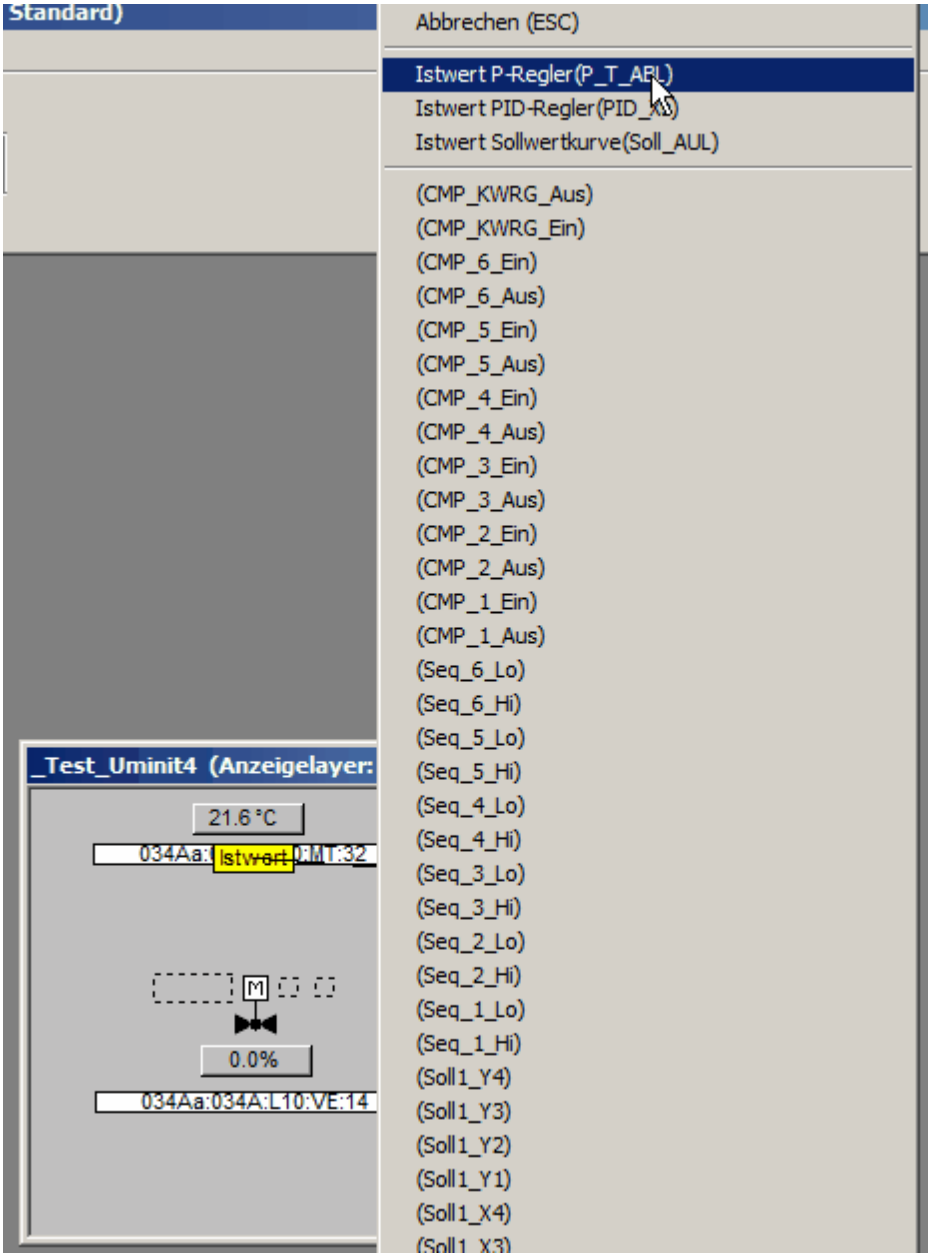
1. Select the Design view. Do so by clicking on  either in the toolbar or in the screen.

2. Click on  in the Drawing Tools toolbar.

3. Click on the measurement object with the left mouse button and select the item ActualValue(ActualValue) in the pop-up menu.
The pop-up menu is explained in full in the chapter Pop-up menus.



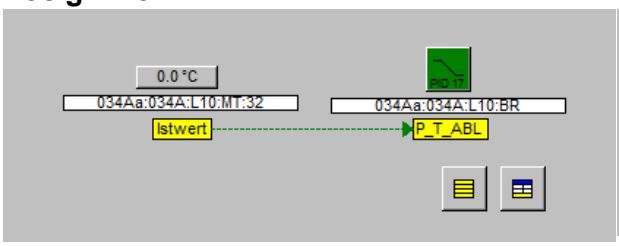
4. Click on the PID object with the left mouse button and select the item Actual Value P Controller (P_T_ALB) in the pop-up menu:



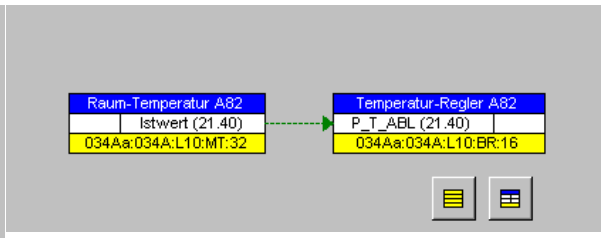
The creation process can be cancelled at any time by pressing the ESC key.

The two views should then look as follows:

Design view



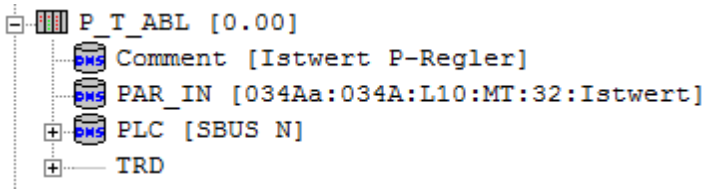
Link Boxes view



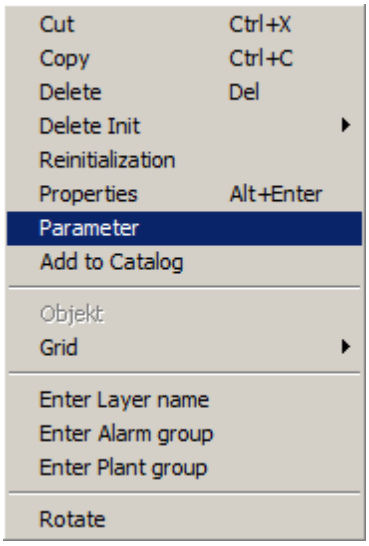
As the example shows, both the link objects in the Design view and the link boxes in the Link Box view are created.

The newly created link objects can be repositioned. Further details on this can be found in the chapter Moving link objects.

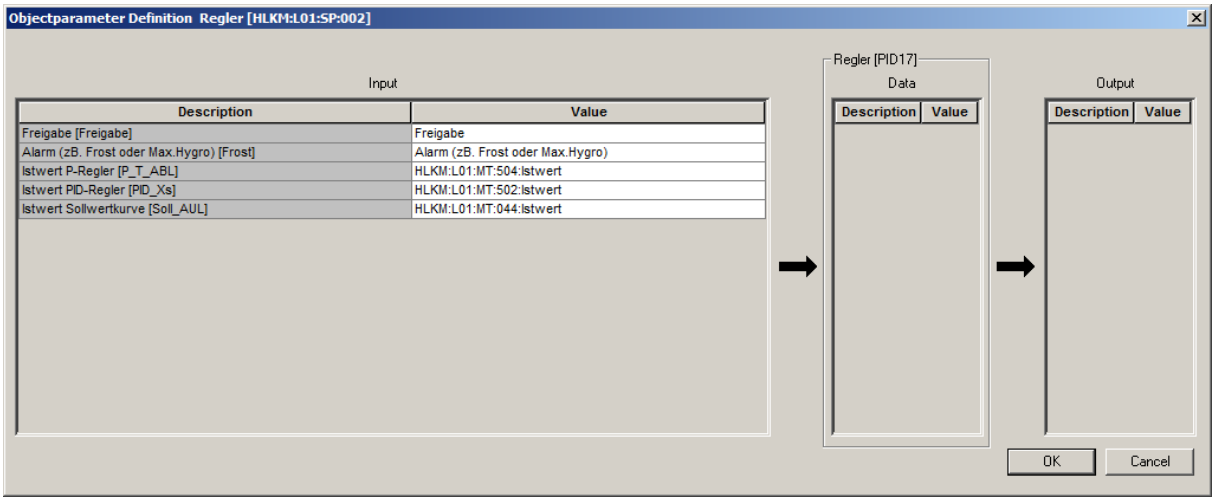
The PAR_IN data point of the PID object is also created in the DMS:



The DMS links can also be displayed in table form. To do so, click on the object with the right mouse button and select the item "Parameters" in the pop-up menu:



The following window opens:

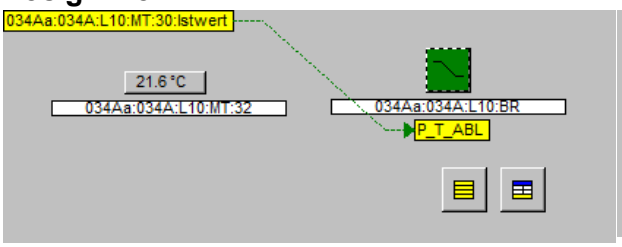


The DMS link created is displayed on the first line. It can be edited directly in the table. For the P_T_ABL input, the DMS name of another measurement object should be entered. In the present example, "034Aa:034A:L10:MT:30:ActualValue" has been entered.

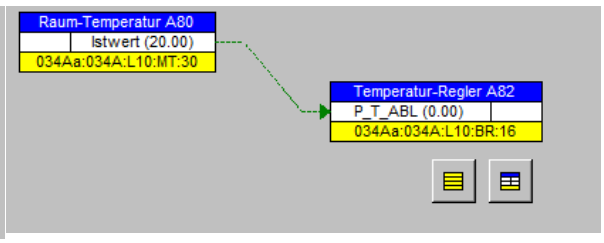
If the window is exited using the OK button, the link objects are subsequently created again new, provided that the DMS data point which the link references exists.

The change made above then results in the following views:

Design view



Link Boxes view

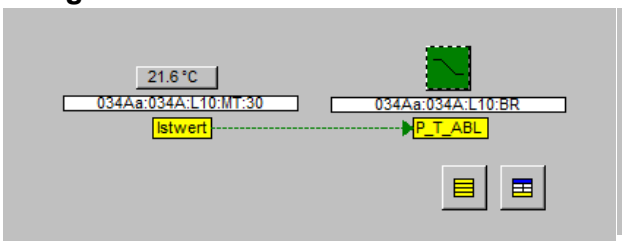


Because the measurement object "034Aa:034A:L10:MT:30" is missing, the complete DMS name is indicated in the Design view. In both views, the link objects are placed in the upper left corner.

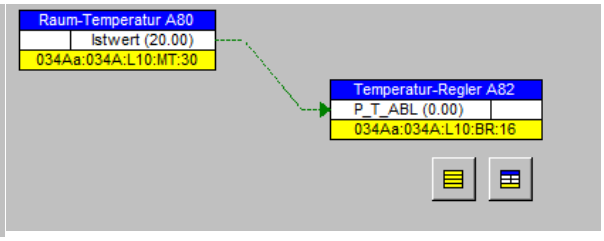
If an invalid DMS name is entered in the parameter table, the error is displayed in a message window after exiting the Parameters view (see the chapter on Checking DMS links).
The link objects and the DMS link are not created in this case.

If the measurement object is reinitialised from "034Aa:034A:L10:MT:32" to "034Aa:034A:L10:MT:30", the link objects are refreshed in the Design view:

Design view



Link Boxes view



The link object "ActualValue" is placed below the measurement object "034Aa:034A:L10:MT:30". The Link Boxes view does not change.

A further example of reinitialising can be found in the chapter Reinitialising objects.

Example 2:

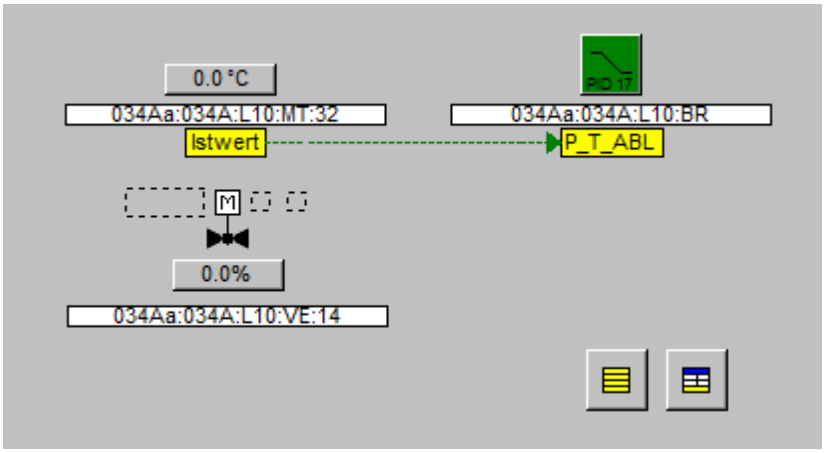
Objective
This example is intended to apply the screen created in example 1. In addition, control signal Seq. 4 (Seq_4_Y) of the PID controller should control the TargetValue(StrGr_Soll) of a valve which is derived from VEN01.

Solution

Step 1: Create the valve

1. Select the Design view. Do so by clicking on  either in the toolbar or in the screen.
2. Load a VEN01 object from the library or the catalog and reinitialise it.
If the valve contains DMS links, delete them in the Parameters dialog (Input column).

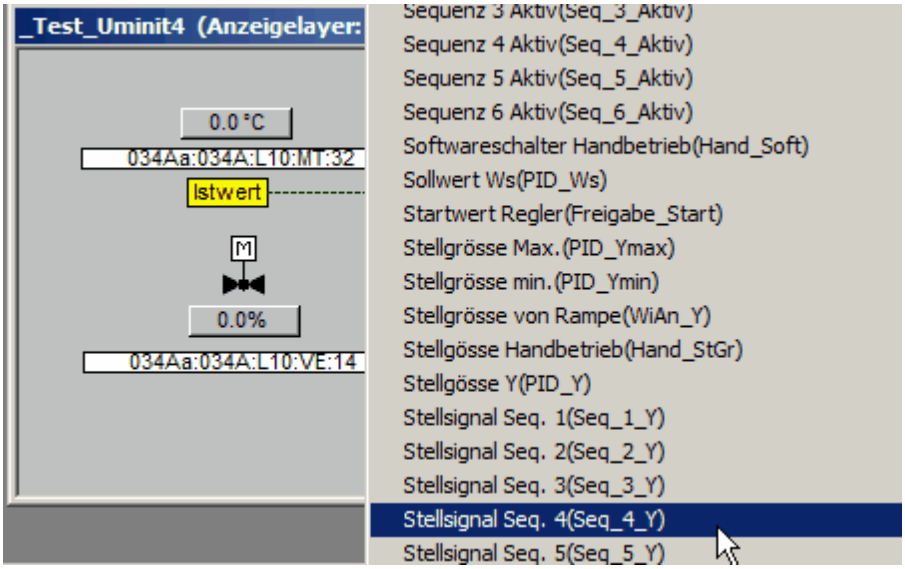
The image should then look about like this:



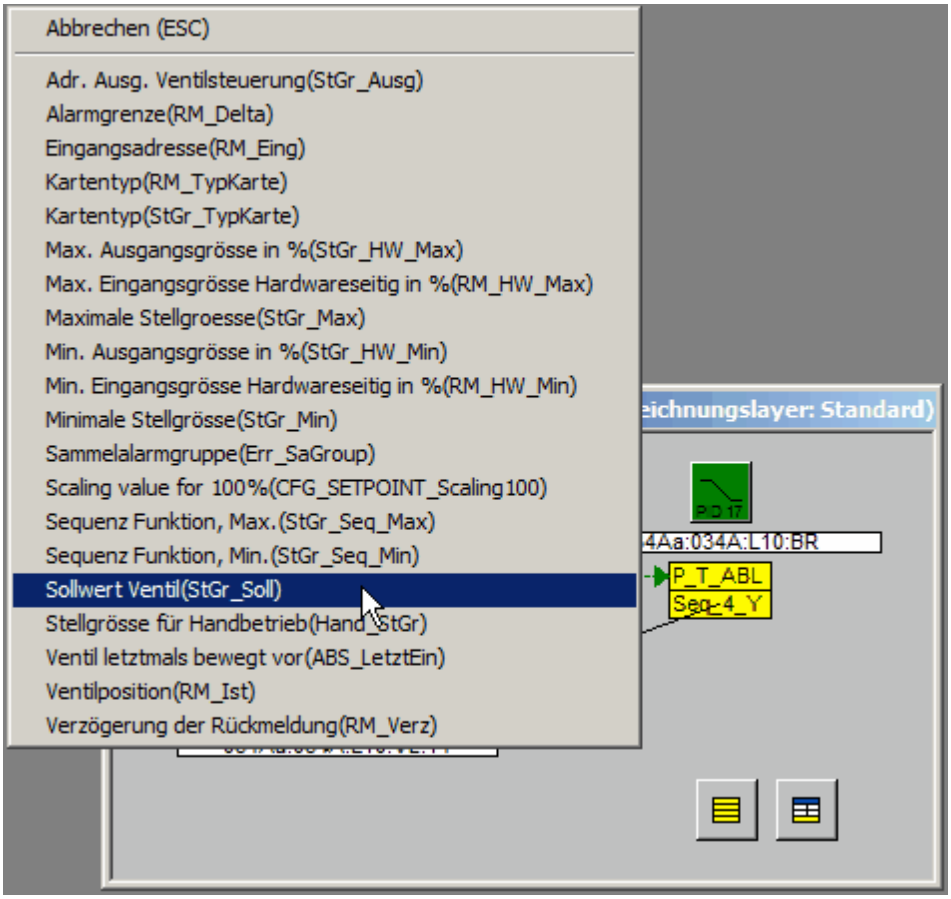
Step 2: Link the BMO object

1. Click on  in the Drawing Tools toolbar.

2. Click on the PID object with the left mouse button and select "Control signal Seq. 4 (Seq_4_Y)" in the pop-up menu:

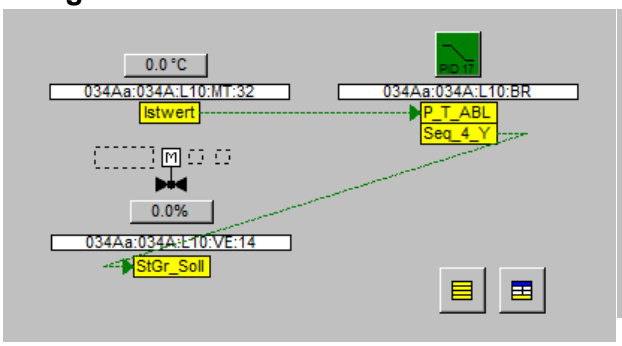


3. Click on the valve object with the left mouse button and select "Valve TargetValue(StGr_Soll)" in the pop-up menu:

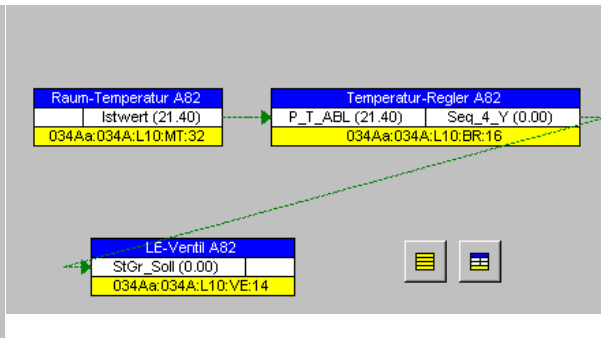


This produces the following result:

Design view



Link Boxes view



As can be seen, the DMS link for the PID controller is displayed in a common box in both views.

Example 3:

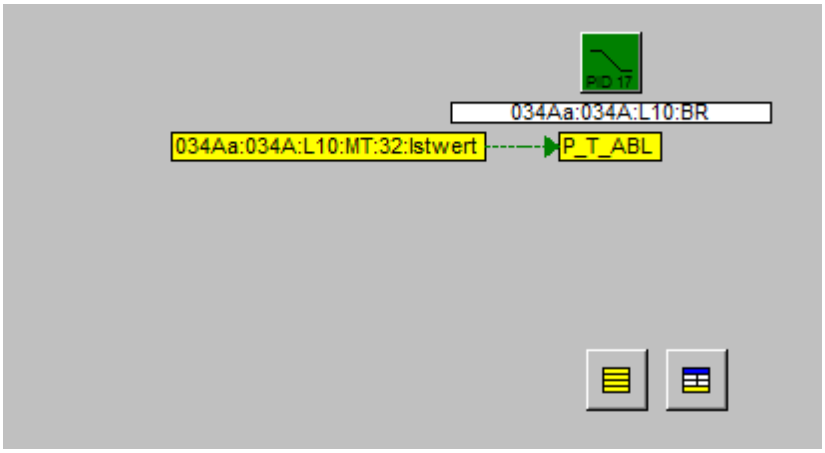
Objective:
Example 2 will be continued. The screen needs to be changed so that it only contains the PID controller without the links created in examples 1 and 2. The DMS links from example 1 are to be created, i.e. the actual value of a measurement object should control the actual value P controller input (P_T_ALB) of the PID controller.

Solution:

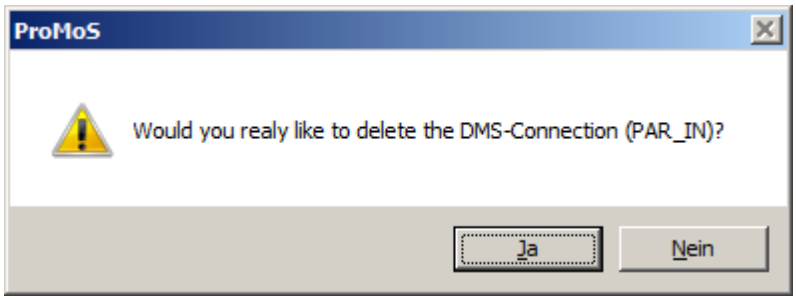
Step 1: Delete the measurement object and the valve as well as all links

1. Select the Design view by clicking on  either in the toolbar or in the screen.
2. Select the measurement object and the valve together.
3. Press the Delete key to delete the selected objects.

The screen now only contains the PID controller and the DMS link to the measurement object:

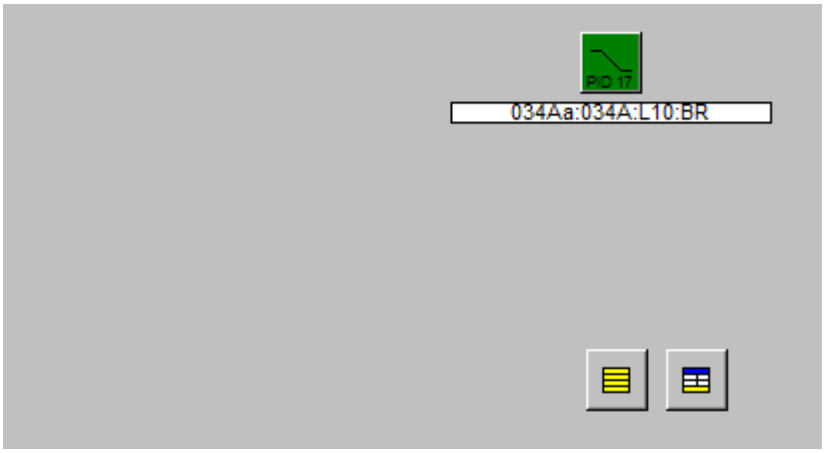


4. Click on the link object "Actual value" or on "P_T_ABL".
5. Press the Delete key to delete the DMS link.
6. Answer the subsequent question



with Yes.

Now the initial state for subsequently performing linking has been established:

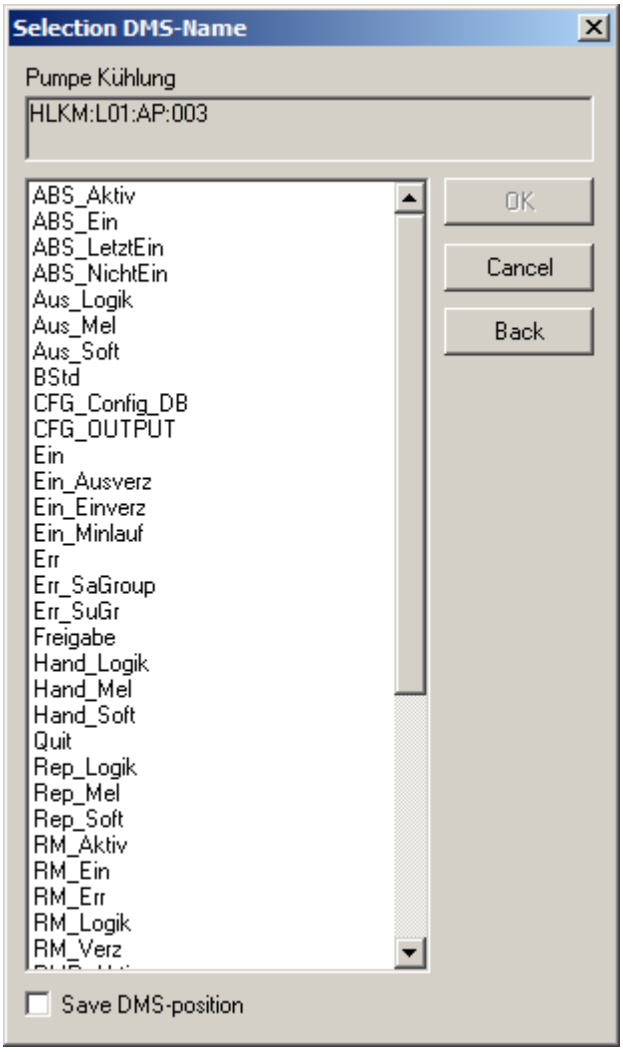


Deleting BMO and link objects is treated fully in the chapter Deleting objects.

Step 2: Link the BMO object

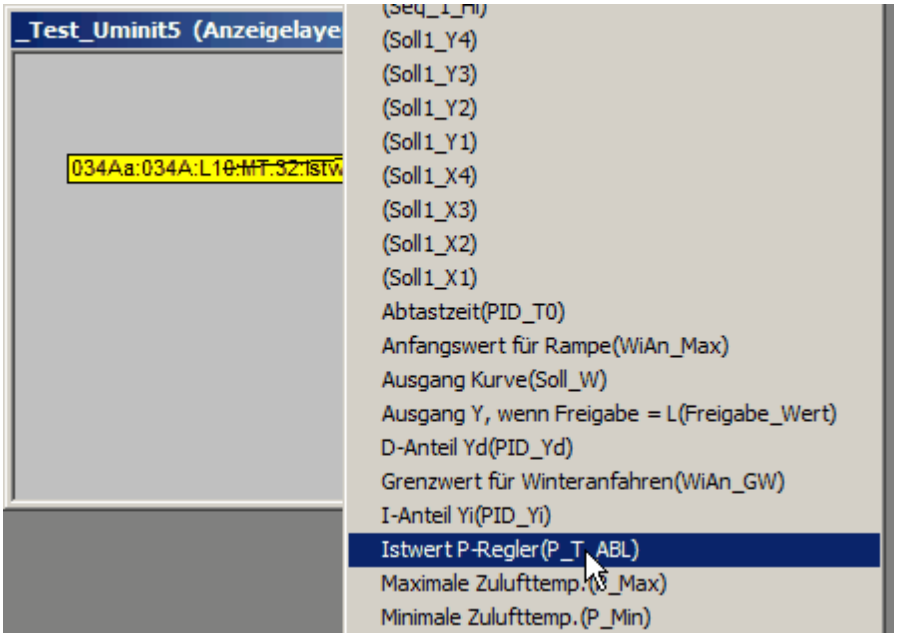
1. Click on  in the Drawing Tools toolbar.

2. Click on a free area in the screen with the left mouse button. The selection dialog for DMS names then appears. Scroll to the measurement object and then select "Actual value":



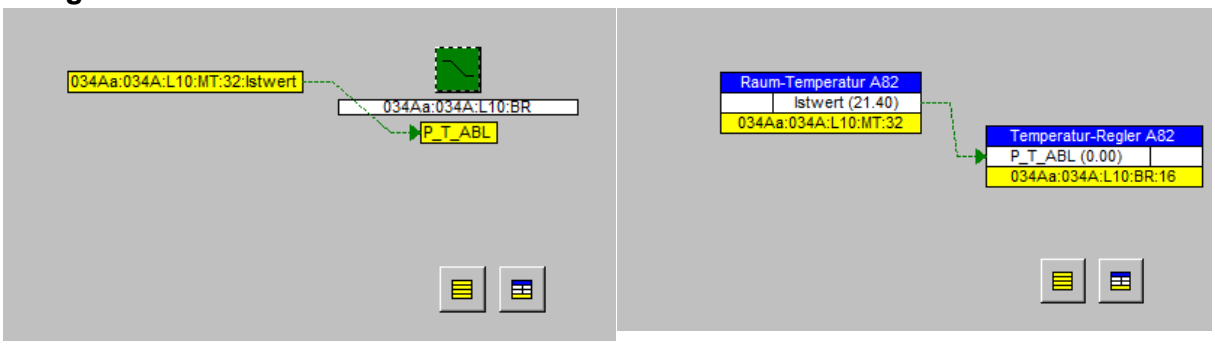
3. After making the selection, exit the dialog with OK.

4. Click on the PID object with the left mouse button and select "Control signal Seq. 4 (Seq_4_Y)" in the pop-up menu:



The following two screens should now have been created:

Design view **Link Boxes view**



The link boxes match those of example 1. Now the full DMS name is displayed in the Design view for the actual value link object.

Example 4:

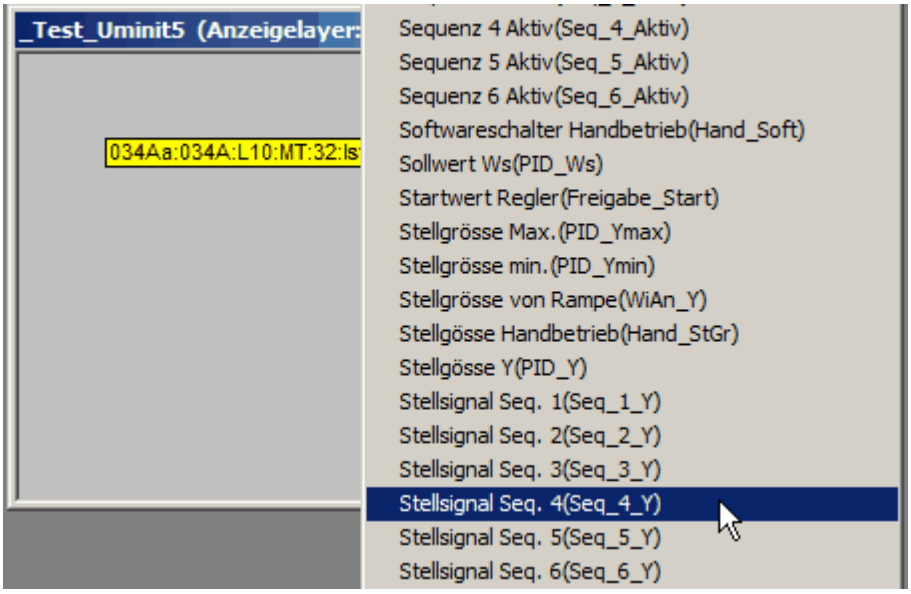
Objective
This example is based on the last screen from example 3. Control signal Seq. 4 (Seq_4_Y) of the PID is now to control the TargetValue(StrGr_Soll) of a valve (DMS link from example 2). The link is to be realised without creating the valve in the screen.

Solution

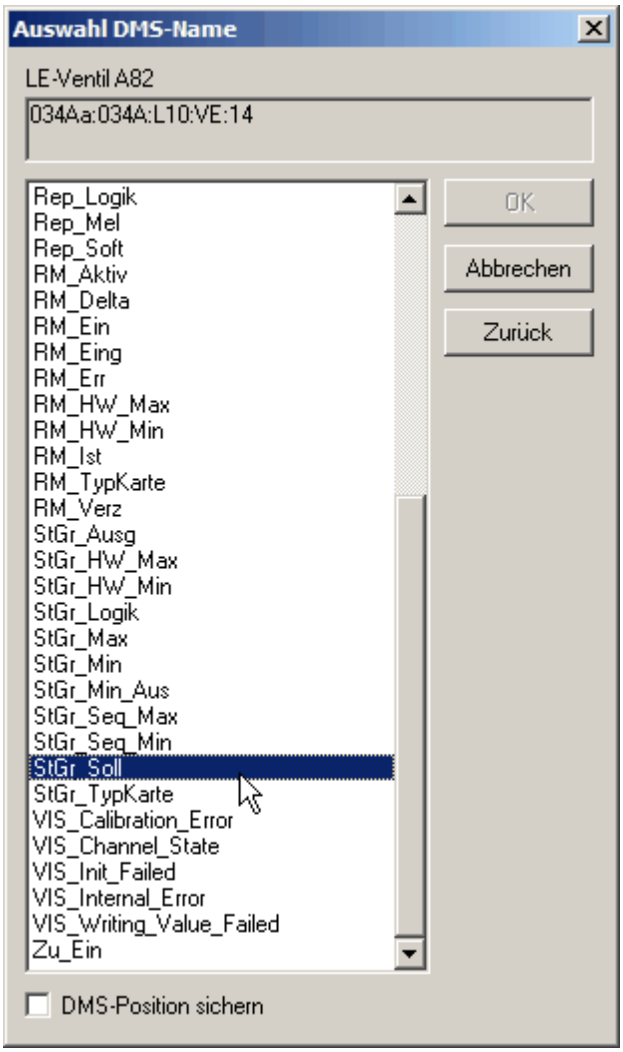
1. Select the Design view. Do so by clicking on  either in the toolbar or in the screen.

2. Click on  in the Drawing Tools toolbar.

3. Click on the PID object with the left mouse button and select "Control signal Seq. 4 (Seq_4_Y)" in the pop-up menu:



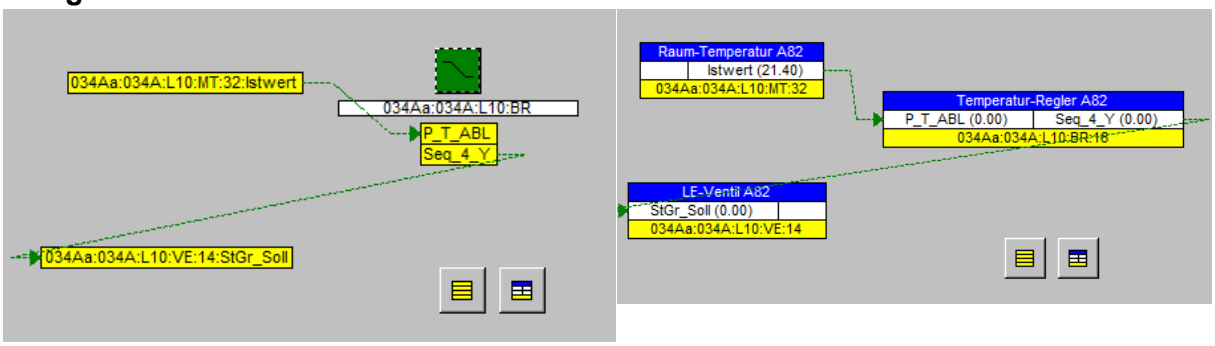
4. Click on a free area in the screen with the left mouse button. The selection dialog for DMS names then appears. Here, select the TargetValue(StrGr_Soll) of the valve:



5. After making the selection, exit the dialog with OK.

This results in the following two views:

Design view **Link Boxes view**



The link boxes match those of example 1. Now the full DMS name is displayed in the Design view for the StGR_Soll link object.

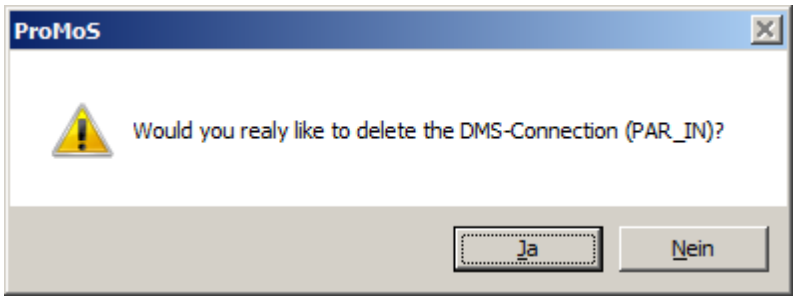
Example 5:

Objective
The link between the valve and the PID controller in the last screen is to be deleted and then created again in the Link Boxes view.

Solution

Step 1: Delete the DMS link in the Link Boxes view

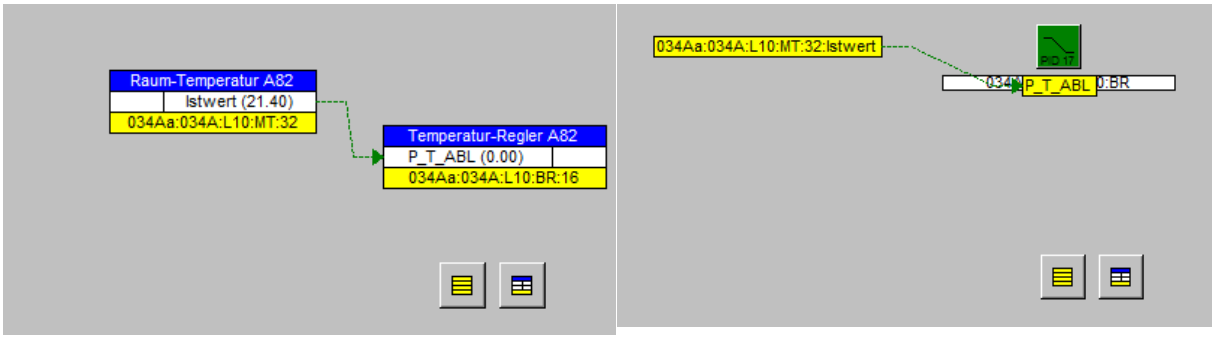
1. Switch to the Link Boxes view . Do so by clicking on  either in the toolbar or in the screen.
2. Click on the entry StGr_Soll in the link box of the valve or select the entry Seq_4_Y for the PID controller.
3. Press the Delete key to delete the DMS link.
4. Answer the subsequent question



with Yes.

This results in the following two views:

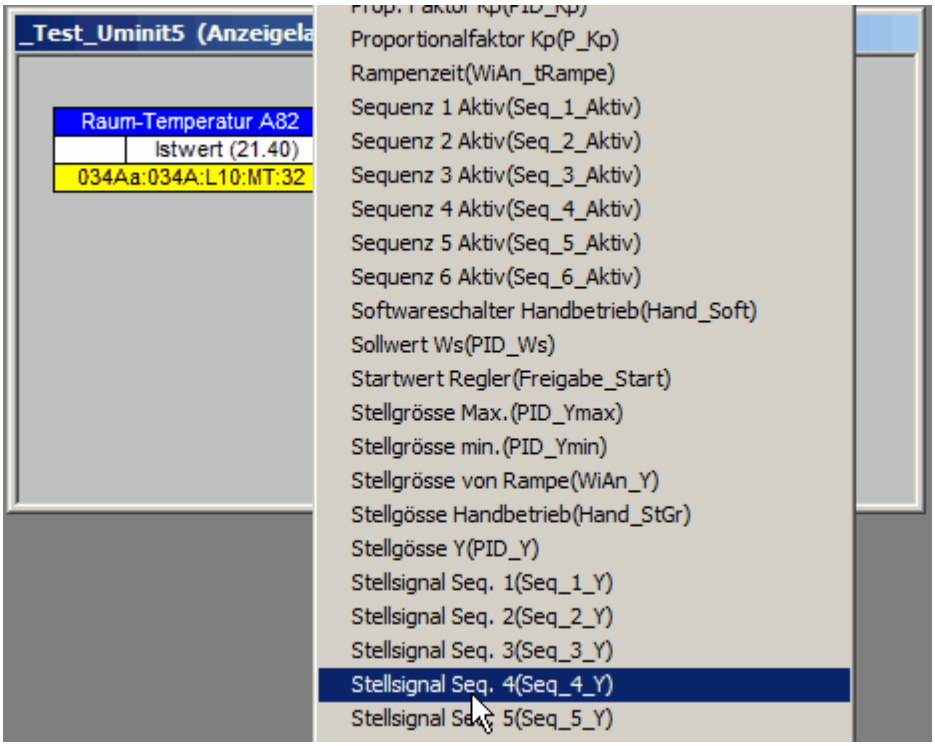
Design view **Link Boxes view**



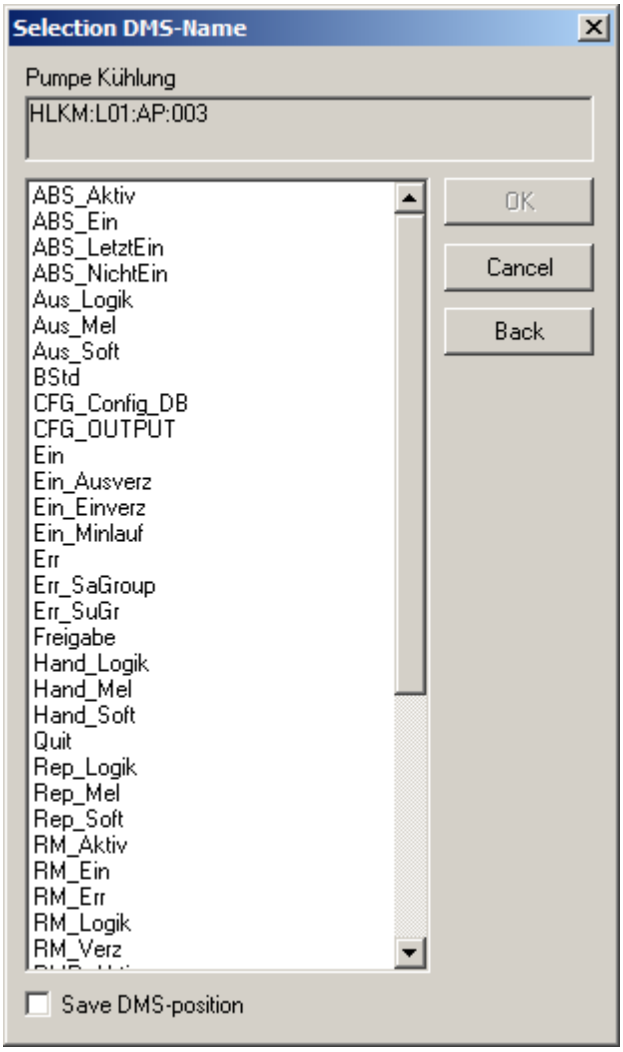
Step 2: Link the BMO object

1. Click on  in the Drawing Tools toolbar.

2. Click on the PID object of the link box with the left mouse button and select "Control signal Seq. 4 (Seq_4_Y)" in the pop-up menu:



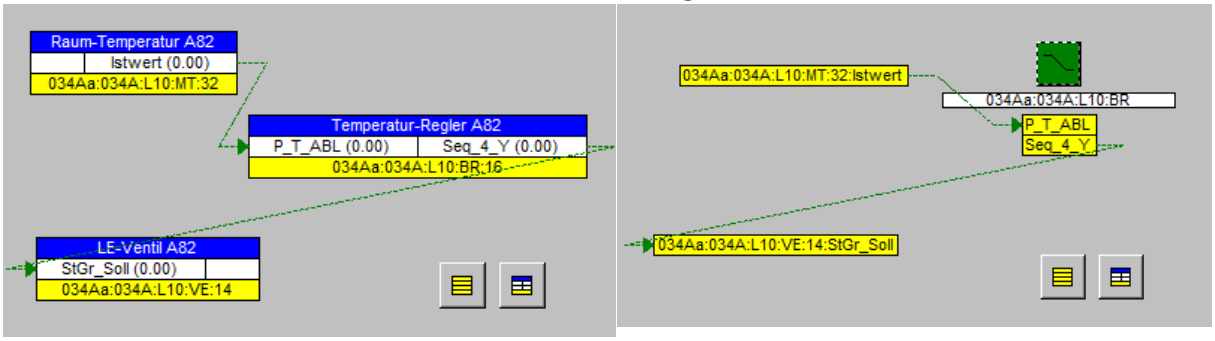
3. Click on a free area in the screen with the left mouse button. The selection dialog for DMS names then appears. Here, select the TargetValue(StrGr_Soll) of the valve:



4. After making the selection, exit the dialog with OK.

This will produce the same result as in example 4:

Link Boxes view **Design view**



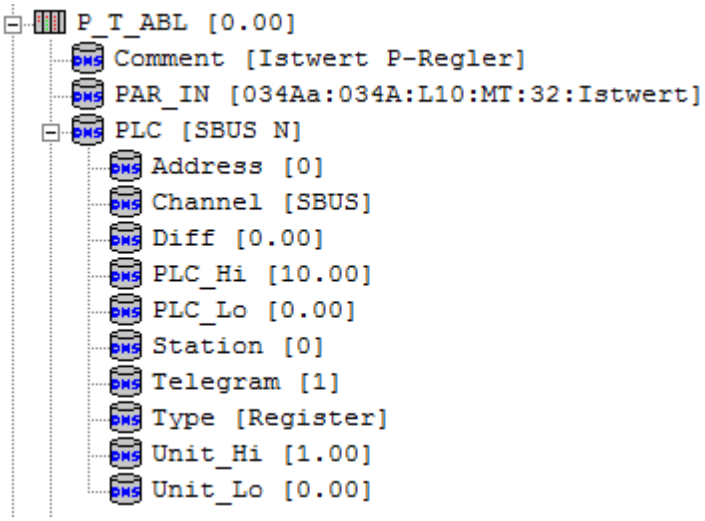
10.3.17.2.2.1 Pop-up menus

It is important to distinguish here between the pop-up menu of the output object and that of the input object.

The only item they have in common is the uppermost item, "Cancel (ESC)". When this is

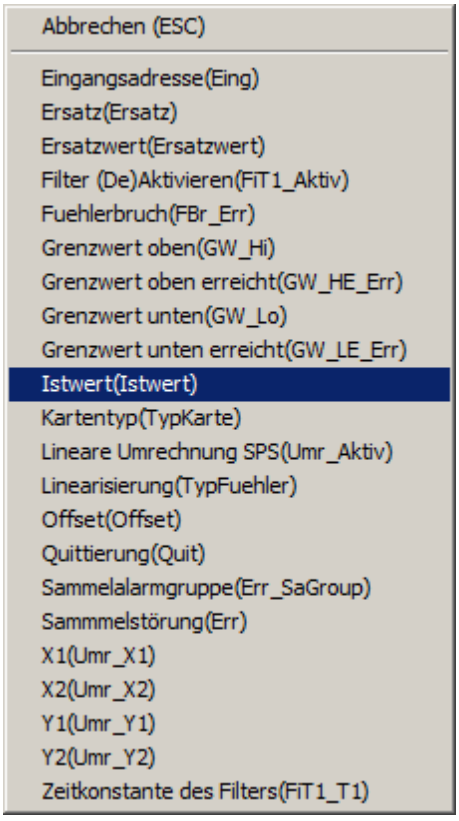
clicked or the ESC key is pressed, the link is cancelled.

After a hyphen, a list of the signals follows which are communicated on the PCD. These signals can be recognised in the DMS based on the attribute "PLC" (one level further down than the signal itself):



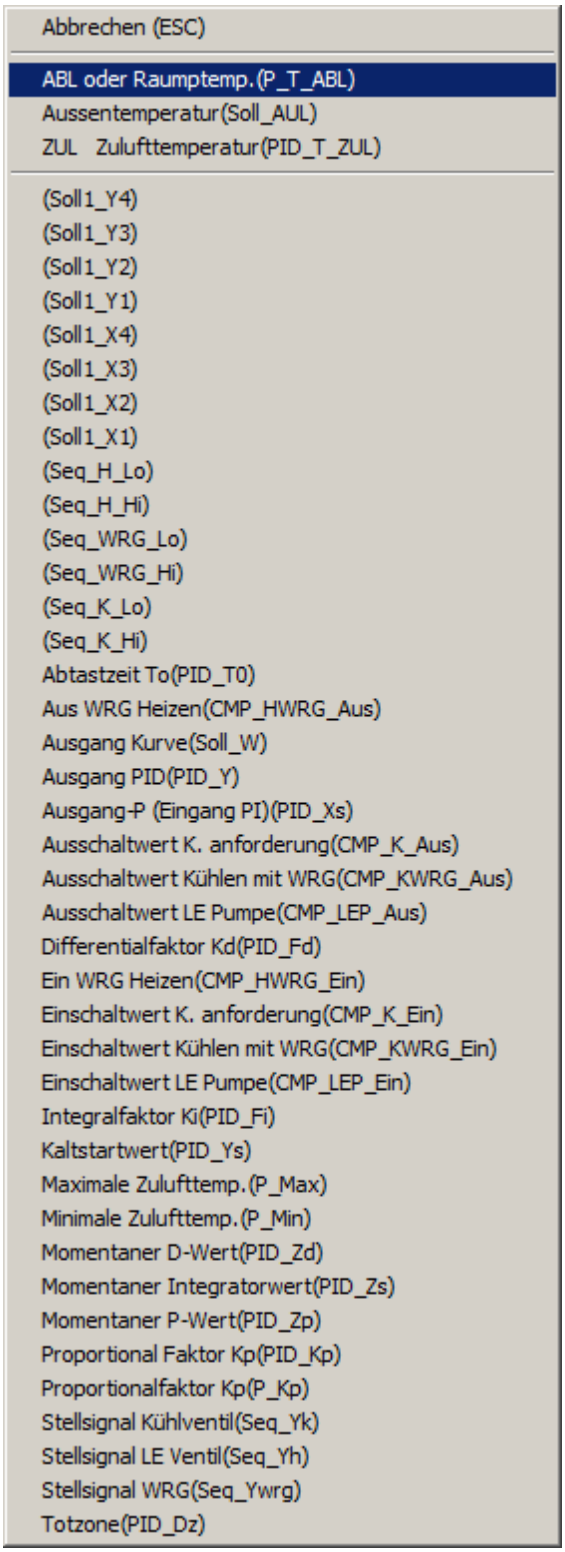
The menu item consists of the signal description (data point comment) and the signal name in parentheses (DMS attribute). For the data point shown, this results in a menu item "Actual value P controller (PT_ABL).

For the **Output object**, all signals, that is both the analogue and the digital signals, are listed alphabetically in the pop-up menu. For example, the **Output menu** of an MES01 object looks as follows:



After a selection has been made here, the **Input object** menu then only displays signals with the same data type.

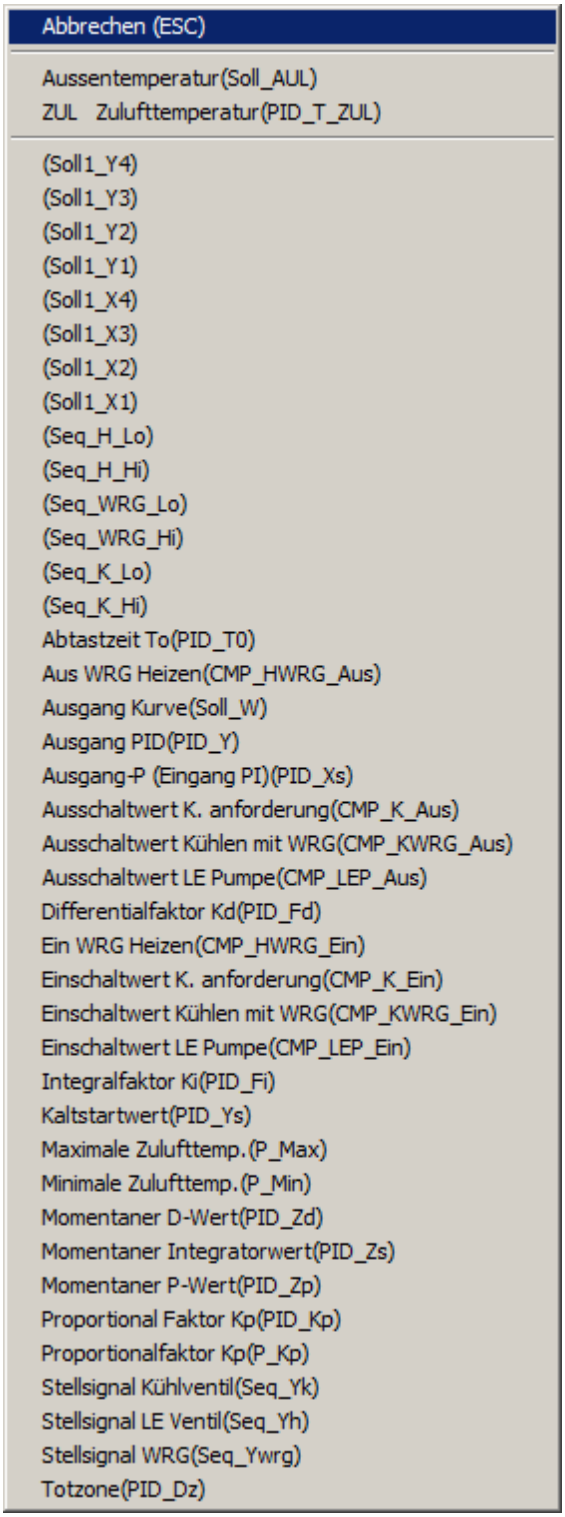
In the pop-up menu of an MES01 object, for example, the analogue signal "ActualValue(ActualValue)" of the data type FLT has been selected. Afterward, only FLT signals are listed in the input menu. If you are dealing with a PID16 object in the output object menu, the following menu is displayed:



It can be seen that in this menu, a further subdivision has been made. First, the signals are listed for which the data point PAR_IN is present in the DMS, then the remaining signals follow.

Furthermore, because one input signal can always only be linked with one output signal (1:1 link), only those signals which are not yet linked are listed in the input menu.

If, for example, the actual value of the measurement object has been linked to the P_T_ABL input of the PID controller, the item "P_T_ABL" will no longer be available the next time linking is performed:



In the output menu, however, the "Actual value" item remains available the next time it is called because one output can control several inputs (1:n link).

10.3.17.3 Moving link objects

Link objects can be moved in the Design and Link Boxes views

1. Moving in the Design view

Link objects are grouped in blocks in this view:

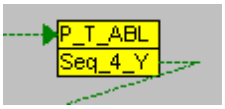
Mono-block
With only one link object

1. The link object is associated with a system object

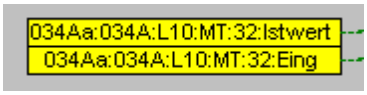
2. The screen contains no associated system object

Multi-block
With several link objects

1. The link objects are associated with a system object



2. The screen contains no associated system object



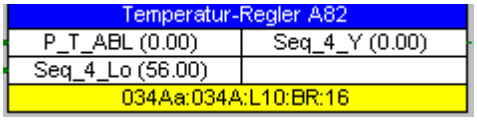
If the block is associated with a BMO object, it automatically moves along with the object when the object is moved.

In addition, the block can be moved individually as well. To do so:
1. Select the block using a left mouse click.
2. Move the block to the desired location using the arrow keys or the mouse.

Furthermore, link objects can be moved within a multi-block. To do so:
1. Select the entry which is to be moved using the left mouse button.
2. At the same time, press the Ctrl key and the arrow keys "UP" or "DOWN".

2. Moving in the Link Boxes view

Link boxes have the following appearance:



They can be moved in their entirety. To do so:
1. Select the link box by means of a left click either on the header (blue) or the footer (yellow).
2. Move the block to the desired location using the arrow keys or the mouse.

Furthermore, the entries within the white area (inputs on the left, outputs on the right) can also be moved. To do so:
1. Select the entry to be moved using the left mouse button.
2. At the same time, press the Ctrl key and the arrow keys "UP" or "DOWN".

10.3.17.4 Deleting Objects

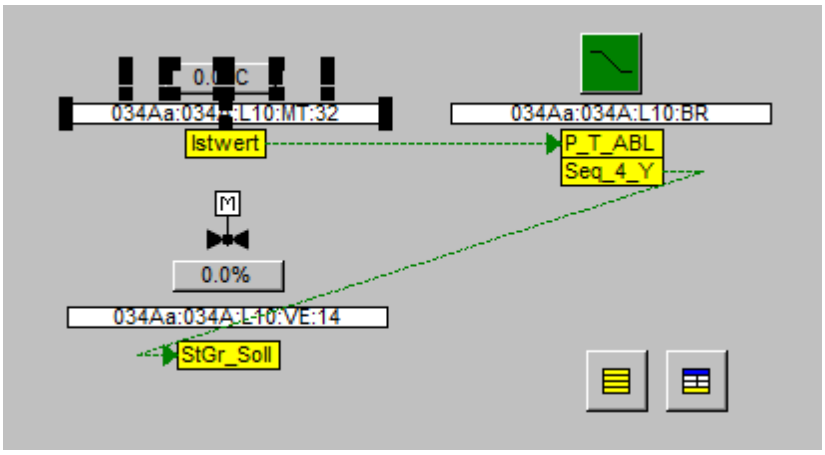
Both BMO objects and link objects can be deleted. Link objects can be deleted both in the Design and the Link Boxes views.

The procedure will be explained using simple examples.

Examples 1 and 2 show how to delete BMO objects.
Examples 3 and 4 show how to delete link objects in the Design view.
Example 5 explains the deletion of link objects in the Link Boxes view.

Example 1:
The basis here is example 3 in the chapter Graphical editor.

The screen consists of 3 objects:
1. Measurement object 034Aa:034A:L10:MT:32", derived from MES01
2. PID object "034Aa:034A:L10:BR:16", derived from PID17
3. Valve object "034Aa:034A:L10:VE:14", derived from VEN01

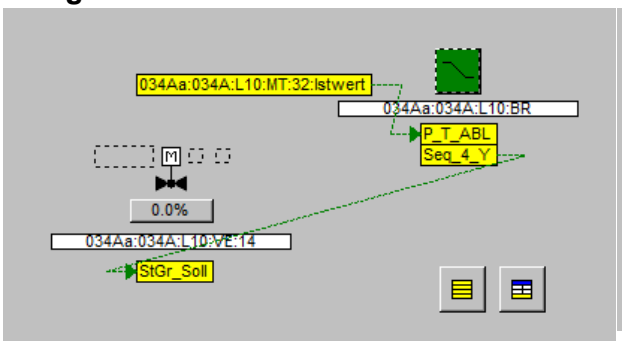


We are in the Design view. The measurement object is to be deleted.

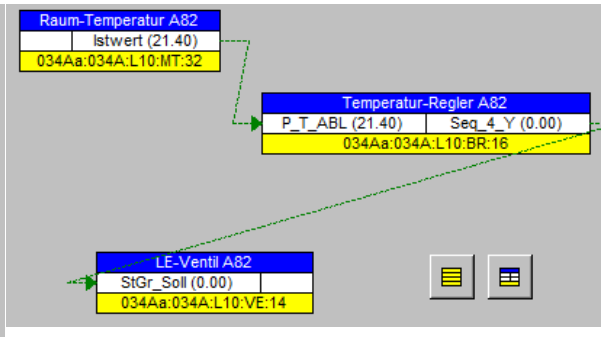
To do so:
1. Select the measurement object with a left mouse click (see figure).
2. Press the "Delete" key.

Afterwards, the views should look like the following:

Design view



Link Boxes view



When deleting BMO objects, the associated link objects are not deleted along with them, even

if they were selected at the same time. Link objects are only deleted if they are selected individually (see examples 3 and 4).

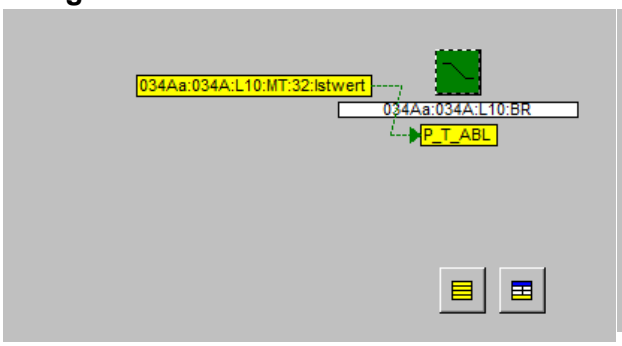
Because the measurement object no longer exists, but the DMS link still does, the link continues to be displayed in both views.
In the Design view, the complete DMS name is now displayed (instead of only the attribute).

Example 2:
This example is based on the last screen from example 1. We are still in the Design view. Now the valve is to be deleted.

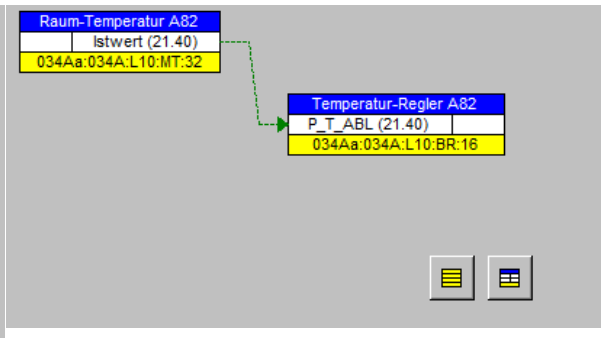
To do so:
1. Select the valve with a left mouse click.
2. Press the "Delete" key.

This results in the following views:

Design view



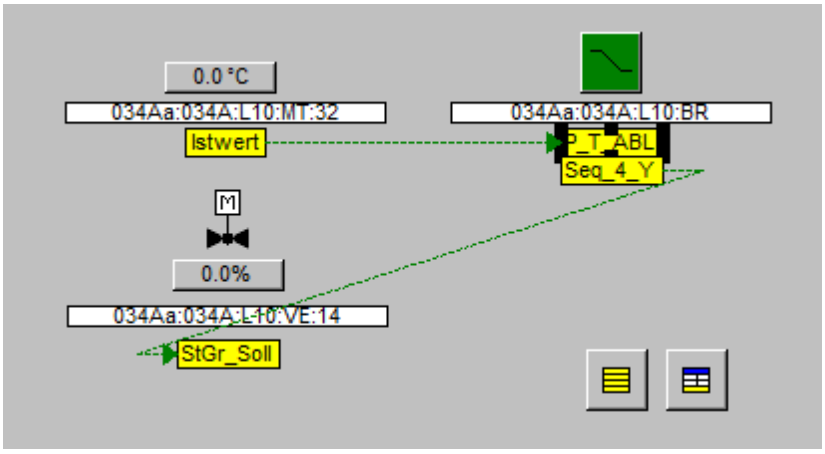
Link Boxes view



The valve has been deleted in both views along with the associated link objects.

Example 3:

This example is based on the initial screen from example 1:

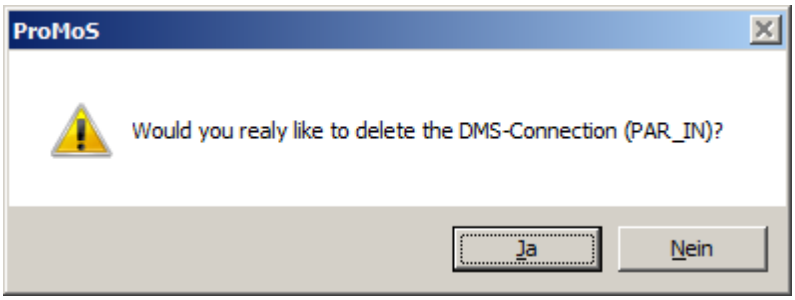


The DMS link between the measurement object and the PID controller is now to be deleted.

To do so:
1. Make a left mouse click on the actual value of the measurement object or on P_T_ABL of the PID controller (see figure).

2. Press the "Delete" key.

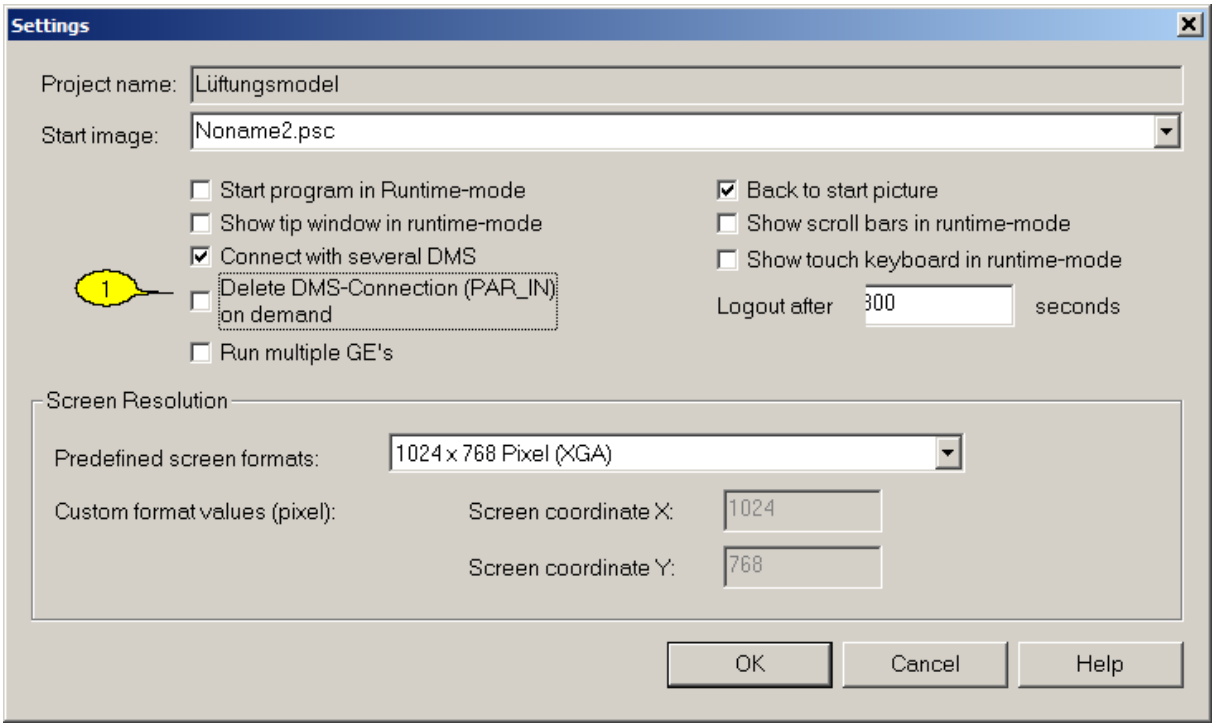
There will be a confirmation prompt, whether you really want to delete the DMS link:



<Yes>
The link object and the DMS link are deleted.

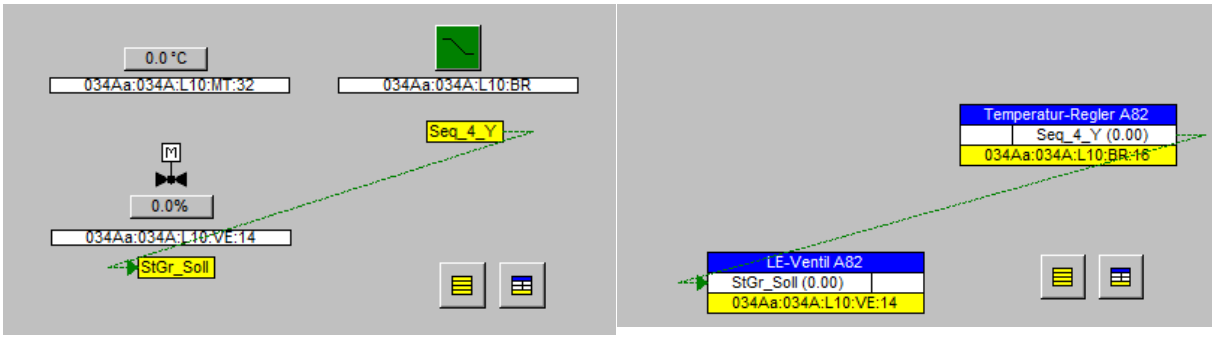
<No>
The link object and the DMS link are not deleted.

This dialog box can be hidden. To do so, open the Project Settings dialog through the **"Project Settings > Project Setting"** menu
and deactivate the option **"Only Delete DMS Link (PAR_IN) After Confirmation Prompt" :**

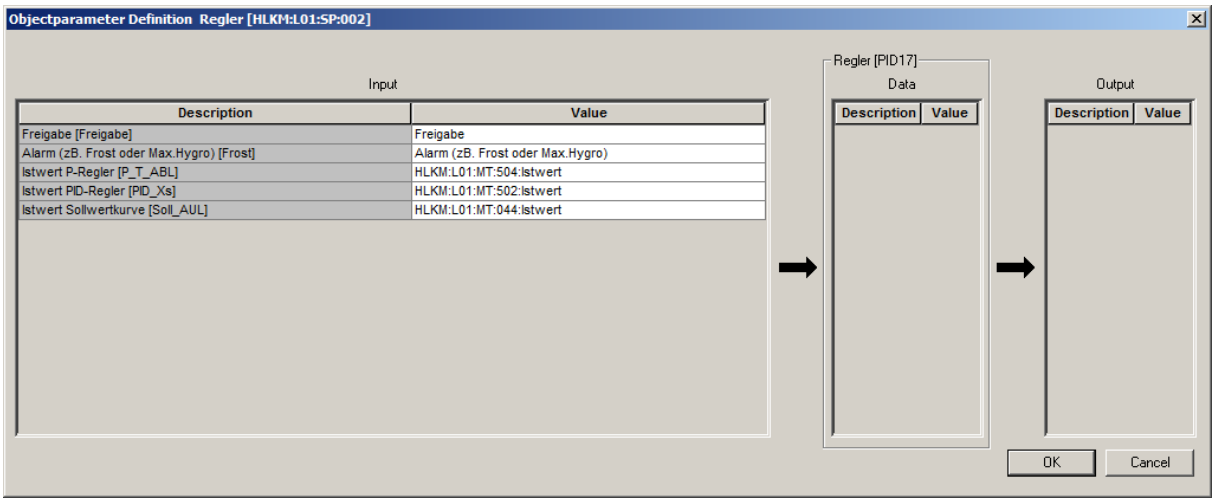


When **Yes** is clicked, the selected link object of the PID controller (P_T_ABL) and link object connected to it belonging to the measurement value (ActualValue) are deleted from both views:

Design view **Link Boxes view**

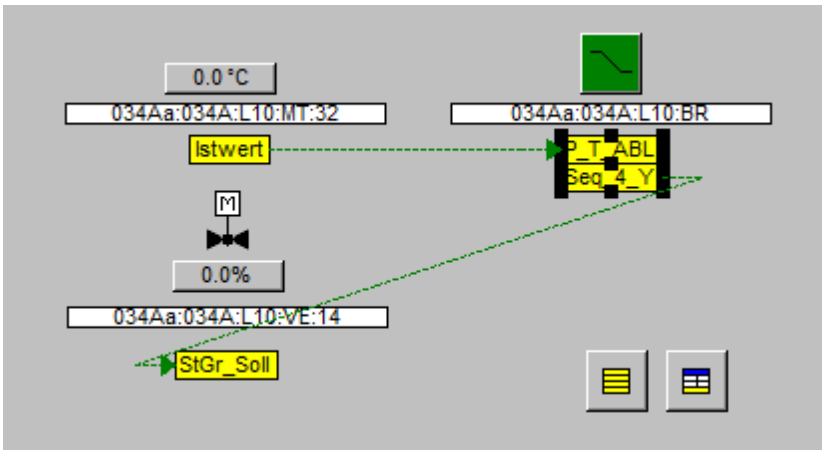


If the parameters are viewed, it can be seen that the input parameter of the PID controller has been deleted from the DMS also:



Example 4:
It is possible to delete not only a single link, but also several links at the same time.

As an example, in the initial screen from example 1, all of the links of the PID are to be deleted:



To do so:
1. Mark the yellow block of the PID by dragging the mouse pointer while holding down the left mouse button (see figure).

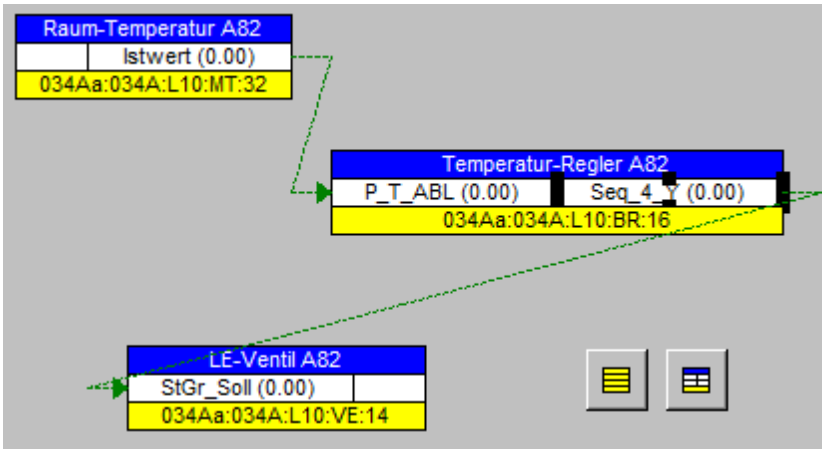
2. Press the "Delete" key.
3. Answer the question "Do you want to delete the DMS link (PAR_IN)?" with **Yes**.

This deletes all link objects of the Design and Link Boxes views:

Design view **Link Boxes view**

Example 5:

This example is based on the initial screen from example 1: The Link Boxes view should be activated now:



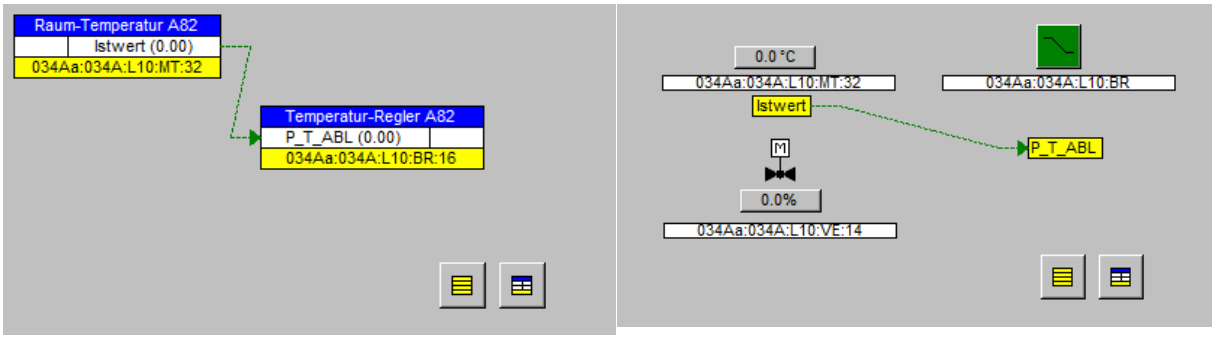
If the actual value of the measurement value object is deleted, the result will be the same as in the previous example.

Instead of deleting the actual value, we instead wish to delete the link between the PID controller and the valve.

To do so:
1. Left click on the Seq_4_Y of the PID or on the StGr_Soll of the valve (see the figure).
2. Press the "Delete" key.
3. Answer the question "Do you want to delete the DMS link (PAR_IN)?" with **Yes**.

The following two views result:

Link Boxes view **Design view**



All links of an object can also be deleted at the same time for the link boxes as well.

A single link box must be marked by clicking on the header (blue area) or the footer (yellow area) of the link box.

Several link boxes can be marked by dragging the mouse pointer while holding down the left mouse button.

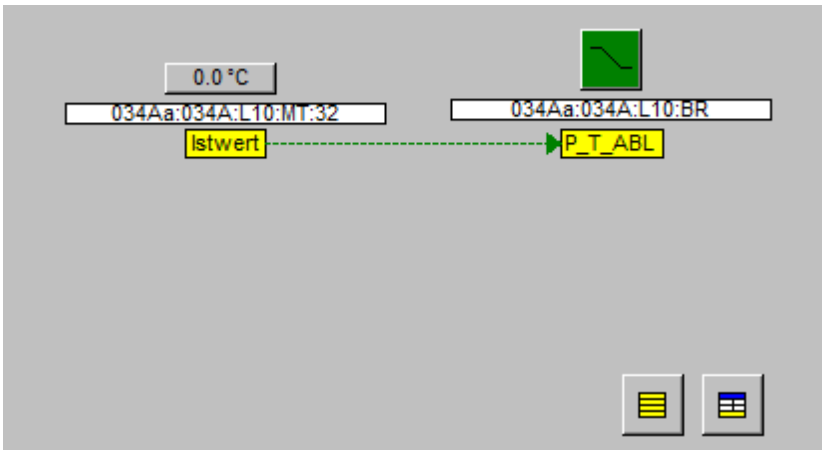
10.3.17.5 Reinitialising Objects

The reinitialising of objects is explained here based on a simple example.

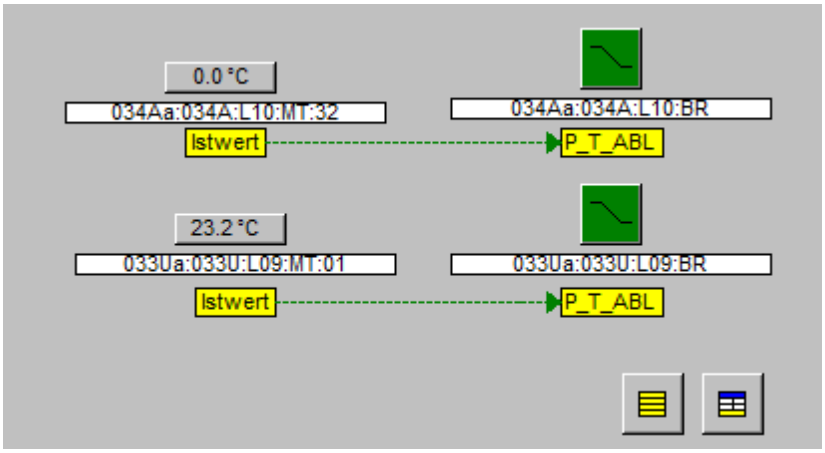
Objective
The initial screen is to consist of 2 objects:

1. A measurement object which is derived from MES01.
2. A PID controller which is derived from PID17.

The actual value of a measurement object should control the actual value P controller input (P_T_ALB) of the PID controller.



After completing the example, a 2nd measurement object and a 2nd PID object should have been created in the screen which have the same link as the two objects of the initial screen:



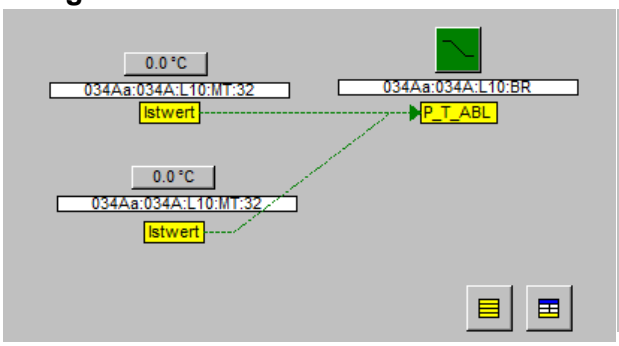
Solution

Step 1: Create the initial screen
The initial screen is created in a manner similar to **example 1** in the chapter Graphical editor.

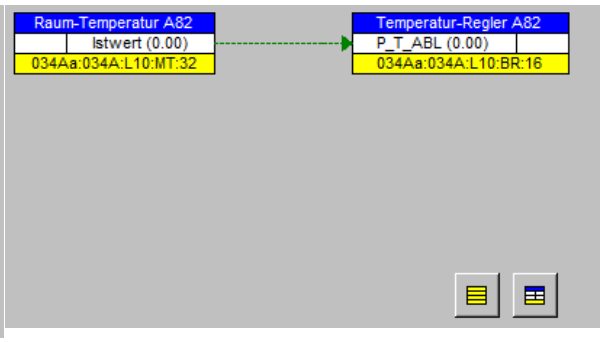
Step 2: Copy the measurement object
1. Select the measurement object with a left mouse click.
2. Press the keys "Ctrl+C" and then "Ctrl+V" to copy and paste the object
 or press "Ctrl+D" to duplicate the object.
3. Move the copied measurement object to a free area.

These actions should deliver the following result:

Design view



Link Boxes view

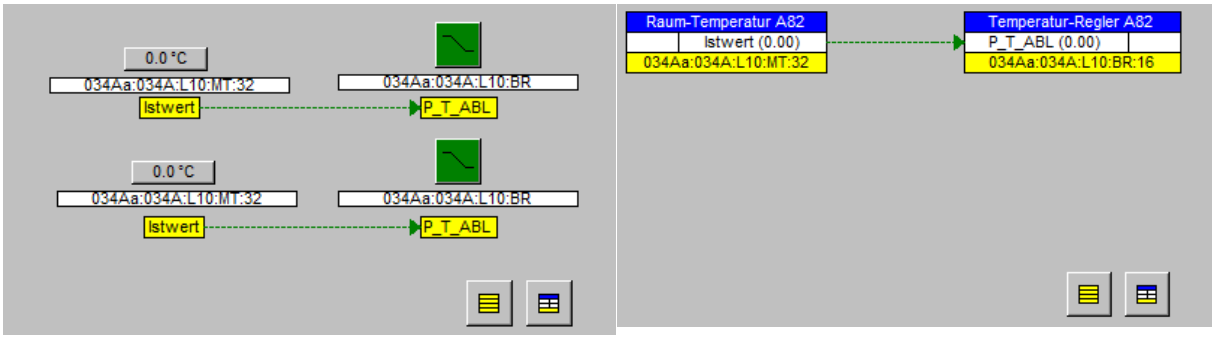


In the Link Boxes view, nothing will have changed because no new DMS link was created. In the Design view, a new actual value link object has been added. It is connected to the P_T_ABL input of the PID controller.

Step 3: Copy the PID controller
1. Select the PID object with a left mouse click.
2. Press the keys "Ctrl+C" and then "Ctrl+V" to copy and paste the object
 or press "Ctrl+D" to duplicate the object.
3. Move the copied PID object to a free area.

These actions deliver the following result:

Design view **Link Boxes view**

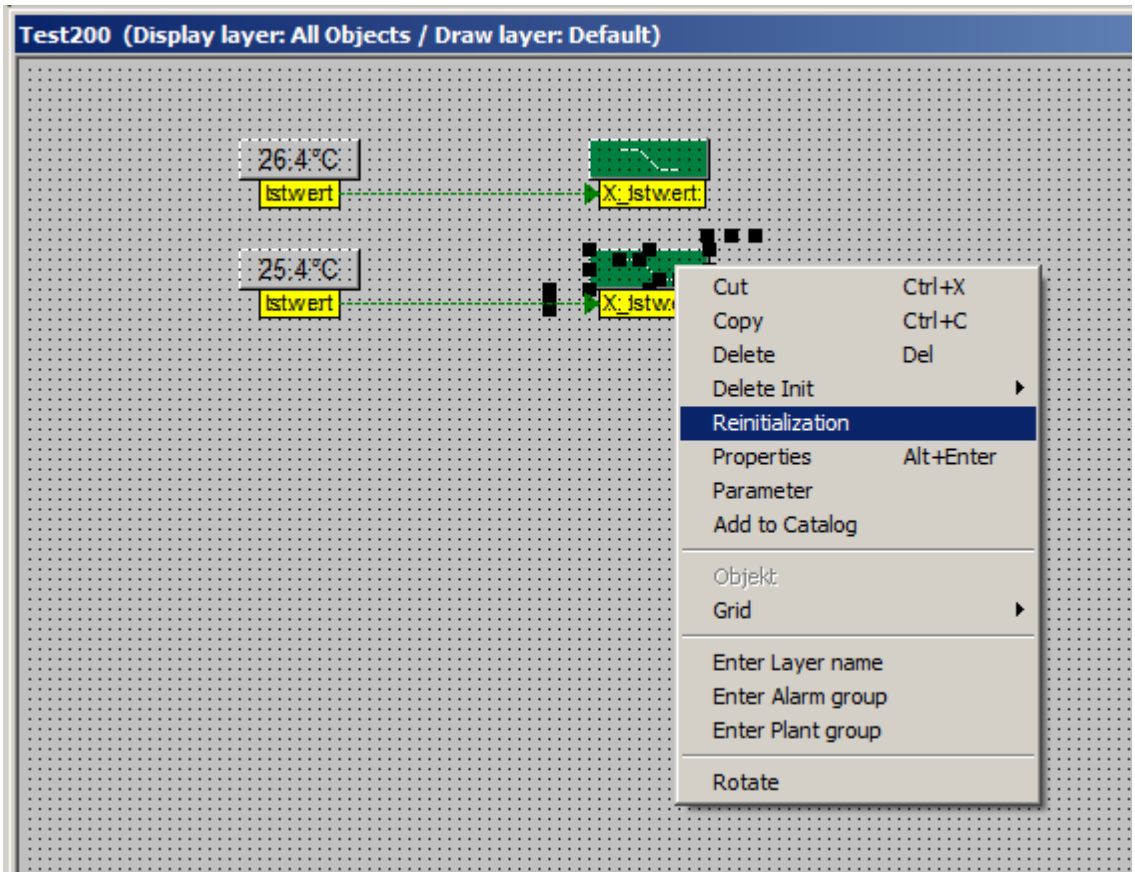


In the Link Boxes view, again, nothing will have changed. This time as well, no new link object was created.

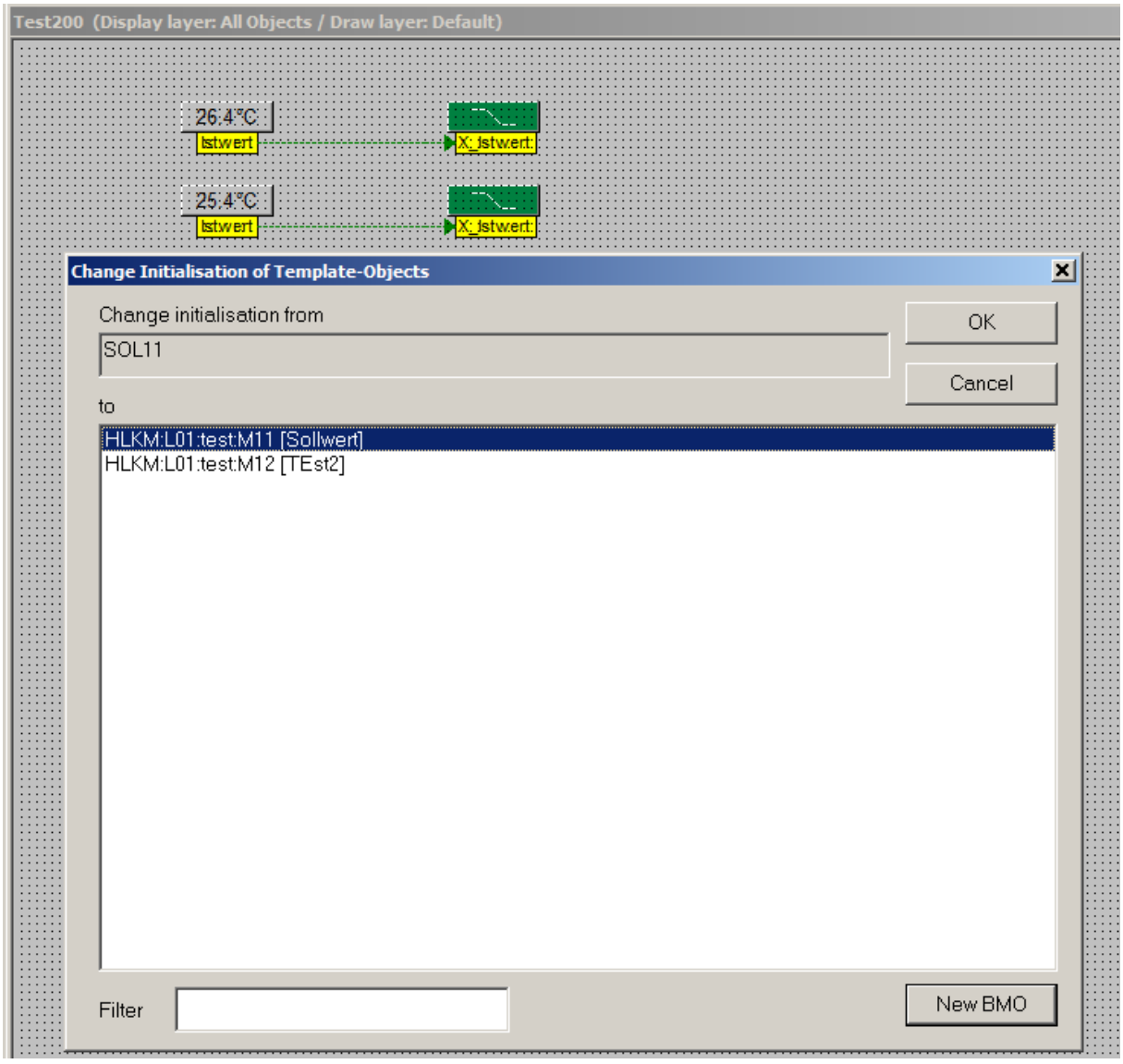
In the Design view, the P_T_ABL link object of the copied PID object has been connected to the actual value link object of the previously copied measurement object.

Step 4: Reinitialise the copied PID controller

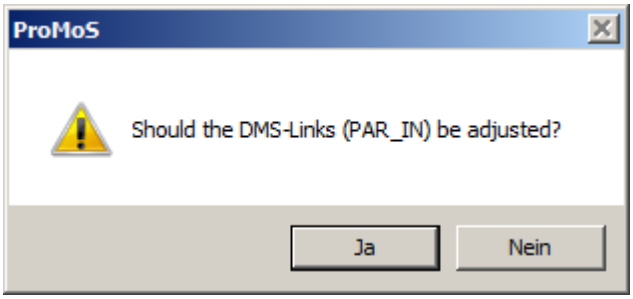
1. Right click on the copied PID object.
2. In the pop-up menu, select the item "Reinitialisation".



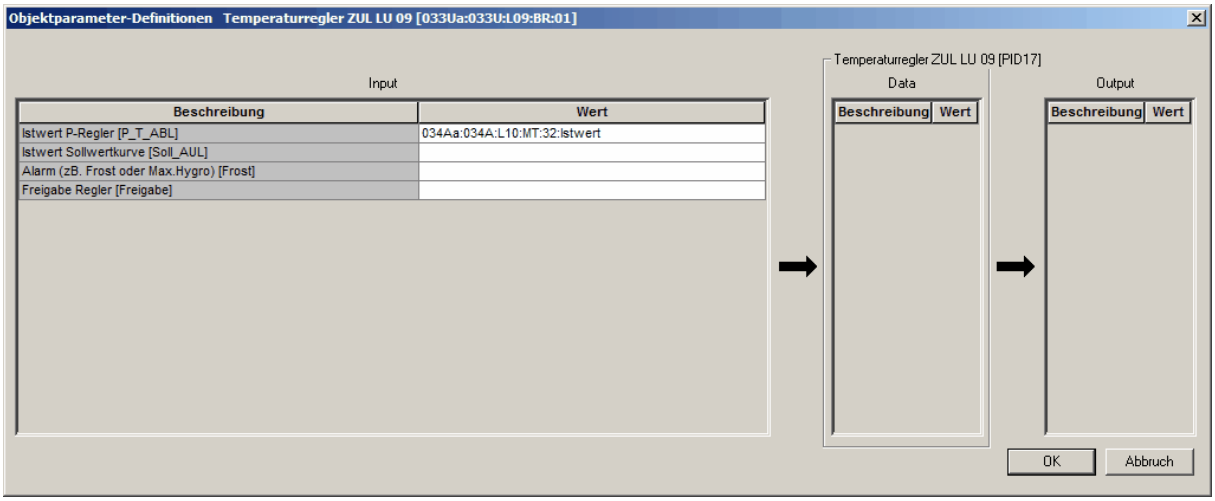
3. Select a new BMO object in the reinitialisation dialog which opens.



4. Answer the subsequent question with **Yes** to apply the input parameters from the linked measurement object.



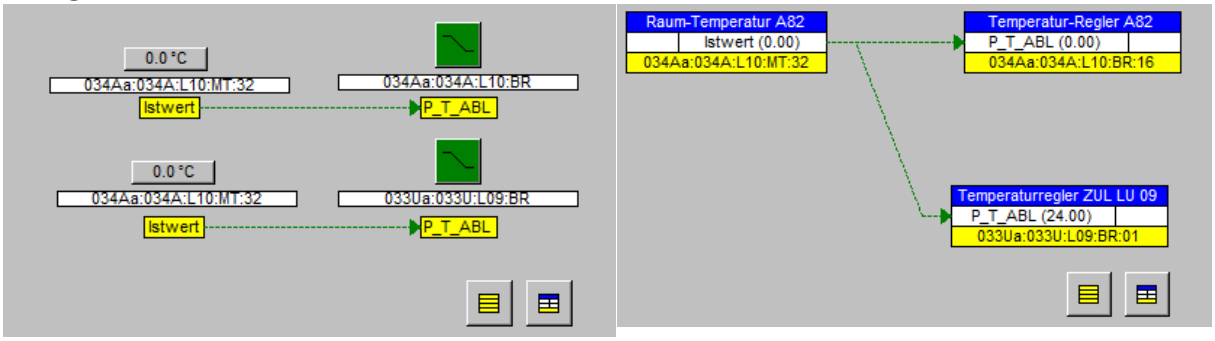
3. Afterwards, the Parameter Definition dialog will open. The first line shows that the reinitialised PID object "033Ua:033U:L09:BR:01" is now linked with the measurement object "034Aa:034:L10:MT:32:ActualValue".



4. Close the Parameters dialog by clicking on **OK**.

The two views should now look as follows:

Design view **Link Boxes view**



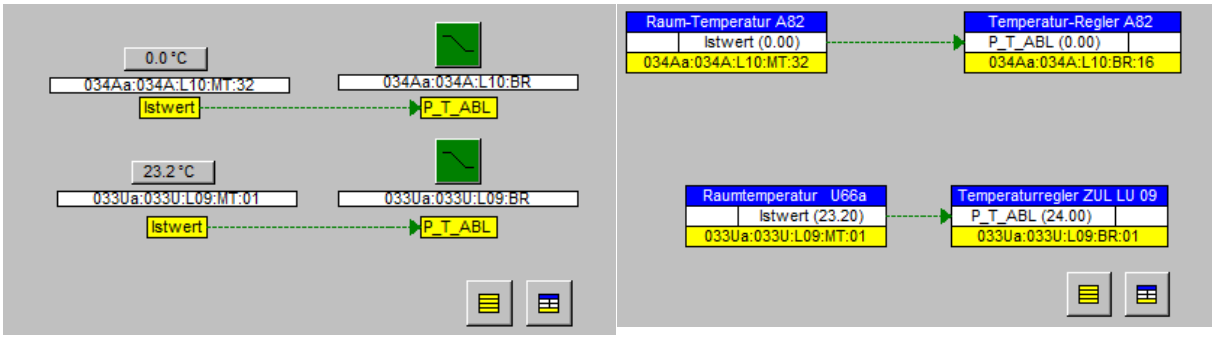
The fact that one new DMS link was created can be seen best in the Link Boxes view. Here, a new link box has been added.

Step 5: Reinitialise the copied measurement object.
This is done a manner similar to the reinitialisation of the PID controller which was just shown in detail.

1. Right click on the copied measurement object.
2. In the pop-up menu, select the item "Reinitialisation".
4. Select a new BMO object in the reinitialisation dialog (operating resource object) which opens.
5. Answer the question "Should the DMS links (PAR_IN) be adapted?" with **Yes**.
6. Close the Parameters dialog by clicking on OK.

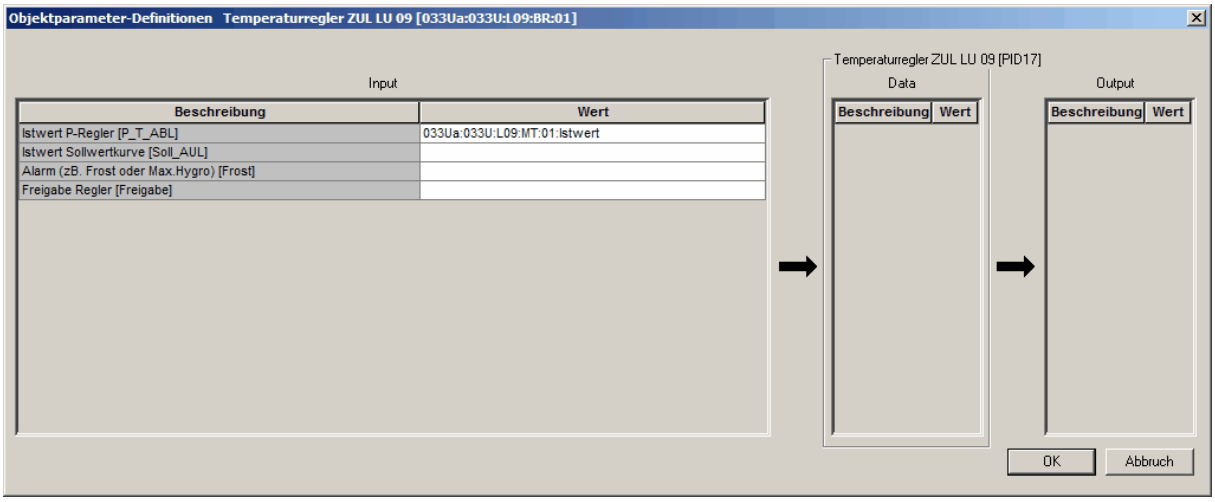
We have now reached the end of the example with the desired result:

Design view **Link Boxes view**



Two link objects were created in the Design view and two link boxes were created in the Link Boxes view. These belong to the newly-created system objects.

The creation of the link in the DMS was also successful, as is evidenced by the Parameters dialog of the created PID object:



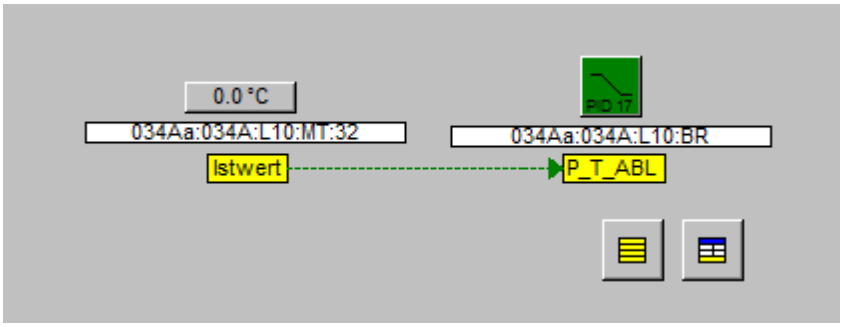
10.3.17.6 Creating and inserting macros

The example in the chapter Reinitialising objects could also have been accomplished using a macro consisting of the reinitialised measurement object and the PID controller.

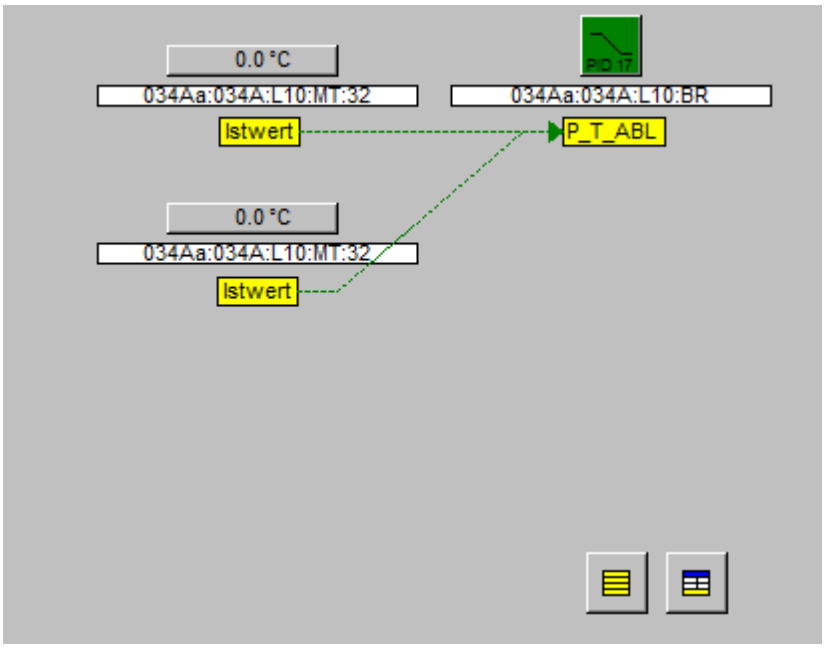
The concept of macros and their creation and use was already introduced in the chapters Macros, Loading macros and Saving macros. The following example is intended to show what happens with the DMS links when a macro is reinitialised.

Objective

This example is based on the screen created in **example 1** in the **chapter Graphical editor**:

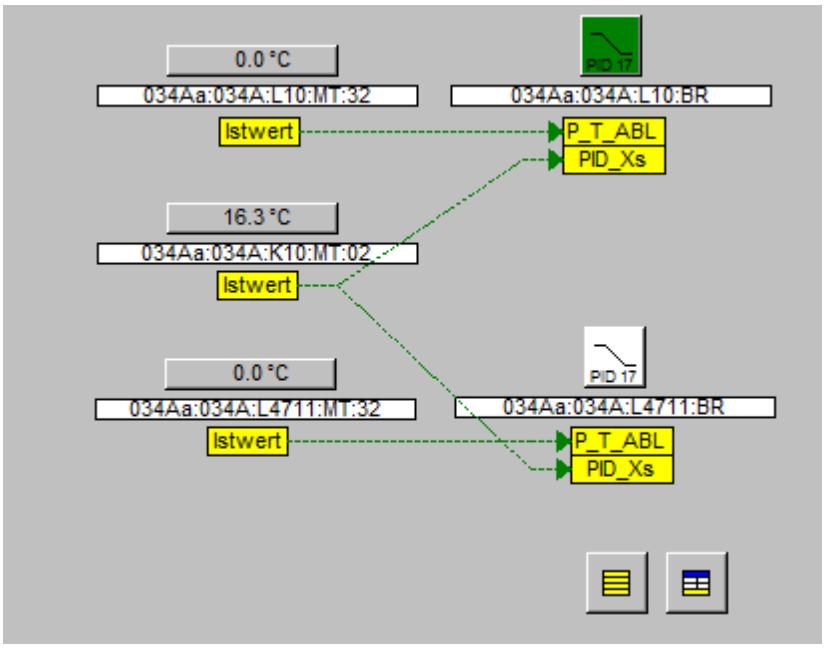


1. The PID_Xs input of the PID controller is to be controlled by the actual value of a further measurement object:



2. A macro is to be created from the first measurement object and the PID controller.
3. The macro created is then to be loaded and reinitialised.

The result should then look something like this:



Solution

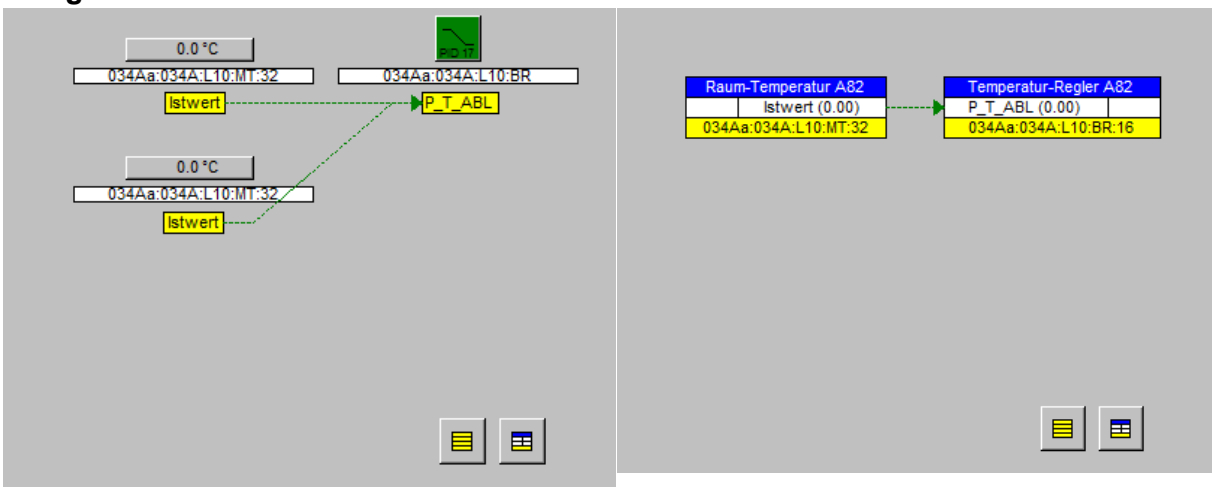
Step 1: Create the initial screen

The initial screen is created in a manner similar to **example 1** in the chapter Graphical editor.

Step 2: Copy the measurement object

1. Select the Design view by clicking on  either in the toolbar or in the screen.
2. Select the measurement object with a left mouse click.
3. Press the keys "Ctrl+C" and then "Ctrl+V" to copy and paste the object
   or press "Ctrl+D" to duplicate the object.
4. Move the copied measurement object to a free area.

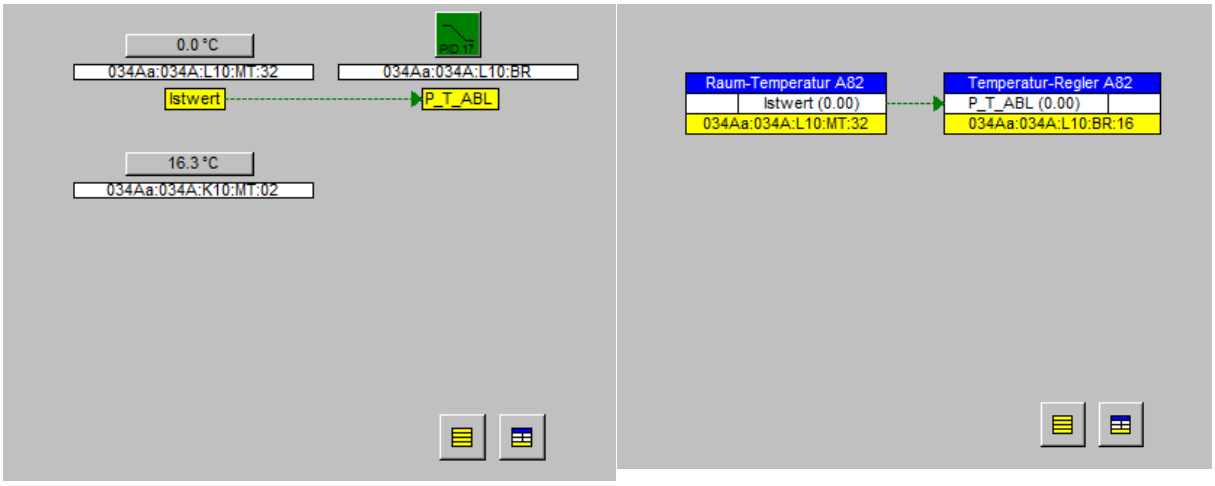
Design view **Link Boxes view**



Step 3: Reinitialise the copied measurement object

1. Right click on the copied measurement object.
2. In the pop-up menu, select the item "Reinitialisation".
3. Select a new BMO object in the reinitialisation dialog which opens. When doing so, be sure that the new BMO object is not on the same level as the BMO objects of the initial screen. In our example, the common level between the PID controller and the first measurement object is "034Aa:034A:L10". Consequently, in this example, a measurement object located on level "034Aa:034A:K10" will be selected.
4. Answer the subsequent question "Should the DMS links (PAR_IN) be adapted?" with **No**.

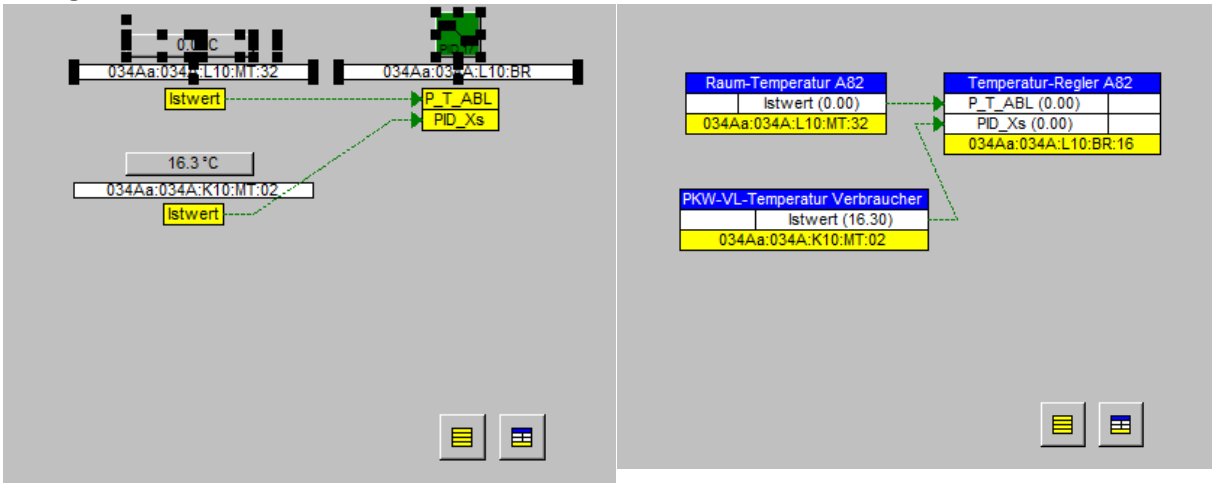
Design view **Link Boxes view**



Step 3: Link the BMO object

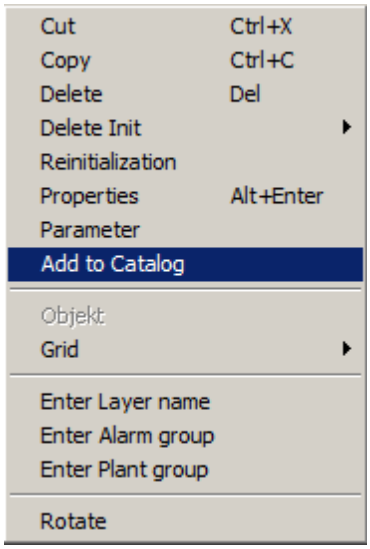
1. Click on  in the Drawing Tools toolbar.
2. Click on the copied measurement object with the left mouse button and select the item ActualValue(ActualValue) in the pop-up menu.
3. Click on the PID controller with the left mouse button and select the item in the pop-up menu.

Design view **Link Boxes view**



Step 4: Save the macro in the catalogue

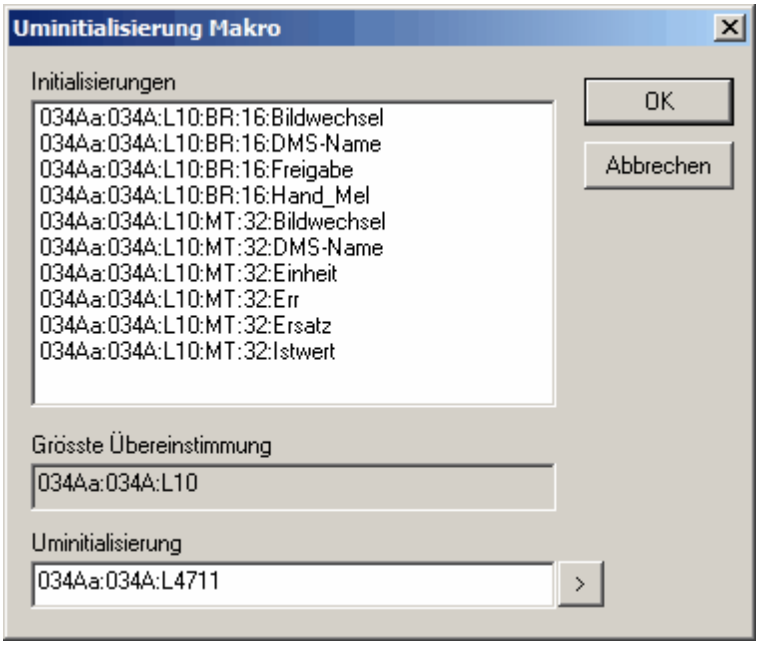
1. Open the Saia Visi.Plus Library (catalogue) by clicking on the **View > Catalogue Bar** menu.
2. Mark the measurement object of the initial screen and the PID controller by dragging the mouse pointer while holding down the left mouse button (see figure).
3. Open the pop-up menu with a right mouse click and select the option Add Object to Catalogue:



The measurement object and the PID controller are now saved together as a macro in the catalogue. Link objects which may have been copied as well during selection are not saved.

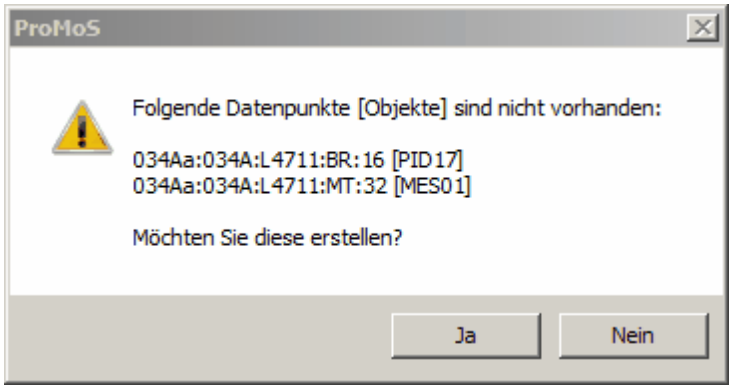
Step 4: Load the created macro and reinitialise

1. Click on the newly-created macro in the catalogue and drag it to the screen.
2. Enter the new DMS data point in the "Reinitialisation" field in the reinitialisation dialog:

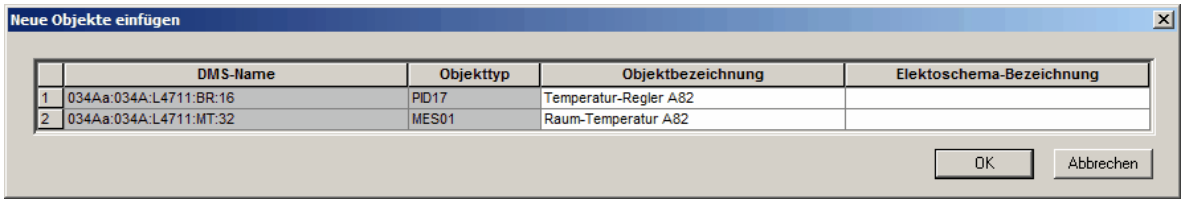


Please specify a data point here which does not yet exist in the DMS.

3. Answer the subsequent question with Yes:



4. If needed, change the object designation in the next dialog:



The object designation is applied to the macro objects. Then exit the dialog with OK.

The data points for the new PID17 and MES01 objects are now created. In addition, the DMS links are updated.

The macro contains two DMS links:

1. To the measurement object located on the same level as the PID controller (link ActualValue -> P_T_ABL).
   In this case the DMS link is updated. For PAR_IN, the "Greatest match" is replaced by "Reinitialisation".

2. To the measurement object located on a different level than the PID controller (link ActualValue -> PID_Xs).
   In this case, the link to the original object remains unchanged.

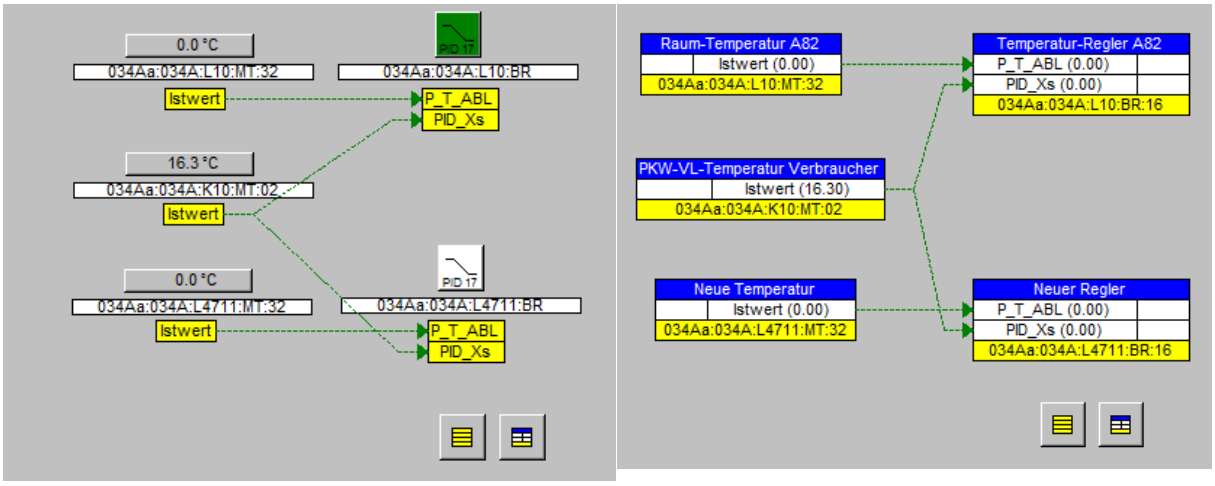
These changes are summarised for the present example in the following table:

Signal	PAR_IN of the macro (PID controller)	PAR_IN of the reinitialised PID controller
P_T_ABL	034Aa:034A:L10:MT:32:ActualValue	034Aa:034A:L4711:MT:32:ActualValue*
PID_Xs	034Aa:034A:K10:MT:02:ActualValue	034Aa:034A:K10:MT:02:ActualValue**

* **Greatest match** "034Aa:034A:L10" has been replaced by **Reinitialisation** "034Aa:034A:L4711"
** No match between **Greatest match** "034Aa:034A:L10" and **PAR_IN** "034Aa:034A:K10:MT:02:ActualValue", therefore, no replacement.

This then looks as follows in the two views:

Design view **Link Boxes view**

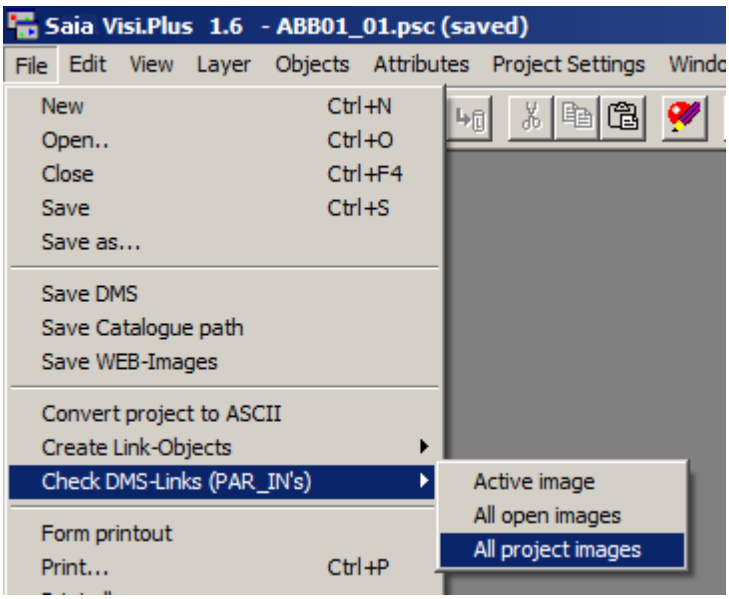


10.3.17.7 Checking DMS links

It sometimes occurs that data points are renamed without the DMS links (PAR_INs) being modified accordingly. Until now, it was very time consuming to track down these errors.

Consequently, the GE now provides a function which checks whether the data point referenced by the DMS link (PAR_IN) still exists in the DMS.

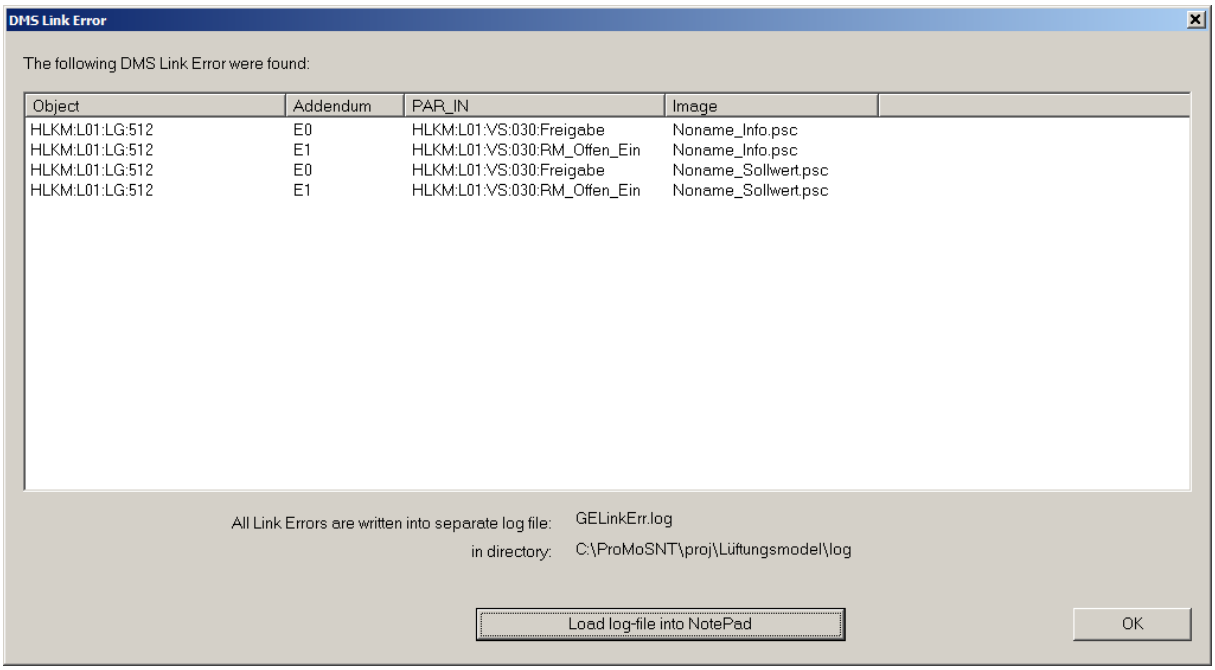
This function can be called up through the **"File -> Check DMS Links (PAR_INs)"** menu:



In the pop-up menu, it is possible to specify whether the **Active diagram, All open diagrams** or **All diagrams of the project** should be checked.

If "All diagrams of the project" has been selected, a progress bar appears.

If a DMS link error is found during the course of the check, it will be output in a message window upon conclusion:



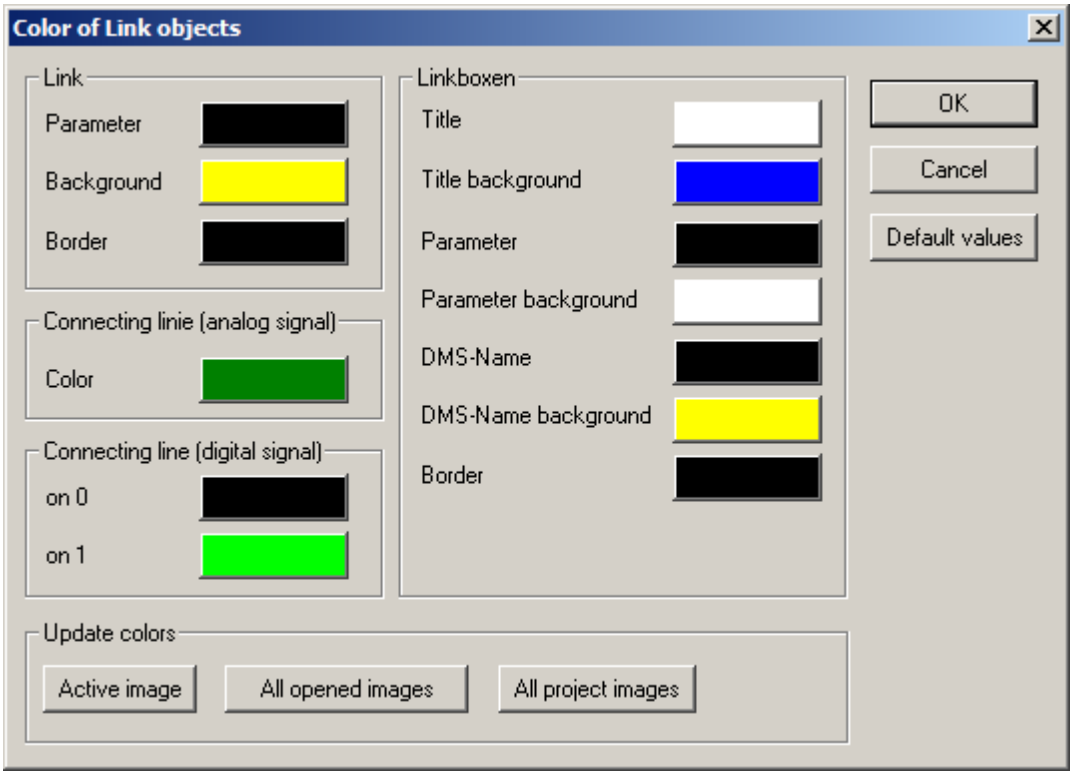
<Load log file in NotePad>
The errors are stored in the file GELinkErr.log. This button can be used to open the log file directly from the message window.

< OK>
The message window is closed.

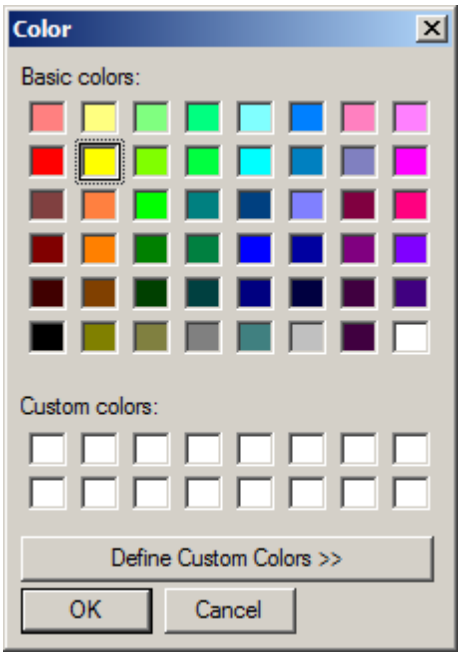
The table or the log file contains all the information necessary to be able to eliminate the error quickly and with precision.

10.3.17.8 Changing colours

The standard colours can be changed via the menu **"Attributes > Specify link object colours"**:



The colour can be changed by clicking on one of the coloured buttons. This opens the colour selection window:



Now the desired colour can be selected here.

In order to apply a desired colour change, one of the three buttons under "Update colours" must be actuated. "OK" will not cause a change.

10.3.18 Displaying documents in the GE/WebServer

The following document types can be displayed in both the GE and the WebServer:

- **PDF** (Adobe)
- **TXT** (ASCII editor, Notepad)
- **HTM**, **HTML** (web browser, HTML reader)
- **DOC** (Office, Word)
- **PPS** (PowerPoint)
- **XLS** (Excel)
- **SXW** (OpenOffice)
- **RTF** (RTF editor, Wordpad)

If filenames contain blank spaces, the entire expression "<name with blank space>" must be written (in quotation marks).

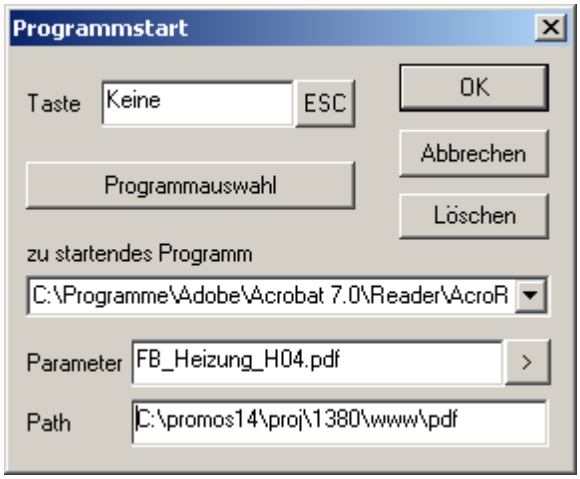
Since incompatibilities with Internet Explorer repeatedly arise on the client side, we urgently recommend dispensing with blank spaces in filenames!

Example of a PDF display

Program to be Started:
Name of the program (preferably with full path) in which the document is displayed.
For pdf documents: "C:\Program Files\Adobe\Acrobat 7.0\Reader\AcroRd32.exe"

Parameter:
Name of the document (without path).
e.g.: "Manual_en.pdf"

Path
Path statement of the document.
e.g.: "C:\Visi.Plus\proj\<ProjName>\www\pdf"

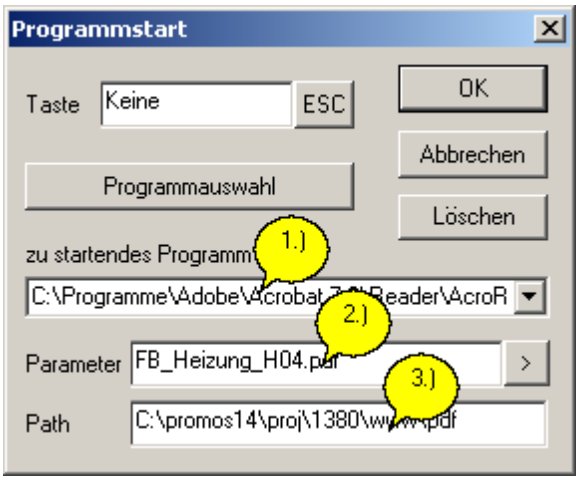


In order to avoid document files appearing twice within the project, or having to be

copied, we recommend placing them in respective folders in \www from the start, and using these path references in GE_images.

10.3.18.1 Examples

The following examples are compatible with **WebServer**, meaning that the document is also opened on the client side in the respective program or is interpreted in a new Explorer window. This depends on the settings of the client PC ("Select the program to open this file").



PDF: 1) "<path>\Reader\AcroRd32.exe" e.g. "C:\Program Files\Adobe\Acrobat 7.0 \Reader\AcroRd32.exe"
2) "<Doc.Name>.pdf" "Manual_en.pdf"
3) "<Doc.path>" "C:\Visi.Plus\proj\<ProjName>\www\pdf"

TXT: 1) "C:\Windows\System32\notepad.exe"
2) "ReadMe.txt"
3) "C:\Visi.Plus\proj\<ProjName>\www\txt"

HTML / HTM:

In order to ensure web server compatibility, another setting must be selected in the GE:

1) "C:\Program Files\Internet Explorer\IEXPLORE.EXE C:\Visi.Plus\proj\Phonak\www \html\Read-me-WEB-Interface.html"
2) blank
3) blank

PPS: PowerPoint doc.
1) "C:\Program Files\Microsoft Office 2000\Office\POWERPNT.EXE"
2) "Demo.pps"
3) "C:\Visi.Plus\proj\<ProjName>\www\pps"

DOC: Word doc.
1) "C:\Program Files\Microsoft Office 2000\Office\WINWORD.EXE"
2) "Demo.doc"
3) "C:\Visi.Plus\proj\<ProjName>\www\doc"

10.3.18.2 Compatibility with WebServer

For the documents to also be displayed in the WebServer, they must be available in the ..<proj>\www\<doc.extention>\.. folder.

e.g. document = “Manual_d.pdf”, localisation (path): “C:**Visi.Plus**\proj\Phonak\www\pdf\Handbuch_d.pdf”.

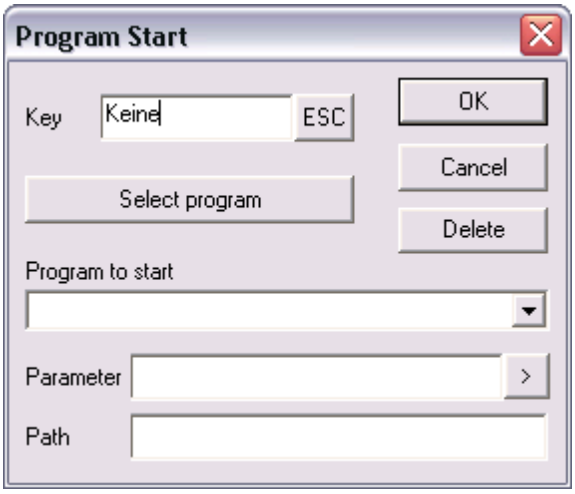
Examples:

PDF documents: `C:\`*Visi.Plus*`\proj\IhrProject\www\pdf\Handbuch.pdf`
Word documents: `c:\`*Visi.Plus*`\proj\IhrProjekt\www\doc\Dokument.doc`
Excel documents: `c:\`*Visi.Plus*`\proj\IhrProjekt\www\xls\Dokument.xls`

To avoid the need to have the document files twice or copied twice in the project, it is recommended to place them in appropriate folders in the \www at the outset and these path references in GE_images.

10.3.19 Web links in the GE/WebServer

The link is defined via the INIT program start action, with which the following fields must also be entered:



Program to be Started:
Name of Internet Explorer (preferably with full path)
e.g.: "C:\Program Files\Internet Explorer\IEXPLORE.EXE"

Parameter:
Name of the link, web address, etc. If the link only has the form "www.name.ext", 'http://' is added by default.
If the link also includes the protocol type (recommended!), such as http://", "https://" or "ftp://", the entire term must be written in quotation marks ("...").
This serves to differentiate between the types DMS Name and Link and prevents the DMS ":" delimiter being replaced by the "_" character in the link name.

Path:
remains empty

10.4 The Alarm Viewer (ALMView)

The Alarm Viewer is used to display alarms which are issued from the PDBS database and, if applicable, to allow them to be acknowledged. The Alarm Viewer can be opened simultaneously on multiple workstations (depending on the licence).

Since the Alarm Viewer is dependent on the DMS, Alarm Manager and PDBS, these modules must also be open on the computer.

10.4.1 Starting the program ALMView

The Alarm Viewer is usually started via an initialised button in the process diagram. However, it can also always be started with a file manager (e.g. Windows Explorer), like any other Visi.Plus module.

Brief instructions for initialising the Alarm Viewer to a button:
In order to start the Alarm Viewer via a button on the display, proceed as follows:

- *After a "**Button**" graphics object has been set in the process diagram, it must be clicked once with the mouse in order to be selected.*
- *Then open the context menu with the right mouse button and select "**Attributes**".*
- *Under the **Action** attribute, press the Initialisation button (right column), select **Start Program** and then the command **Alarm Program**.*
- *Switch the GE to runtime mode and test the button.*

As opposed to the Alarm Manager (alarm management), the Alarm Viewer does not have to be running permanently, but rather only if the user wants and chooses to start it.

The Alarm Manager, on the other hand, must always be running if alarms are to be detected!

10.4.1.1 Program_parameter_AlmView

Filtering parameters for can be issued when ALMView is opened.

In the preceding image, for example, the parameter /DMS=SS027C has been issued. This means that only alarms whose DMS names begin with SS027C are shown in the Alarm Viewer.*

The following parameters (filters) can be assigned:

Parameter	**Example**	**Filter by**
/DMS=	/DMS=SS027C*	DMS name
/TXT=	/ TXT=*incoming*	Text
/PRI=	/PRI=2	Alarm priority
/GRP	/GRP=3	Alarm group
/LIN=	/LIN=300	Max. number of display lines is 300
/Z=		Alarm status
	/Z=1	only incoming alarms are displayed.
	/Z=0	only acknowledged alarms are displayed.
	/Z=-1	only outgoing alarms are displayed.
/H	/H	The tab with the historical alarms is displayed after AlmView is started.

Multiple parameters (filters) can be issued at once. The parameters must be separated from one another by a blank space:

Example: /DMS=SS027C /Pri=2*
Only alarms whose DMS names begin with SS027C and have the priority 2 are displayed.

It is not possible to enter multiple words separated by blank spaces:

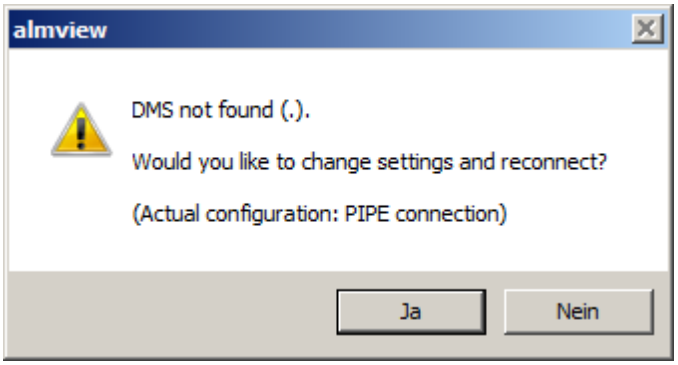
Example: /TXT=*Incoming sensor break **does not work!**

Additional information about the use of filters can be found in the chapter Filter settings.

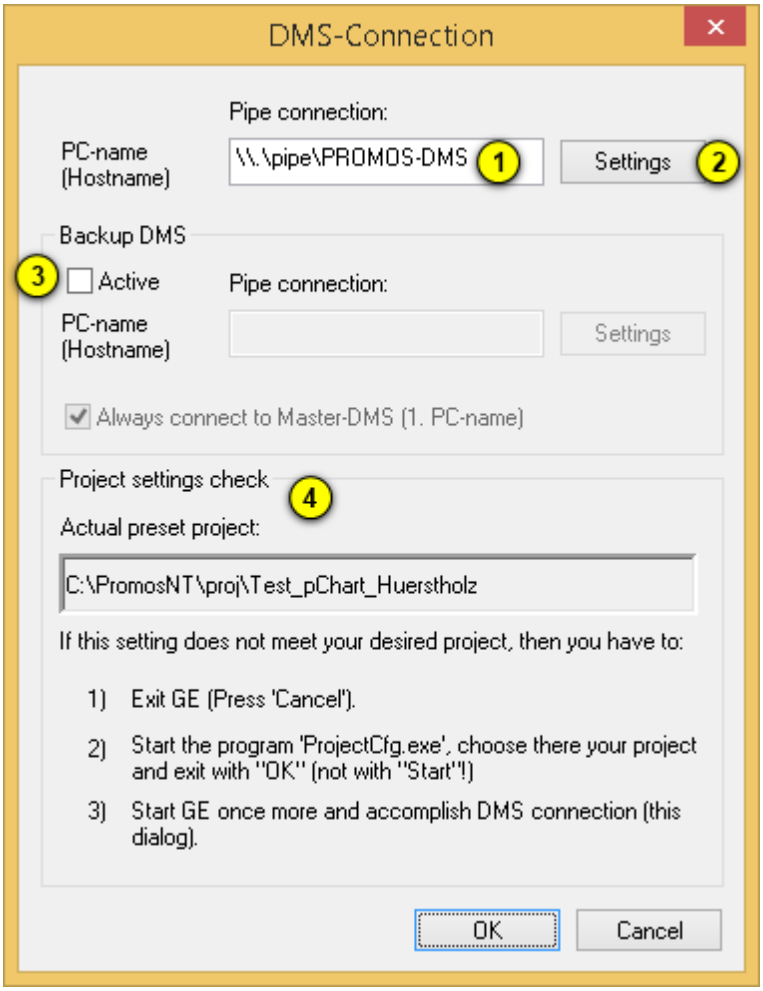
10.4.1.2 ALMView remote

The AlmView can be started even if the DMS is not running on the same PC as the AlmView. To do so, a remote connection must be established.
The connection can be established via pipe or TCP/IP: Pipe connections are more reliable, however the ports are suppressed by many firewalls. It is recommended that a pipe connection be used for a local network.

When starting for the first time, the following message appears:



When the **<Yes>** button is actuated, the configuration window for the remote connection will open.

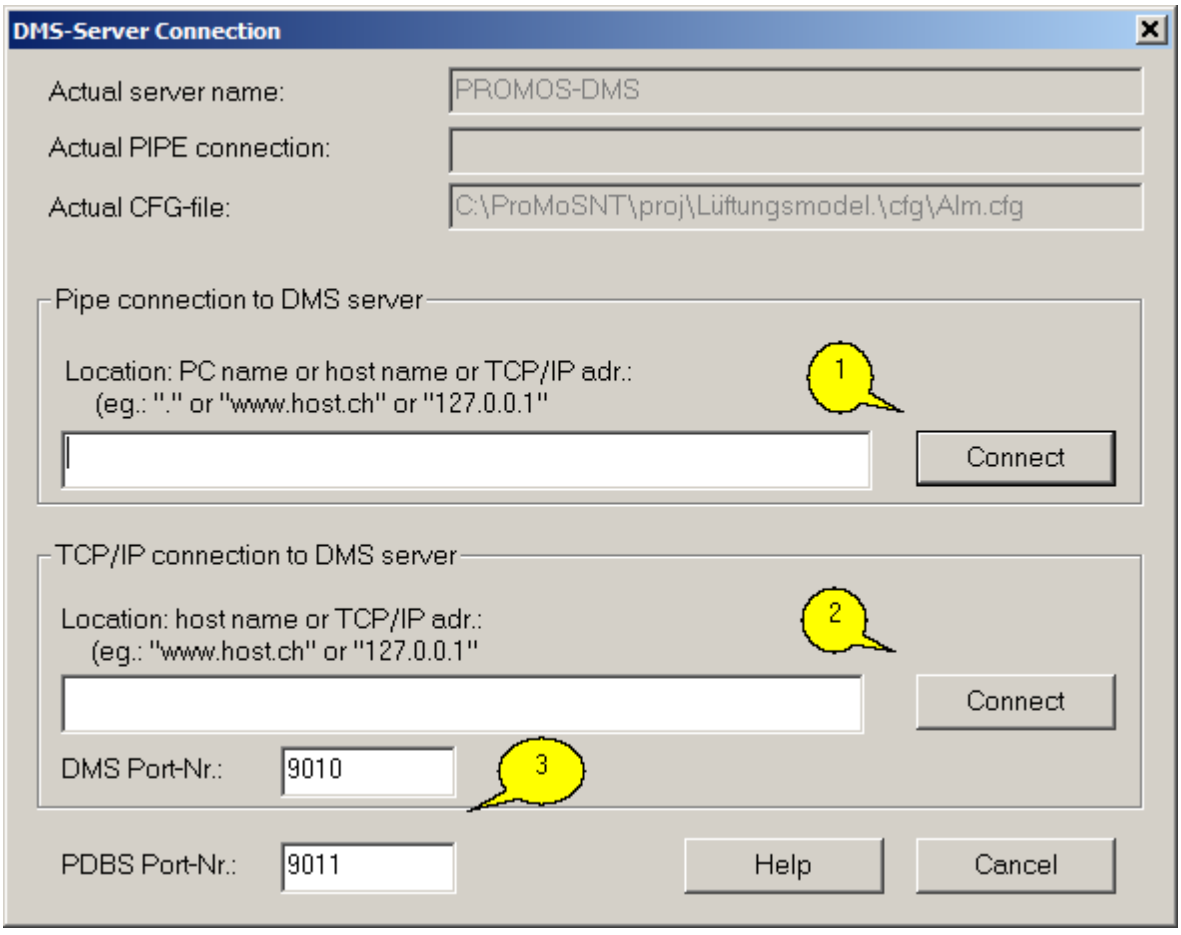


1) Location:
Name or IP address of the computer where the DMS is running. A dot is used here to refer to the local PC.

2) Settings:
Use this button to open the settings window. See the following image in this regard:

3) Backup DMS:
If the connection to the set DMS fails, the 2nd DMS takes over. At the moment, this is a quasi-redundancy. The two DMS do not exchange any data with each other.

4) Project settings:
The project path which has been set in the ProjectCfg is shown here.



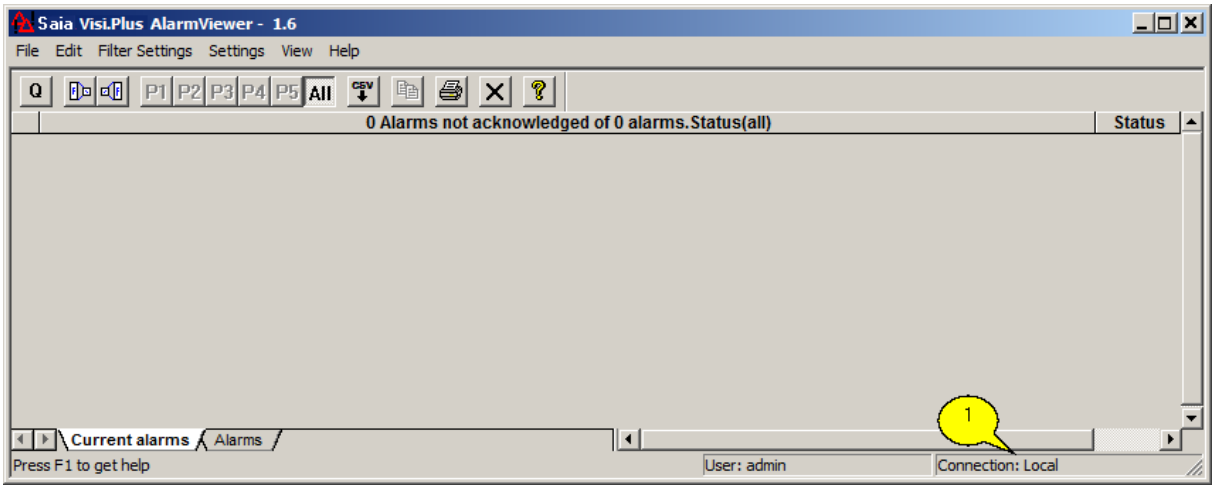
1) Pipe connection:
If a pipe connection is established, the IP address or name of the corresponding PC is shown here. Actuating the **<Connect>** button causes the GE to establish a pipe connection to the DMS. If the establishment of the connection fails after 3 tries, the attempt is aborted automatically.

2) TCP/IP connection:
If a TCP/IP connection is established, the IP address or name of the corresponding PC is shown here. Actuating the **<Connect>** button causes the GE to establish a TCP/IP connection to the DMS. If the establishment of the connection fails after 3 tries, the attempt is

aborted automatically.

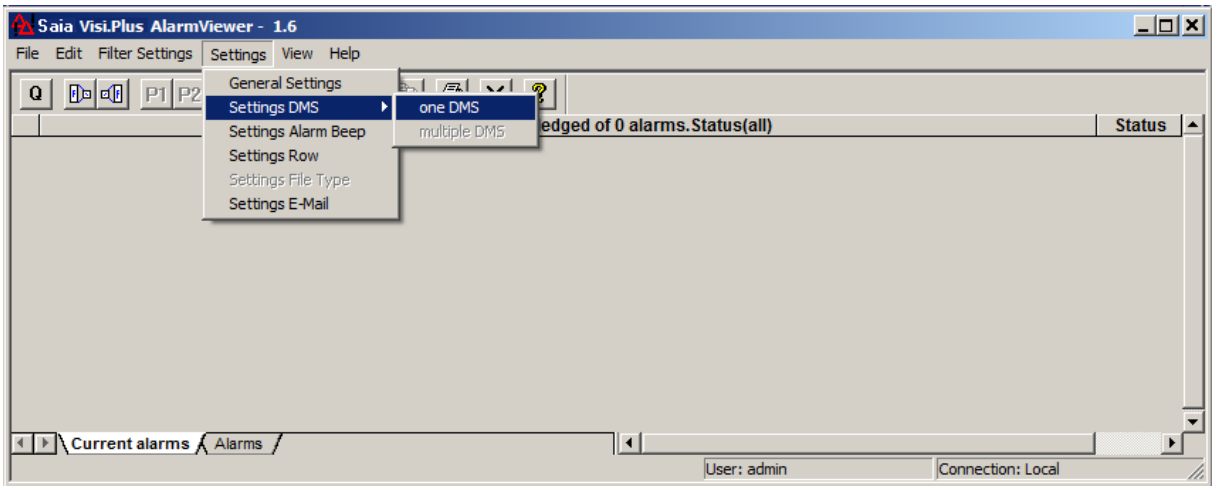
3) Port number:
The port number for the DMS as well as PDBS can be specified here. The default given is 9010 for the DMS and 9011 for the PDBS.

After the "OK" button is pressed, the Alarm Viewer appears, showing the data of the master PC:



1) Connection:
IP address of the master PC

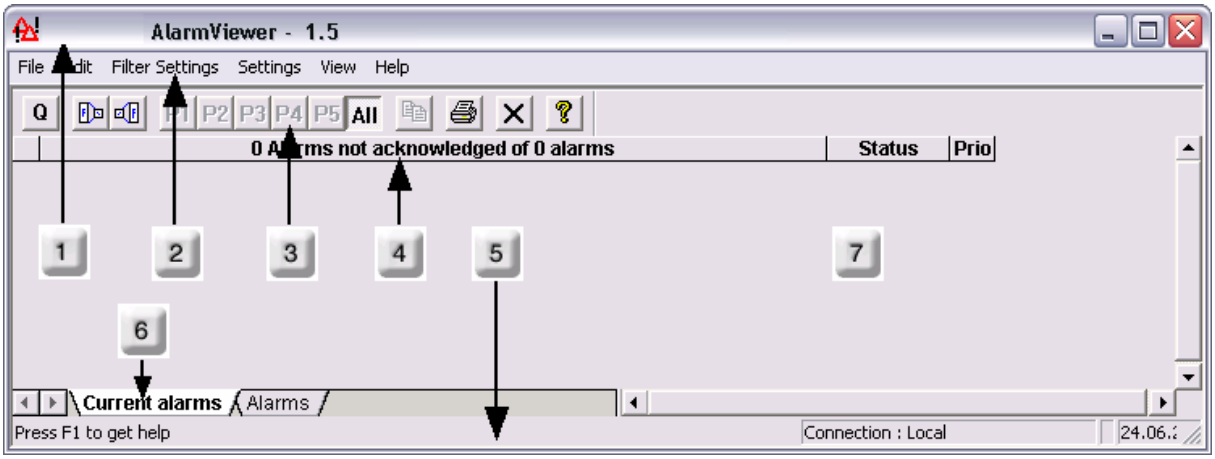
The network setting can also be made under the menu item "Settings -> DMS Settings":



It might be necessary for the drive to be mapped on the master PC in order for the pipe connection to operate trouble free. For more information about this, see the chapter Remote GE

10.4.2 The ALMView user interface

The window contents of the Alarm Viewer (ALMView):



1	Title bar	Contains the name of the current window.
2	Menu bar	The menus are sorted according to basic functions.
3	Toolbar	Buttons for important commands and functions.
4	Column name	Also shows a statistical overview of the alarms.
5	Status bar	Displays module status (program) and the meaning of the buttons in the toolbar.
6	Switching tab	The Alarm Viewer offers four different views that you can switch between ("Current Alarms" and "Alarms" as well as "Current Maintenance" and "Maintenance")

The Maintenance (or Service alarms) tab is only active if at least one maintenance alarm has been set in the PET.This can be defined in the Alarm column in the PET in the alarm configuration properties window.

7	Display window	The alarms are shown depending on the filter setting.

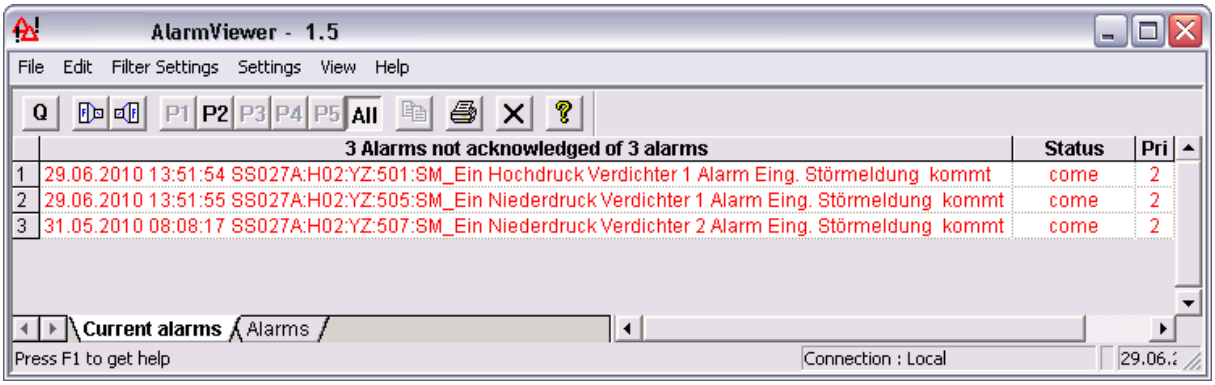
10.4.3 "Current Alarms" switching tab

In the "**Current Alarms**" view, all current alarms are displayed.

The alarms are typically displayed in three different ways:

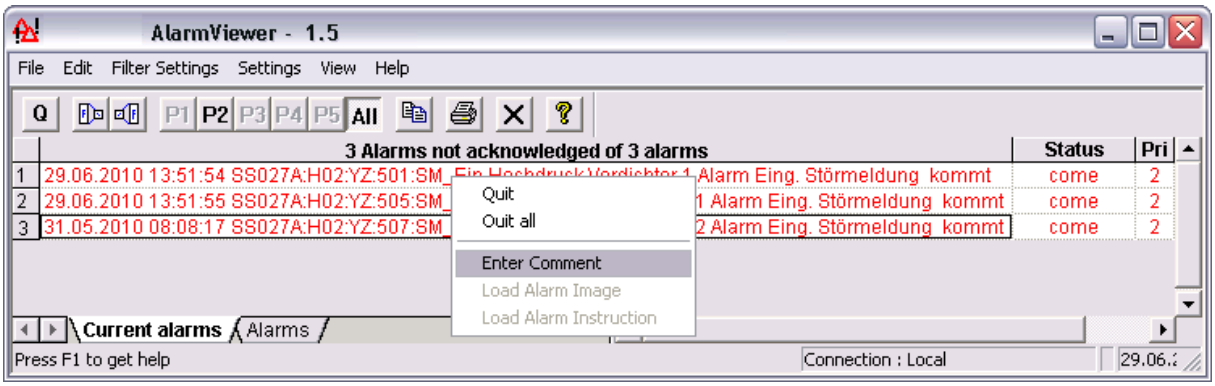
- Bright red: Pending alarm, not yet acknowledged, incoming status
- Blue: Alarm no longer pending, not yet acknowledged, outgoing status
- Dark red: Alarm pending, already acknowledged, ack. status

The colours can be defined in the menu Settings -> Line Settings. See the chapter Line settings



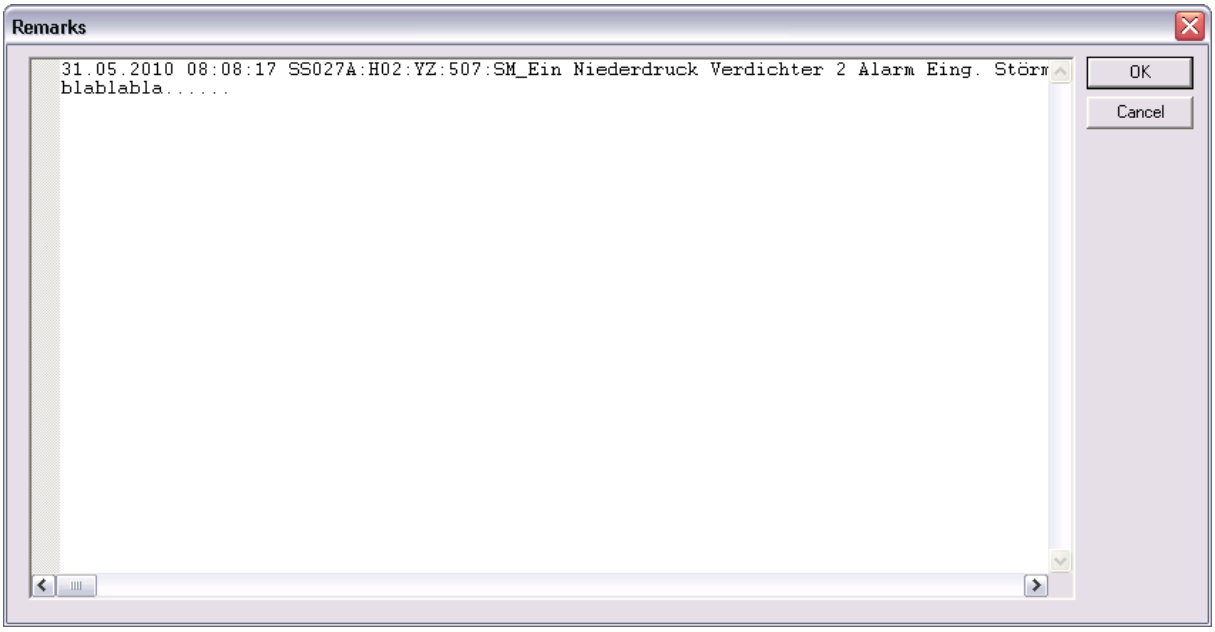
In the "**Current Alarms**" view, all alarms that have ***not been acknowledged*** yet are listed.

By pressing the right mouse button on an entry, a submenu appears, which enables ***acknowledgment*** of an individual alarm (**Ack.**) or ***all alarms,*** in addition to the conventional method.



It is also possible to enter a ***comment*** here.

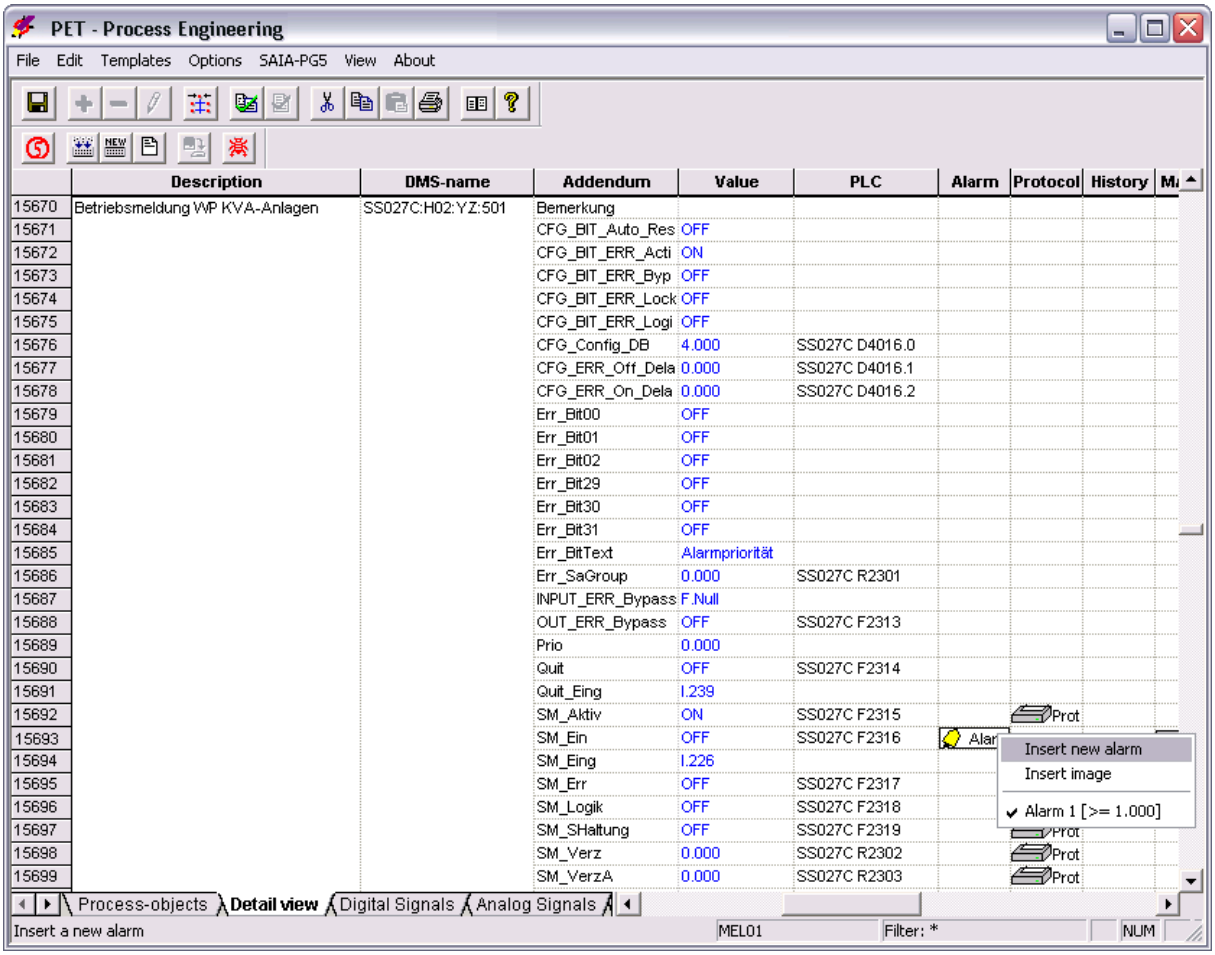
If ***Enter Comment*** is activated in the "Current Alarms" tab, a window in which any text can be entered as a comment opens and the text is accepted using **<OK>**. These comments may be useful, for example, if research into past alarms must be carried out. The comments can be called up at any time in the Alarms Tab.



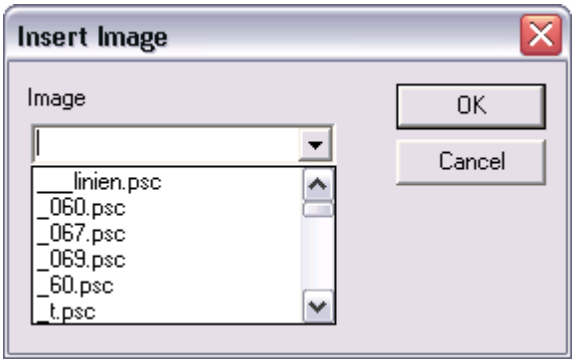
By clicking on the menu entry **Load Image**, an image for an alarm can be loaded in the Graphical Editor. The menu entry is only selectable if the alarm was appropriately configured in the PET.

The procedure for this is described in the following:

- Open the PET.
- Select the thumb index Detail View
- Find the data point and click with the left mouse button on the same line in the "**Alarm**" column.



- Select the entry **Insert Image** in the context menu. The following window appears:



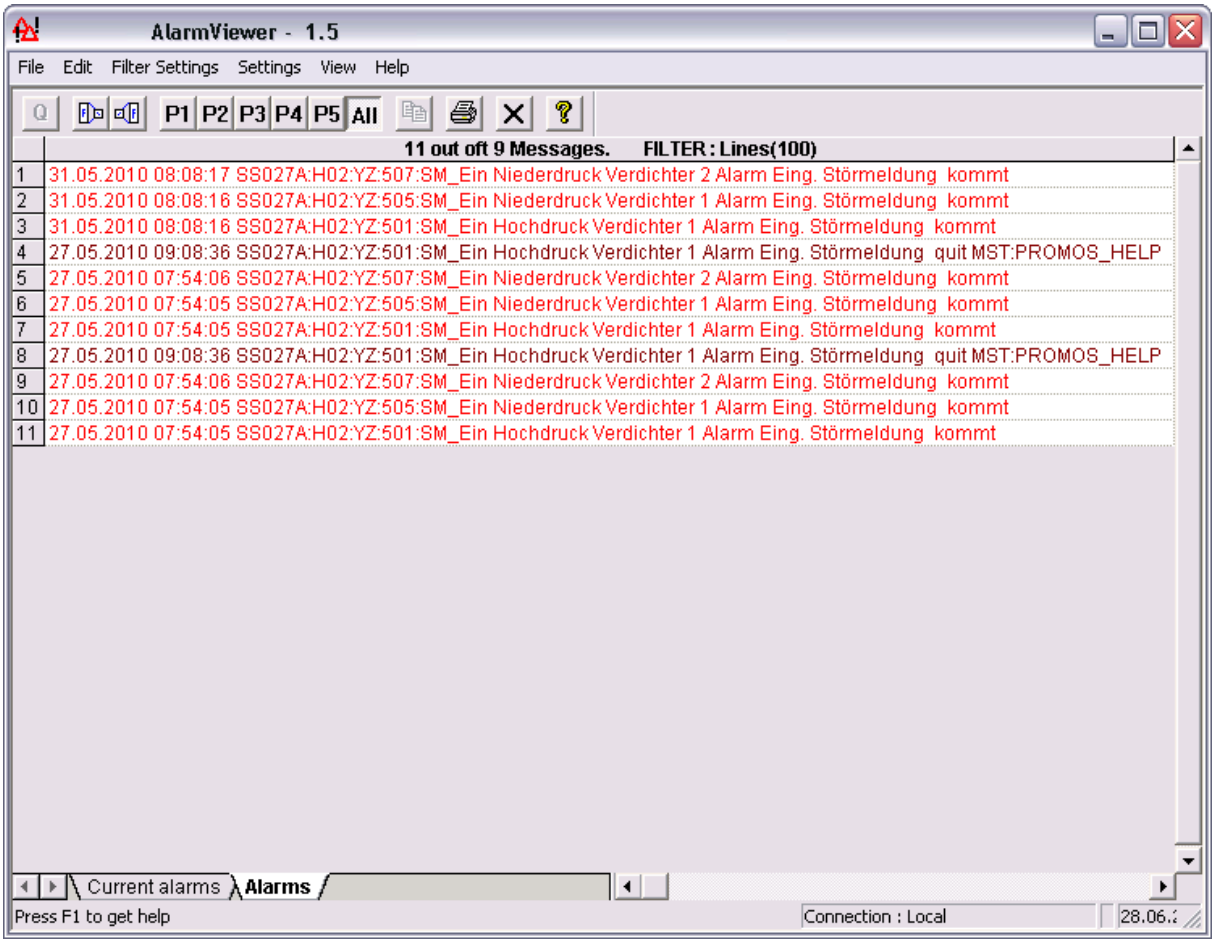
- Select the respective process diagram where the system object is located from the list in the dropdown menu.
- Confirm the entry with **OK**.

Now, if an alarm arrives for the data point that was just configured, the stored process diagram can be loaded in the GE by clicking the right mouse button in the Alarm Viewer and selecting the menu entry **Load Image**.

The Graphic Editor must be started.

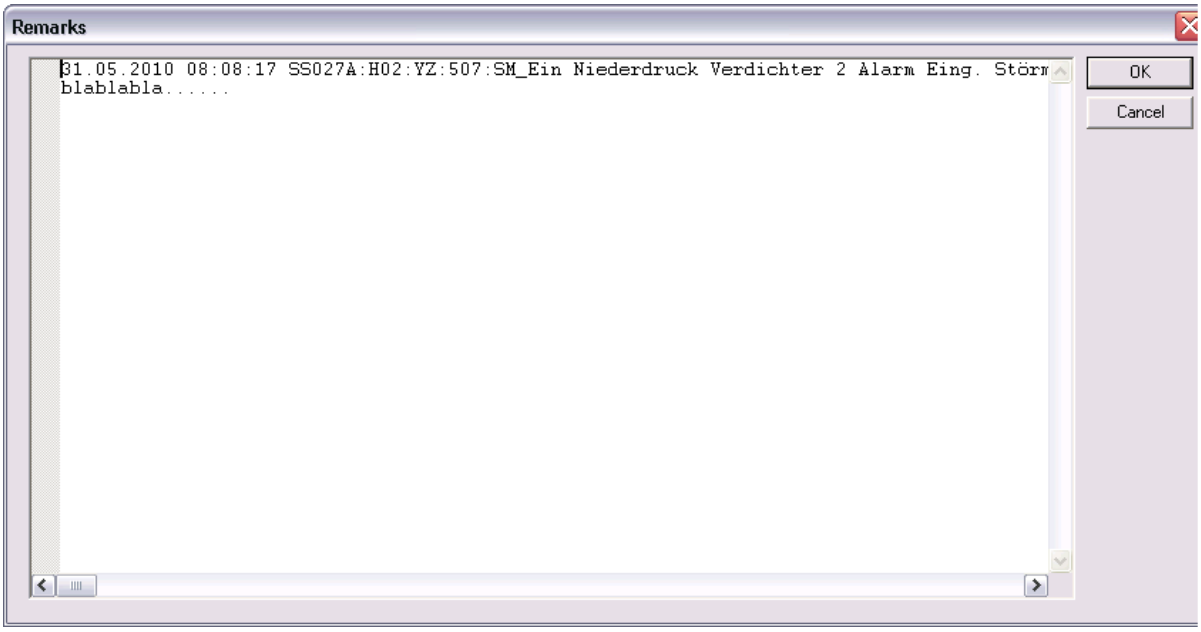
10.4.4 "Alarms" switching tab

In the "**Alarms**" view (see thumb index above the window status bar), all logged alarms are displayed:



The number of messages contained in the Visi.Plus database is shown in the column heading (in the preceding image there are 32 entries).

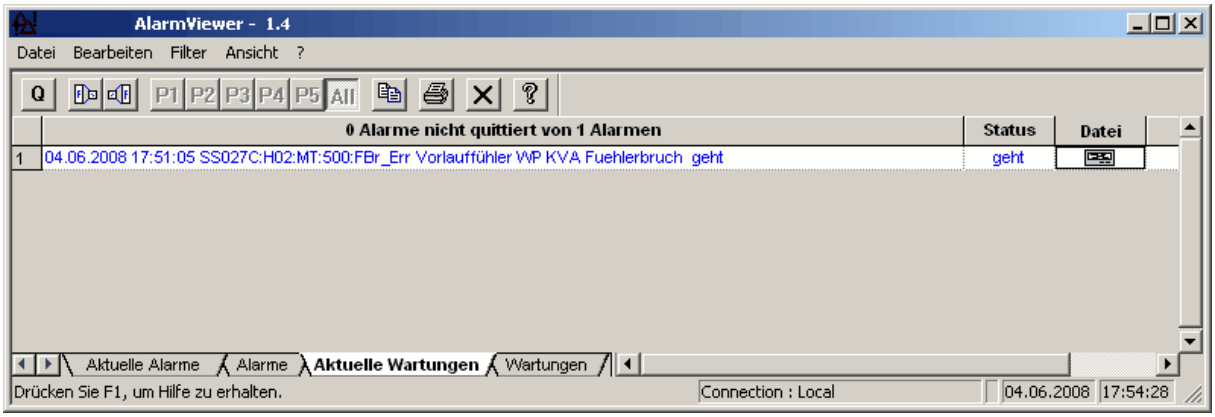
Comments that have been entered can be viewed in the "Alarms" tab by right-clicking on an entry.



*If **no** comment has been saved, the following message appears when the right mouse button is pressed:*

10.4.5 "Current Maintenance" switching tab

In the "**Current Maintenance**" view, all current maintenance alarms are displayed:



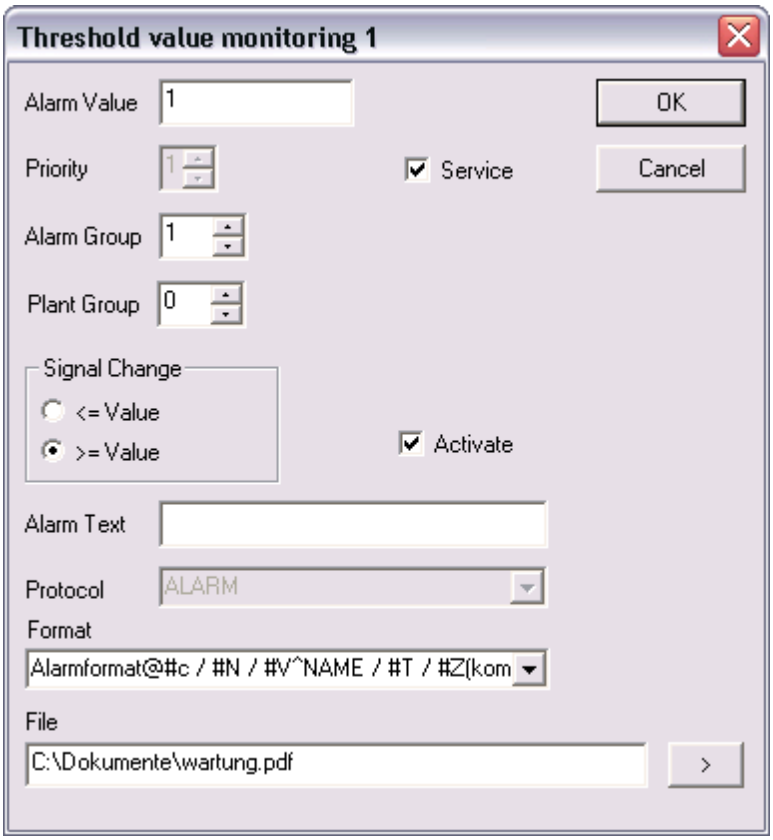
The maintenance alarms are handled the same way as normal alarms. The only difference is the following definition:

- As soon as the maintenance checkmark is activated in the PET, no priorities can be entered. The maintenance alarm is automatically assigned Priority 6.

Maintenance alarms are used with devices which, for example, must be replaced after a specific number of operating hours or when maintenance work is necessary after a specific number of operating hours.

The functionality is the same as with the "Current Alarms" switching tab. Therefore this chapter should also be consulted.

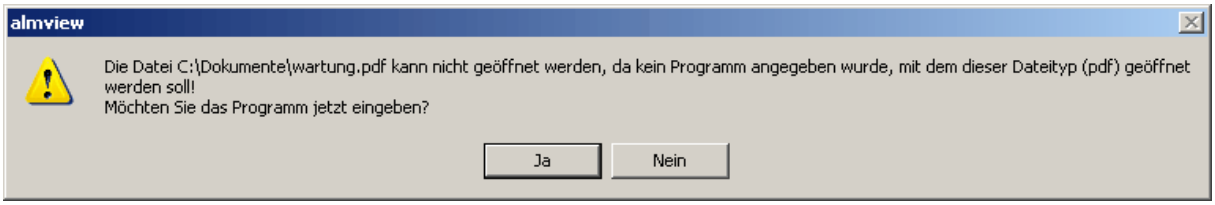
The following image shows the configuration window (PET) for threshold monitoring, which was defined for the maintenance alarm:



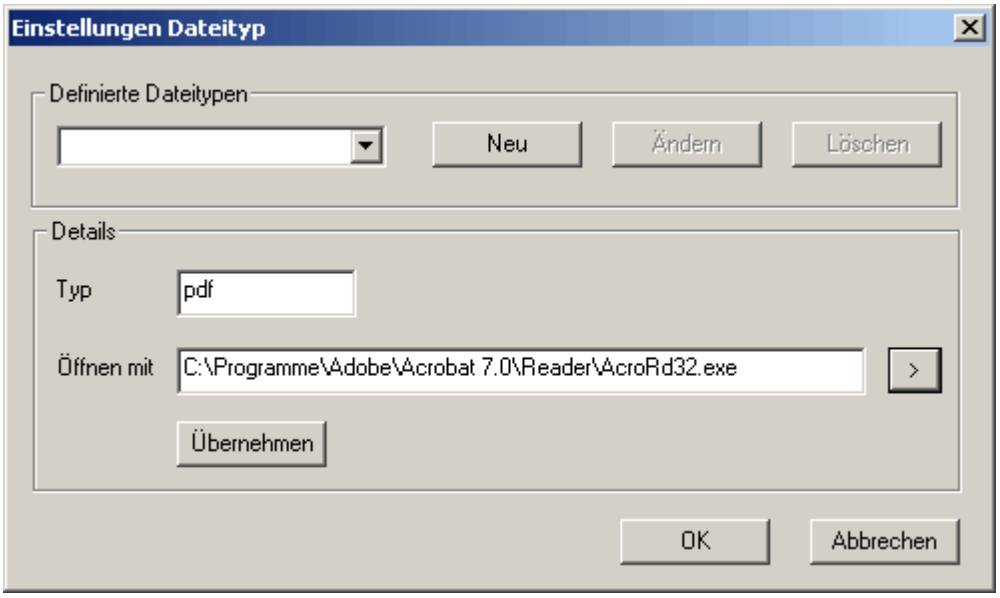
A file can be assigned to each alarm in the dialog under **File**, which may include information for the maintenance to be carried out, for example. This can then be opened in Alarm Viewer.

If the file is also to be displayed in a web browser, the PDF file must be saved in the www\pdf directory.

If such a file is available for the alarm, it is displayed in the Alarm Viewer by the icon in the File column. The first time you click on this icon, the following dialog appears:



Next click on **Yes** and insert the new file type as well as the program with which this file type should be opened:



<New>
Enables the input of a new file type and the program with which this file type should be opened. The button to the right of the **Open with** input field allows you to browse for the program.

<Change>
An entry selected under "Defined File Types" can be changed.

<Delete>
Deletes the selected entry under "Defined File Types".

<Adopt>
The newly created file type is accepted into the "Defined File Types" list.

<OK>
Adopts the entered values and closes the dialog.

<Cancel>
Closes the dialog without adopting the entered values.

After entering the values, the dialog is exited with **OK**. The icon must now be clicked again. The indicated file is opened with Acrobat Reader with the settings from the preceding dialog.

10.4.6 "Maintenance" switching tab

In the "**Maintenance**" view (see thumb index above the window status bar), all logged maintenance alarms are displayed.

The functionality is the same as with the "Alarms" switching tab. Therefore this chapter should also be consulted.

10.4.7 File menu

Alarms can be acknowledged, users can log in, and other operations can be executed in the "File" menu of the Alarm Viewer (ALMView).

10.4.7.1 Acknowledge

In order to acknowledge an alarm, the respective alarm must be selected by left-clicking on the relevant alarm title. Then the "**Acknowledge**" command can be used in the "**File**" menu or the "**Acknowledge**" icon in the toolbar can be clicked on.

It is also possible to acknowledge the current alarm by ***double-clicking*** it.

10.4.7.2 Acknowledge All

Using the command "**Acknowledge All**", all pending alarms are acknowledged.
The "**Acknowledge All**" command can only be used if the checkbox "**Global Acknowledgement with 'Q' Key**" is marked in the menu "**View**" **>** "**Settings**".

*In order to be able to use the "**Settings**" command in the "**View**" menu,*a user with administrator rights must *be logged in.*

10.4.7.3 Login <CTRL+L>

By default, any user can operate the process diagrams and enter or modify data.
If only specific users should have **access rights** to specific functions and inputs, the corresponding **levels** (groups) should be defined in the "**User Management**" module and the relevant graphics objects must be initialised accordingly.

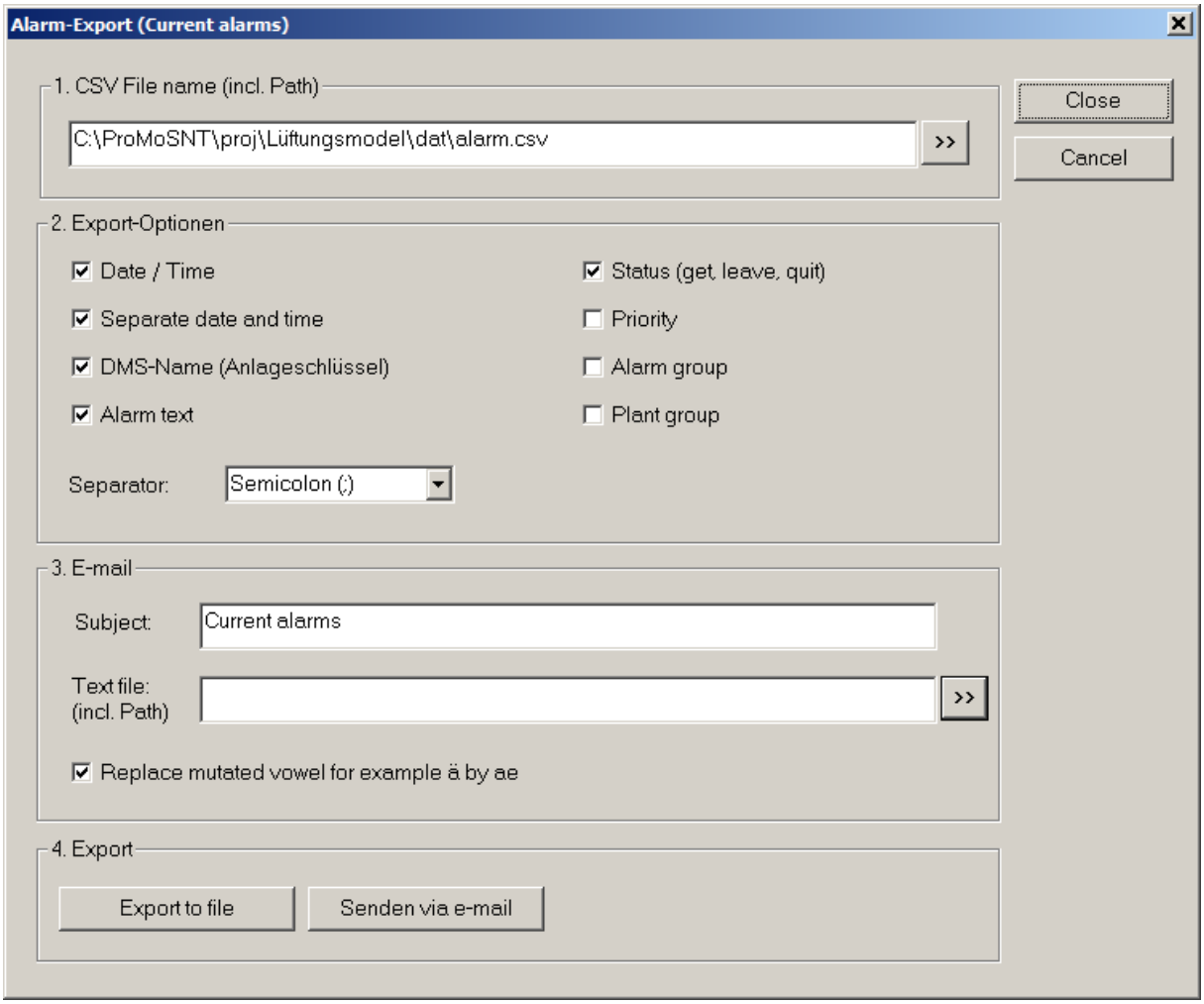
If the user would also like to make an entry that is protected by access rights (pages can also

be prevented from being called up), the user must log in using "**File > Login**" in the menu or the key combination **<CTRL+L>**. This causes an input dialog with the User Name and Password input fields to open. As soon as a user is logged in, all of the actions they perform are logged.

The login can take place at any time in Visi.Plus.
With appropriate configuration of the AlmViewer (see the chapter Settings) what system group an alarm belongs to is also taken into account. After successful login, the user is only shown the alarms whose alarm group they have access rights to.

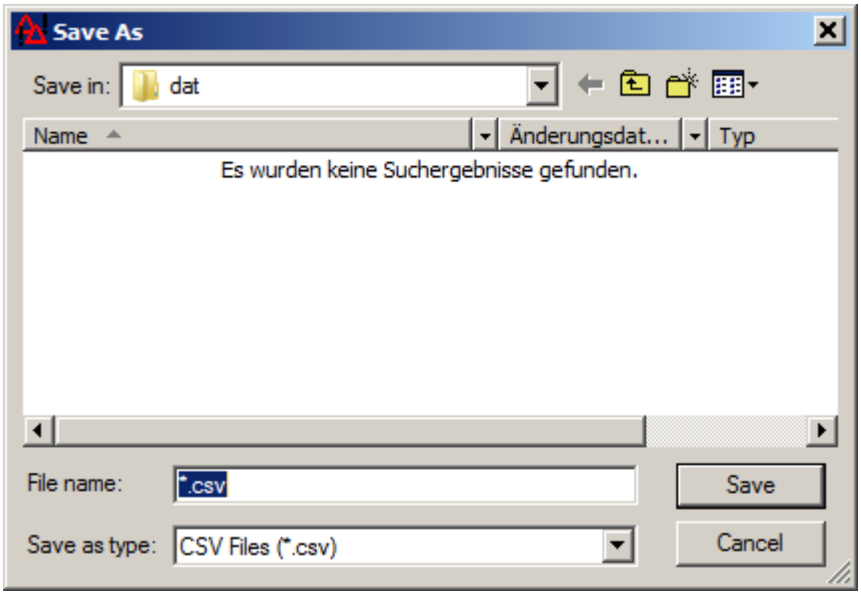
10.4.7.4 Export Alarm Data

In this menu, the alarms can be exported. It is possible to export the alarm data as a CSV file or send them via e-mail.



1) CSV file name:

The name and the path for the storage location can be entered in this text field.When the >> button next to the text field is clicked, the **Save file as** dialog window opens:



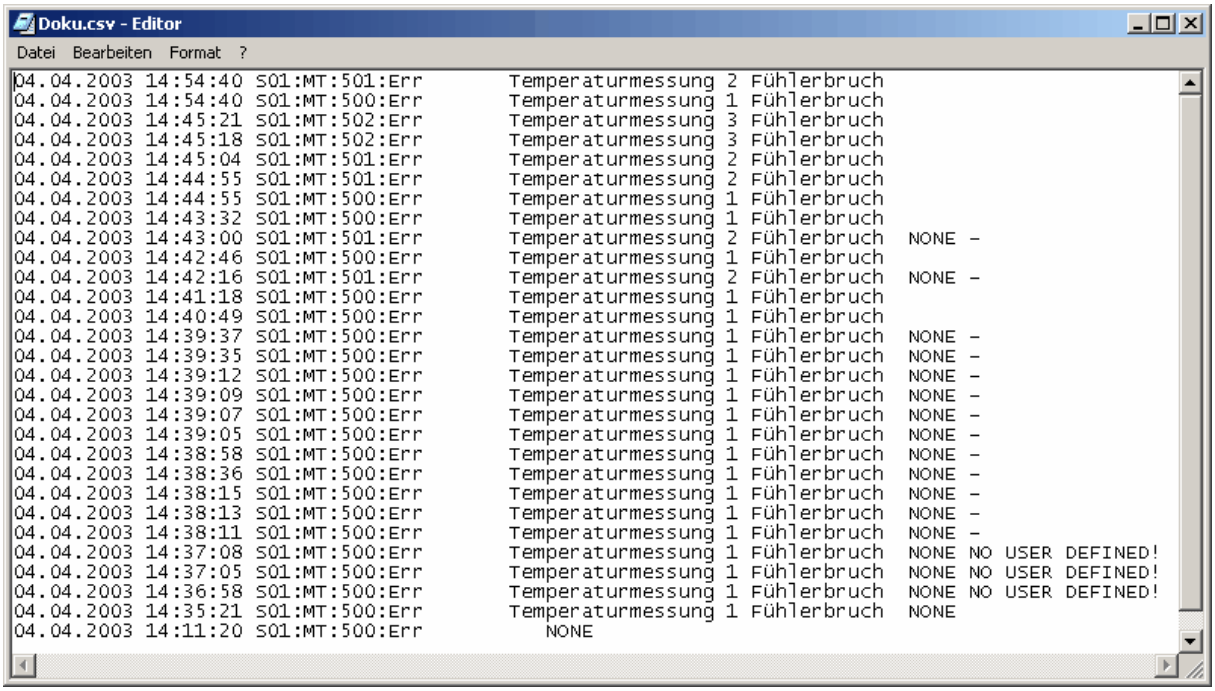
This dialog works in the usual way.

File type: Currently only the file format *.**csv** is supported.

2) Export options:
Various settings for the display of the exported alarms can be configured here. The configuration options correspond quite well to those found in the **"Settings -> General Settings"** menu for the display in Alarm Viewer. For more on this, see the chapter General Settings

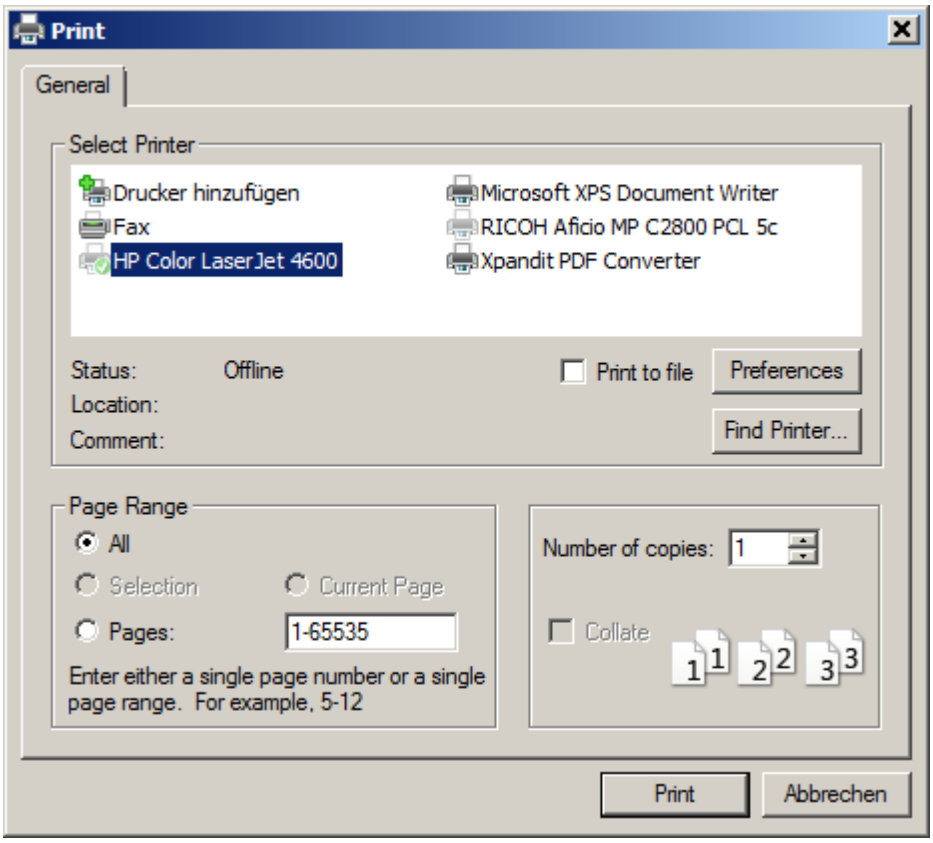
3) E-mail:
Here an attachment for an e-mail can be added and a subject line can be formulated. When the checkbox is marked, the special characters ä, ü, ö are replaced by ae, ue, oe respectively.

4) Export:
When the **<Send via e-mail>** button is actuated, an e-mail with the alarm data is sent. The e-mail configuration can be made under **"Settings > E-mail settings"**. When the **<Export to file>** button is actuated, a csv file is generated.
The **csv** file can be viewed by any text editor or can be easily imported to Excel tables.



10.4.7.5 Print <CTRL+P>

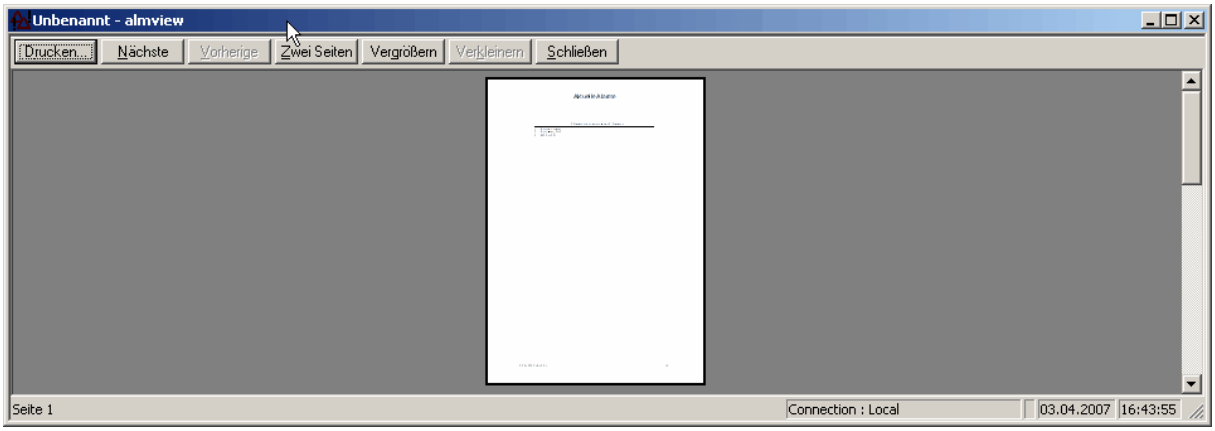
Alarm lists can be printed by clicking on the print icon  in the toolbar, via the menu command of the same name under "File" or with the key combination <CTRL+P>.



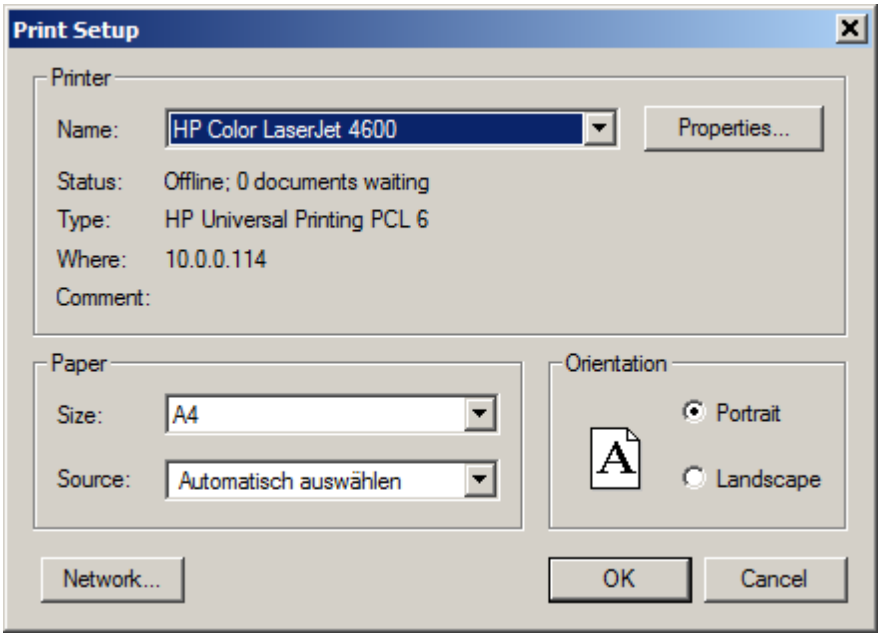
This is a PC operating system dialog and should be interpreted accordingly.

10.4.7.6 Print Preview

Before printing out an alarm list, it is possible to show a preview of the upcoming printout. This may be helpful for adjusting and optimising the borders, for example.



10.4.7.7 Printer Settings



This is a PC operating system dialog and should be interpreted and configured accordingly.

10.4.7.8 Exit

 This menu item exits the Alarm Viewer and closes the window.

10.4.8 Edit menu

Since alarms must not be edited, only the Copy function is needed.

10.4.8.1 Copy <CTRL+C>

In order to copy alarms to the clipboard, the "**Copy**" command can be used or the key combination **<CTRL+C>** can be pressed. The alarms can then be pasted into a text editor, for example, and edited further from there.

10.4.9 Filter menu

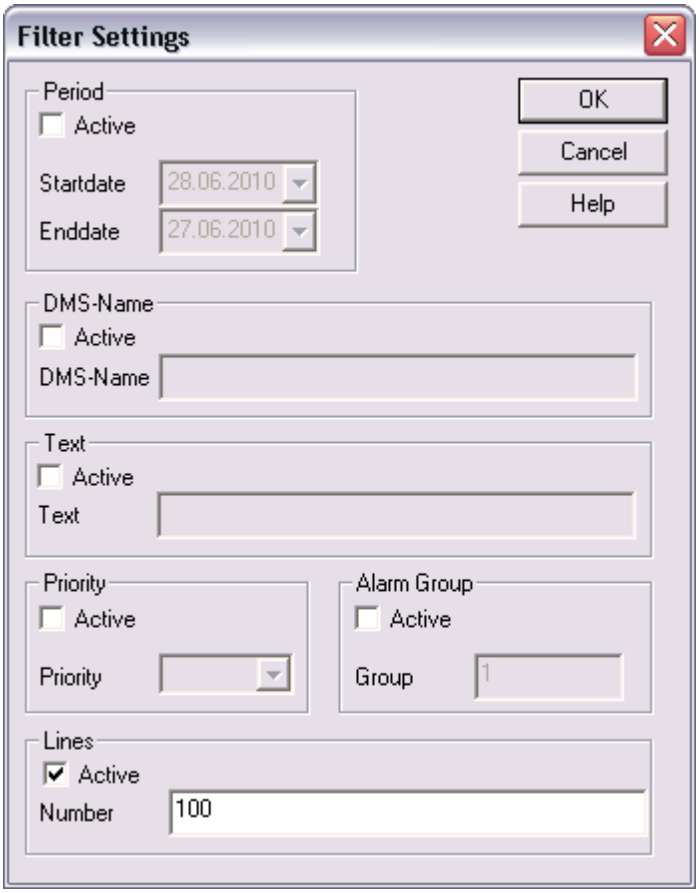
Filters are used to list only selected alarms, thereby limiting the list. For example, this includes filters for a time range, DMS names or part of it, priorities and alarm groups.

*The filter function lines only functions in the **Alarms** view.*
*In the **Current Alarms** view, this **filter rule** is not observed.*

10.4.9.1 Filter Settings

When a project contains many alarm entries, it is difficult to quickly find a specific alarm again. With the use of a **filter**, only the alarm entries that fulfil defined filter criteria are displayed. In order to set a filter, the menu command "**Filter > Filter Settings**" or the icon **Set Filter**  in the toolbar can be used.

The filter is only activated if the **Active** checkbox is marked. Only then is it possible to make entries.



1) Time Range:
Filters for the alarm entries between the indicated date and time values.

2) DMS Names:
Only the entries that match the indicated DMS name or part of it are displayed.
Placeholder "*" see below.

3) Text:
Only the entries that precisely match the text entered are shown.

Placeholder "*" (wild card)"
If the text entered is only part of the alarm name, it must be surrounded with an "*" on each end.

It is not possible to enter multiple words separated by blank spaces:
Example: /TXT=*Incoming sensor break **does not work!**

4) Priority:
Filters for alarms with the indicated priority. Has no significance for maintenance alarms.

5) Alarm group:
Filters for the indicated alarm group.

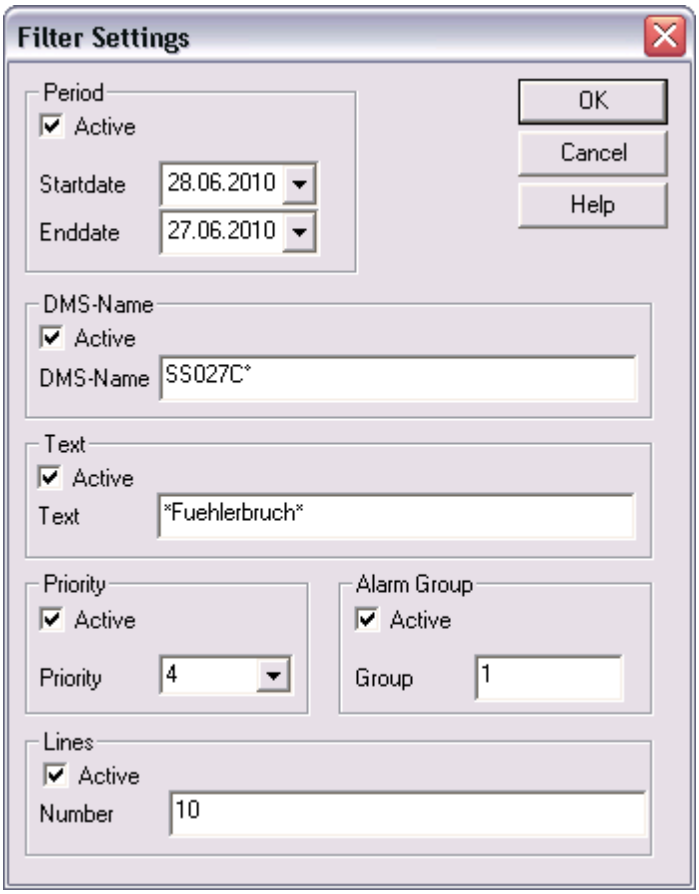
6) Status:
Filters the alarms according to your status. Depending on what is activated, either **coming**,

going or **acknowledged alarms** are displayed.

7) Lines:
Limits the number of lines that should be displayed.

Example:

If the filter is set as in the following figure:



it means that only the first 10 alarms are displayed for which:

- the alarm time is between 04/06/08 and 05/06/08
- the DMS name begins with SS027C
- the text contains the word sensor_break at any point
- the priority of the alarm is 4
- the alarm belongs to Alarm Group 1

The asterisk (*) serves as a placeholder for DMS names and text strings, meaning any text can take its place. Along with the asterisk (*), the question mark (?) can also be used as a placeholder. Any single character can then appear in its place.

Example:

DMS Name Filter = ??027C*

means that

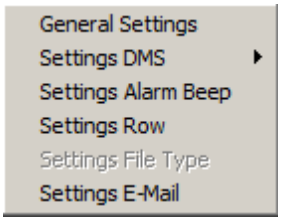
- the first two characters may be any character
- 027C must follow the first two characters
- any text can follow 027C.

10.4.9.2 Delete Filter

In order to delete all set filters, the menu command "**Filter > Delete Filter**" or the icon **Delete Filter**  must be used.

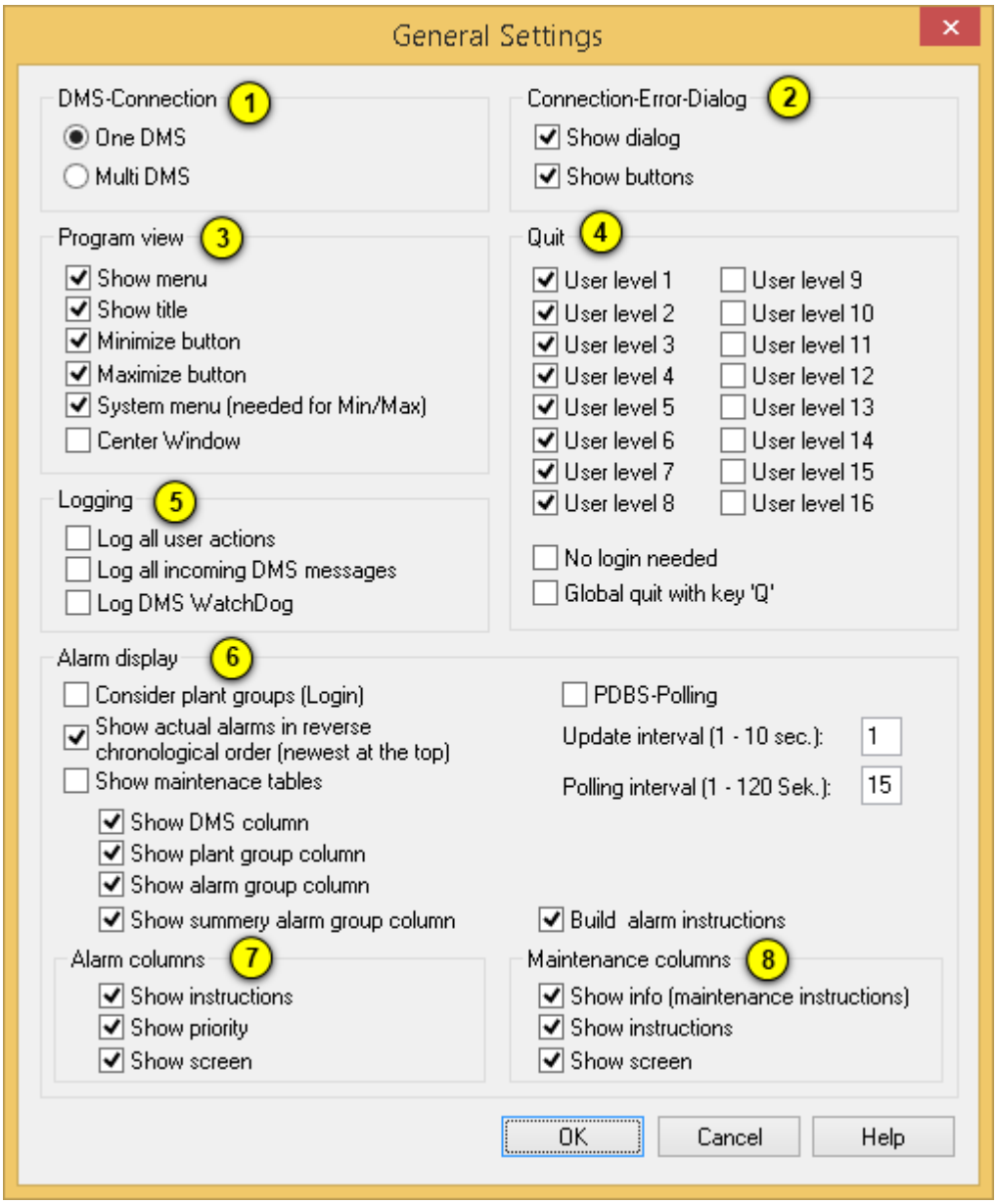
10.4.10 Settings menu

In the "**Settings**" menu, various settings for communication, display, alarm notifications, etc. can be made.



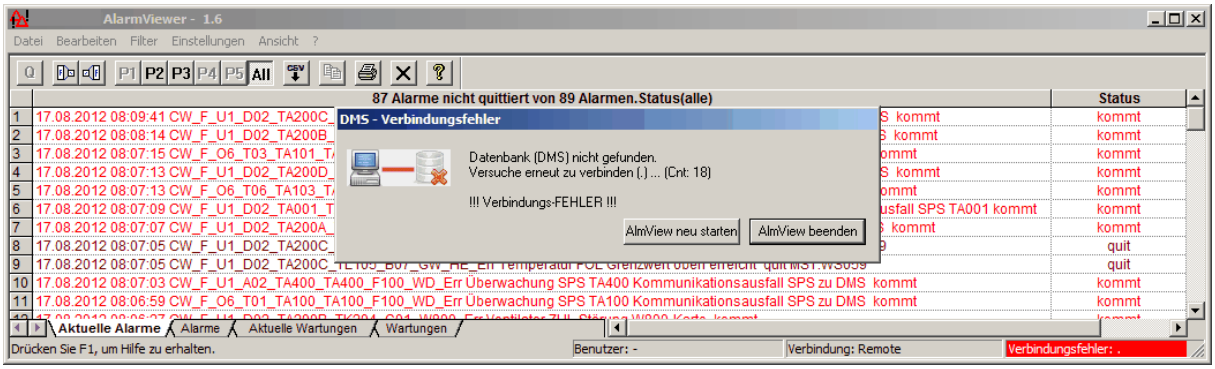
10.4.10.1 General Settings

This command can only be called up if a user with administrator rights is logged in.



1) DMS Link:
This sets whether the Alarm Viewer should connect to Mulit-DMS or only to one. The settings for the connections can be made in the menu**"Settings > General Settings"** and are described in the chapter DMS Settings.

2) Connection Errors Visible:
If these checkboxes are selected, a warning message is generated if the connection to the DMS is lost. Depending on the selection, either with or without a button. The message is generated 5 minutes after the interruption of the connection. The time can be specified under the item DMSTimeout in Alm.cfg of the corresponding project. The connection error is also displayed on the status line.



3) Program View:

Display Menu
Switches the menu bar on or off. In order to switch the menus on again, click on the icon for the Alarm Viewer in the title bar with the left mouse button. A pop-up menu appears in which the settings can be called up again.

Show Title Bar
Switches the heading column on or off in the "**Current Alarms**" view.

Minimise
Deactivates the minimise icon button (top left in the title bar of the Alarm Viewer).

Maximise
Deactivates the maximise icon button (top left in the title bar of the Alarm Viewer).

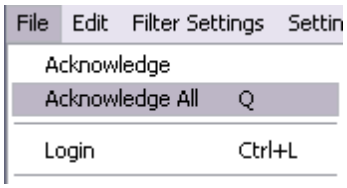
System Menu
Deactivates the icons buttons (in the title bar). This means the Alarm Viewer can no longer be closed.

4): Acknowledgement

Acknowledgement
Here you can set the access rights that a logged-in user must have to be allowed to acknowledge alarms. If, for example, User 2 is activated, it means that the user can only acknowledge the alarms if Access Level 2 was assigned to the user in pUser.

No Login Necessary
If activated, any user can acknowledge alarms without having to log in.

Global Acknowledgement with the "Q" Key
Enables the menu command "**File > Acknowledge All**"



as well as the  **Acknowledge All Alarms** button.

5): Logging:

Log all Actions (User)
If this checkbox is selected, all actions of the users are logged.

Log all Incoming DMS Messages
If this checkbox is selected, all incoming DMS messages are logged.

Log DMS Watchdog
If this checkbox is selected, all DMS watchdog messages are logged.

6): Alarm View

Account for System Groups (Login)
If this checkbox is marked, the system group of the alarm (definable in the PET) is compared with the system levels (definable in pUser) of the user on login. The alarm is only displayed if these match. If, for example, an alarm was assigned to System Group 2, it is only shown to the logged-in user if the user is entered in pUser with System Level 2. The idea in doing so is that the logged-in user is only shown alarms and log entries of system parts for which they are responsible.

The following example should demonstrate the utility of this filter:

A mountain railway consists of a mountain station and a valley station. Employee A is responsible for the valley station. Employee B is responsible for the mountain station. Therefore Employee A is only interested in the alarms for the valley station. Employee B, on the other hand, is only interested in those for the mountain station. The boss of A and B (e.g. the manager) wants to be informed about all alarms.

In order to achieve this, the system group is set to 1 in the PET for the alarms pertaining to the valley station. System Group 2 is selected for the alarms for the mountain station. System Level 1 must be activated in pUser for Employee A, System Level 2 for Employee B and both System Levels 1 and 2 for the Boss.

Now, when the Alarm Viewer is started, Employee A is only shown the alarms for the valley station after logging in, however those of the mountain station are not shown, because this employee does not have the appropriate access rights. These can only be viewed by Employee B or the manager.

 *If System Group 0 has been defined for an alarm, the access rights of the logged-in user are irrelevant. The alarm will be shown to them in any case.*

Display Current Alarms in Reverse Chronological Order
If this checkbox is selected, the alarms are displayed in reverse chronological order. This causes the oldest alarms to listed at the top.

Display Maintenance Tables
If this checkbox is activated, a tab containing "**Current Maintenance Tasks"** and

"**Maintenance Tasks"** is displayed in the Alarm Viewer.

Display DMS Column
If this checkbox is activated, the DMS name is displayed for the respective alarm or for the respective maintenance task (see the figure below).

Display System Group Column
If this checkbox is activated, the system group is displayed for the alarm or for the maintenance task (see the figure below).

Display Alarm Group Column
If this checkbox is activated, the alarm group is displayed for the alarm or for the maintenance task (see the figure below).

Create Alarm Instructions
The user must have administrator rights for this option.
When activated, every alarm data point **"..ALM:Alarm1"** has **"ExtAlmText"** appended to it, making: "**..ALM:Alarm1:ExtAlmText**"
The name of the RTF or text file can be entered here which describes the associated alarm or contains alarm instructions.

7): Current Alarm Column

Display Alarm Instructions
Switches the Alarm Instructions column on or off in the "Current Alarms" view. If the attribute "..ALM:Alarm1:ExtAlmText" = filename with instructions for the selected alarm exists in the associated alarm DP, a button is displayed in the Alarm Instructions column of the current line, which loads this file in a pop-up RTF viewer window.

Display Priority Column
Switches the priority column on or off in the "**Current Alarms**" view. Priority 6 is issued internally for maintenance alarms. It does not make sense to display the priority. Therefore the "**Maintenance**" view has no Priority column. For the same reason, selection of priorities through the toolbar is not possible either.

Priorities can be set or reset in the Alarm Viewer. To do so, click in the Prio column and a drop-down list opens automatically. It is possible to select priorities from 1 - 5. Priority 6 is reserved for maintenance and can only be set in PET.exe. If a new priority is set, it is written directly to the DMS and is also visible in the PET. Administrator rights are required to change a priority.

Display Alarm Diagram
Switches the direct jump labels to the corresponding screen on or off in GE. The screens can be linked automatically in the PET.

8): Current Maintenance Alarm Column

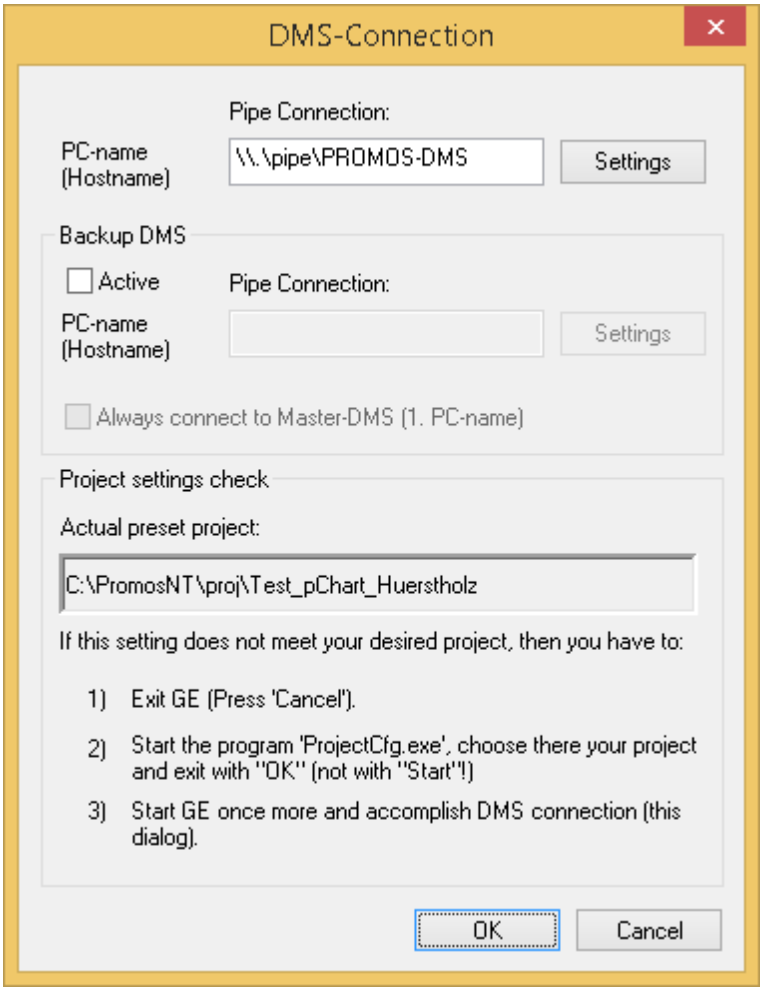
Display Info (maintenance instruction)
Displays the Info column for maintenance tasks and current maintenance tasks.

Display Alarm Instructions
Displays the Instructions column for maintenance tasks. If the attribute "..ALM:Alarm1:ExtAlmText" = filename with instructions for the selected alarm exists in the associated alarm DP, a button is displayed in the Alarm Instructions column of the current line, which loads this file in a pop-up RTF viewer window.

Display Alarm Diagram
Switches the direct jump labels to the corresponding screen on or off in GE. The screens can be linked automatically in the PET.

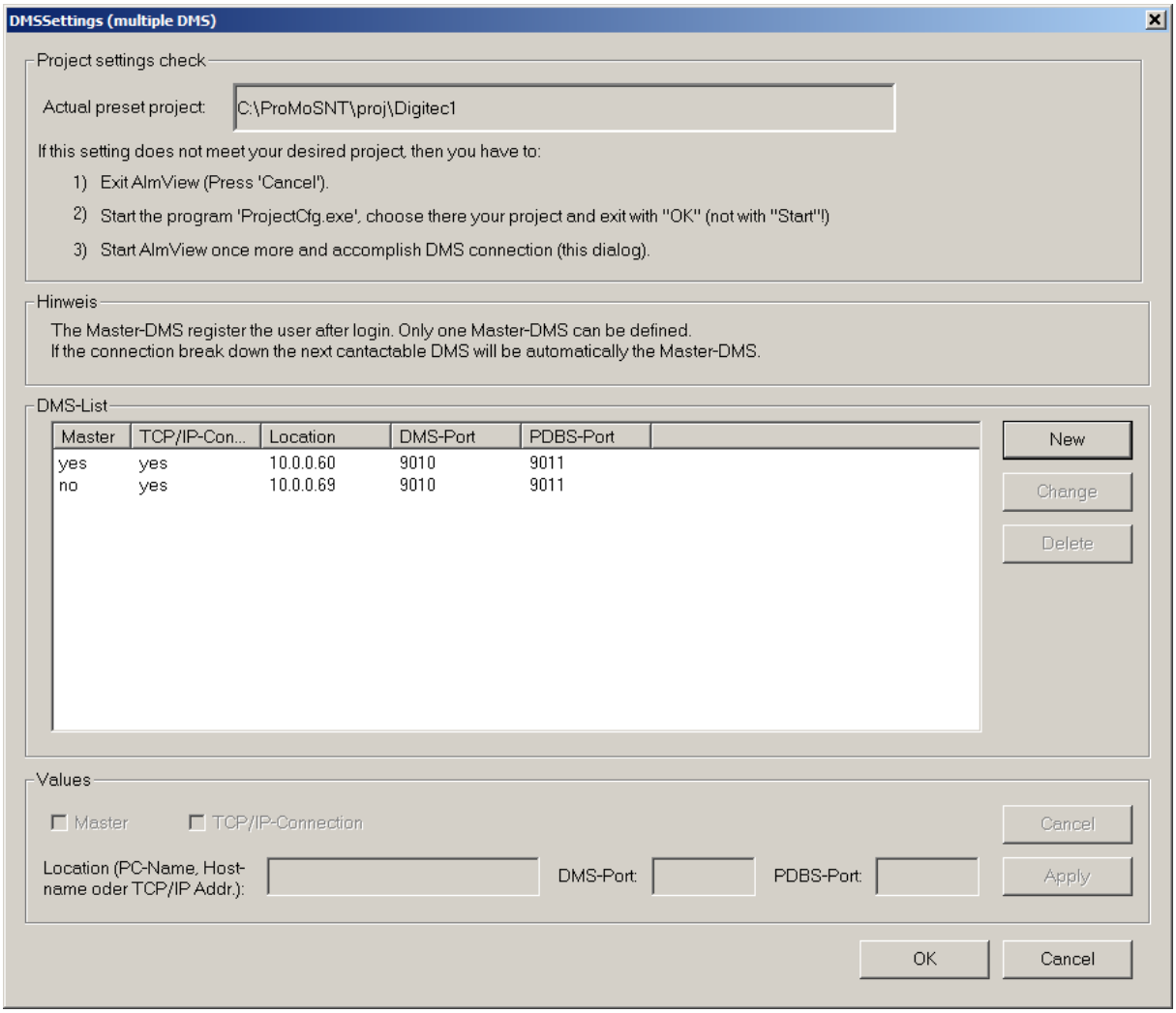
10.4.10.2 DMS Settings

If the checkbox **"Connect to a DMS"** under the menu item **"Settings -> General settings"** is activated, the following setting window appears under the menu item **"DMS settings"**:



The settings for the connection to the DMS can be made here. A detailed description can be found in the chapter ALMView remote.

If the checkbox **"Connect to several DMSes"** is activated, the following setting window appears:



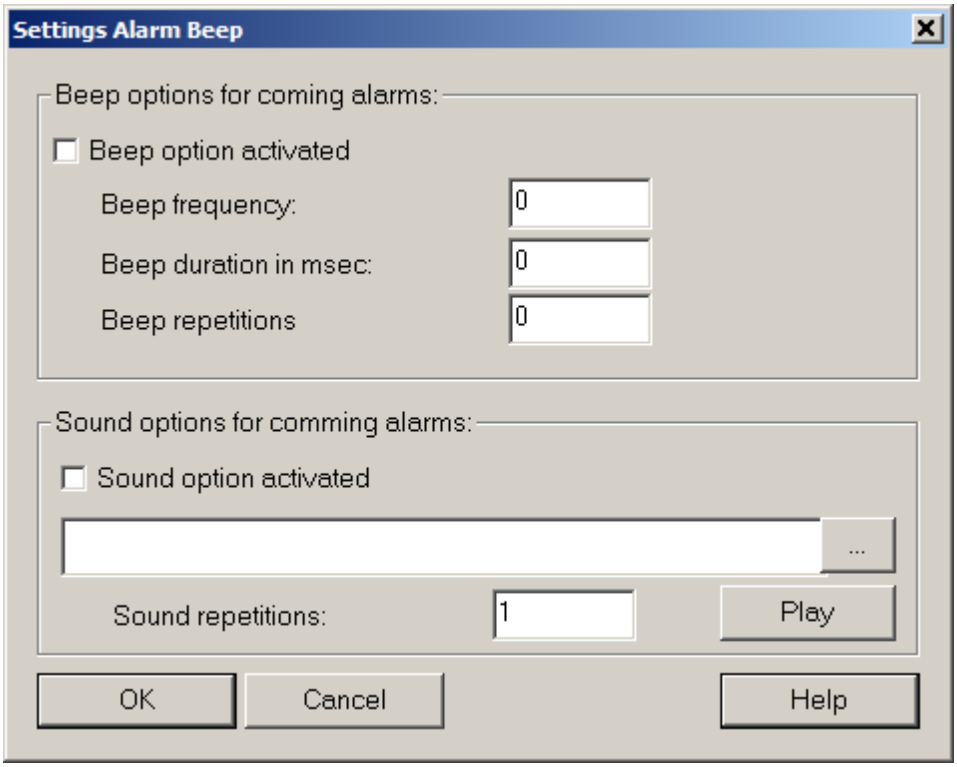
The function of several DMSes in the Alarm Viewer is the same as in the graphical editor. A detailed description of a connection to several DMSes can be found in the chapter Multi DMS.

The dot- sign "." for the connection is only working for Pipe. With TCP/IP it won't work. For local connection under TCP/IP the setting localhost or IP-Address 127.0.0.1 is used.

If the Multi DMS settings were configured in GE or pChart, the configuration is read from the MultiConn.cfg file at startup. If this was not the case, the GEConn.cfg file, if available, is copied to MultiConn.cfg. Otherwise, the original configuration is copied from the Alm.cfg to the MultiConn.cfg file. In all three cases, the UseCommMultiCfgFile entry in the Settings section is set to 1 in Alm.cfg. AlmView now uses the configuration from the MultiConn.cfg file. If the MultiConn.cfg file was newly created, it can now be used by GE and pChart. Therefore, only one configuration file has to be created for all three modules. If an individual configuration is required that is saved in the Alm.cfg file, the UseCommMultiCfgFile entry must be set to 0.

10.4.10.3 Alarm Beep Settings

The settings in regard to the behaviour of the signal tone (beep) are made here.



Beep Option Activated:
When this checkbox is marked, incoming alarms are signalled with a signal tone (beep).

Beep Frequency:
The pitch (frequency) of the beep can be adjusted here.
e.g. 400 for an audible, quiet, low ton, or e.g. 4000 for a somewhat loud, high tone.

Beep Duration
Here you can indicate how long the signal tone should last in milliseconds.

Beep Repetitions
Here you can indicate how many individual beeps the signal tone should consist of.

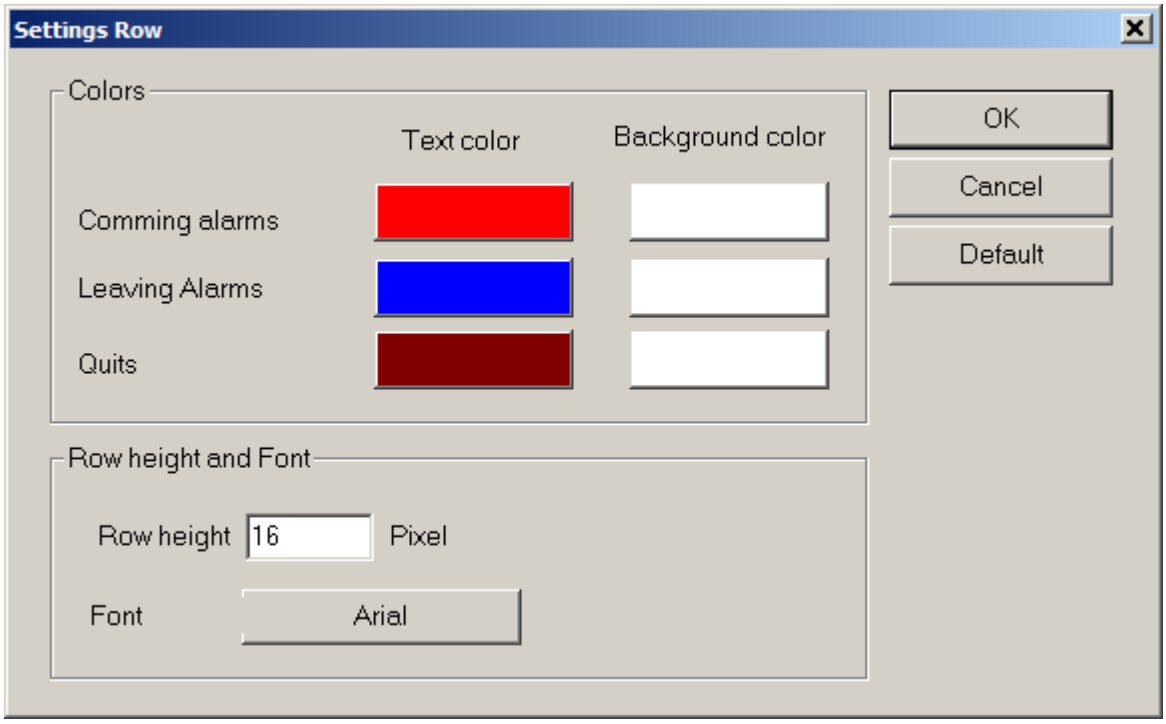
Sound Option Activated
Use this checkbox to activate the sound option. When activated, an audio file is played when there is an alarm. At the moment, audio files in the wav format are supported. The path for the desired file can be entered below the checkbox.
Only one of the two checkboxes may be activated!

Sound Repetitions
Specify here how often the audio signal should be played. Use Play to play the file for test purposes.

By clicking the **<OK>** button, the settings are adopted. If the **<Cancel>** button is pressed, the window is closed without adopting the settings.

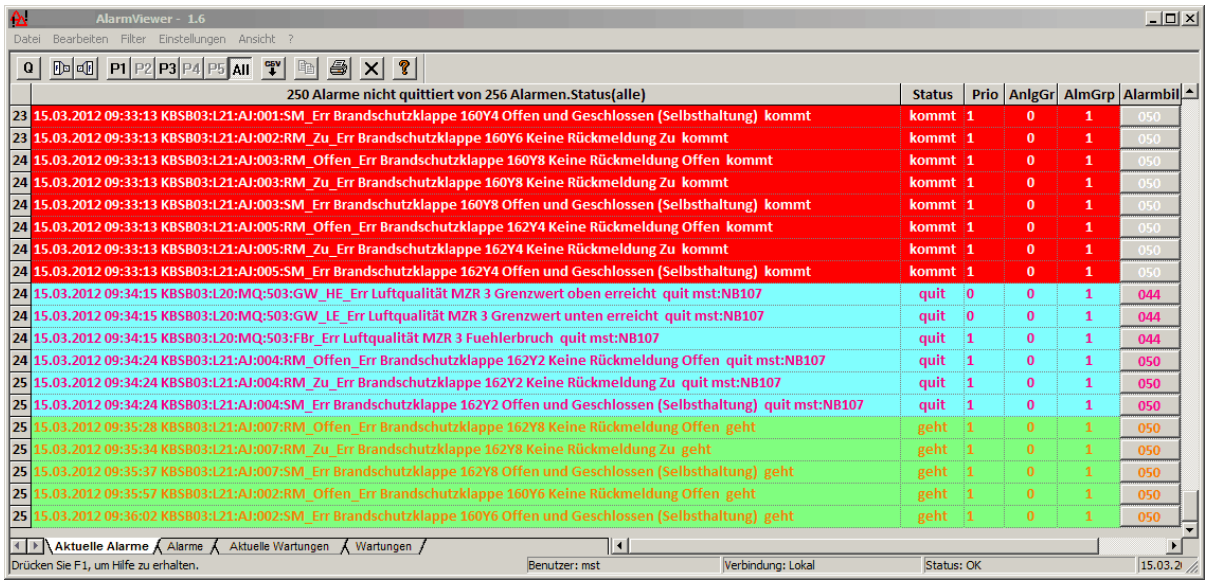
10.4.10.4 Line settings

The lines can be formatted in this setting window.



Colours

The colours can be configured individually according to the customer's wishes. The text colour as well as the background colour for incoming, outgoing and acknowledged alarms can be defined.



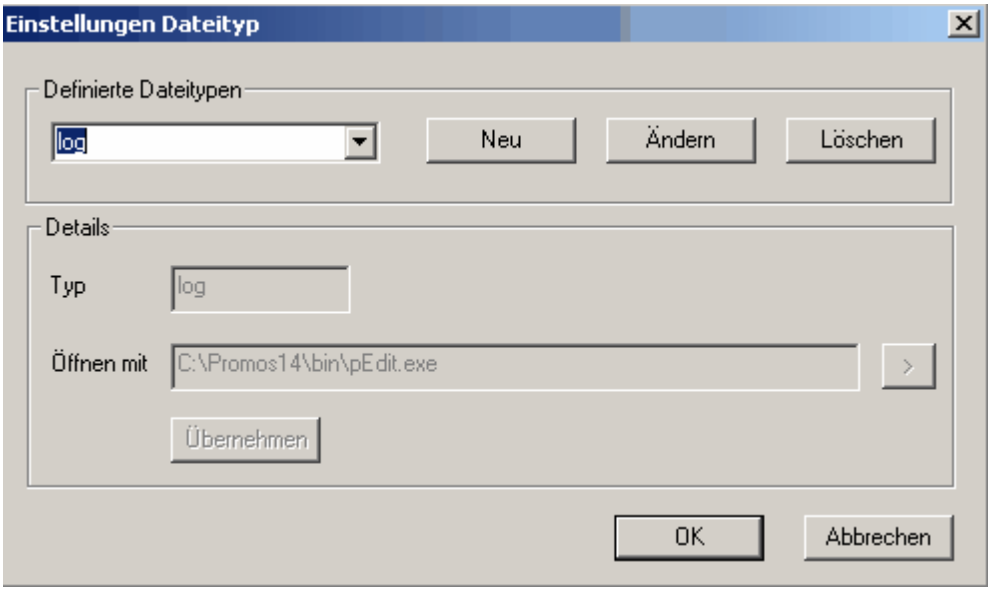
Line height

Specifies how large the line height of the individual alarm entries should be. This is useful when touch panels are used. This makes it easier to mark the alarm line.

Font

When the button is actuated, the configuration window for the font opens. Here it is possible to set the desired font, font style and the font size. The line height is matched to the selected settings automatically.

10.4.10.5 File Type Settings



File Type Settings are only available in the "Current Maintenance" register. As soon as a maintenance alarm has been defined in the PET, the menu item is activated and can be selected.

It is then possible to assign an alarm to circuit diagrams, other diagrams or even images and photos. Any program can be configured to use a file type.

Section: Defined File Types

<New>
Enables definition of a new file type in the Details area.

<Change>
Enables changes to a listed file type that has already been defined.

<Delete>
Deletes a listed file type that has already been defined.

Section: Details

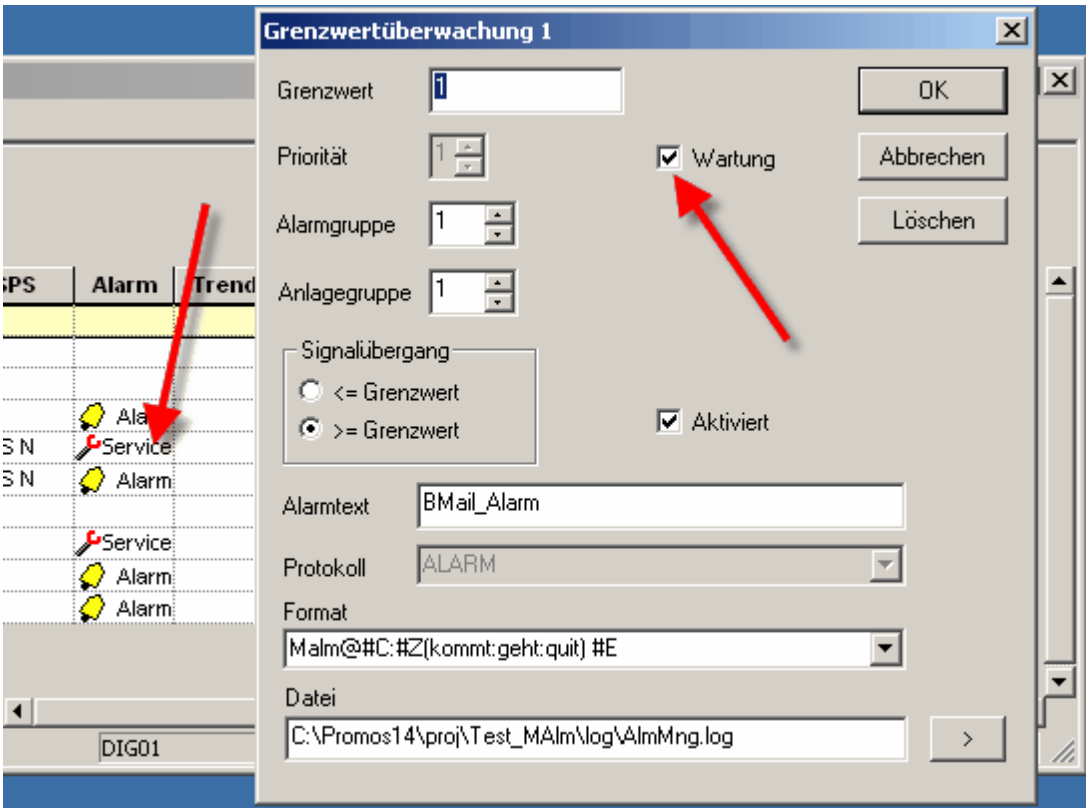
Type
Here you must enter what type of file it is.
(e.g. LOG for log files)

Open with
The directory in which the program for opening this file type is located must be indicated here.
(e.g. C:\Program Files\TextPad\TextPad.exe)

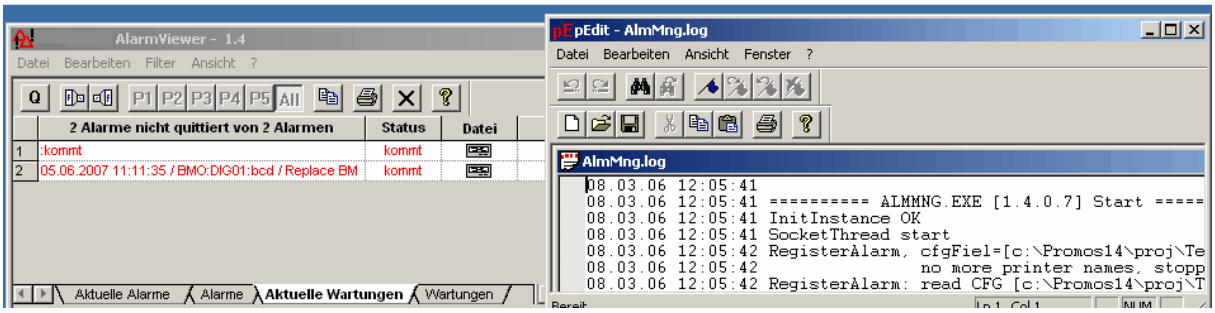
The following images demonstrate the use of the File Type function:

A data point is defined in the PET as a service alarm by setting the checkmark for Maintenance.

Here an additional file can be assigned to the maintenance alarm by indicating this in the "File" field: (e.g. here the log file of the AlmMng)



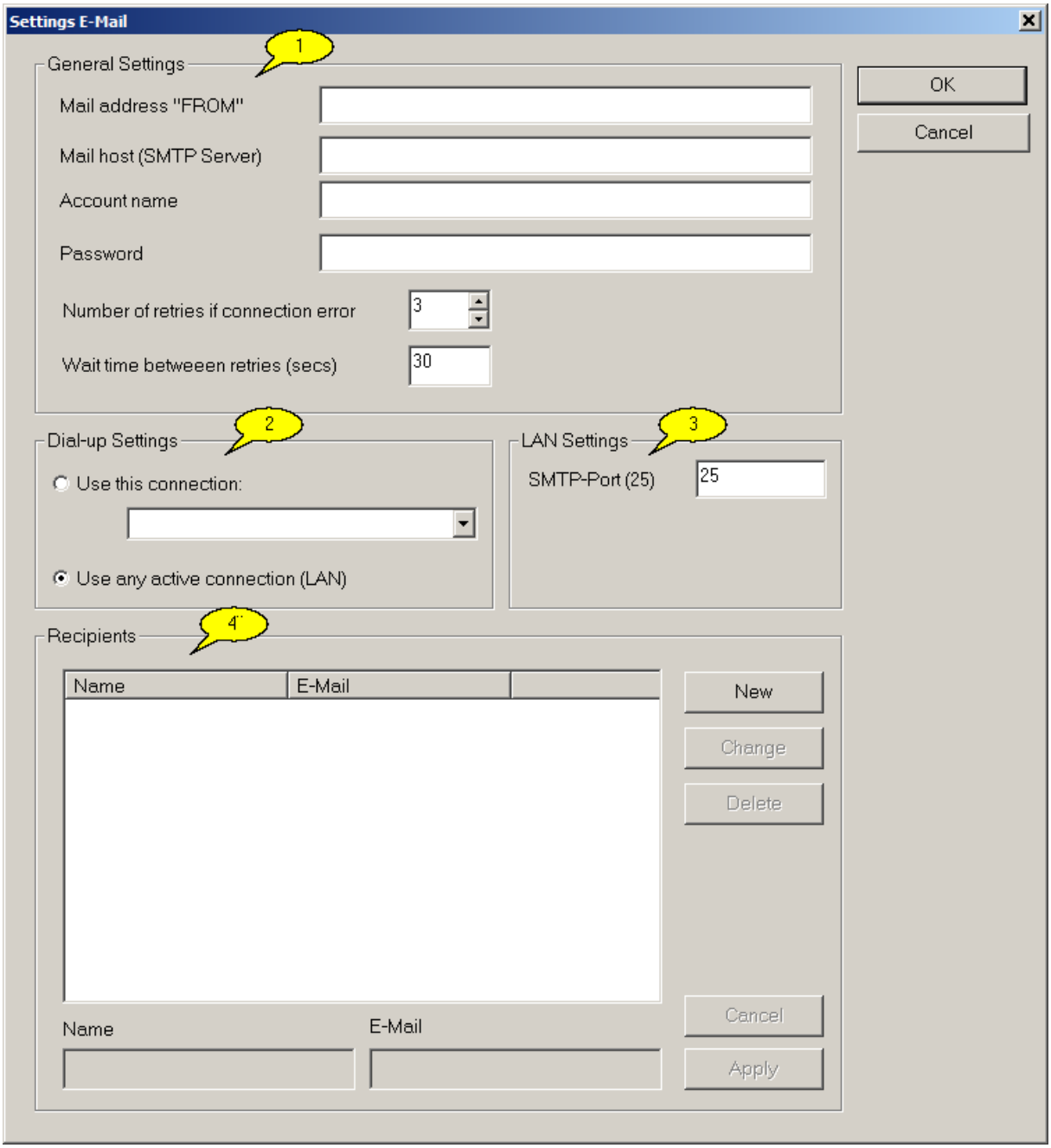
If this alarm occurs, it is shown in the Alarm Viewer in the register "Current Maintenance". Clicking on the icon in the "File" column opens the file defined under the alarm in the PET with the program defined for this file type under "File Type Settings":



Here the log file *AlmMng.log* was opened with the program pEdit.exe.

10.4.10.6 E-Mail Settings

This window can be used to make the settings for exporting alarms via e-mail. The data can be sent via the **"File > Export Alarm"** menu.



1): General Settings

The data settings for the sender can be made in the General Settings. In addition to the e-mail address, account name and password of the sender, the outgoing mail server data of the respective provider are also required. This information can be obtained from your provider. The larger providers are also listed on the Internet.

Example of account information for Hotmail:

E-Mail Address:	Example@hotmail.de
Mail Host (SMTP Server):	smtp.live.com
Account Name:	Test Example
Password:	12345

How often the connection set-up should be repeated can be specified in the case of connection repeats.

2): Remote Transmission Settings
The type of remote transmission must be specified here. The default is for the **"Use any Active Connection (LAN)"** checkbox to be marked.

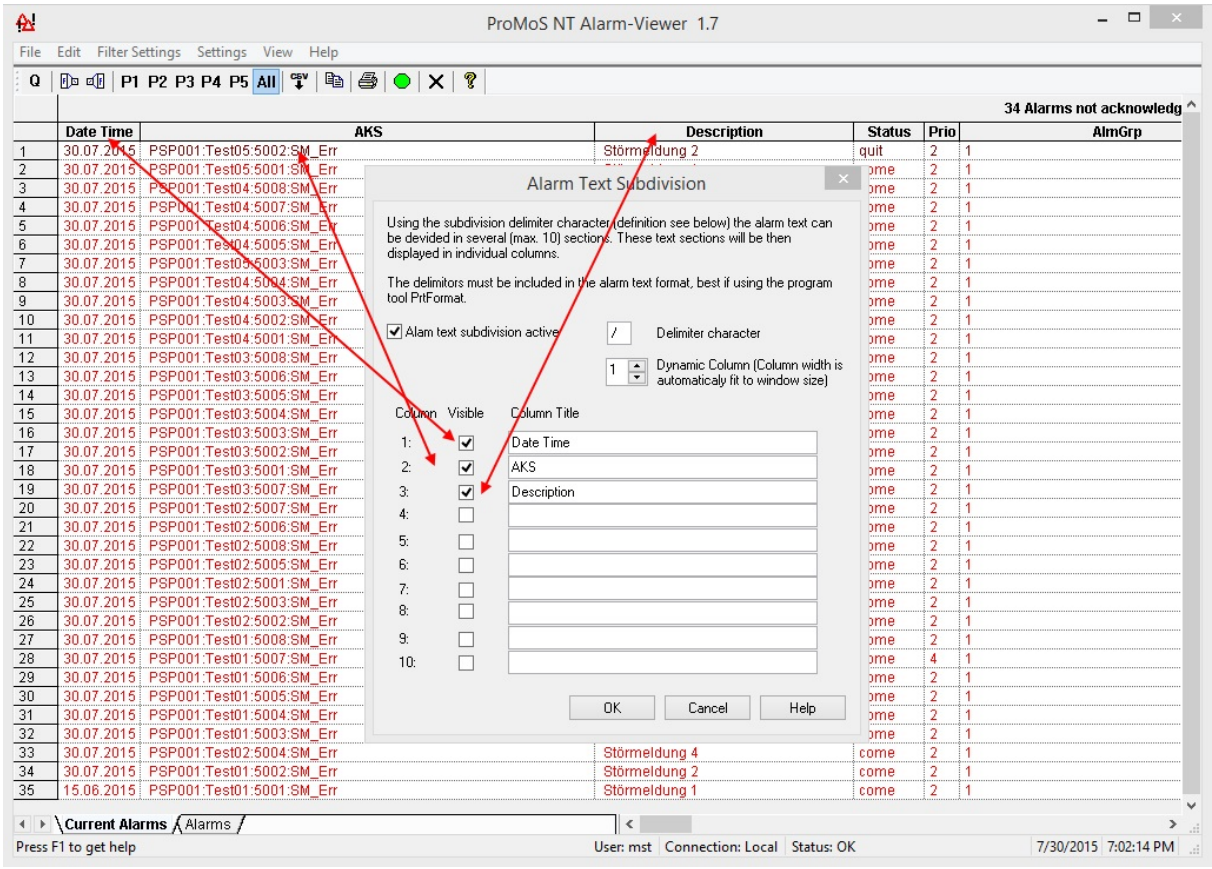
3): LAN Settings
The port at which the outgoing mail server is located must be specified here. For our example, it is 587.

4): Recipient
The recipients can be recorded here. A new recipient can be created here by activating the **"New"** button. To do so, the corresponding name and the e-mail address must be entered in the text field. An existing user can be modified using the **"Change"** button, while **"Delete"** removes a user.

10.4.10.7 Alarm text subdivision setting

Alarm text subdivision is used to define and display additional customer-specific columns in the AlarmViewer view.

In the alarm text subdivision dialog, max. 10 additional columns can be defined:



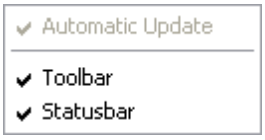
The condition for this is that the alarm text is divided into the same number of text sections

according to the number of visible columns.
The text sections are separated from each other with the selected separator character.

See 1st line in the above ex.: Alarm text format in DMS = "Alarm format@#c / #V^NAME / #V^description / #V^alarm-text / #Z(come:go:quit)"
- Separator is "/"
- 1st section = "#c" is displayed in the column "Date Time"
- 2nd section = "#V^NAME" is displayed in the column "Date Time"
- 3rd section = "#V^description" is displayed in the column "Description"

10.4.11 View menu

In the View menu, the symbol and status bars can be displayed or hidden and automatic refreshing can be activated or deactivated.



10.4.11.1 Status bar

Switches the status bar on or off.

10.4.11.2 Toolbar

Switches the toolbar on or off.

10.4.11.3 Automatic Update

The file Alm.cfg is found in the directory ***C:\Visi.Plus\proj\your_project_name\cfg***.

If this file is opened with a text editor and in the [Settings] area, the entry **MultAlmView=1** is added, multiple instances of AlmView are allowed. That means multiple Alarm Viewers can be opened simultaneously.

e.g. *1 AlmView* for current alarms and *1 AlmView* for all alarms. In this case, ***"Automatic Update"*** must be activated for the first AlmView for current alarms (see below).

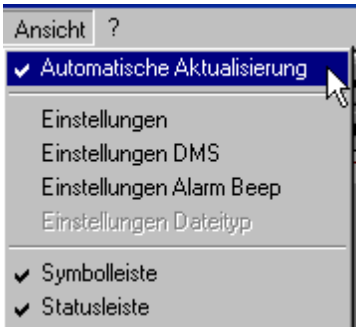
Extract from the Alm.cfg file with the required entry:

```
[Settings]
MultAlmView=1
[Settings Remote]
Redundant DMS=0
Master=.
Slave=
[AlarmColWidth]
Number=20
Alarm=588
Status=60
DMSName=200
Prio=30
AnlGrp=50
AlmGrp=50
```

MultAlmView=0 means normal operation, or only one instance.

If MultAlmView=1 is set, a new menu is activated:

View > Automatic Update (can be switched on/off)



If ***"Automatic Update"*** is deactivated, no alarms will be updated in real time in this instance of AlmView. That means displayed alarms and current filters will not be replaced by incoming and outgoing alarms.

With **MultAlmView=1** set, the menu "Automatic Update" is set to active in AlmView by default, so that any potential filters cannot be deleted.

10.4.12 Alarm instructions

A file which contains a description or alarm instructions for this (or these) alarm(s) can be assigned to one (or more) alarms. The file format ist .txt.

The storage location for alarm instruction files is the ..\<proj>\<projekt>\RTF folder. The filename is contained in the corresponding alarm DMS data point "**..ALM:Alarm1:ExtAlmText**".

10.4.13 Help menu

The **Visi.Plus online help** and current information about the Graphical Editor version are found in the "**?**" menu.

10.4.13.1 Help Topics

The Visi.Plus online help is accessed through the **"Help"** menu, via the corresponding <Help> button or by pressing the **<F1>** key.

10.4.13.2 About ALMView...

The version number of the Alarm Viewer and other information are displayed using this menu item.

Important
The version number must always be indicated in case of support queries.

We recommend our internet forum for support queries. This offers two advantages: First, help is available more quickly; and secondly, other users can benefit from the entries.

10.5 The Log Viewer (prtview.exe)

The Log Viewer (Log View) is used to show events that have been saved by the PDBS in a user-friendly way.

10.5.1 Starting the program

It is best to start the Log Viewer via an initialised button in the Graphical Editor or in the conventional manner, directly from the default **/bin** directory in the folder **c:/Visi.Plus**.

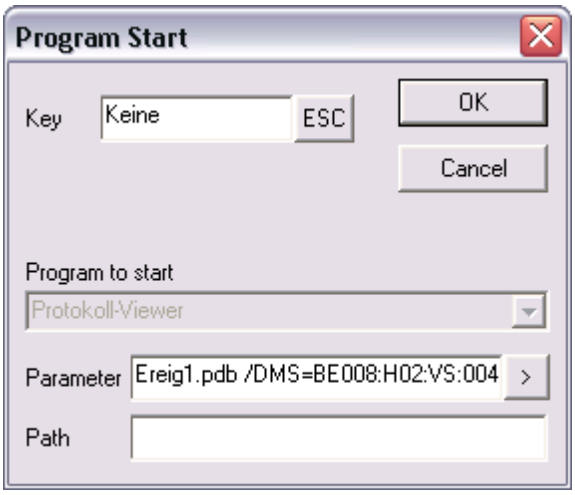
***Brief instructions for initialising the Log Viewer to a button**:*

- *After a "**Button**" graphical object has been placed in the process diagram, it must be clicked on once with the mouse in order to be selected.*
- *Then open the context menu with the right mouse button and select "**Attributes**"*
- *In the open Attributes window* under "**Action**" in the right column (Initialisation column), select "**Program Startup > Logging**" (prtview.exe from the **bin** directory is started)
- Start the Log Viewer with **<CTRL>** + **<button>**

Unlike the Log Manager, the Log Viewer does not have to be left running continuously. The events are only displayed with the log viewer.

If the Log Viewer (prtview.exe) is started with a log file specified in the Parameter section, this file is opened automatically.

Example of inputting parameters:

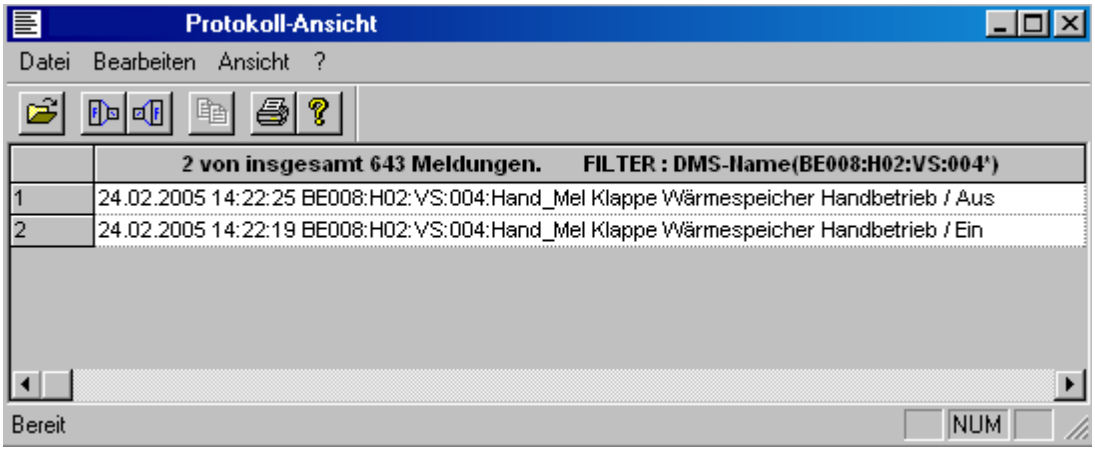


Parameter:
.pdb
Opens the log .pdb from the ***\dat*** *directory of the current project.*

*Event1.pdb /DMS=BE008:H02:VS:004**
Opens the log *Event1.pdb* with a filter on the respective DMS name (do not forget the asterisk * as a placeholder).

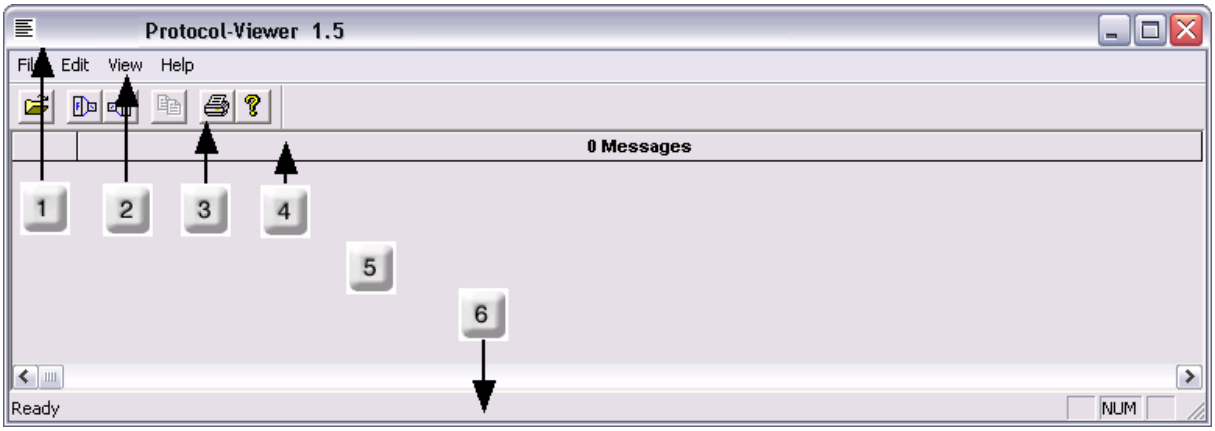
Path:
If the project is running on a different partition than Visi.Plus, the path of the corresponding directory can be entered here. In general, the storage location of each program to be started is entered here.

Now click on the initialised button in the GE and prtview.exe is started. The view may look like the following:



10.5.2 The PRT user interface

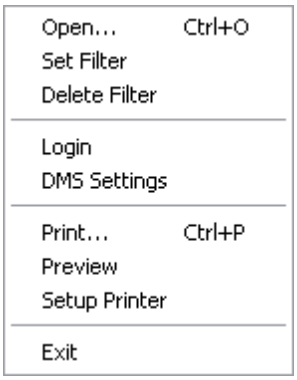
The user interface of the Log Viewer (Log View):



1	**Title bar**	Contains the name of the current window
2	**Menu bar**	The menus are arranged according to basic functions and are explained in the following.
3	**Toolbar**	Displays icon buttons for some important commands and functions available in the **PRT**.
4	**Header**	Indicates how many events are logged
5	**Debug View**	All registered events are displayed here sequentially.
6	**Status bar**	Displays module (program) status and the meaning of the buttons in the toolbar

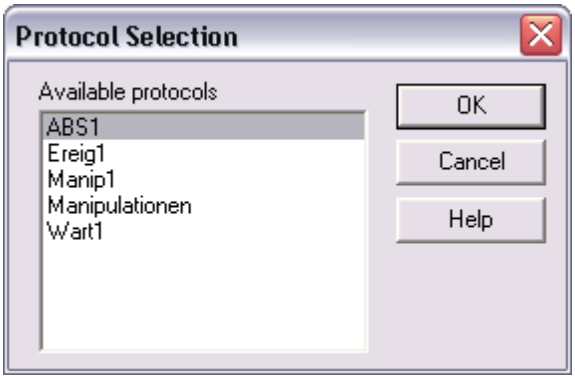
10.5.3 File menu

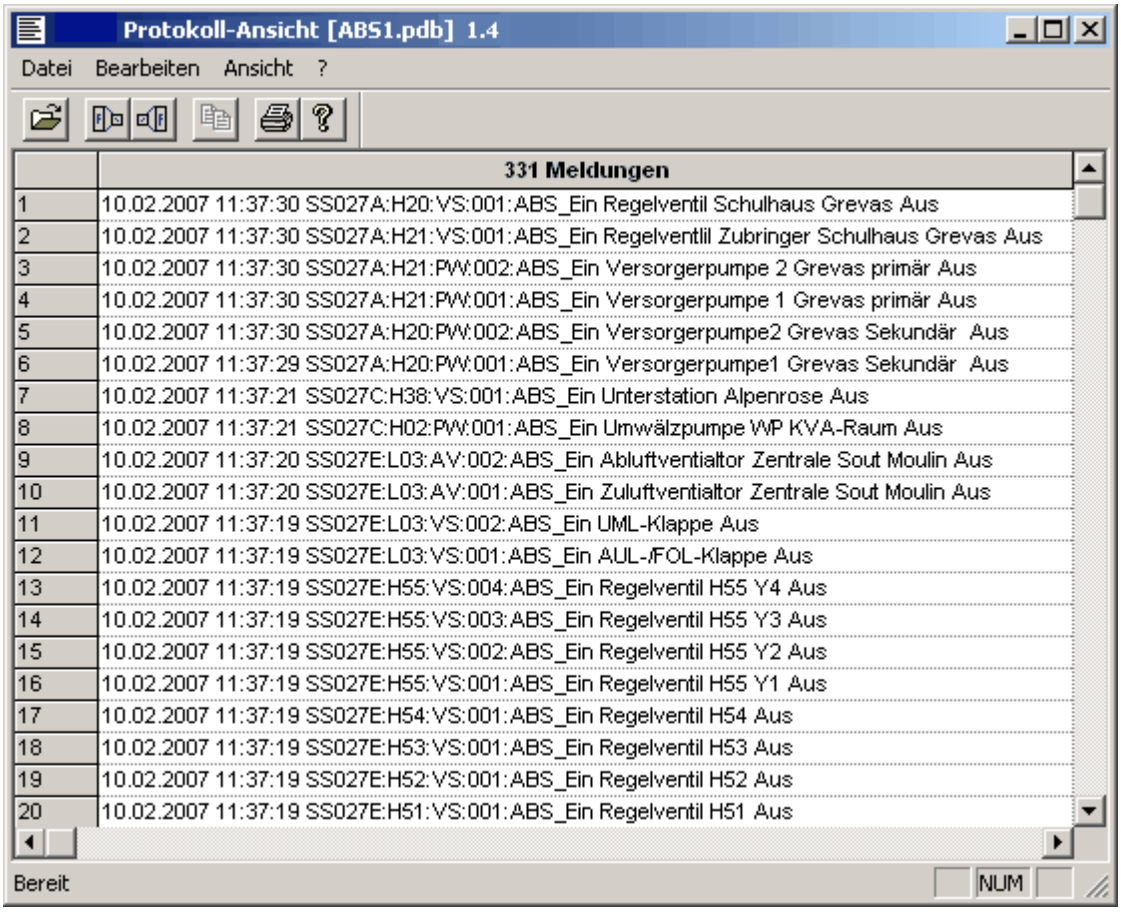
The "**File**" menu offers various options for opening, logging on, filtering and exiting the Log Viewer (PRTView).



10.5.3.1 Open

Events that are already logged can be loaded and displayed with the menu command **File >Open** or by clicking on  or with the key combination **<CTRL+O>**. Select the log file to be opened in the **Log Selection** window.

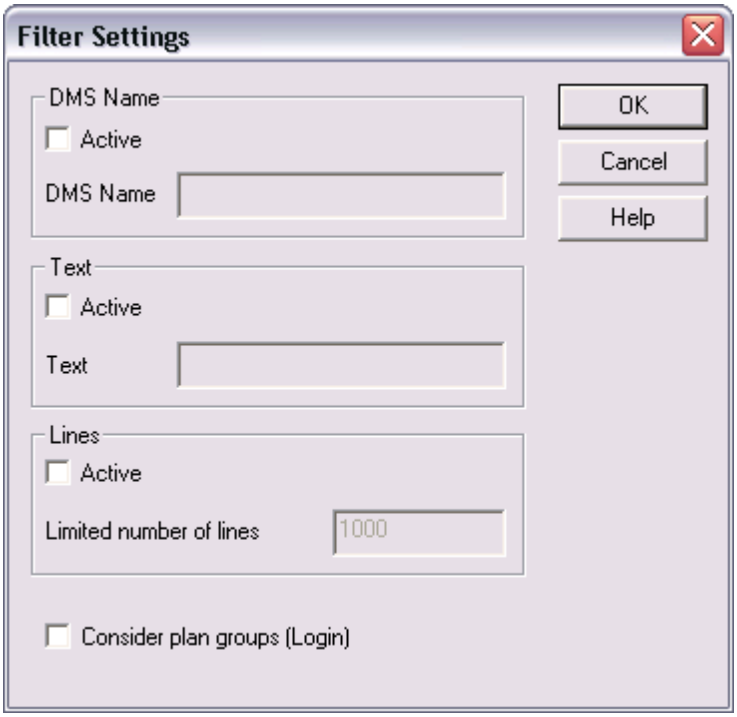




The name of the open document is shown in the title bar. In the preceding image, the **log file "ABS1.pdb"** was opened. The contents of the lines depends on the defined log format.

10.5.3.2 Set Filters

If many events are logged in a project, it is difficult to quickly find a specific event again. Therefore it makes sense to filter the events. The Filter dialog is accessed with the menu command **File > Set Filter** or by clicking on the button  :

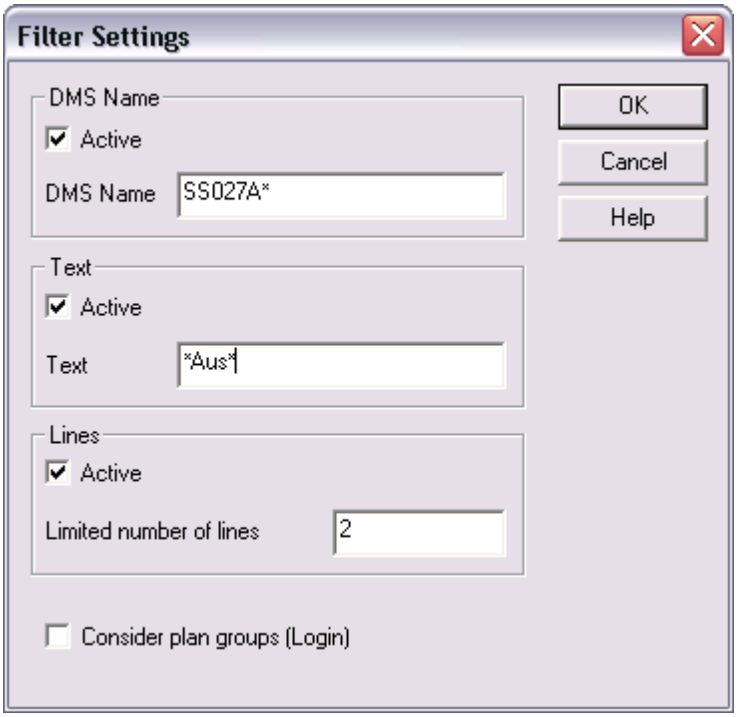


Four different types of filter can be set:

- DMS name filter
- Text filter
- Line filter
- System groups filter

The filters can be combined as desired. The filter is only activated if the ***Active*** checkbox is marked. It is also only possible to enter the filter with this option selected.

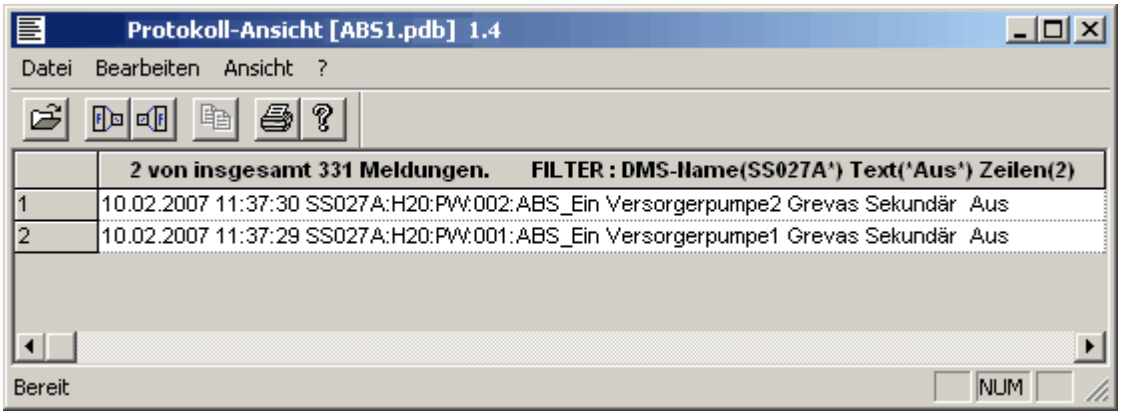
If a filter is set, only vmessages that fulfil the applied filter rules are still shown.



If the filters are set as in this image, this means that only the last 2 messages are shown, where:

- the DMS name begins with SS027A
- the message contains the word "Off" in any position

The result of this filter could then look something like this:



The asterisk "*" serves as a placeholder for DMS names and texts, meaning any text can take its place. Along with the asterisk "*", the question mark "?"can also be used as a placeholder. Any single character may appear in this position.

Example:

Filter DMS name = ??027C* means

- the first two characters may be any character
- 027C must follow the first two characters
- any text can follow 027C

The system groups filter can only be applied by users with administrator rights.

If the **Account for system groups (login)** checkbox is marked, at login the system group of the log (definable in the PET) is compared with the system levels (definable in pUser) of the user. The log entry is only shown if these match.If, for example, a log was assigned to System Group 2, the log entry is only shown to the the logged-in user if the user is entered in pUser with System Level 2. The idea in doing so is that the logged-in user is only shown log entries of system parts for which they are responsible.

The following example should demonstrate the utility of this filter:

A mountain railway consists of a mountain station and a valley station. Employee A is responsible for the valley station. Employee B is responsible for the mountain station. Therefore Employee A is only interested in the log entries for the valley station. Employee B, on the other hand, is only interested in those for the mountain station. The boss of A and B (e.g. the manager) wants to be informed about all log entries.

In order to achieve this, the system group pertaining to the valley station is set to 1 in the PET for the logs. System Group 2 is selected for the logs for the mountain station. System Level 1 is activated in pUser for Employee A, System Level 2 for Caretaker B and both System Levels 1 and 2 for the boss.

Now, when the Log Viewer is started, Employee A is only shown the log entries for the valley station after logging in, however those of the mountain station are not shown, because this employee does not have the appropriate access rights. This can only be called up by Employee B or the manager.

 *If the System Group 0 has been defined for a log, the access rights of the logged-in users are irrelevant. The log entry will be shown to them in any case.*

10.5.3.3 Delete Filter

In order to delete the applied filter, the command "**Delete Filter**" must be selected from the menu or the button  pressed.

10.5.3.4 Login <CTRL+L>

Opens the login window (pLogin.exe). Logging into prtview is necessary so that it can be verified which system logs should be shown if the checkbox ***"Account for system groups (login)"*** is marked.

10.5.3.5 Exit

This menu item exits the Log Viewer.

10.5.4 View

The toolbar and status bar of the Log Viewer (PRTView) can be switched on and off in the "**View**" menu.

10.5.4.1 Toolbar

Switches the toolbar on or off.

10.5.4.2 Status bar

Switches the status bar on or off.

10.5.5 Help menu

The **Visi.Plus online help** and current information about the Graphical Editor version are found in the "**?**" menu.

10.5.5.1 Help

The Visi.Plus online help is accessed in the menu via **"? > Help"**, via the  button or by pressing the **<F1>** key.

10.5.5.2 About PRTView...

Indicates the version of the Log Viewer being used.

Important
The version number must always be indicated in case of support queries.

We recommend our internet forum for support queries. This offers two advantages: First, help is available more quickly; and secondly, other users can benefit from the entries.

10.6 Project Data Editor (pList.exe)

pList.exe is a debugging tool that can be used for editing all project data in a list mode. The displayed list can also be limited by using the filter function on the DMS name.

10.6.1 Starting the program

pList.exe is usually started using an initialised button in the Graphical Editor, however, it can also be started from the default directory **C:/Visi.Plus/bin**, like any other program.
pList can only used if a user is logged in. Manipulating datapoints is only possible with administrator rights.

***All** data in Visi.Plus can be edited with pList (thresholds, alarms, etc.). Therefore the following information is important:*
*If a button that starts pList is created in the Graphical Editor, it must be ensured that the button can only be selected if the user has corresponding rights. **Otherwise any person could manipulate safety-relevant data in pList**.*

***Brief instructions for initialising pList to a button**:*

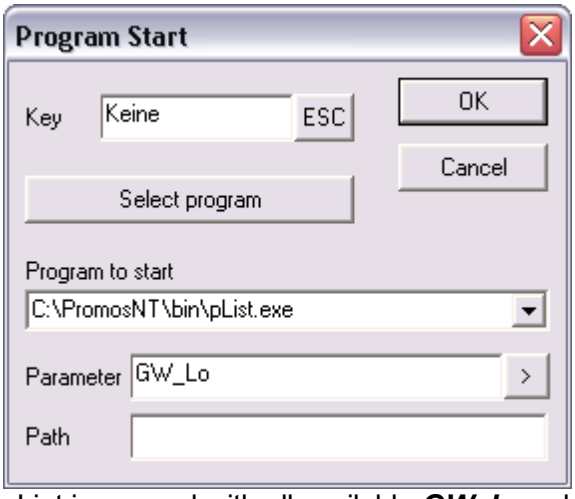
- *After a "**Button**" graphical object has been placed in the process diagram, it must be clicked on once with the mouse in order to be selected.*
- *Then open the context menu with the right mouse button and select "**Attributes**"*
- *In the open Attributes window* under "**Action**", select "**Program Start > *Any Program***" in the right column (initialisation column).
- *Click on the "**Program selection**" button in the dialog window and select **pList.exe** in the Visi.Plus default installation directory*
- ***Start pList** with **<CTRL>** + **<Button>***
- *Initialise the attribute **Visibility** to the DMS name "**System:User:[PCName]:Admin**". The button is then only visible if the user who is logged in has administrator rights.*

*If the program pList is started with indication of an attribute as a parameter, a list is automatically created with this **selection**:*

Example of setting parameters:
Parameter: GW_Lo
pList then automatically lists all DMS names with the attribute **GW_Lo**.

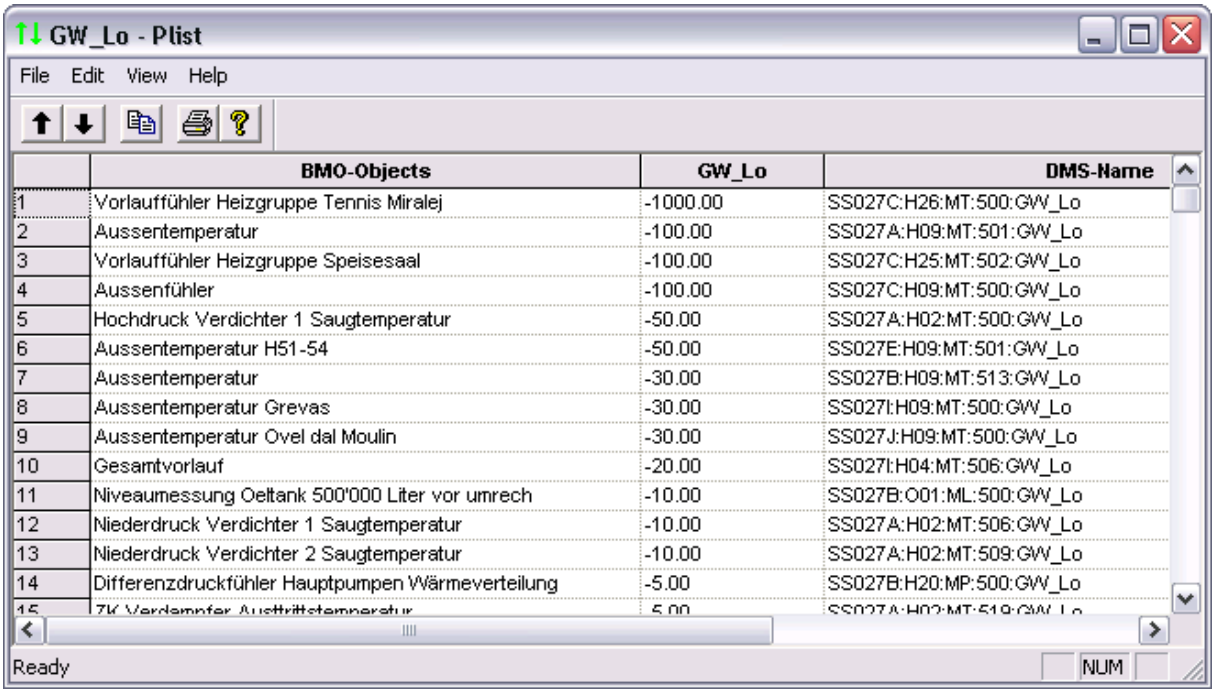
Example of setting filters:

If you indicate the call **/F:<*DMSname*>** (*or even/**F=<DMSname>***) in addition to or separately after the program to be started, this DMS name is used as a filter and only this name is displayed when the program is started.



pList is opened with all available ***GW_Lo*** values.

If **/F:BE008** were also set as a filter, only the DMS names beginning with this string would be listed.



10.6.1.1 Program parameters

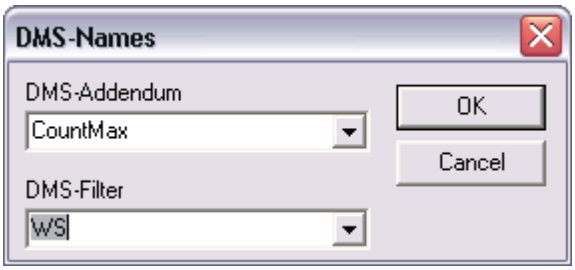
The program pList.exe can be started with various program parameters:

DMS Attribute	Attribute that should be displayed in the list
/F:	Filter
/DMS=	PC name or IP address of the PC running the DMS

Example: `plist Actual value /F:SS027B` (shows only the actual value on the PLC SS027B)

10.6.2 The PList user interface

When starting pList, the DMS data to be edited must be selected if this was not already given as a parameter on startup.

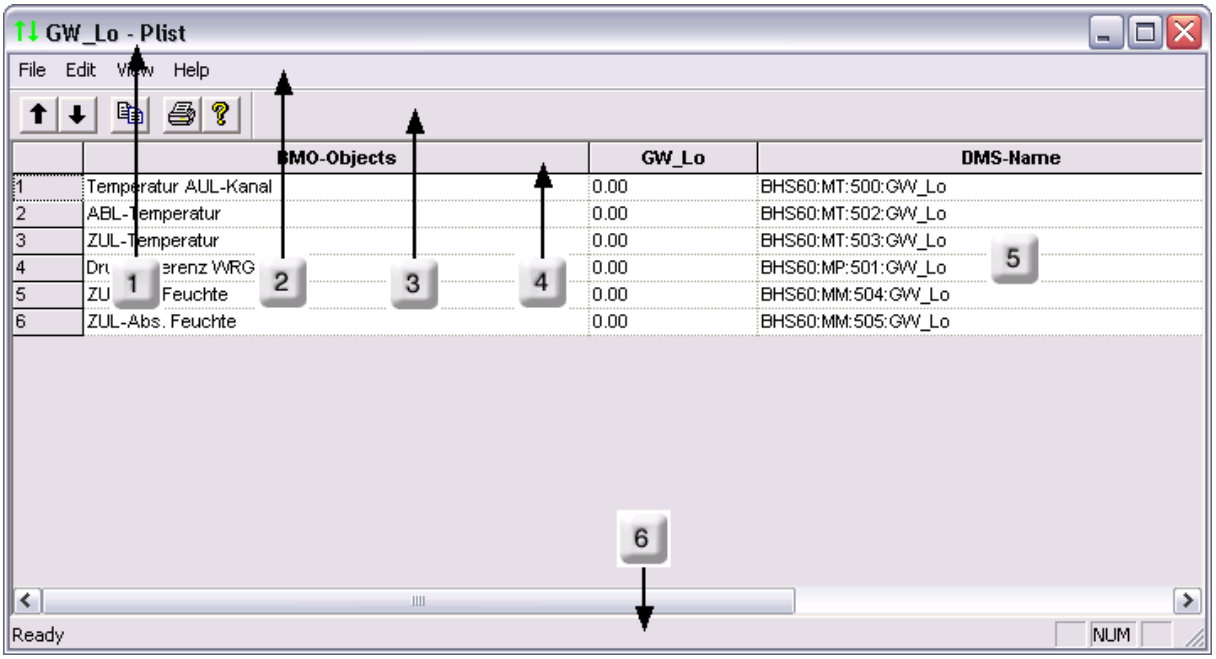


DMS Attribute
By clicking on the dropdown menu, all attributes are shown.
We understand an attribute to be the following:
A motor object with the DMS name "**S01:AK:000**" is present in the project. All system objects which the motor object now exhibits, such as manual, automatic, control, etc., are defined as attributes.

DMS Filter
The list can be narrowed down by indicating a part of the DMS name.

Example:
"**WIT:WKM**:"

The window contents of pList

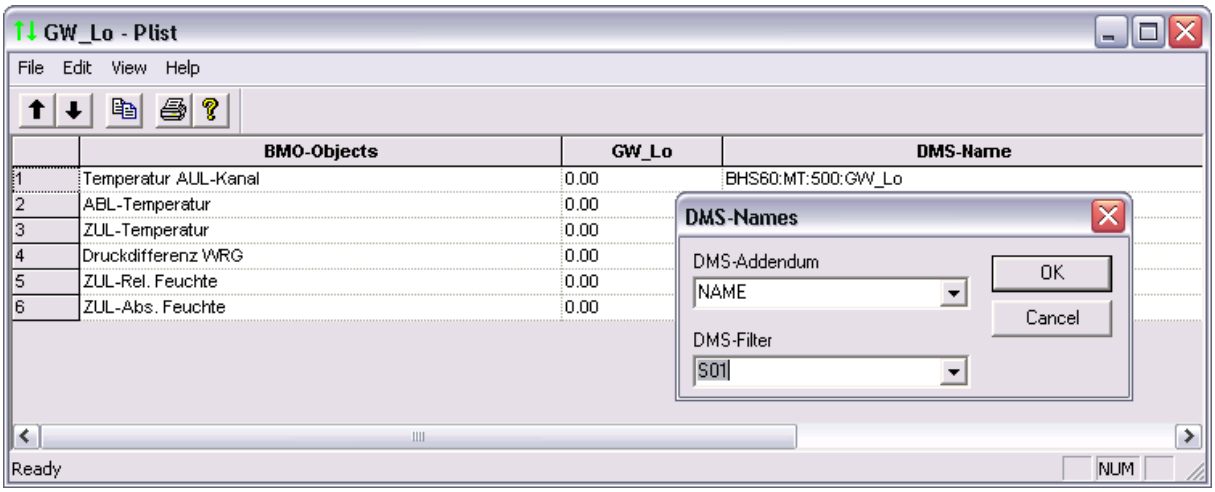


1	**Title bar**	Contains the name of the current window.
2	**Menu bar**	The menus are sorted according to basic functions.
3	**Toolbar**	Shows icons for some important commands and functions of pList.
4	**Header**	Names the columns for the data points.
5	**Debug View**	All data that is found is displayed here sequentially.
6	**Status bar**	Displays the module (program) status and the meaning of the buttons in the toolbar.

The list was created with the **DMS attribute HOp** and the **Filter WIT** (see small window **Searched DMS Data** in the bottom right of the image). As you can see, only the data that has the "**Attribute**", **HOp** and **WIT** in the "System" is displayed.

Contents of the three columns from left to right:
Name of the resource, **DMS attribute** (in the preceding image "**HOp**") and complete **DMS name**

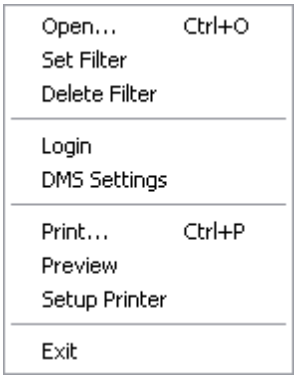
pList is particularly well suited to changing object names.
*When project data is entered in the PET, it must be given a name (object name). Visi.Plus stores this object name in the DMS attribute **NAME**. If you then want to rename to the object, display the attribute **NAME** in pList and change the object names.*



In order to allocate the same value to the selected data, select it with the mouse and then click the right mouse button!

10.6.3 File menu

The "**File**" menu offers various options for opening, logging on, filtering and exiting the Log Viewer (PRTView).



10.6.3.1 New Query <CTRL+N>

 In order to start a new query, select the command "**File > New Query**" or press the key combination **<CTRL+N>**.

 *In order to be able to start a new query, the user must have administrator rights.*

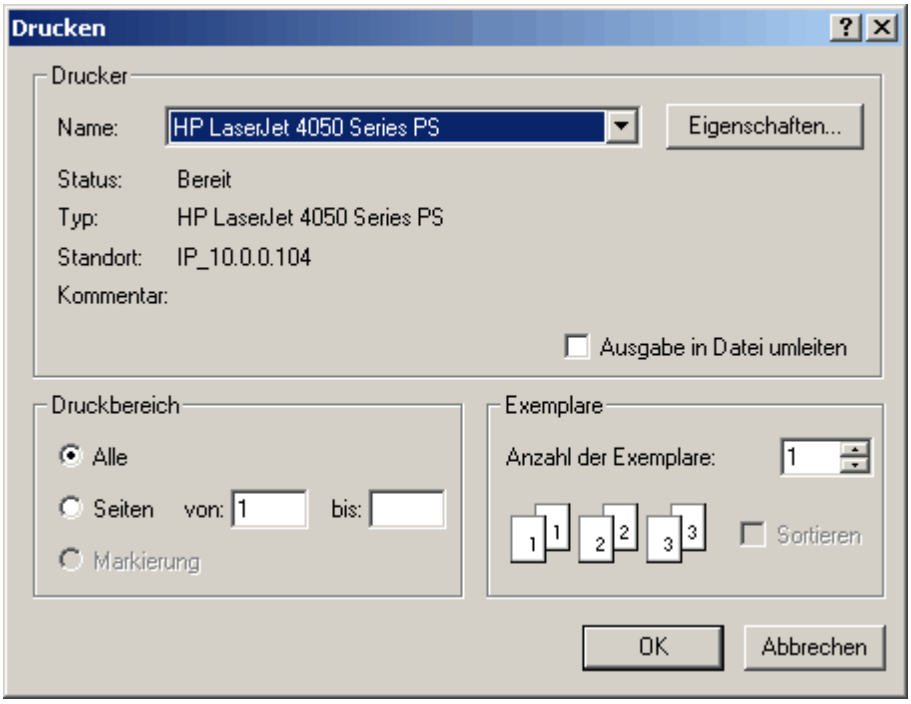
10.6.3.2 Save

In order to save the project data edited in the DMS (in the computer's main memory), use the "**Save**" command. All values in the DMS are saved to the hard disk.

 *If pList is exited **without saving**, the edited data points are **not** saved.*

10.6.3.3 Print... <CTRL+P>

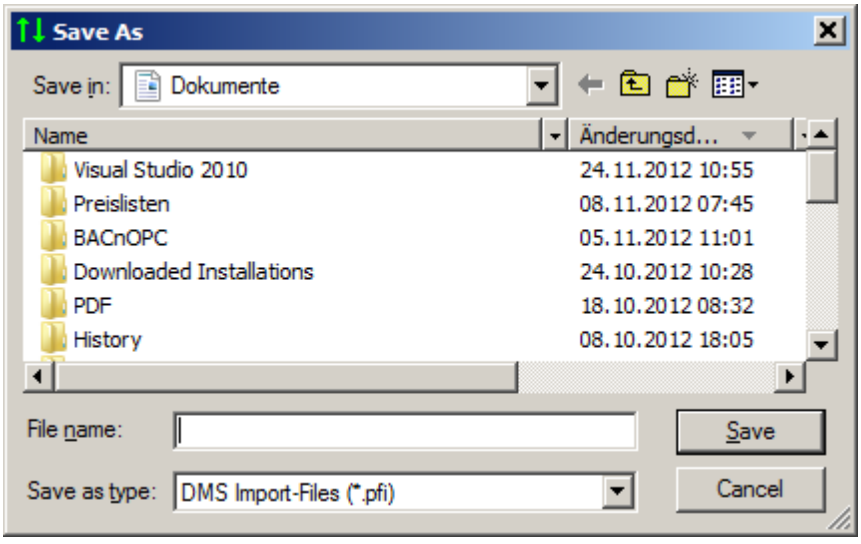
In order to print out the data displayed in pList, click on the  icon on the toolbar. Alternatively, select the "**Print**" command in the **"File"** menu or press the key combination **<CTRL+P>**.



This is a PC operating system control window and should be used accordingly.

10.6.3.4 Export

With this function the **pList list** can be saved to a file. The data is stored in **ASCII text format** and can thus be viewed and processed further with any text editor or similar program.

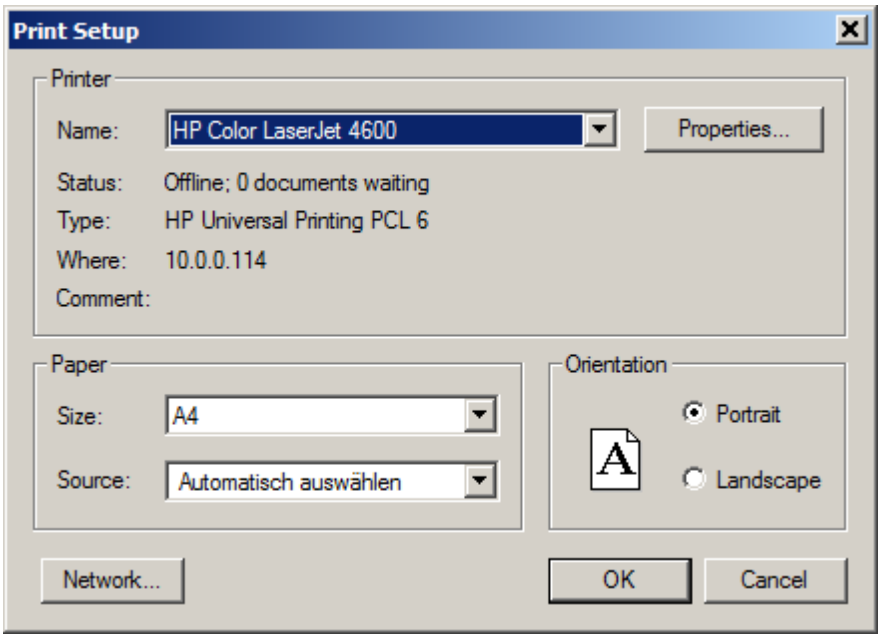


pList generates a ***.pfi file.** This file can be used in the Recipe Management module. More about this topic can be read in the **chapter** **PmosFilePicker**.

10.6.3.5 Print Preview

Before a pList list is to be printed, it makes sense to preview the printout.
This can be achieved with the "**Print Preview**" command. This may be helpful for adjusting and optimising the margins, for example.

10.6.3.6 Printer Settings



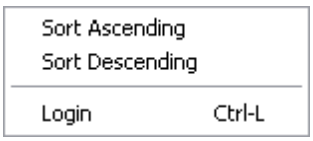
This is a PC operating system control window and should be used accordingly.

A network printer can be added by clicking on the **<Network...>** button.

10.6.3.7 Exit

This menu item exits pList.

10.6.4 Edit menu



10.6.4.1 Sort Up

 Sorts the displayed list in ascending order based on the middle column.

10.6.4.2 Sort Down

 Sorts the displayed list in descending order based on the middle column.

10.6.4.3 User Login <CTRL+L>

The window for user login appears.

10.6.4.4 Additional functions via keyboard

Copy <CTRL+C>

 Copies a selected pList range to the clipboard.

Paste <CTRL+V>

Inserts the contents of the clipboard at the cursor position.

 *To sort a pList list according to the columns "**Resource**" or "**DMS Name**", double-click on the respective column or header.*

10.6.5 View menu

The toolbar and status bar can be switched on or off in this menu.

10.6.5.1 Toolbar

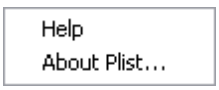
Switches the toolbar on or off.

10.6.5.2 Status bar

Switches the status bar on or off.

10.6.6 Help menu

The **Visi.Plus online help** and current information about the Graphical Editor version are found in the "**?**" menu.



10.6.6.1 Help

The Visi.Plus online help is accessed via "**? > Help**" in the menu, via the  button or by pressing the **<F1>** key.

10.6.6.2 About pList

Displays the version of pList being used.

Important
The version number must always be indicated in case of support queries.

We recommend our internet forum for support queries. This offers two advantages: First, help is available more quickly; and secondly, other users can benefit from the entries.

10.7 The Time Switch Program (CLKCfg.exe)

Events at a specific time can be controlled with the Time Switch Program (CLKCfg). The Time Switch Program allows the switching of binary DMS elements. Any equipment can be switched at specific times. The switch times can be specified to the minute.
The switch times can be edited in the PET, in CLKCfg or in the GE.

ATTENTION:
Safety-relevant processes should not be processed by CLKCfg. If the computer crashes, the processes defined in CLKCfg can no longer be executed!
Therefore PLC-supported time switch functions are preferred.

10.7.1 Starting the program CLKCfg.exe

The Time Switch Program is usually started using an initialised button in the Graphic Editor, however, it can also be started like any other program from the default **/bin** directory.

Brief instructions for initialising the CLKCfg to a button:

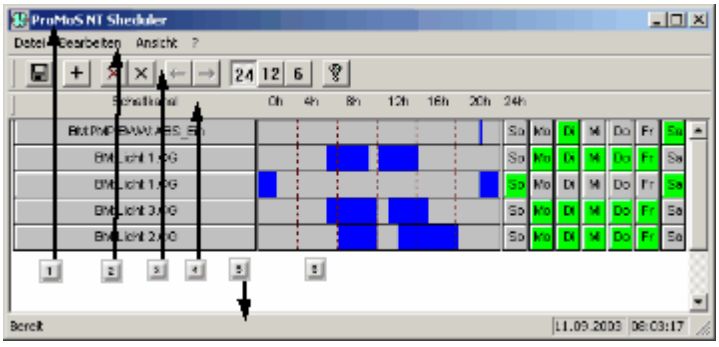
- *After a "**Button**" graphical object has been placed in the process diagram, it must be clicked on once with the mouse in order to be selected.*
- *Then open the context menu with the right mouse button and select "**Attributes**"*
- *In the open attributes window* under "**Action**", select
  "**Program Start > *Time Switch Program***" in the right column (Initialisation column).

- *Start **CLKCfg** in the GE's edit mode with **<CTRL>** + **<Key>***

As opposed to the Time Switch Manager, the Time Switch Program does not always have to be running for the switch processes to be executed. The processes are only managed with the Time Switch Program. The Time Switch Manager, on the other hand, must always be running, otherwise no switch processes are executed.

10.7.2 The Time Switch Program user interface

The window contents of the Time Switch Program (CLKCfg.exe):



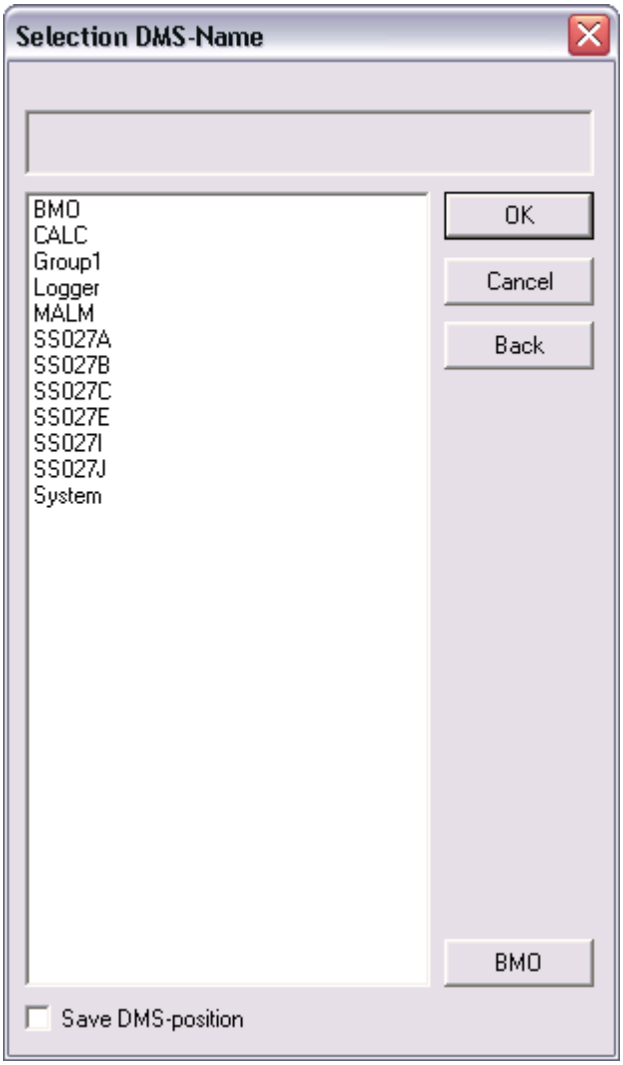
1	**Title bar**	Contains the name of the current window.
2	**Menu bar**	The menus are sorted according to basic functions.
3	**Toolbar**	Shows icons for some important commands and functions that can be selected in CLKCfg.
4	**Header**	Designates the switch channel and the switch time.
5	**Status bar**	Displays module status (program) and the meaning of the buttons in the toolbar.
6	**Display window**	All defined switch processes are displayed here.

The switched times that have been entered are shown in the dialog in blue. The relevant days of the week are shown in green. Therefore it is immediately visible at which times the signal is switched on or off.

10.7.3 Insert/Edit New Switch Channel

In order to define a new switch channel, use the menu command "Edit**"** > **"**Add New Channel**" or click on the icon "**New Switch Line**"** +.

The "Select DMS Name" dialog window opens:



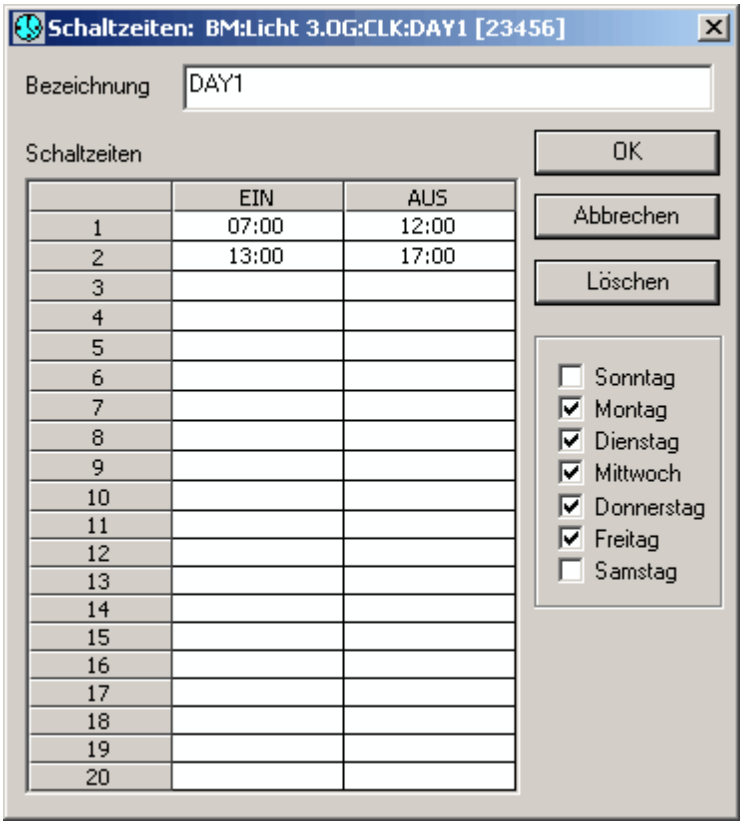
The binary DMS data points that should be switched are gradually compiled from the DMS name components listed.

After selecting a data point, a new switch line is created with the selected data point.

The switch time can be registered two different ways:

- The switch-on time for a switch section (blue) is defined by clicking and holding the left mouse button within the time range 0:00h–24:00h on the data point line and the switch-off time is defined by releasing the mouse button. The "Switch times" dialog then opens.

- By clicking on the desired data point in the "Switch Channel" column. The "Switch Times" dialog window opens.



Designation
The switch process can be given a name here.

Switch Times
Up to 20 times for switching on and off can be defined. The preceding image shows switch on time from 7:00am to 12:00pm and from 1:00pm to 5:00pm, Monday to Friday.

Days
To the right, below the **<Delete>** button, the days of the week when the indicated times should be switched can be defined.

Delete
To delete a switch line, it must first be selected by clicking on the number to the right of the switch-on time.

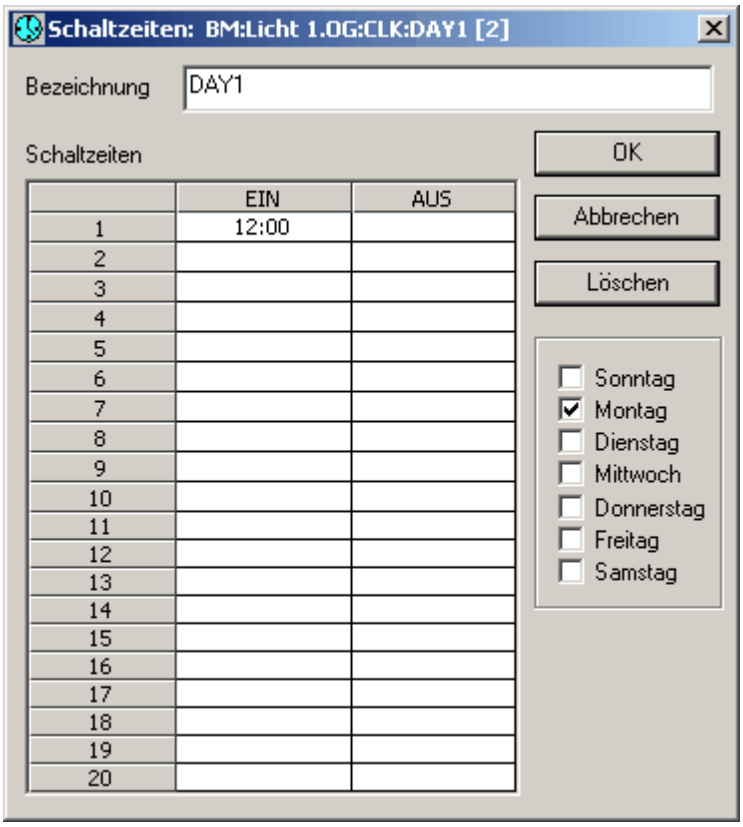
The switch times and days can be modified at any time.
The switch times are shown in the display window in the order in which they were entered.
It is not currently possible to sort the switch times after the event.

10.7.4 Switch times over multiple days

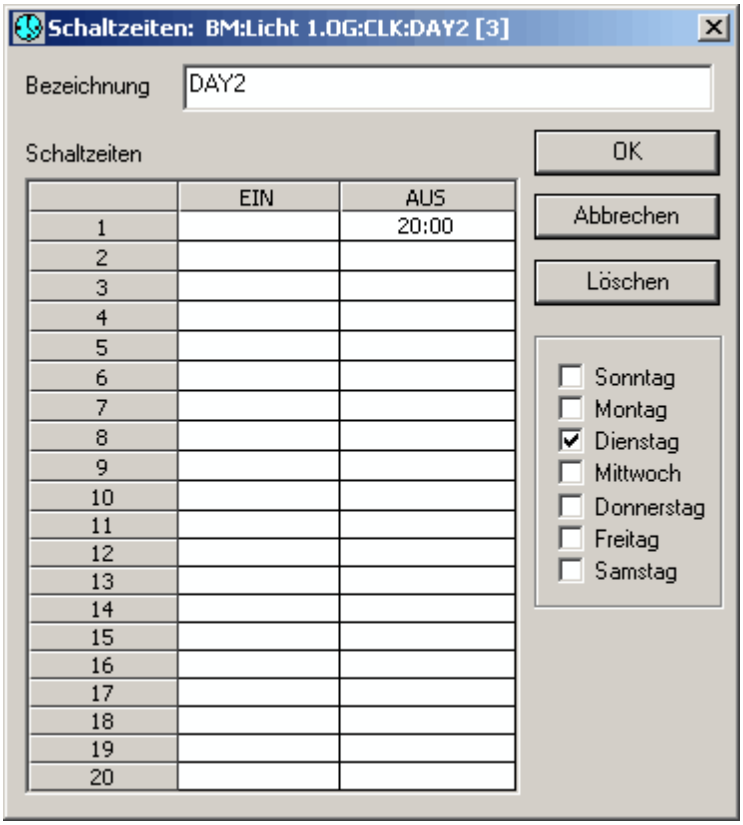
It should be noted that on-times over several days cannot be realised with the settings described in the previous chapter. In order to define an on-time over multiple days, two switch channels must be set up for the same signal.

The switch-on time is defined in the **On** field in the first switch line. The day of the week must

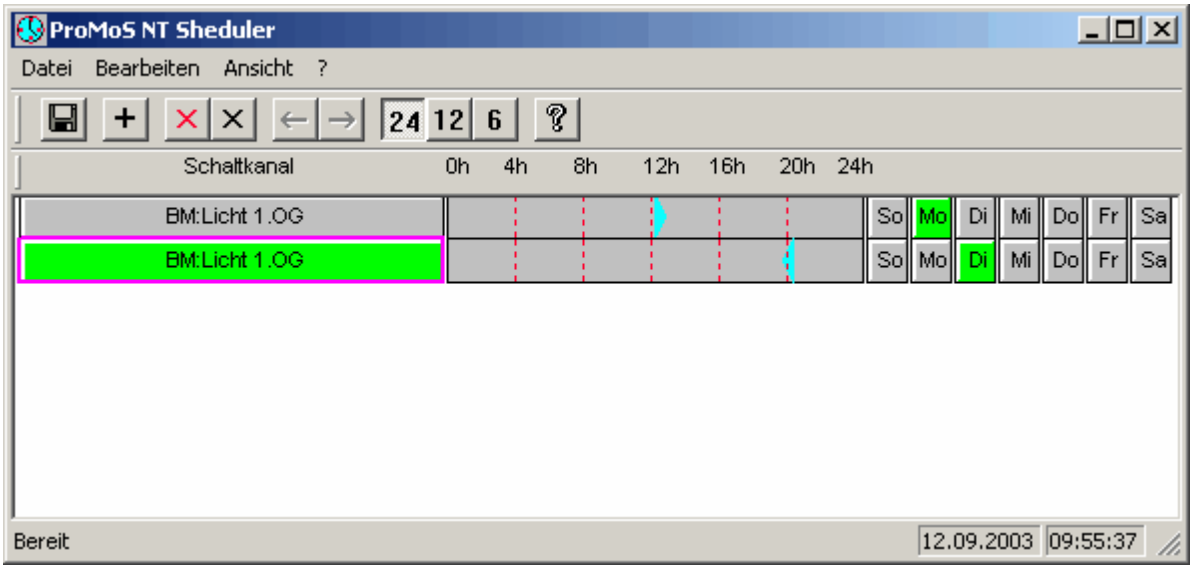
also be indicated. In the example, switching on takes place at 12:00pm on Monday:



The switch-off time is defined in the **OFF** field in the second switch line. The day of the week must also be indicated. In the example, switching off takes place at 8:00pm on Tuesday:



A switch process lasting for several days is shown in the window by two arrows pointing towards one another:



10.7.5 Deleting an existing switch channel

There are two different ways to delete a switch channel:

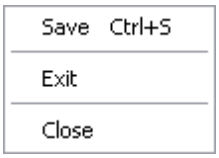
Only a marked switch line can be deleted by selecting the menu command "**Edit > Delete**

Marked Line" or the  Delete Line **icon.**

 An entire switch channel is deleted and all switch processes integrated into the selected DMS name along with it. Alternatively, select the menu command "**Edit > Delete Entire Channel**".

10.7.6 File menu

Options for saving, closing and exiting the Time Switch Program are available in the File menu.



10.7.6.1 Save <CTRL+S>

The switching processes can be saved with the save icon  on the toolbar, by pressing **<CTRL+S>** or via the menu item **File > Save**.

10.7.6.2 Close

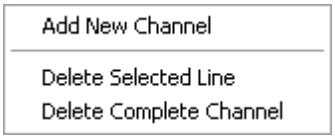
This menu item only closes the time switch program window. It does not exit the program; it is only minimised to the taskbar.

10.7.6.3 Exit

Exits the main window of the Time Switch Program.

10.7.7 Edit menu

Options for inserting and deleting switch channels are available in the Edit menu.



10.7.7.1 Add New Channel

Adds a new time switch channel. The  button has the same function.

10.7.7.2 Delete Marked Line

The marked switch line is deleted. The icon  in the toolbar also deletes the marked switch line.

10.7.7.3 Delete Entire Channel

An entire switch channel that is added to a DMS data point is deleted. The icon  in the toolbar also deletes the entire switch channel.

10.7.8 View menu

The menu items in the "View" menu affect the visual representation of the Time Switch Program.

10.7.8.1 24/12/6 hours

The view settings of the time bar can be changed. This can also be done by clicking on  in the toolbar.

10.7.8.2 Toolbar

Switches the toolbar on or off.

10.7.8.3 Status bar

Switches the status bar on or off.

10.7.9 Help menu

The Visi.Plus online help and current information about CLKCfg are found in the "**?**" menu.

10.7.9.1 Help

The Visi.Plus online help is called up through the menu **"? > Help"**, by clicking on the button  or by pressing the **<F1>** key.

10.7.9.2 About CLKCfg

Displays the current version of the Time Switch Program CLKCfg.

 *Important*
The version number must always be indicated in case of support queries.

 *We recommend our internet forum for support queries. This offers two advantages: First, help is available more quickly; and secondly, other users can benefit from the entries.*

10.8 Graph diagram output (pChart.exe)

With the program **pChart**, already saved historical data (from the PDBS) can be displayed, evaluated or exported.

10.8.1 Starting the program

Requirement:
DMS and PDBS must have been started!

***pChart** can also be started as an independent program. Only the three standard DLLs (pmos, etc.) from the **/bin** directory are necessary for this.*
For pChart to be started as an independent program, these must be present in the same save location as the program itself.
On startup, you will prompted to select and configure a connection (pipe, socket) to the computer on which the data to be displayed is located. You can read more on this in the chapter pChart remote

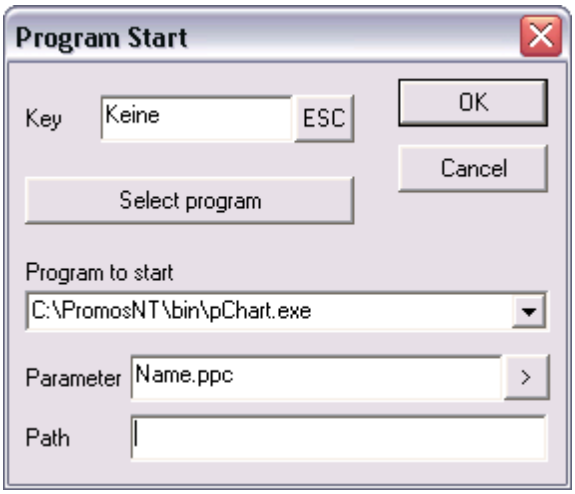
The program **pChart** can be started via an initialised button in the Graphical Editor

or

using Windows Explorer like any other program.
After a standard installation, the file **pChart.exe** is located in the directory "**c:\Visi.Plus\bin**".

The program can also be started with a parameter as long as the Program Start action has been initialised via a button:

Parameter: Name.ppc
(Name of the pChart file)



In the process, the current data based on the configuration is automatically loaded into **Name.ppc** and displayed (in this case "**Name.ppc**" is an existing trend graph view created by pChart, or a saved file).

In order to be able to represent trend data in the program pChart over months, the backup function must be activated. In order to create backups of trend data, ***PDBS*** *must be activated in the Program module under the menu command "**Settings > Backup**" (for information, see the appropriate chapter).*

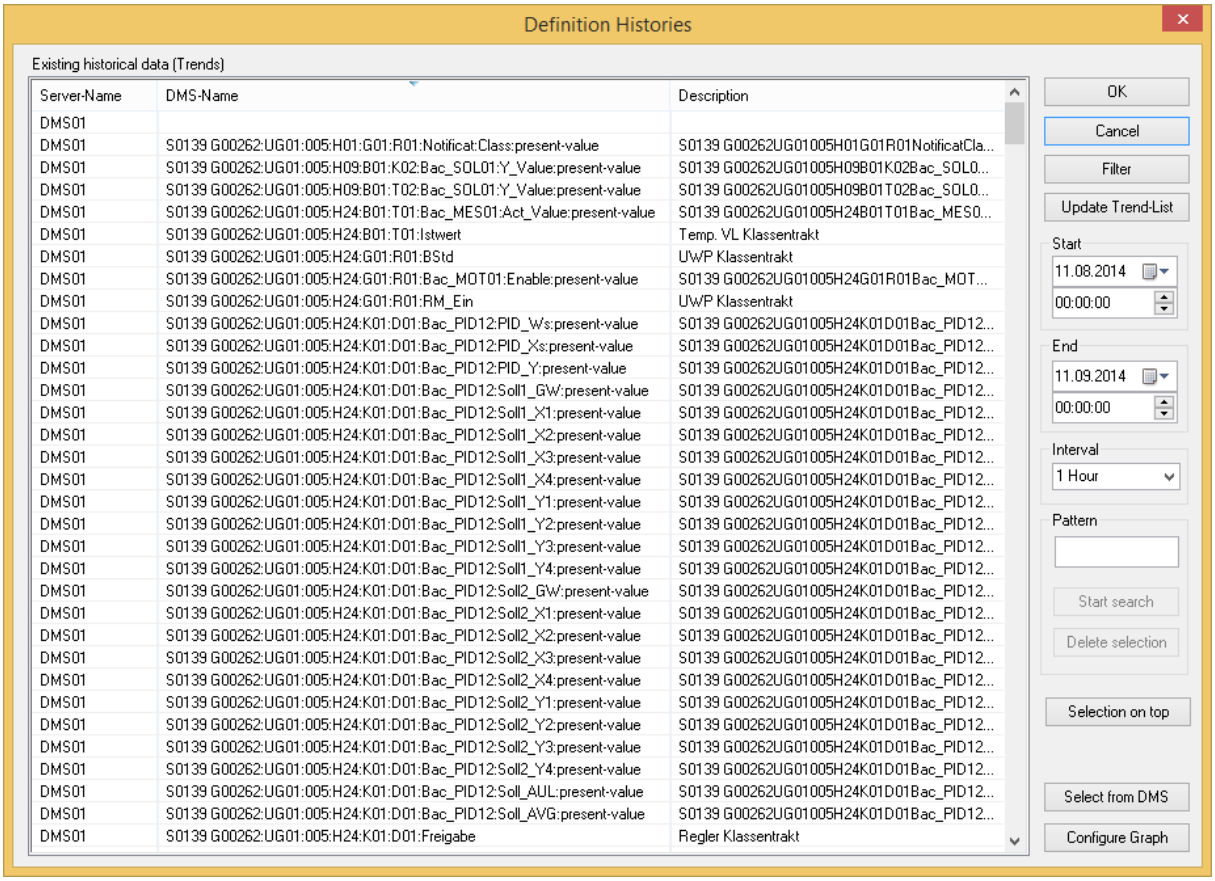
The definition data in the charts (in the GE as well as in pChart itself) is loaded from the RAM insofar as possible, the rest is automatically loaded from the hard disk of your computer.

10.8.2 pChart updates

The pChart menus have been revised. Restructuring of the curve selection and configuration menus in particular has taken place.

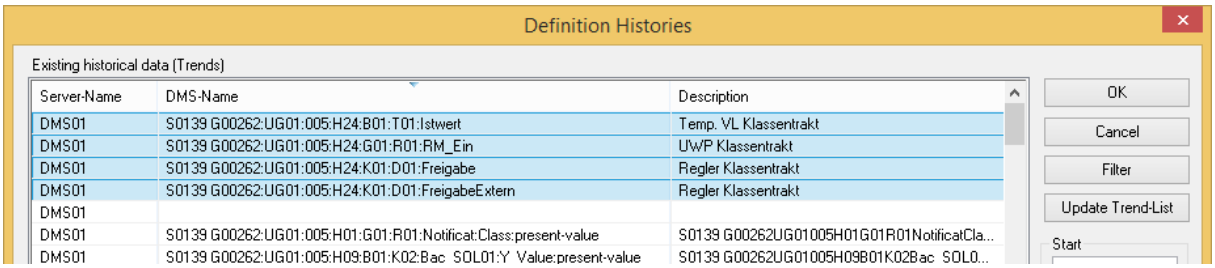
Curve selection menu:

- Server name: If several DMSs are available, they can be sorted according to this column.
- DMS name: This is where the data point's DMS name is indicated.
- Description: Consistent with the object description in PET.



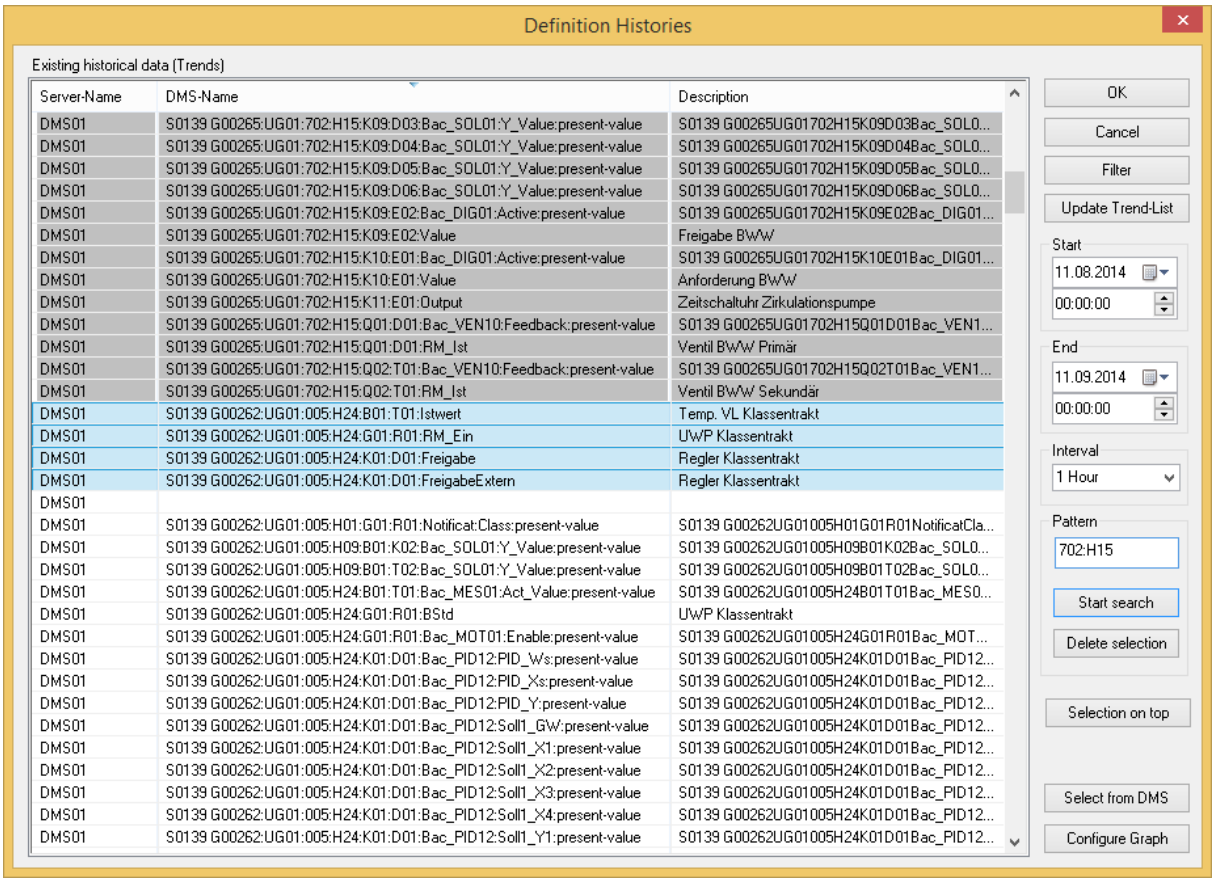
Selection follows a different process than in older versions. The data points selected are the

ones marked in blue. Hold down either the Shift or Ctrl button to select more than one. When opened up, data points already in use are listed uppermost! The <Move selection up> button moves the selected data points so that they are listed uppermost.



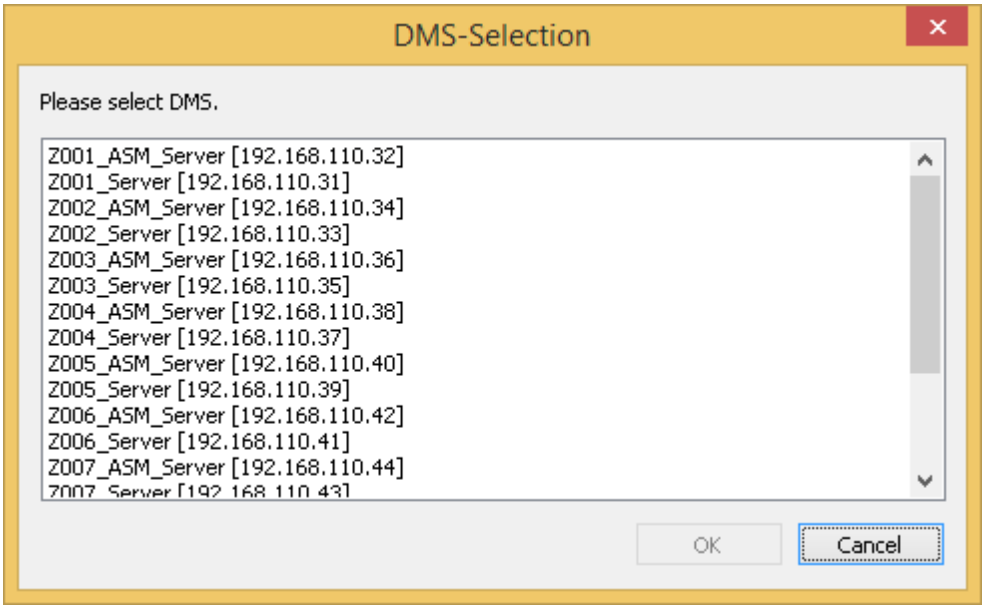
Search string:

The *star* (*) stub is already integrated straight into the search box and no longer needs to be added when searching for a partial term. Search results are each marked in grey and displayed uppermost.



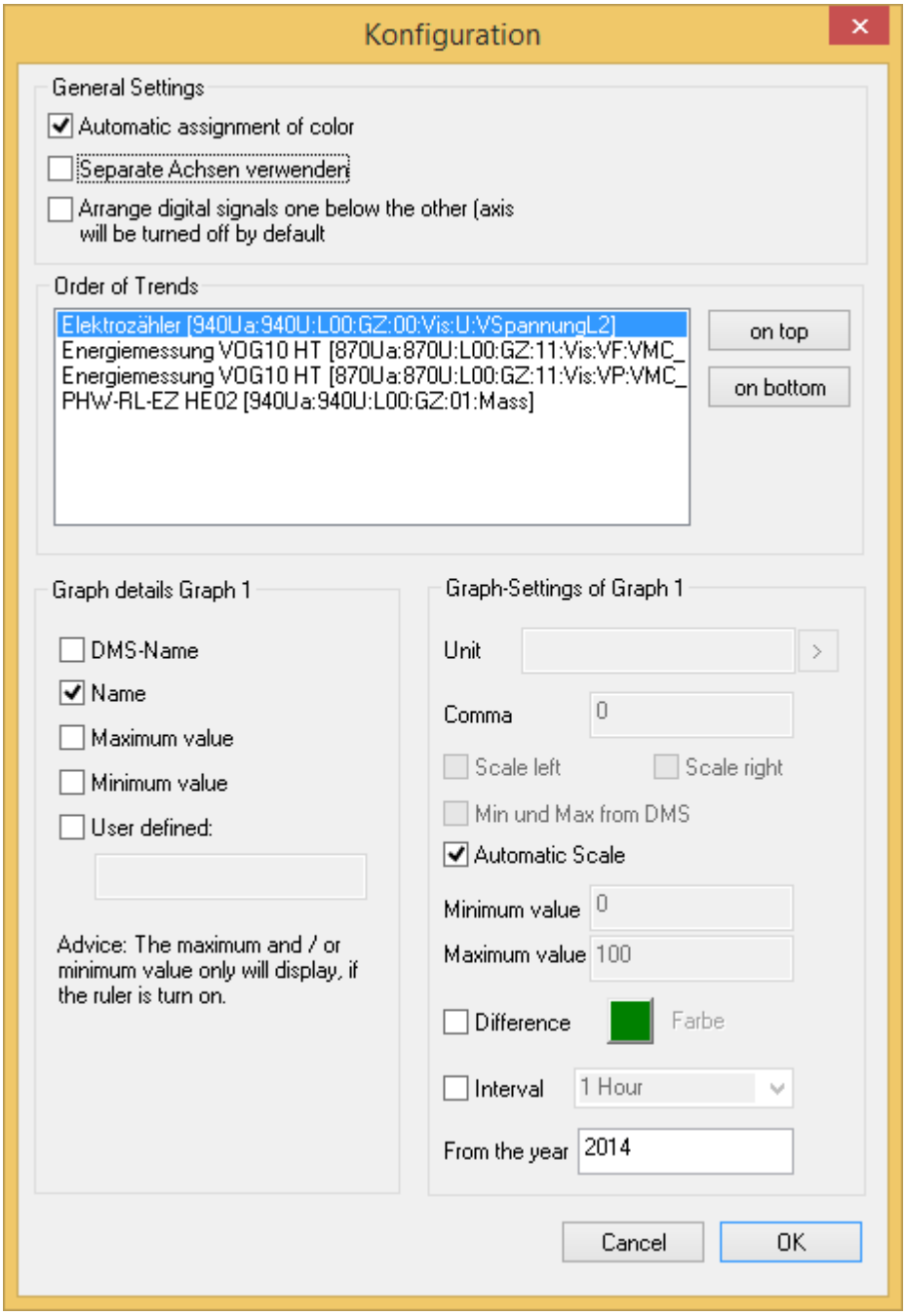
DMS selection:

If more than one DMS is available, different DMSs must be selected before a data point can be selected.



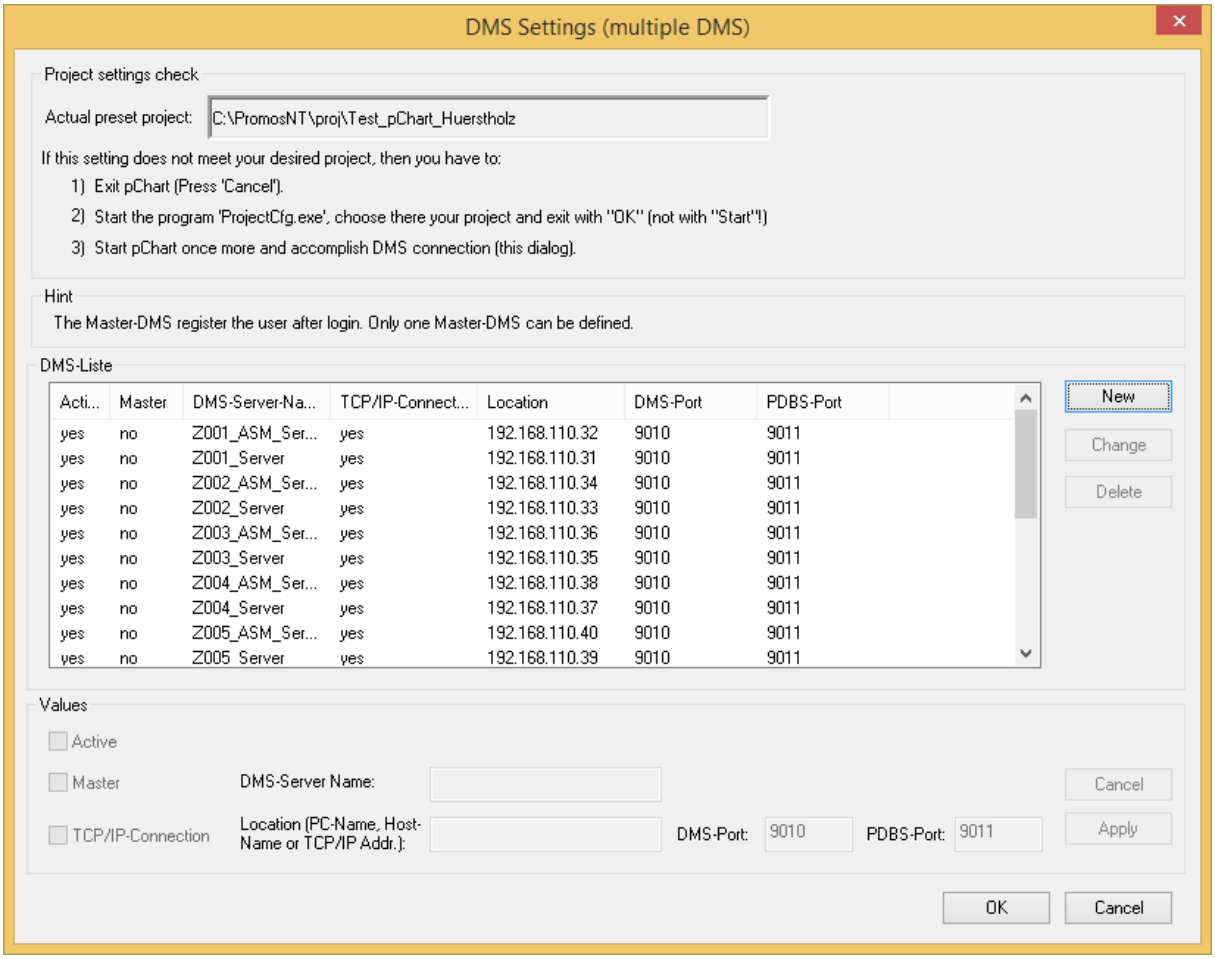
Configuring curves

The selection window for configuring curves has been reorganised. The global settings valid for all curves are at the top. Below are the settings for the individual curves. These settings differ for each curve selected.

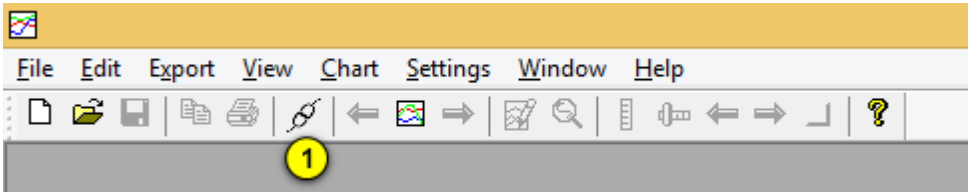


Verbindungseinstellungen

Die DMS-Verbindungseinstellungsmöglichkeiten sind identisch wie im GE und AlarmViewer aufgebaut:



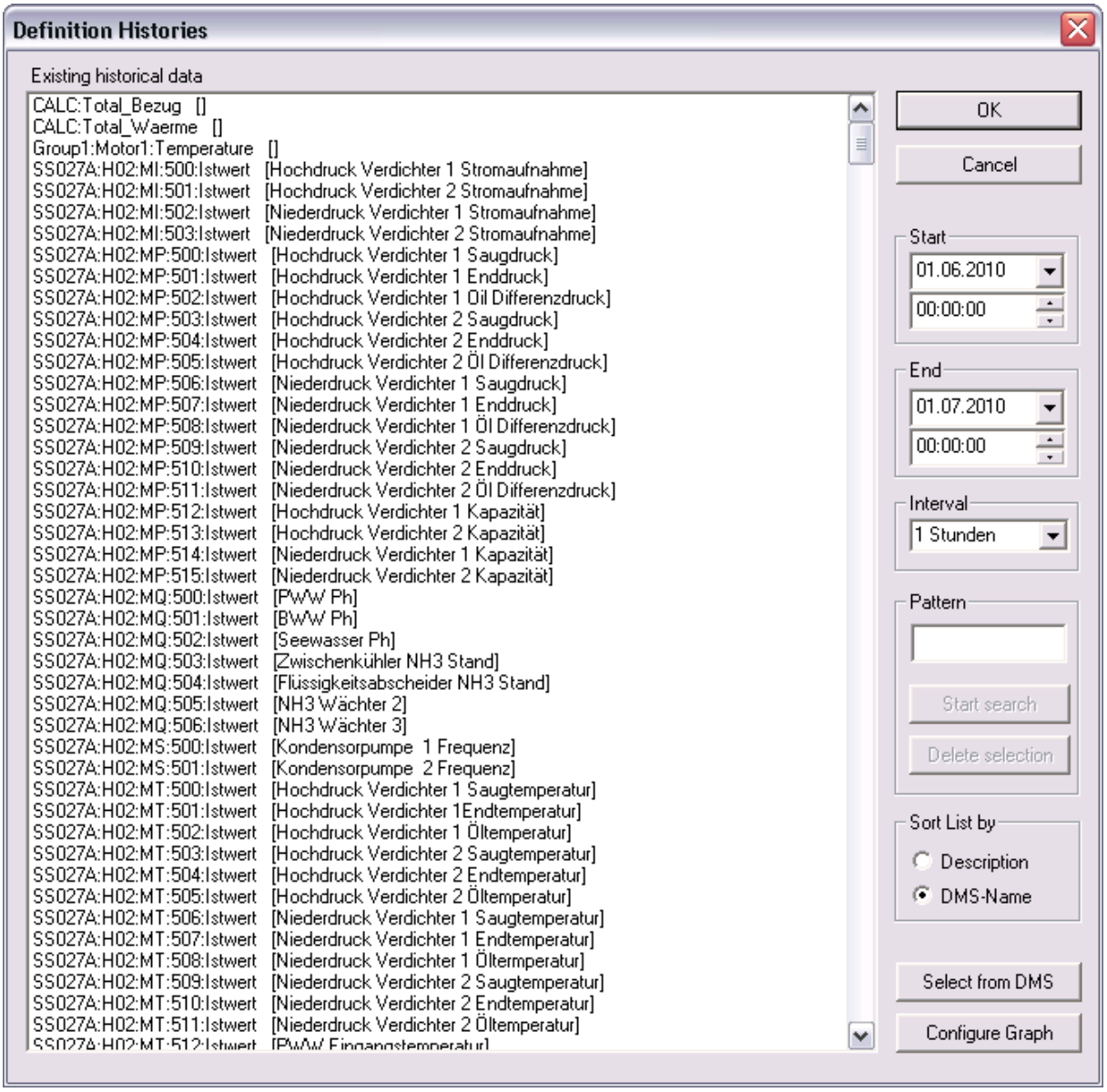
It is already possible to define several DMS connections. However this feature is still in the beta phase and is only available for testing purposes.



1 The toolbar features a new “Connect to DMS/PDBS” icon. By clicking on this button, pChart re-establishes the connection to DMS/PDBS.

10.8.3 The pChart user interface

Through the menu item **Edit > Set Graph Details you can access** the window "Graph Display Definitions".



All available **trend data** in the project is listed by **pChart** in the selection window. From this list you can choose which trend data should be shown by pChart. In the image above the trend data for various temperatures has already been selected (marked in blue).

Start
In the **Date** and **Time** input fields the beginning of the graphical trend data display can be defined by **pChart** through the dropdown menu or with the keyboard.

End
Defines the time for the end of the representation.

Interval
Defines the time intervals at which the data should be loaded from the database. The data is automatically interpolated. If, for example, a value should be obtained from the database every hour (8:00, 9:00, 10:00, etc.) and only data from 8:50 and 9:20 is available, pChart interpolates the value for 9:00.

Search String
The Search String function allows you to perform a search based on specific information. In most cases the asterisk (*) must be used as a placeholder. If, for example, you want to select all heating pumps, you must enter the search string *heating pump*. If the asterisks are omitted, nothing is selected.

Along with the asterisk (*), the question mark (?) can also be used as a placeholder. Any **single** character can then appear in its place.

Example:

Search string = ??028*

means that
- the first two characters may be any character
- 028 must follow the first two characters
- any text can follow 028

<Filter>
Filters the objects by PLC (first gradation in the AKS).

<Update trend>
Updates the trend.

<Start Search>
After the search string has been entered, the list is searched and the lines that are found are marked blue.

<Deselect All>
This deselects the lines which were selected.

<Sort List By>
Sorts the list of data points by DMS name or name.

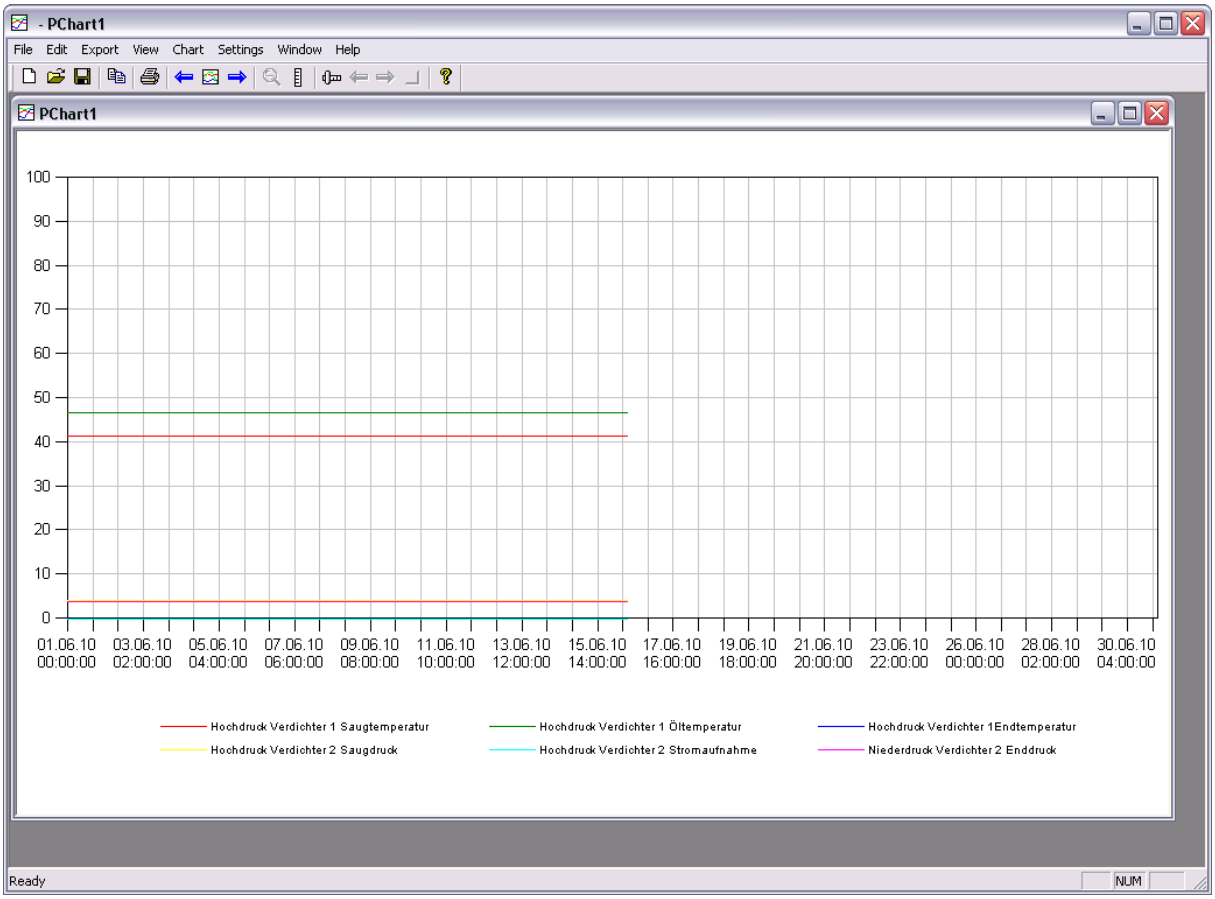
<DMS Selection>
With this button a DMS name can be selected to show its trend data. If a sub-tree is selected, pChart searches all historical data within the selected sub-tree and displays it.

<Configure Graphs>
Opens the Graph Configuration window, just like the menu item Graph/Configuration.

<Cancel>
Closes the definition dialog in pChart.

<OK>
Displays the saved data.

In the image below an example with multiple pChart graphs is shown:



If no values are present for a trend graph, this is indicated with an asterisk in the legend next to the respective DMS name or name.

Scroll back ten days in the graph.

Scroll forwards ten days in the graph.

Opens the "Graph Display Definition" screen. Additional trend graphs can be added or removed.

Zoom is switched off.

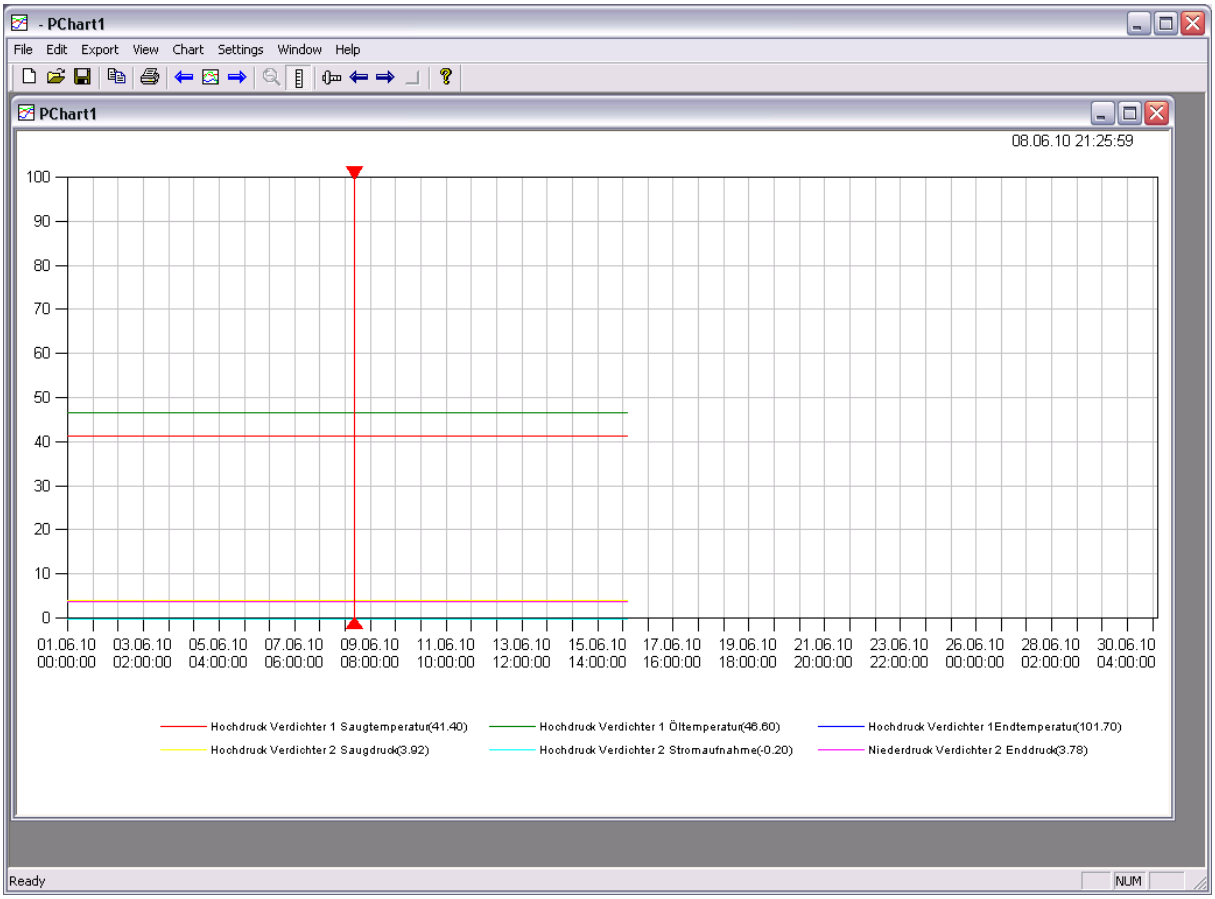
Zoom function

The trend graphs can be zoomed in horizontally. To do so, in the diagram, click and hold the left mouse button and drag a box over the diagram area that should be zoomed in on. The chart is updated after you release the mouse button.

With or a right mouse click on the diagram zoom can be switched off again.

Version 1.5 has been enhanced with a graph ruler and Autoplay functions.

These functions can be selected using icons in the toolbar. Therefore the user interface in Version 1.5 or higher has the following appearance (chart with graph ruler shown):



 Shows and hides a graph ruler.

 Starts the graph ruler in reverse direction (Autoplay).

 Starts the graph ruler in forward direction (Autoplay).

 Stops the graph ruler.

 Defines the speed at which the graph ruler travels across the diagram during Autoplay.

Graph ruler function

The graph ruler is displayed when you click the left mouse button in the diagram area, though the mouse may not be moved, otherwise it will zoom. The graph ruler (red line with two triangles) is now shown at the X-position where the mouse was clicked.

If you click on  to show the graph ruler, the graph ruler is positioned at the left edge of the diagram.

The values at the respective graph ruler position are displayed in the legend. The time is shown to the left above the diagram. If no value is available a dash (-) is displayed.

The graph ruler can now be moved back and forth with the left and right arrow keys. In doing so, the time and the displayed values should change. If the graph ruler reaches the right or left edge of the diagram range, the contents displayed in the trend window automatically moves forward or back along the time axis.

The graph ruler can also be moved with the mouse. To do so, left click on the upper or lower triangle of the graph ruler and move the mouse while holding down the mouse button. The graph ruler now follows the mouse pointer. When the mouse button is released, the graph ruler stops moving.

The graph ruler is switched off by clicking in the diagram area with the right mouse button. If zoom has been switched on, it is also switched off now.

The settings for the graph ruler can be found in the chapter Graph Ruler

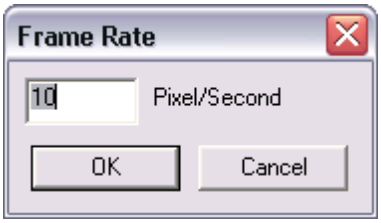
Autoplay function
It is possible to have the graph ruler move automatically either forwards or backwards across the diagram area (the so-called Autoplay function).

When clicking on  the graph ruler moves to the left to earlier times. In doing so, the time and the displayed values should change. When the graph ruler reaches the left diagram area, the contents of the trend window are automatically shifted along the time axis.

When clicking on  the graph ruler moves to the right to later times. In doing so, the time and the displayed values should change. When the graph ruler reaches the right edge of the diagram area, the contents of the trend window are automatically shifted along the time axis.

By clicking on  the graph ruler is stopped.

The speed at which the graph ruler travels across the diagram can be defined by clicking on  . The following dialog opens:



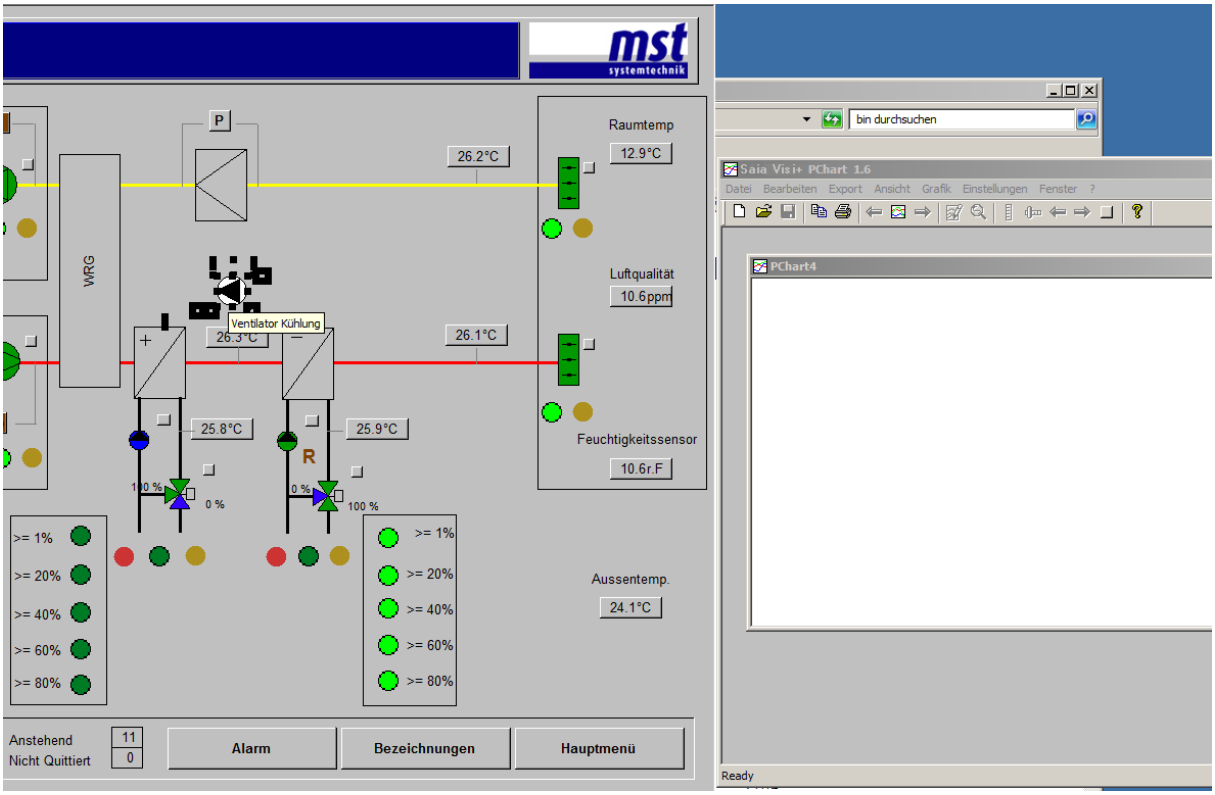
It is preset to 10 pixels/second. The value can be changed via the keyboard.

If the selected playback speed is too high, e.g. if the computer has insufficient memory, the graph ruler will always travel over the range of the diagram at the maximum possible

speed.

10.8.4 Dragging a trend graph from GE to pChart

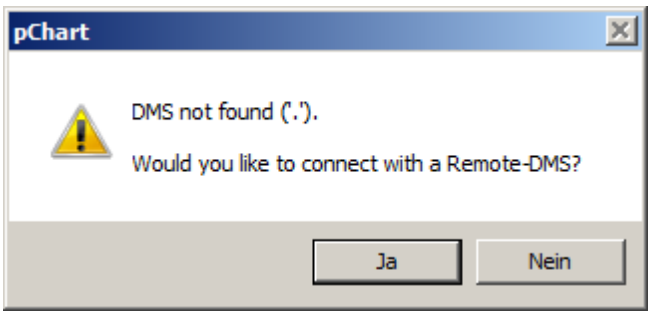
As of version 1.6 it is possible to create trend graphs in pChart from the GE using drag & drop. To do this, the GE must be in run mode. Hold down the right mouse button on an object in the GE and then drag it to a pChart window.



10.8.5 pChart remote

pChart can also be started as a remote connection, as is the case with GE or Alarm Viewer.

If pChart is started without there being a DMS on the same computer, the following message is generated automatically:



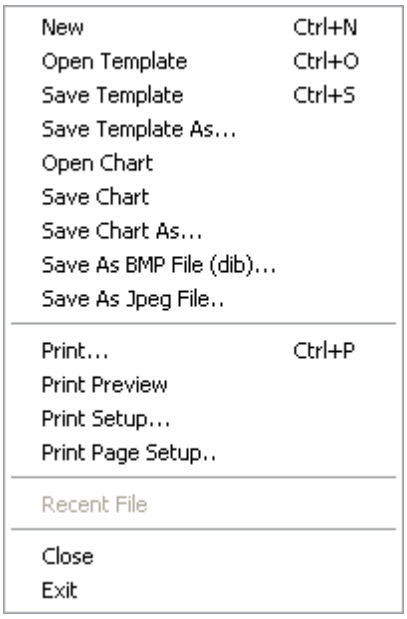
When the **<Yes>** button is actuated, the configuration window for the remote connection will open.

See Topic Connection Settings for information.

10.8.6 The pChart menus

10.8.6.1 File menu

The "File" menu contains file operations and print functions.



New

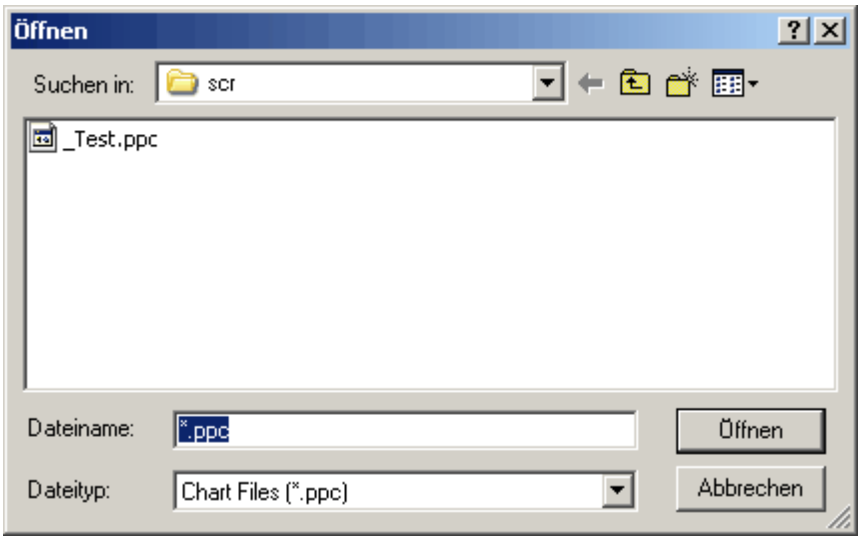
 or **<CTRL+N>** or **"> New"**

Creates a new pChart image. The **Graph Display Definition** dialog window appears.In the "Filename" field the desired name is entered.

Open Template

 or **<CTRL+O>** or **"> Open Template..."**

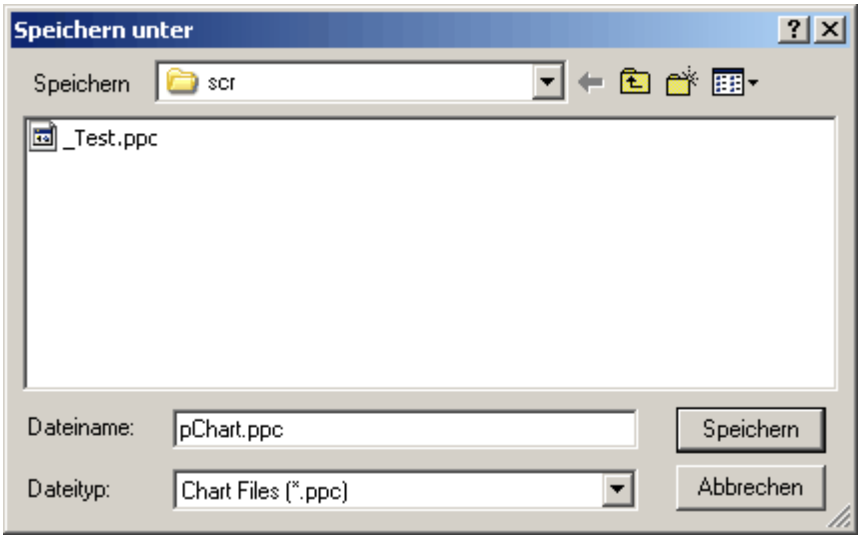
Opens an existing pChart file. In the **Open** dialog window the pChart file to be opened can be selected.



Save Template

 or **<CTRL+S>** or **"> Save Template"**

Saves the current pChart file. Only the configuration data is saved. The **historical data** is stored in the PDBS as before.



Filename

Select the path to be saved and the enter the desired filename in the input field

File Type

Currently only "csv" file types are supported.

Save

The file is saved.

Save Template As...

The same as "Save Template", however with the possibility of entering a different storage location or filename.

Open Graph

If the graph data is completely saved, this data can be displayed again. The time ranges, etc. can no longer be changed.

In the current version, graphs can only be opened if another graph is already open or has been created.

Save Graph

Saves a trend selection 1:1 (**incl. historical data**). The data cannot be changed later.

Save Graph As...

The same as the previous menu item. The filename can be specified.

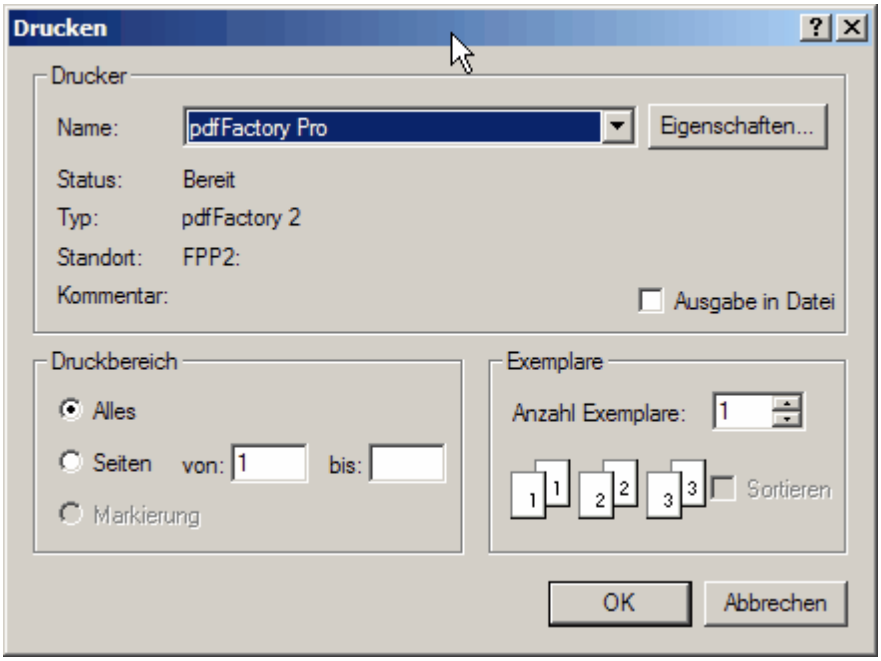
 *The difference between the commands "**Save Template**" and "**Save Graph**" are: The "**Save Template**" command does not save any historical data.*

*The "**Save Graph**" command also saves the historical data for the graph.*

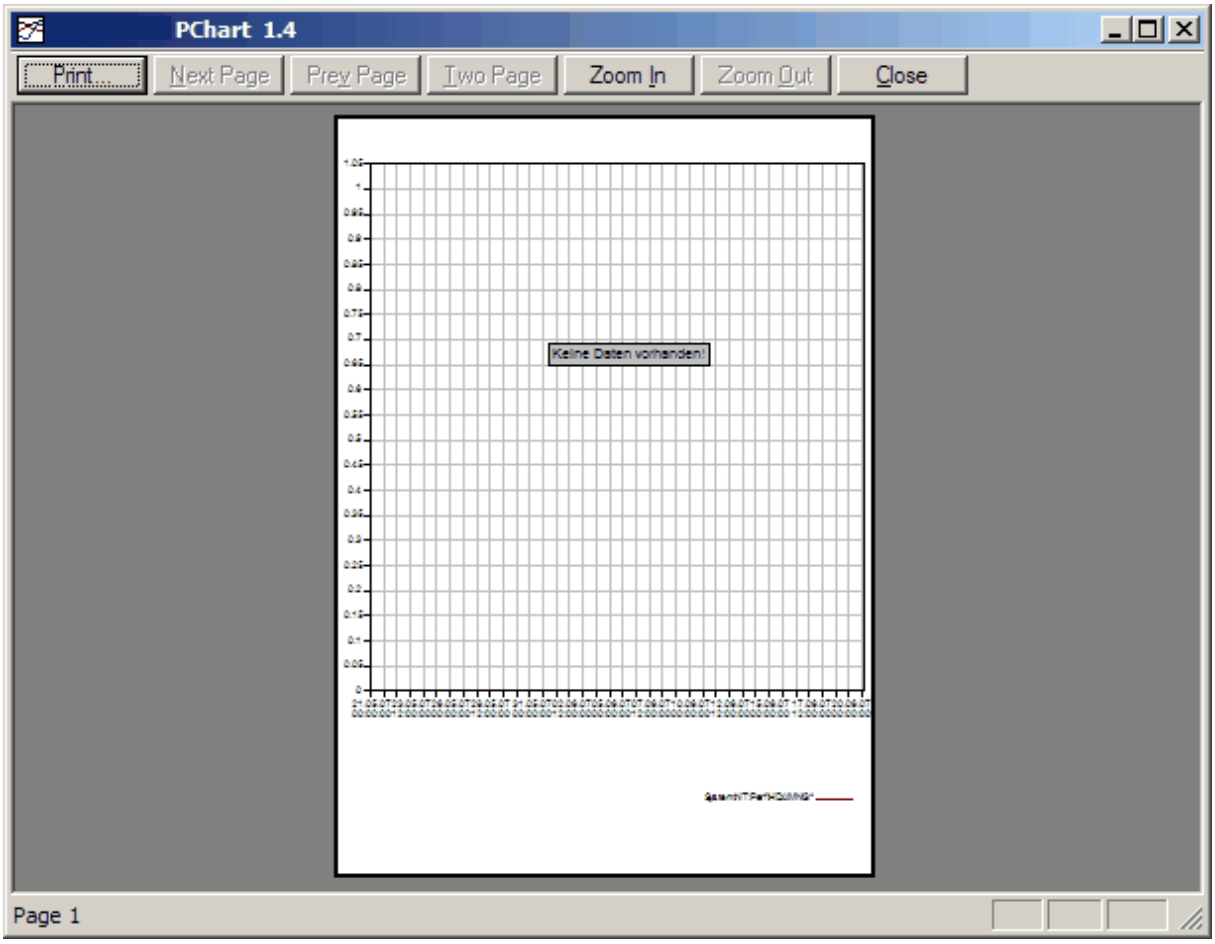
Save As BMP File
The current graph is stored in bitmap format (BMP). The format is suitable for most programs, e.g. for a word processor. The quality of the graph takes a corresponding amount of space on the hard disk.

Save As JPG File
The current graph is stored in JPG format. Compared to the BMP file, significantly less hard disk storage is occupied. Therefore the quality of the graph can be somewhat lower.

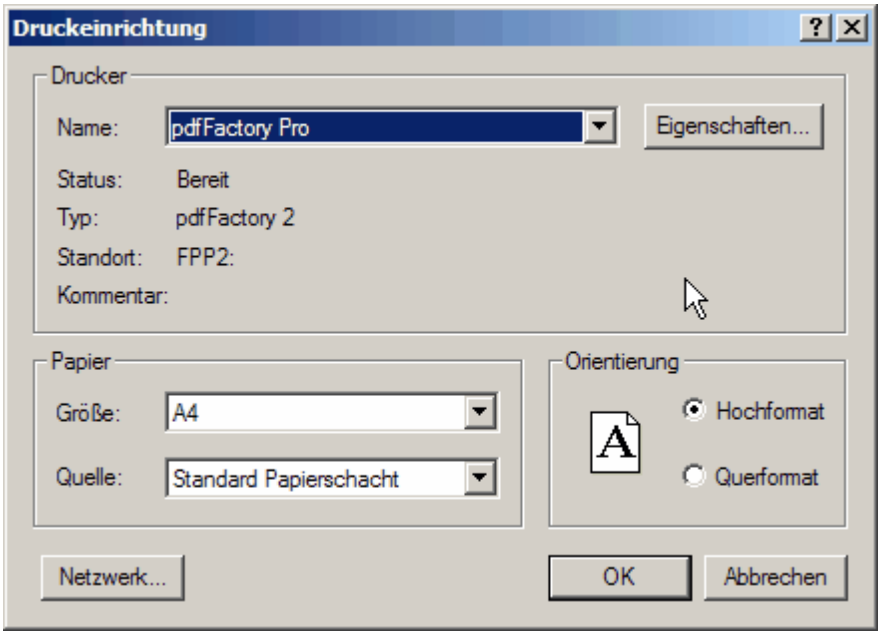
Print
Opens the default Windows dialog for printing the selected trend graph:



Print Preview
Opens the current trend graph in Print Preview.



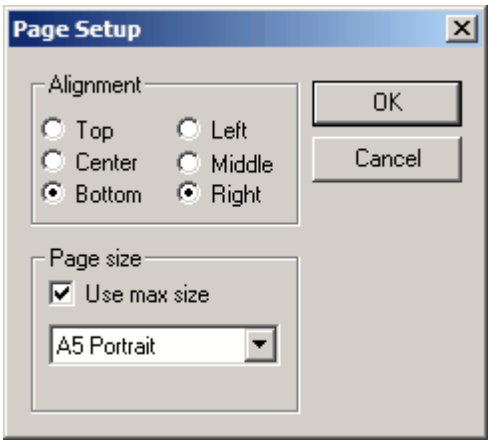
Printer Setup...
Opens the default Windows "Printer Setup" dialog.

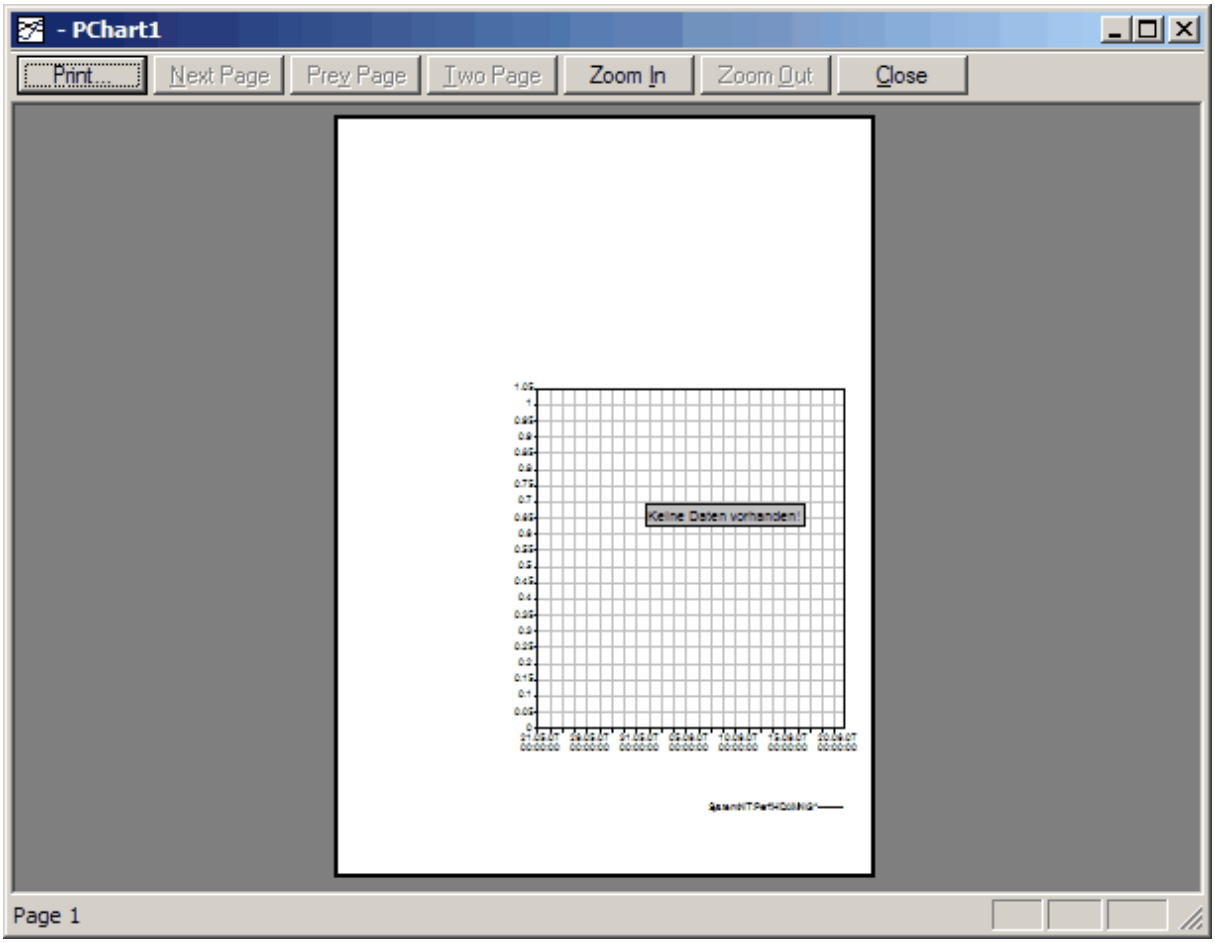


Page Setup...

Opens the settings window for the printer page setup.
Here the alignment of the data on the page and the size of the trend graph to be displayed can be adjusted.

Example: These settings would appear on paper as follows (Print Preview):





Recently Opened Files...
Opens the last edited file.

Close

Closes the graph window. The program **pChart** is not exited by doing so.

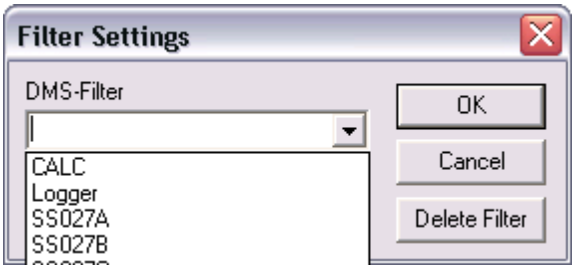
Exit
Closes the graph window and exits the program **pChart**.

10.8.6.2 Edit menu

Copy or <Ctrl+C>
Copies a chart to the clipboard.

or <Alt+Enter> or Set Graph Details
Opens the "Graph Setting Definition" window for selecting the trend data and indicating the start and end points.

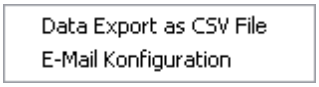
Filter Settings
A DMS filter can be configured here:



The trend data is then filtered accordingly in the "Graph Setting Definition" selection window.

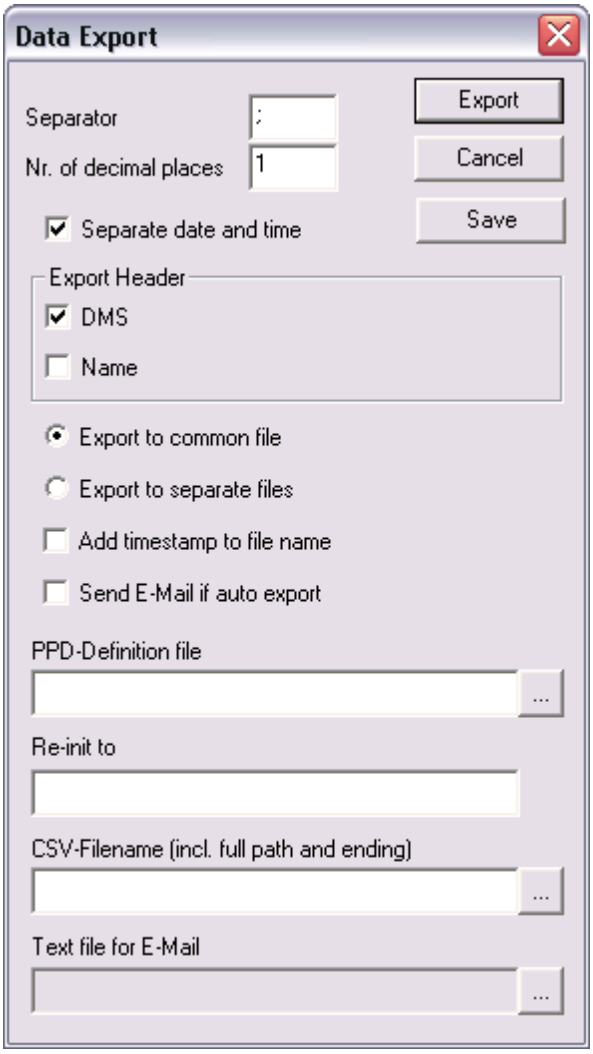
10.8.6.3 Export menu

Trend data can be exported to a csv file using the "**Export**" menu. With Autoexport, the file can be sent as an e-mail attachment (see the chapter Starting pChart using program parameters). An additional input dialog is provided for the input of the e-mail recipient and for configuring the e-mail account.



Data Export As CSV File

Exports the data of the trend graph to a **csv file**. The following dialog opens on selection of the menu item:



Separator
Here the separator that separates the cells of a line can be defined.

Number of Decimal Places
Defines the number of decimals for the exported values.

Also Separate Date and Time
If activated, the date and time are placed in separate columns.

Export Name
The DMS name (DMS and/or name can be exported as a header.

Export to Common File
If multiple sets of trend data are to be exported, they are written to a common file as interpolated values.

Export to Separate Files
The values are read 1 to 1 from the hdb file and written to the csv file (no interpolation).

Attach Timestamp to Filename
The time at which the export was started is attached to the filename. For example,

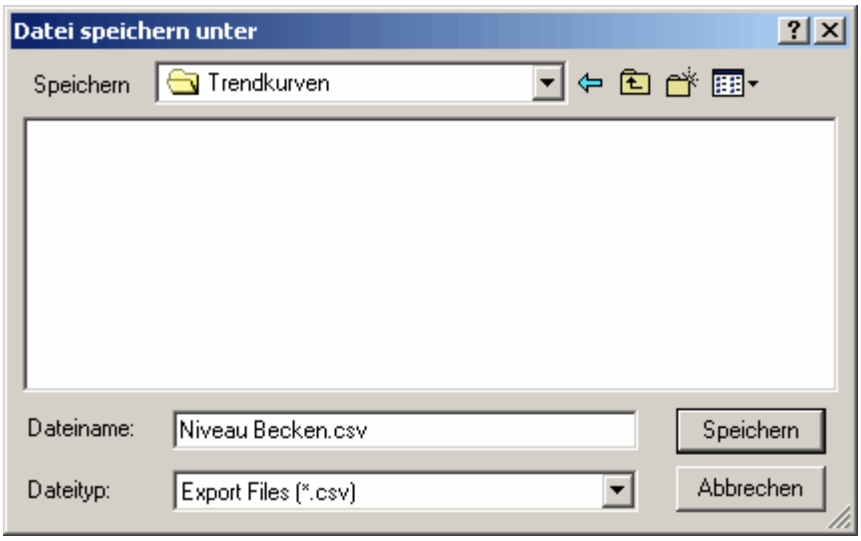
"export.csv" becomes "export_28.05.2008_16.24.05.csv"

Send e-mail on Autoexport
If the data is automatically exported using program parameters, this option defines whether the e-mails should also be generated automatically.

PPD Definition File
The data can be exported in a structured manner. To do so, a template must be defined (see Structured table export).

CSV Filename or Directory Name
Depending on the type of export, meaning in a common file or separate files, either the CSV filename or the directory name is shown in the edit field.

With export to a common file, the CSV filename is entered in the edit field (including complete path and extension). This can be done with the keyboard or by clicking on the button next to the edit field. This causes the following dialog to open:

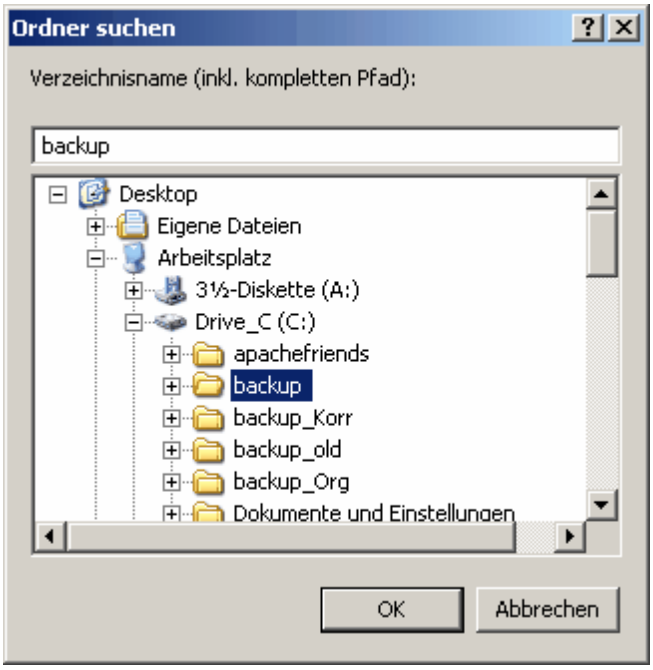


Filename
Select the path to be saved and the enter the desired filename in the input field.

File Type
Currently only "csv" file types are supported.

Save
After saving, you return to the Data Export dialog, where the filename and the full path has now been carried into the edit field.

If the trend values should be saved to separate files, a directory must be indicated. This can be done with the keyboard or by clicking on the button next to the edit field. When you click on the button, the following dialog opens:



You must click through to the respective directory in the tree and then exit the dialog with OK. The directory path is carried over to the edit box. The export files are then saved in the indicated folder under their respective DMS names.

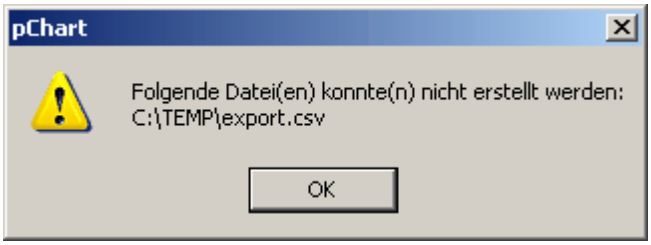
Send e-mail on Autoexport
If this is clicked on, on Autoexport the export file is sent as an e-mail attachment. Autoexport is discussed in greater detail in the chapter Starting pChart using program parameters. Here it should be mentioned that you can start the data export in the background, e.g. through the command line, from a batch file or from the GE.

Text File for e-mail
The text for the e-mail can be taken from a file. The name of this file must be entered with the keyboard or selected via the button.

<Export>
This button starts the export. It can only be clicked on if the chart data is shown. If the data export is completed, the following message window is shown:

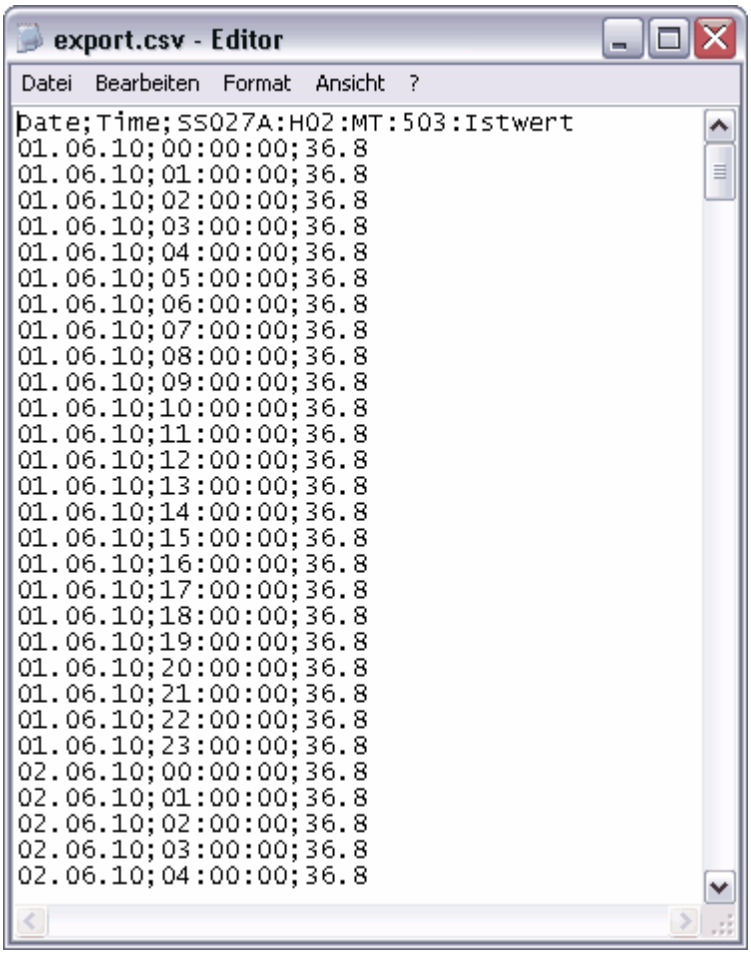
Data export successful:

In case of error the following appears:



The "csv" file can be opened with any editor or a spreadsheet program (e.g. MS Excel).

Output in an editor:



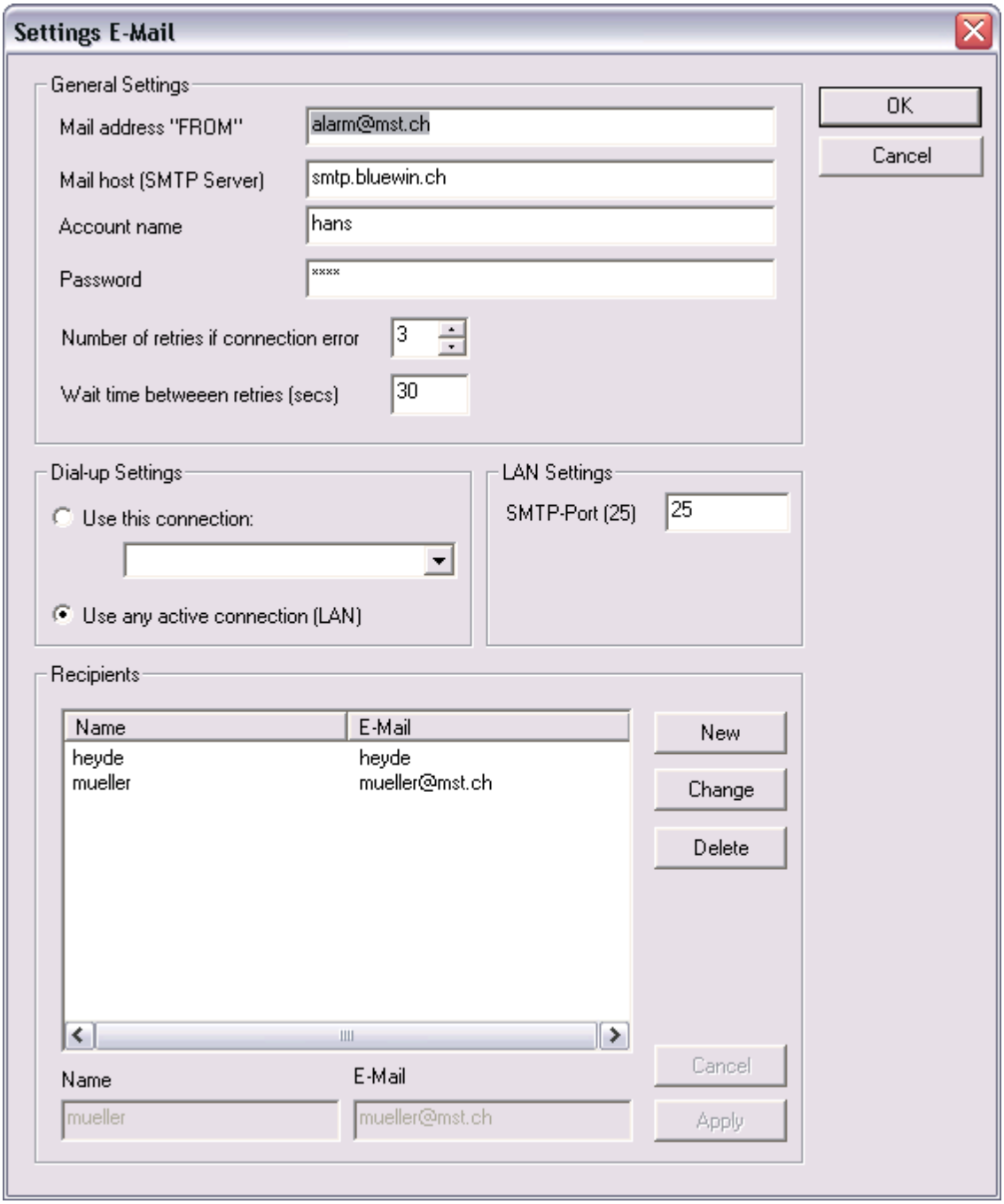
<Save>

This menu item has the same function as  or **<CTRL+S>** or **"File > Save Template"** in the menu. Only the configuration data is saved. The **historical data** is stored in the PDBS as before.

<Cancel>
Closes the dialog.

E-mail Configuration

This window enables you to enter e-mail recipients and to configure the e-mail account. The following dialog opens on selection of the menu item:



General Settings section

Sender e-mail Address

Holds the sender's e-mail address.

Mail Host (SMTP Server)

Here the mail server for outgoing post (SMTP) is entered. If this information is unknown, check with your internet service provider.

Account Name and Password
Here the user can enter an e-mail account and password.

Repetition(s) on Connection Error
In this field, the number of repetitions in case of connection error can be defined.

Wait Time Between Repetitions
Indicates how many seconds are waited before the e-mail is resent.

LAN Settings section

SMTP Port (25)
Defines the port of the mail server (SMTP). Normally this is Port 25. If this information is unknown, check with your internet service provider.

Remote Transmission settings section
In this section the connection settings are configured.

*The information in the preceding image **cannot** be used as displayed. The SMTP server and its port depend on the internet service provider. The ending of the **sender e-mail address** (e.g. ...@mst.ch) must be registered with the indicated SMTP server.*

Recipient section

Here the recipients to whom the e-mail should be sent are indicated.

To do so, click on the **New** button. The fields Name and e-mail can now be edited.

By clicking on the **Adopt** button, the new values are adopted into the recipient list. The entries in the list can then be modified or deleted. The actions Modify and Delete always refer to the active line (highlighted in blue). In order to activate a line, click with the left mouse button on the corresponding line in the recipient list or scroll through the list with the arrow keys if a line is already selected.

<New>
Allows you to enter information on the recipient (name and e-mail address).

<Change>
An already entered recipient can be modified.

<Delete>
Deletes the active line.

<Cancel>
A newly created entry is not adopted into the recipient list.

<Adopt>
A newly created entry is adopted into the recipient list.

<OK>
Closes the dialog, saving the configuration is saved in the project directory under ...\cfg \pChart.cfg in the [e-mail Settings] section.

10.8.6.4 View menu

The toolbar and status bar in pChart can be switched on and off in the "**View**" menu.

Toolbar
Switches the toolbar on or off.

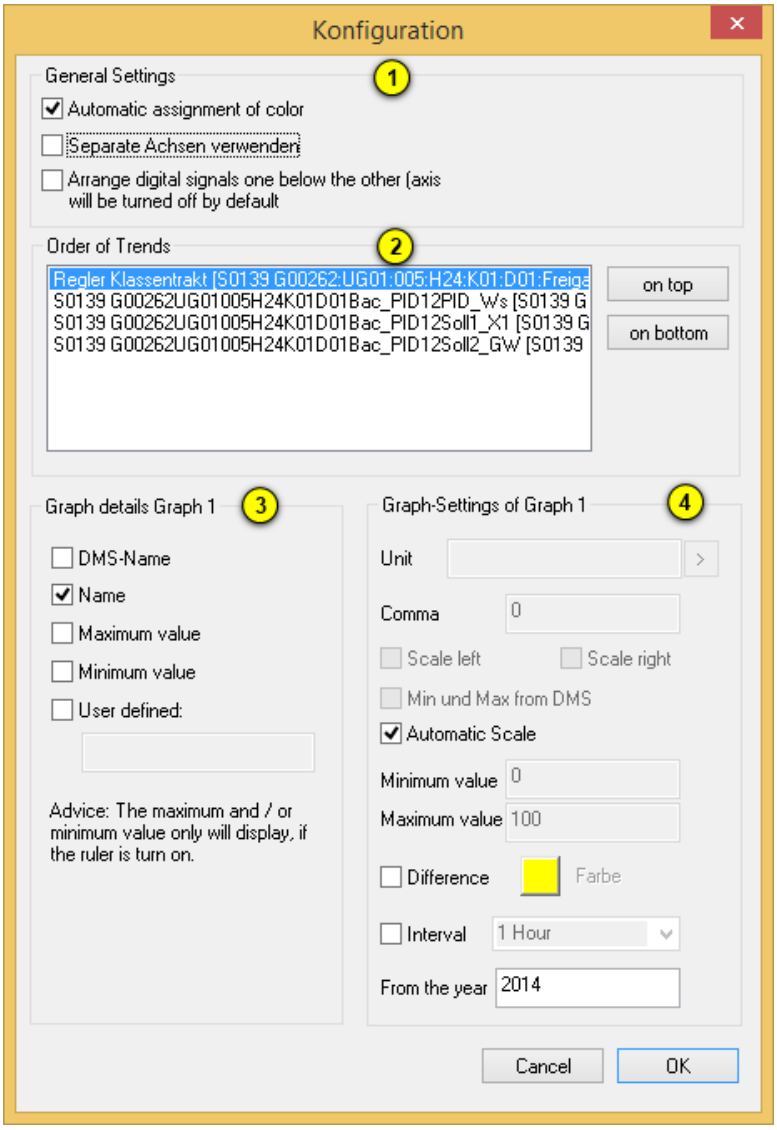
Status bar
Switches the status bar on or off.

10.8.6.5 Graphics menu

Various graphics settings can be configured in the Graphics menu.

10.8.6.5.1 Configuration

Configuration:
Opens the window for configuring the selected data points:



1 General settings

This is where the global settings are defined.

Automatic colour labelling:
Ticking this box automatically assigns colours to the trends. When the box is left unticked, colours can be selected under curve details in the configuration window.
Use separate axes:
Scaling takes place on a separate axis when the box is ticked.
Display digital signals one below the other:
When this box is ticked, the digital signals will be displayed one below the other. This tick box has no bearing with analogue signals.

2 Order of display:

This is where the display order is defined across both interfaces.

3 Curve details for curve:

This is where the keys to curves are configured. This setting is different for every data point. A curve can be selected under item 2

The following settings options are available:

- **DMS name**: AEI code is displayed in the key
- **Description:** The data point description is displayed in the key.
- **Maximum value:** The maximum value of the curve is also displayed in the key.
- **Minimum value**: The minimum value of the curve is also displayed in the key.

4 Graph settings:
This is where the settings for different graphs can be defined:

Unit: This is only active if separate axes are used. This is where separate units can be defined or read from a data point. In each case a unit can only be read from the same DMS!

Decimal places: This is only active if separate axes are used. This is where the number of decimal places is defined for each unit.

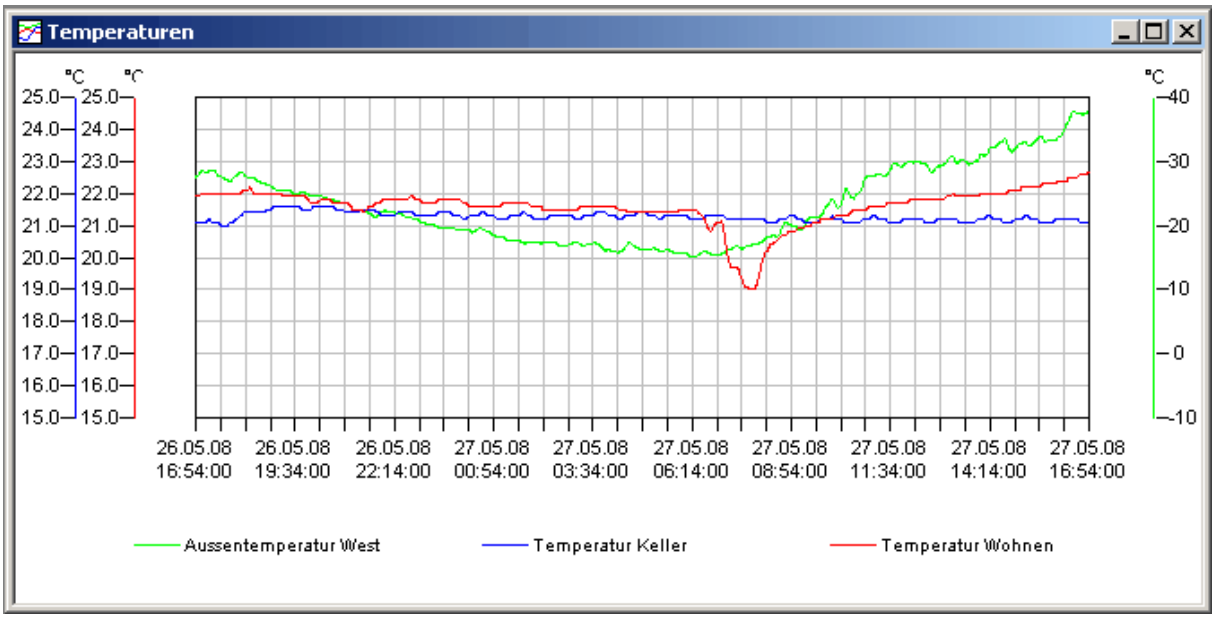
Scaling left/Scaling right:
By ticking one of these boxes, scaling is displayed on the left, right, or on both sides of the pChart. Where no box is ticked, no scaling is displayed. Can only be selected when the “Use separate axes” box has been ticked.

Automatic scaling:
When this box is ticked, scaling can be performed manually.

Difference:
When this box is ticked, the difference from the last historic values is displayed along each curve.

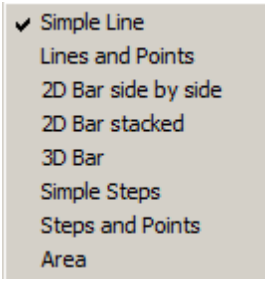
Interval:
This defines the intervals at which the data should be loaded from the database. The data is automatically interpolated.

Example of a diagram with separate axes:

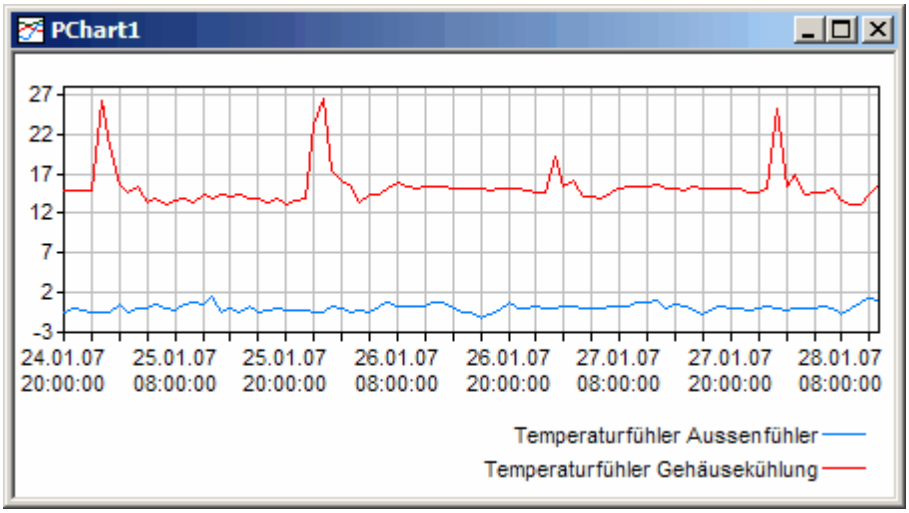


10.8.6.5.2 Type

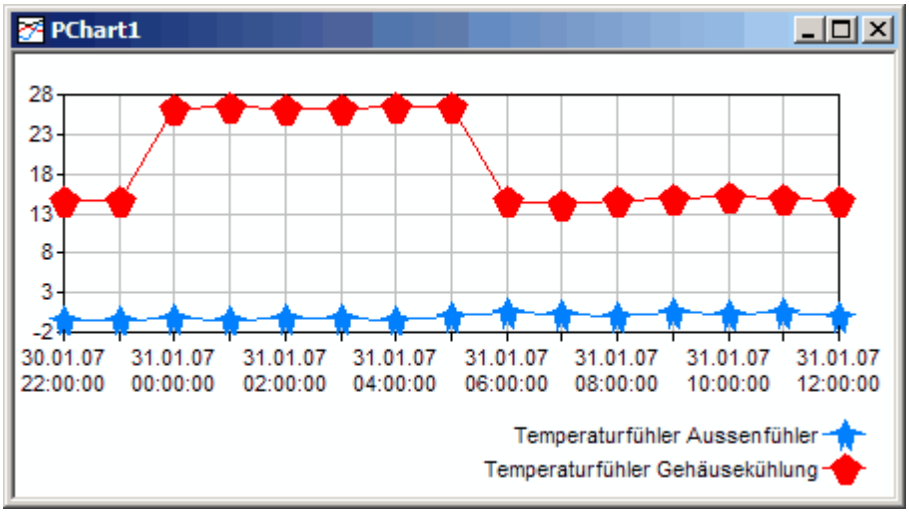
The type of the trend graph can be selected in this menu.



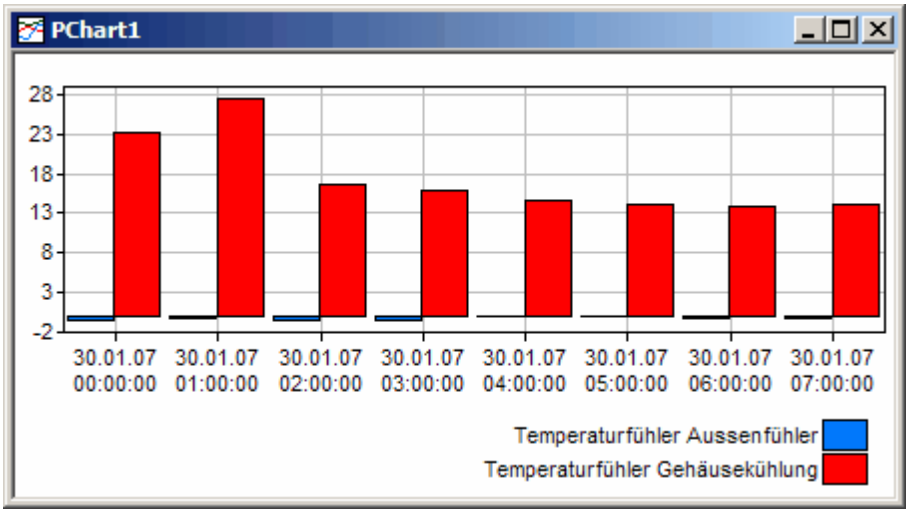
Simple Line
By default, a graph is shown as a simple, conventional rounded-down line.



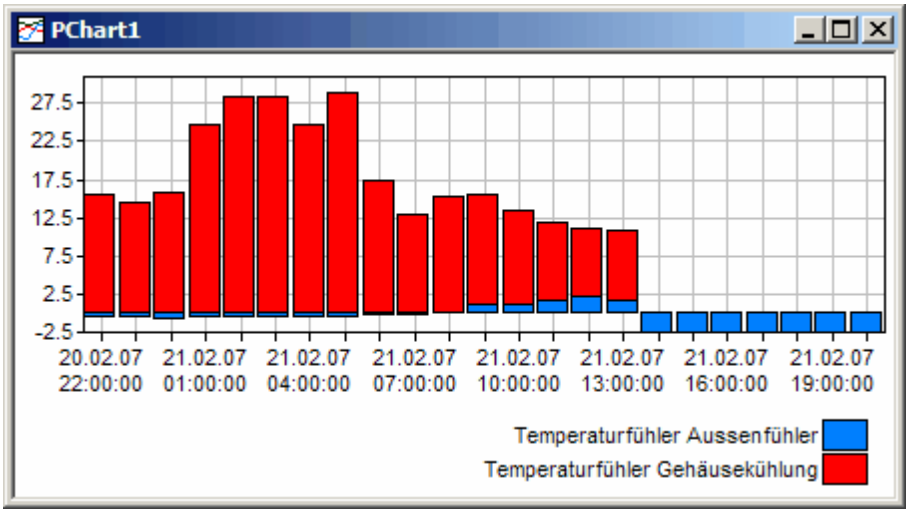
Line and Points
A symbol is placed at a high or low point of the graph. With only one graph this is a point. With multiple graphs, stars, polygons or other symbols are used.



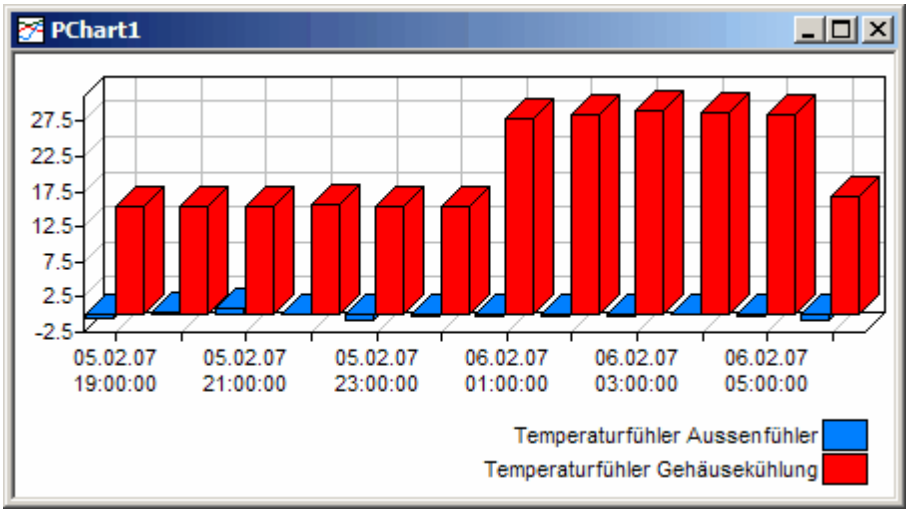
Side-By-Side 2D Bars:
Shows the values as side-by-side 2D bars.



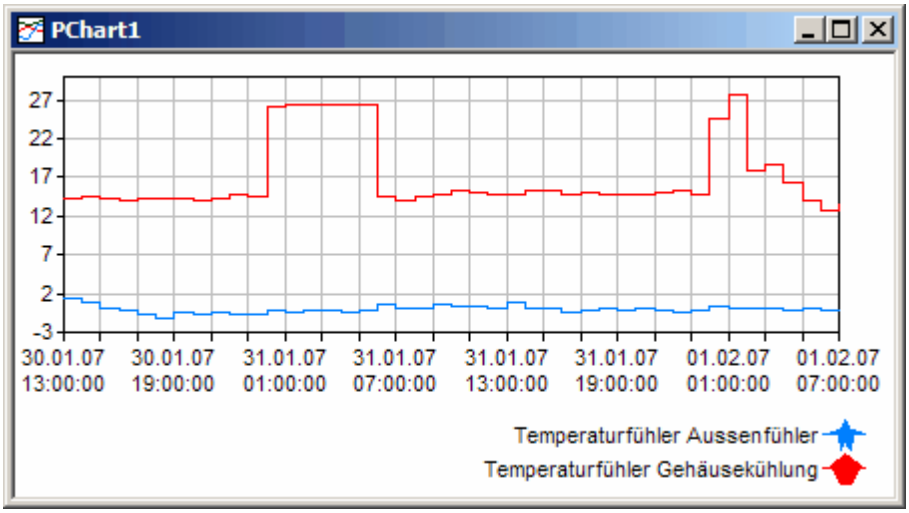
Superimposed 2D Bars:
Shows the values as superimposed 2D bars.



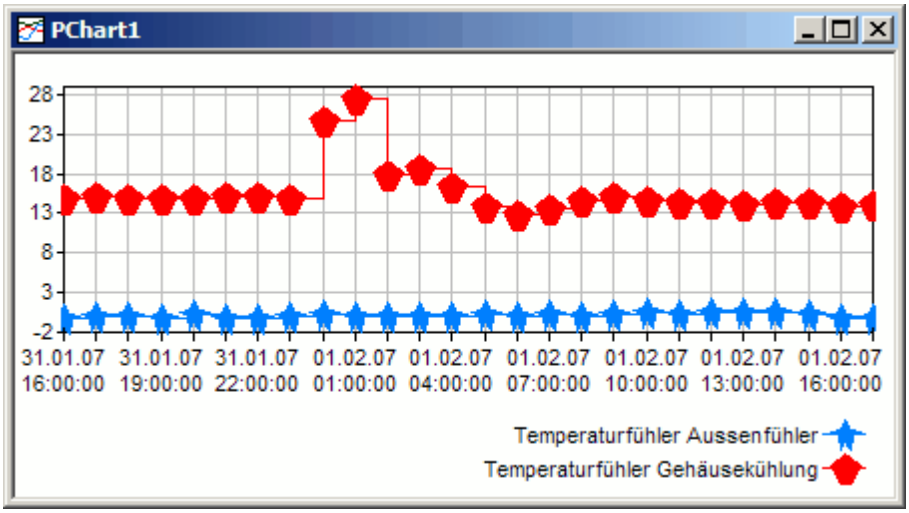
3D Bars:
Shows the values as 3D bars.



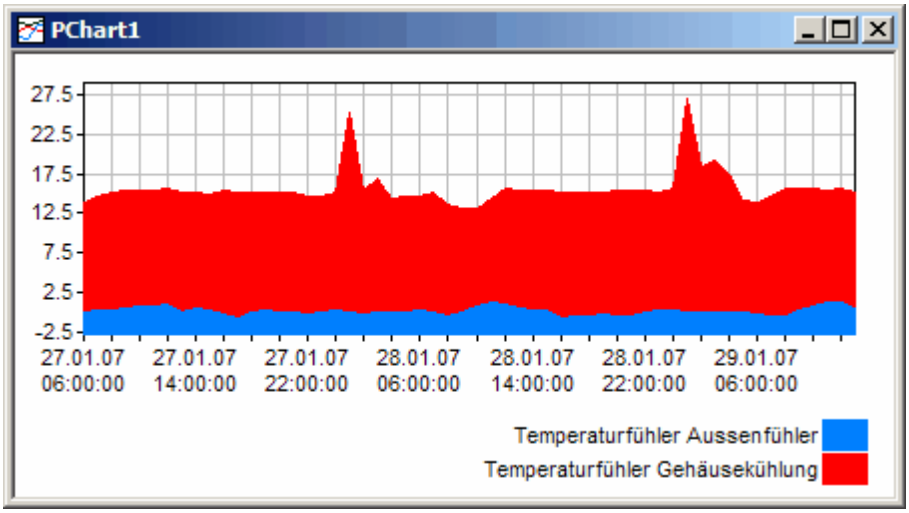
Simple Steps:
The line is displayed in the form of simple steps. The difference from the simple line format is in the gradation of the roundings for changes in value.



Steps and Points:
The same as with Lines and Points, however additionally showing the gradation.



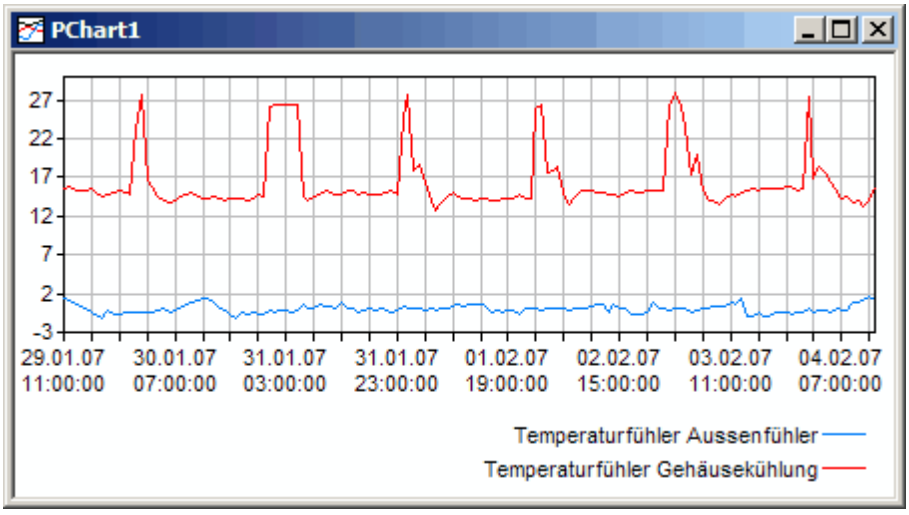
Area:
In this format the area defined by the connected values is filled in.



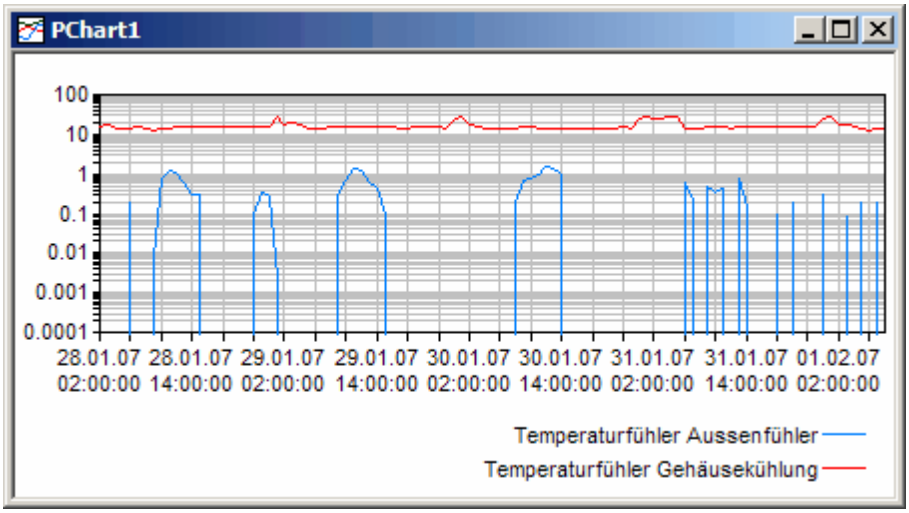
10.8.6.5.3 Y-scale

The scale can be set to linear or logarithmic.

Linear:
Linear representation of the Y-axis

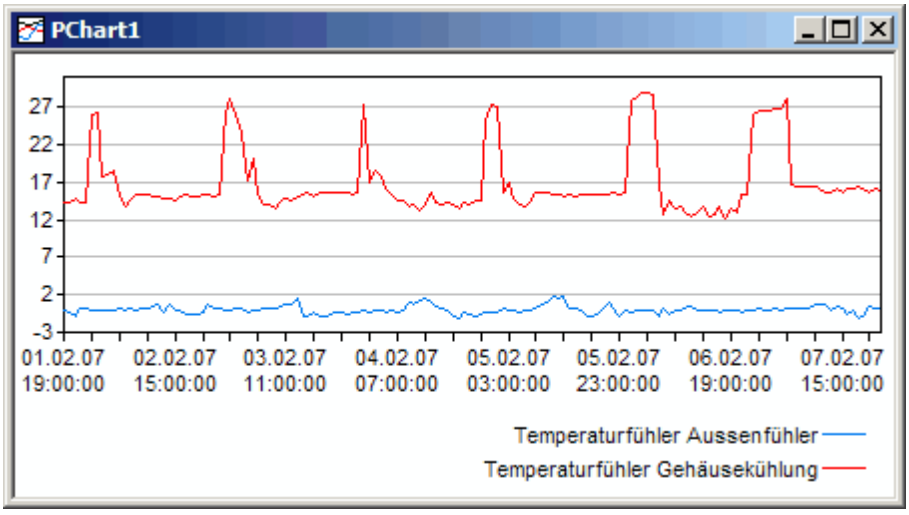


Logarithmic:
Logarithmic representation of the Y-axis.



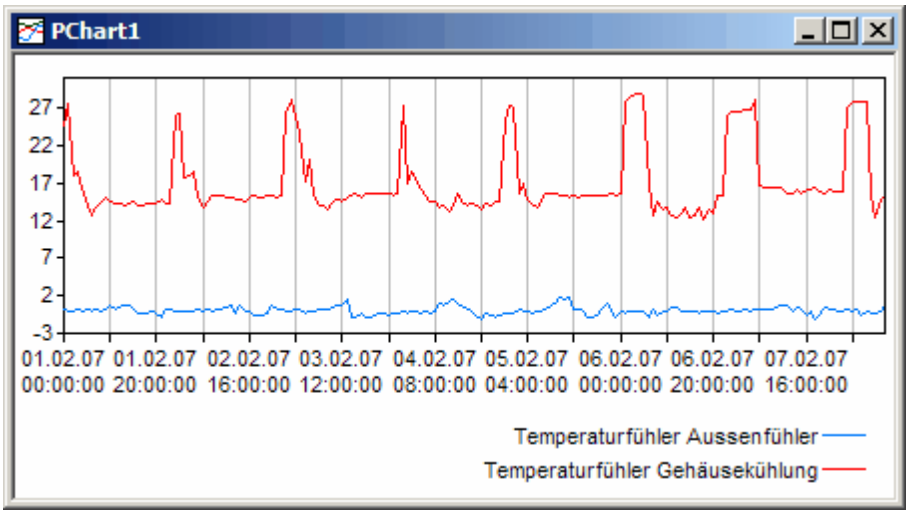
10.8.6.5.4 X-Gridline

Switches the vertical grid lines on or off.



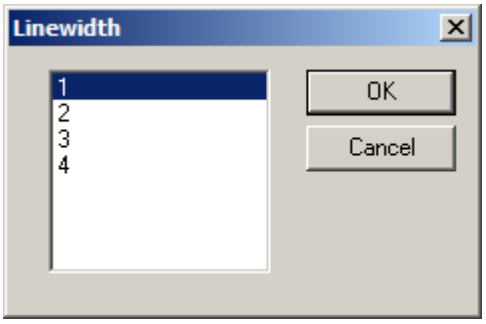
10.8.6.5.5 Y-Gridline

Switches the horizontal grid lines on or off.

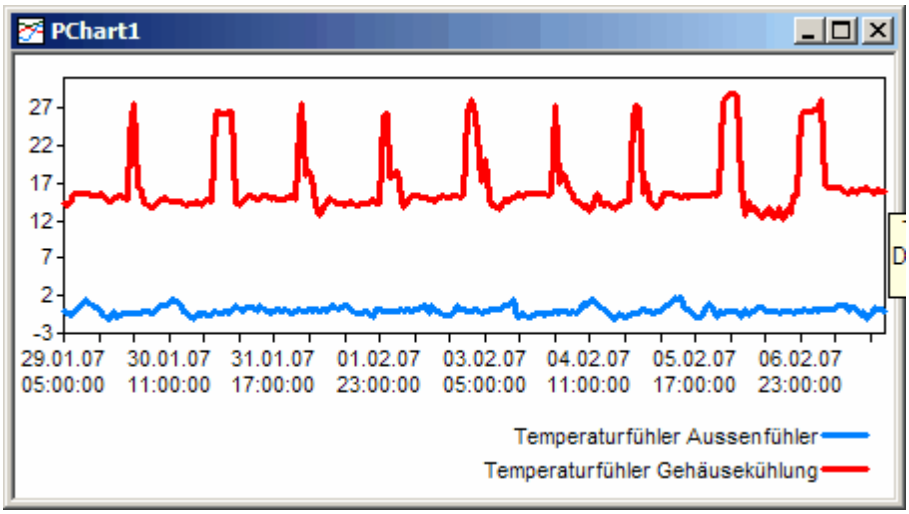


10.8.6.5.6 Line Width

4 different line widths can be selected.

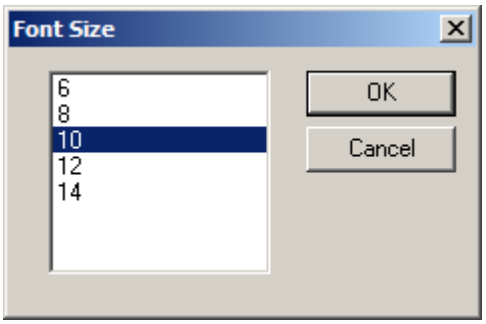


In the following image a line width of 3 was selected.



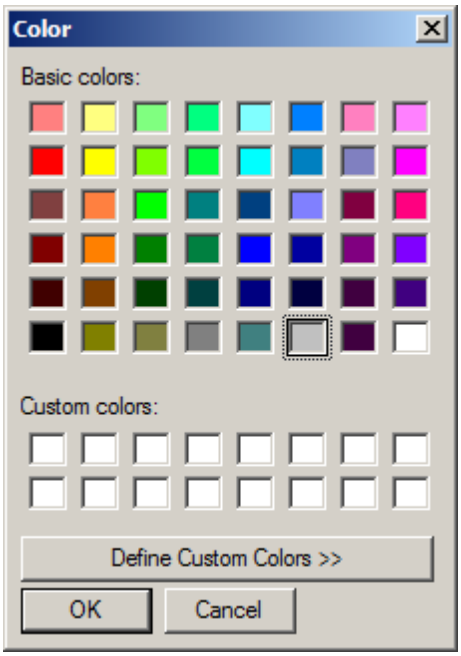
10.8.6.5.7 Font Size

Five different font sizes are available for selection for the designations in diagrams:



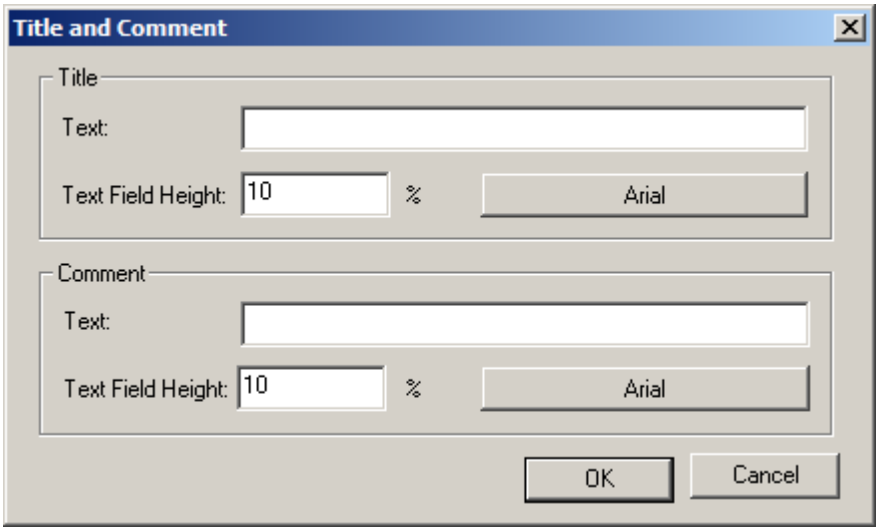
10.8.6.5.8 Grid Colour

Any colour can be selected for the grid here.



10.8.6.5.9 Title and comment

One title and comment can be entered per view for output on the monitor or printer. A backslash causes the title or comment to be output on more than one line. Example: "This is a\ Title" The backslash causes "Title" to be written on the second line.

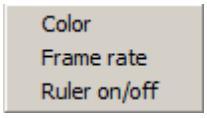


1) Height of the text field:
The height in percent of the text field can be entered here. This must be adapted in dependence on the font size!

2) Font:
When the button is actuated, the configuration window for the font opens. There, the font, font style and font size can be specified.

10.8.6.5.10 Graph ruler

This menu allows various settings for the graph ruler to be made. There are 3 sub-menus available:



Colour:
The colour of the graph ruler can be set here.

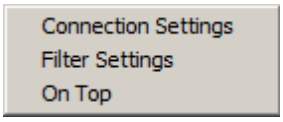
Playback speed:
The speed can be specified here.

Graph ruler on/off:
Hides or displays the graph ruler.

A detailed description of the function of the graph ruler can be found under:The pChart user interface

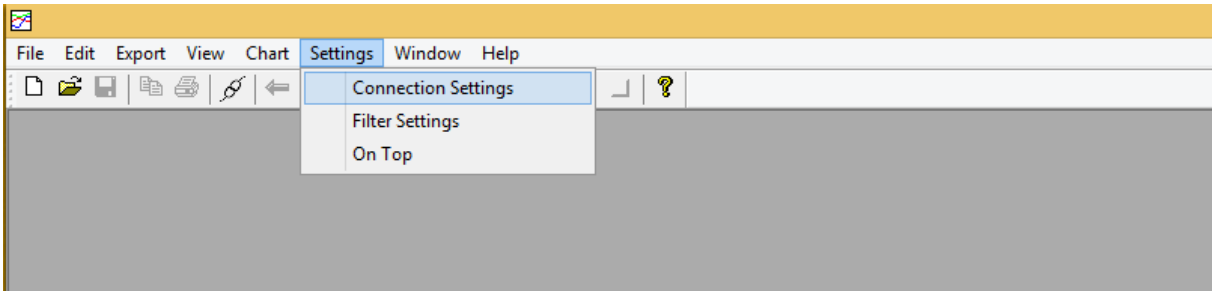
10.8.6.6 Settings menu

The connection and the filter settings can be configured in the Settings menu.

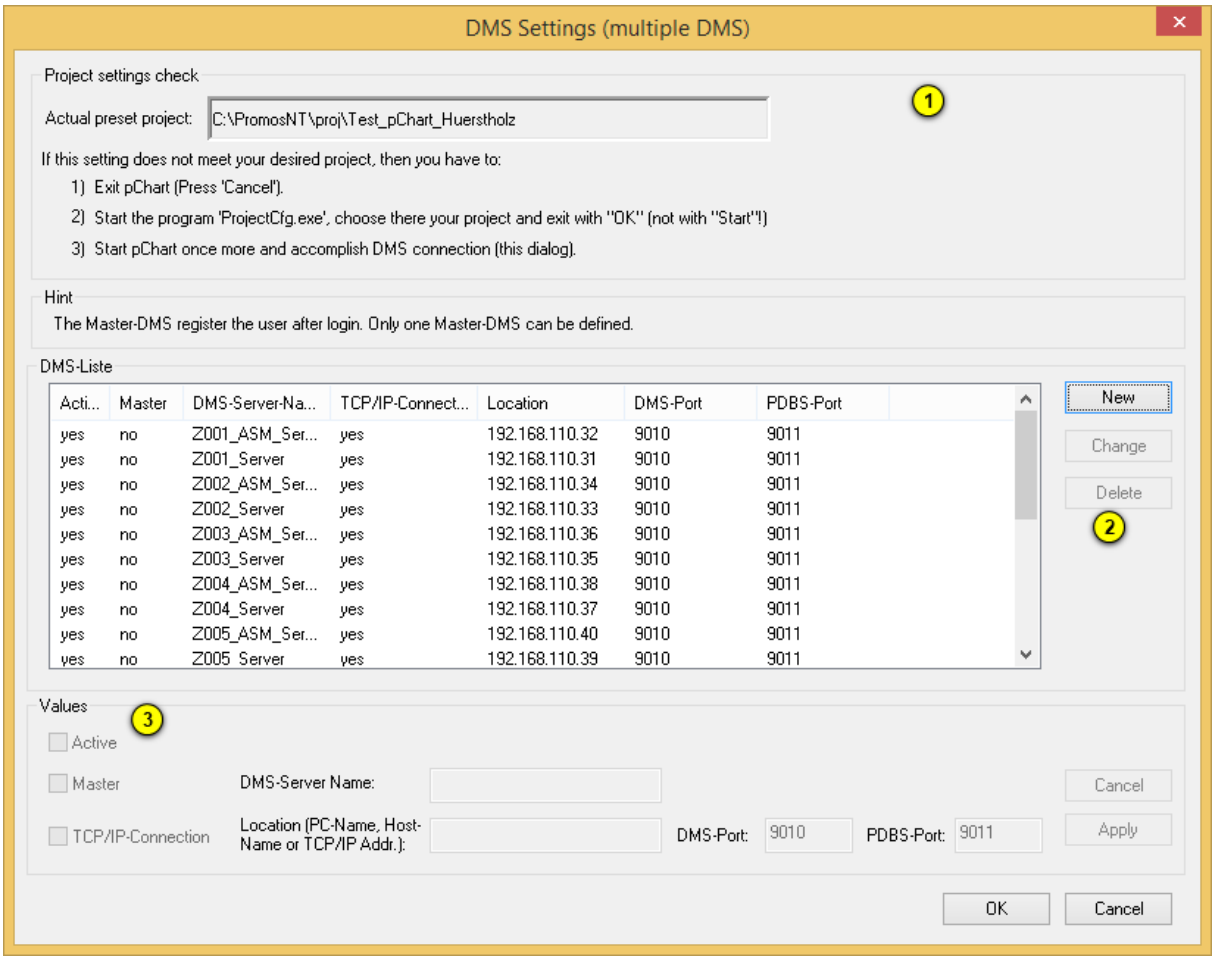


10.8.6.6.1 Connection Settings

Connection settings can be defined using the **"Settings > Connection settings"** menu.



This opens up the settings menu. The layout of the settings menu is identical to that in GE and AlarmViewer.



1 Project settings:
This is where the project settings are listed. This information is secondary for remote connections. The pChart templates are saved in the project folder. The connection settings are also stored in the project folder.

DMS list:
All connections that have been defined are listed here. Clicking on the **<New>** button creates a new connection. The **<Change>** button is used to change an existing connection. Using the **<Delete>** button deletes a connection.

Values:
This is where the various connections are configured. These settings vary for each DMS. These options are only available when either **<New>** or **<Change>** has been activated in the DMS list

Active:
Tick this box to search for this connection on start-up. Otherwise it will be ignored.

Master:
Ticking this box defines this as the master connection. The master connection is important for users logging into the system. Only one connection can be defined as the master connection at any one time.

TCP/IP connection:
Ticking this box establishes the connection to the TCP/IP protocol. Ticking this box connects pChart to the DMS and PDBS via PIPE.

Location:
This specifies the IP address or PC name where the DMS/PDBS is located.

DMS port/PDBS port:
Where the connection to DMS and/or PDBS is established via TCP/IP, the port can also be specified. Usually port 9010 is active for DMS and port 9011 for PDBS.

The **<OK>** button saves the settings and, where necessary, the pChart is rebooted.

The dot- sign "." for the connection is only working for Pipe. With TCP/IP it won't work. For local connection under TCP/IP the setting localhost or IP-Address 127.0.0.1 is used.

If the Multi DMS settings were configured in GE or AlmViewer, the configuration is read from the MultiConn.cfg file at startup. If this was not the case, the GEConn.cfg file, if available, is copied to MultiConn.cfg. Otherwise, the original configuration is copied from the pChart.cfg to the MultiConn.cfg file. In all three cases, the UseCommMultiCfgFile entry in the Settings section is set to 1 in pChart.cfg. pChart now uses the configuration from the MultiConn.cfg file. If the MultiConn.cfg file was newly created, it can now be used by GE and AlmViewer. Therefore, only one configuration file has to be created for all three modules. If an individual configuration is required that is saved in the pChart.cfg file, the UseCommMultiCfgFile entry must be set to 0.

10.8.6.6.2 Filter

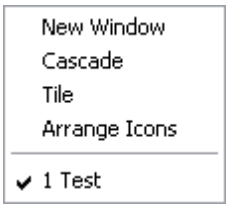
Similar to the procedure in the **"Edit -> Set graph details"** menu, a filter can be set for the PLC here.

10.8.6.6.3 To the foreground

When this menu item is activated, pChart is always kept in the foreground with reference to the graphical editor. This simplifies the drag & drop function of objects to pChart.

10.8.6.7 Window menu

The "**View**" menu presents different options for modifying the appearance of pChart.



New Window
By selecting the command "**New window**", a new window is opened in the current pChart view.

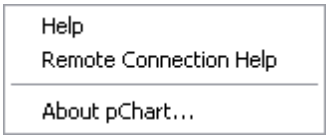
Cascade Windows
The pChart windows are shown in a staggered, cascade arrangement.

Tiled
Displays the pChart windows 'tiled' across the display area.

Arrange Windows
Arranges the windows.

10.8.6.8 ? Menu

The Visi.Plus online help and current information about pChart are found under the "**?**" menu bar.



Help
 or **<F1>** or **"? > Help" in the menu**
Select the Visi.Plus online help.

 **About pChart**
Displays the current version of **pChart**. The version must always be given in case of any support queries!

Should support queries become necessary, the internet forum is preferential to contact via telephone or e-mail. This offers two advantages: First, help is available more quickly; and secondly, other users benefit from the entries.

10.8.7 Starting pChart using program parameters

pChart can be started using program parameters, e.g. from GE or from a batch file. ***Doing so allows, for example, historical data to be exported automatically and immediately sent via e-mail! pChart runs during this as a background process.***

This is the call syntax:
pChart <ppc-file> /A:<MM.YY or MM.YYYY> /E:<ExportFilename>

Parameter Value:

/Y: YYYY
➔ For indicating the year

/M:<MM.YY or MM.YYYY>
➔ For indicating the month

/W:<CW.YY or CW.YYYY>
➔ For indicating the calendar week and the year (CW = calendar week)

/D:<DD.MM.YY or DD.MM.YYYY>
➔ For indicating a date (day)

If no value is entered after the colon, the current year, calendar week or the current date is adopted by default

/O:<Number>
➔ Here you can indicate how many years, months, weeks or days it should scroll forward or back from a specified point in time. With a positive number it scrolls forward. If the number is negative, it scrolls back. /D: /O:-31 means scroll back 31 days.

Please do not confuse the letter O with the number 0 when entering the offset.

/Show
➔ Open pChart with the adjusted settings

Example: *Test.ppc /M:06.13 /Show*
Open the template *Test* in pChart for the month June in the year 2013

/E:<ExportFilename>

➔ For indicating an export filename according to your requirements.

/Z:
➔ Add the current Daten and Time to the Name of the csv- File.

Example: */E:C:\Daten\test_Juni_13.csv /Z:*
Save the csv- file under *C:\Daten* as *Test_15.08.2013_11.12.36.csv*

Let us clarify how to use this parameter using some examples.The current date is 29.05.08. This date falls in calendar week 22.

/Y:	Exports trend data for the current year (2008)
/Y: /O:-1	Exports trend data for the previous year (2007)
/Y:2006	Exports trend data from 2006
/Y: 2006 /O:2	Exports trend data from 2008

/M:	Exports trend data for the current month (June 07)
/M: /O:-1	Exports trend data for the previous month (May 07)
/M:12.06	Exports trend data for December 2006
/M:12.06 /O:2	Exports trend data for January 2007

/W:	Exports trend data for the current calendar week (week 22)
/W: /O:-2	Exports trend data from calendar week 20
/W:14.2007	Exports trend data from calendar week 14 of 2007
/W:14.2007 /O:-2	Exports trend data from calendar week 12 of 2007

/D:	Exports trend data for today (29.05.08)
/D: /O:-1	Exports trend data for yesterday (28.05.08)
/D:11.06.2007	Exports trend data for 11.06.07
/ D:11.06.2007 / O:2	Exports trend data for 13.06.07

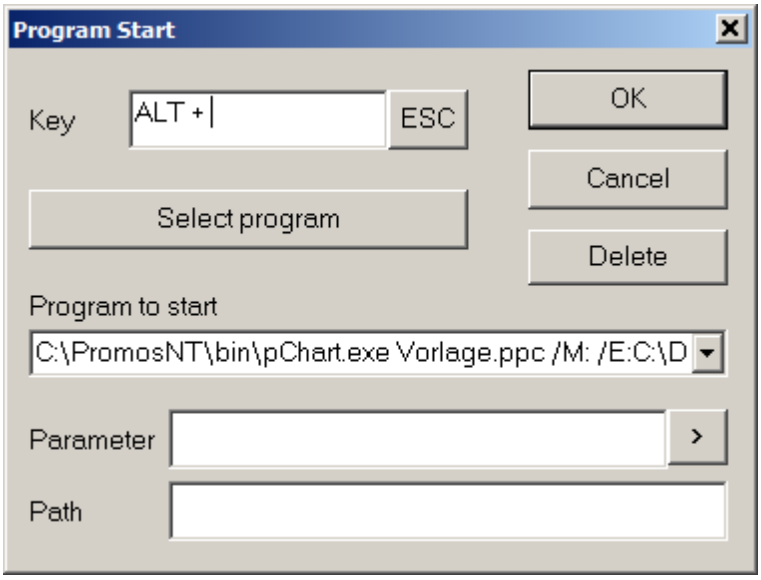
 ***A character space must always be inserted between the individual parameters.***

Trend data can also be exported from within the GE, for example via a switch that must be initialized on Action > Program Start > Any Program. However, the template must previously be created in pChart and saved to the project's *ppc* folder.

There are several different ways of calling from the GE:

Variant 1:

In the entry field under "Program to be Started", pChart can be opened with the following parameter:



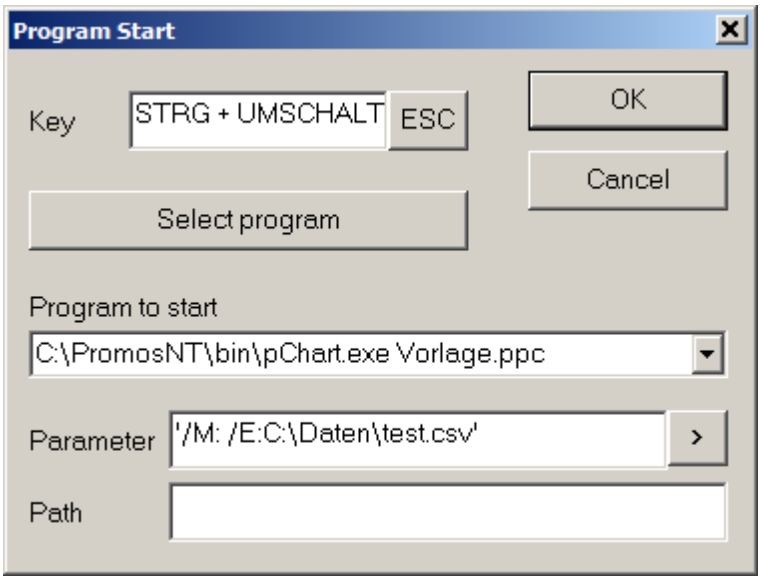
<Path of pChart.exe File> <Name of Template> <Parameter for Data> <Export Path>

Example:
C:\Visi.Plus\bin\pChart.exe Template.ppc /M: /E:C:\Data\test.csv

C:\Visi.Plus\bin\pChart.exe indicates the absolute path where pChart is stored. After a character space comes the name of the template stored in the project's ppc folder. Then come the parameters for the required export data and the path indicating where the file should be saved as a csv file.

The stored template must always be located in the project's ppc folder.

Variant 2:
The parameters are listed in the ***Parameter*** entry field.

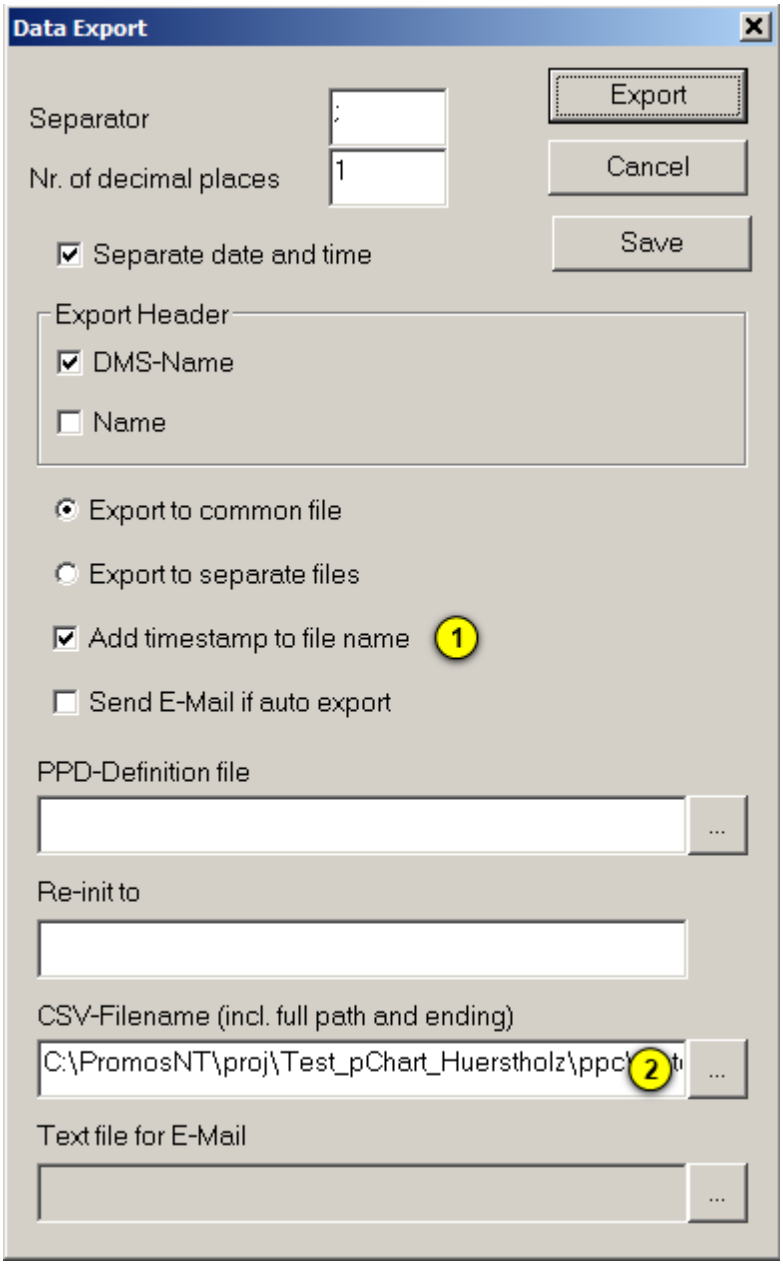


The entire parameter must be set between inverted commas (masking). If there are no inverted commas, when pChart is called any colon characters will be replaced with underscore characters. The call should therefore be written like this: '/M: /E:C:\Data\test.csv'. This instruction is only partially understood by pChart. The data export is missing. So instead, pChart will be started up with the template file test.ppc.

The file name of the pChart template does not always have to be in the entry field under "Program to be Started", but can also be located under "Parameter". The entry under "Parameter" will then look like this: 'Template.ppc /M: /E:C:\Data\test.csv'.

Variant 3:

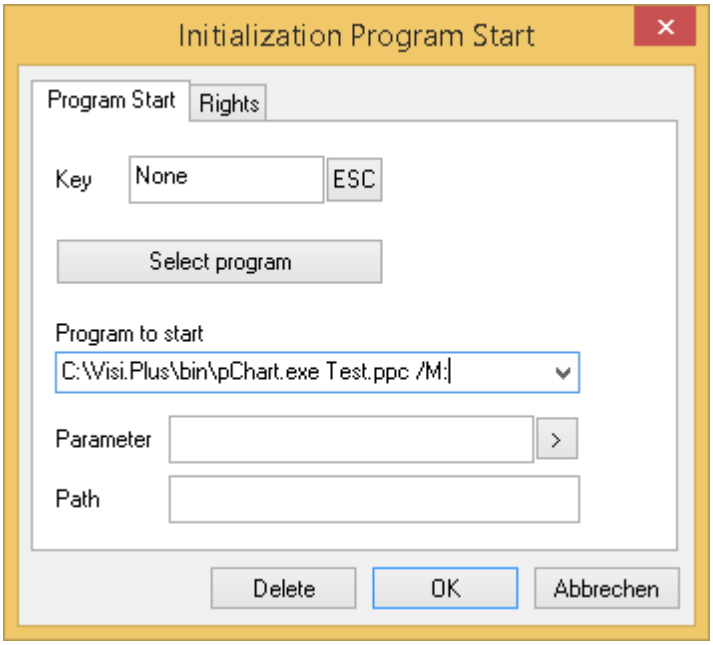
The 3rd variant consists of previously defining the memory path and csv file name in the templates. This has the advantage that the date and time can also be stored in the file name. The settings can be adjusted in the pChart under **Export-> Data Export**.



1 With this checkbox, the time stamp is added to the end of the file name.

2 The memory location for the csv file can be indicated here.

The Program Start looks like this:



Call via a control function:

With the control function, a program can be started by triggering a data point. Parameters can also be transferred at the same time. The control function must be defined as follows:

```
pchart [OFF]
  PRG [PRG(C:\Visi.Plus\bin\pChart.exe Test_2.ppc /Y: /E:C:\Daten\Dillier.csv)]
```

The entry may be defined as in GE variant 1 or variant 3.

10.8.8 Structured table export

10.8.8.1 Introduction

If a table in accordance with the following example must be created by means of data export, this can be realised through the structured table export.

Possible CSV file:

	A	B	C	D	E
1	Bei diesem Versuch wird der Einfluss der Lichtverhältnisse gestestet.				
2	Dauer: 5 Tage				
3					
4	Proportionalband	10,0	s		
5	Nachstellzeit	600,0	s		
6	Minwert	10,0	KLux		
7	Maxwert	60,0	KLux		
8	Sollwert	20,0	°C		
9					
10	Zeit/Datum	H01:B01:M01	H01:B01:M02	H01:B01:M03	H01:B01:M04
11	01.01.1970 00:00	1	2	3	4
12	01.01.1970 00:01	2	3	4	5
13	01.01.1970 00:02	3	4	5	6
14	01.01.1970 00:03	4	5	6	7
15	01.01.1970 00:04	5	6	7	8
16	01.01.1970 00:05	6	7	8	9
17					

The information displayed within the table can be freely defined, down to the range of data exported (in the example, as far as line 11). Using freely definable table entries, either specific text (in the example, the fields A1, A2, A4, C4, etc.) or the contents of any DMS name can be output.

10.8.8.2 Definition file

Three sections are inserted into the PPD definition file. The extension of the table is defined using the ASCII Editor. It is part of the engineering to generate the PPD files required by the customer. The customer can then select it on data export.

```
[Header]
Tag1=A1,D,%s,Trace01:StartStop:Text1

Tag2=A2,T,%s,Start:
Tag3=B2,T,%d.%m.%Y %H:%M:%S,$Start

Tag4=A3,T,%s,End:
Tag5=B3,T,%d.%m.%Y %H:%M:%S,$End

Tag6=A5,T,%s,ProportionalBand
Tag7=B5,D,%3.1f,Trace01:StartStop:Param1
Tag8=C5,T,%s,s

Tag9=A6,T,%s,ResetTime
Tag10=B6,D,%3.1f,Trace01:StartStop:Param2
Tag11=C6,T,%s,s

Tag12=A7,T,%s,MinValue
Tag13=B7,D,%3.1f,Trace01:StartStop:Param3
Tag14=C7,T,%s,KLux

Tag15=A8,T,%s,MaxValue
Tag16=B8,D,%3.1f,Trace01:StartStop:Param4
Tag17=C8,T,%s,KLux

Tag18=A9,T,%s,TargetValue
Tag19=B9,D,%3.1f,Trace01:StartStop:Param5
Tag20=C9,T,%s,°C
```

```
BeginData=A12
ReplaceBMO=Trace01

[StartEnd]
StartDate=ALG:StartStop:StartDate
StartTime=ALG:StartStop:StartTime
EndDate=ALG:StartStop:StopDate
EndTime=ALG:StartStop:EndTime

[File]
Export= ALG:StartStop:Export
```

[Header] section

The individual tags are numbered consecutively, beginning with Tag1.

The structure of a line is as follows:

Tag_xy = Table Field (according to Excel), Export Type, Export Format, Data

Table Field
The Table Field describes the position within the table. The numbering corresponds to the designation in Excel or OpenOffice Calc. The positions stored in the tags do not have to correspond to any specific sequence. Therefore it makes no difference if the lines (A1, B1, C1, etc.) or the columns are stored first (A1, A2, A3, etc.) or if an arbitrary order is chosen (B2, A1, C3, A2, etc.). This means that tags can be added at any time without having to change the numbering of the tags in the PPD file.

Export Type
The Export Type can assume the following values:

- D The subsequent data corresponds to a DMS name. The output must be loaded from the DMS.
- T The subsequent data corresponds to the data and can be output 1:1
- I The subsequent data corresponds indirectly to a DMS value. The DMS name contained there is read
  and its value is used.

Export Format
The Export Format defines the format of the output. The familiar formats from Visi.Plus are supported.

Example:	%5.2f	Floating-point notation with a total of five digits, 2 decimal places
	%s	Text Output

Data
The data is interpreted according to the export type. Text is adopted 1:1. Values are first read from the DMS and then transferred to the table. $Start and $End are reserved words. The start and/or stop times are entered in their position.

BeginData
Determines the line from which the exported data is written.The output of the data always begins with the first column regardless of whether, for example, A10 (1st column, 10th line) or C10 (3rd column, 10th line) is entered. What matters here is only the line number (number), i.e. in

the example shown, the data will be written as of line 10.

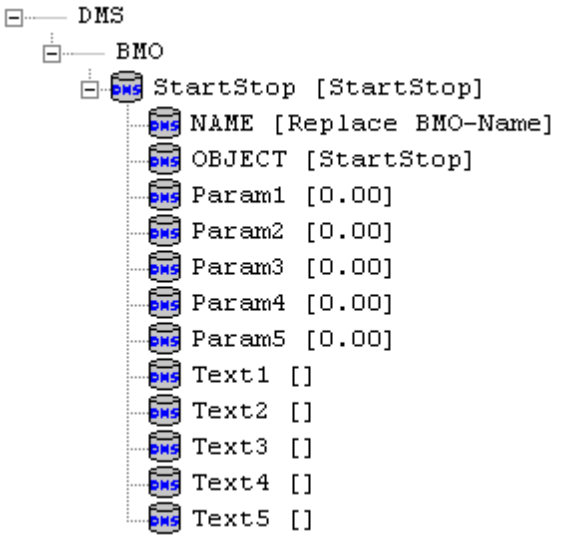
ReplaceBMO
This is where the name is located that takes the place of "BMO" (reinitialisation) when reading the PPD file. If left blank, no replacement takes place. The value is entered in pChart in the Data Export dialog.

[StartEnd] section
Four DMS names that contain the evaluation times (start/stop) can be defined in the [StartEnd] section.
The times are stored in PCD format (according to the input options in the GE). Date and time are stored in two different DMS variables.DMS data points must be used for the date and the time.
According to requirements, reinitialisation - as occurs for parameters and text - is not intended. Therefore they are not reinitialised with a preset "BMO".

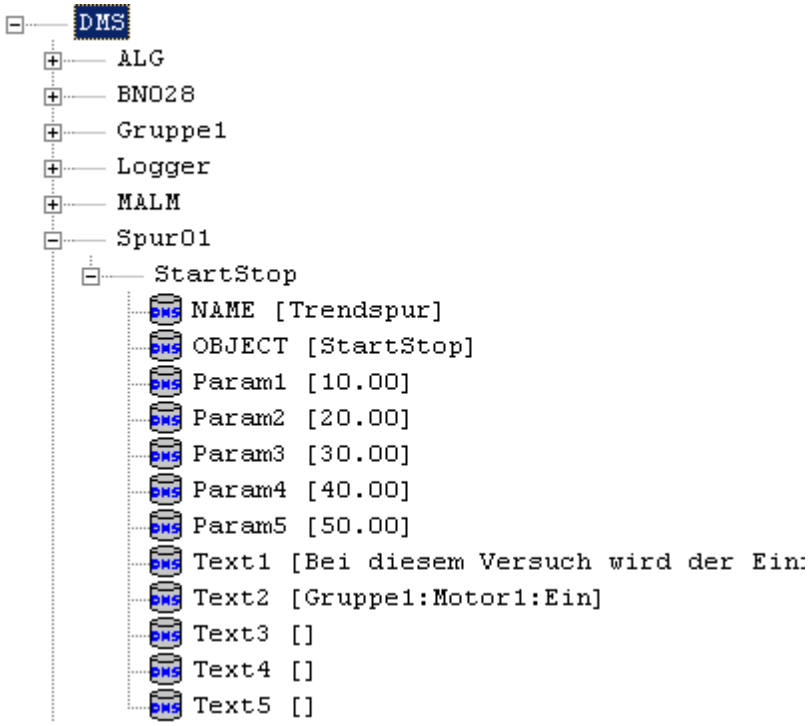
[File] section
The DMS name (type STR), which contains the path for the export file, is defined in this section. If left blank, the entry is taken from the generated PPC file. According to requirements, reinitialisation - as occurs for parameters and text - is not intended. Therefore they are not reinitialised with a preset "BMO".

Inheritance mechanism (reinitialisation)
With this concept, any individual lines and columns can be adjusted individually. In addition, an inheritance mechanism can be realised in simple way (reinitialisation). To do this, an operating resource (BMO) object must first be generated in the PET.

On the basis of the preset PPD file, this appears as follows:



Objects can then be derived from the template object (e.g. Trace01, Trace02, etc.).



Reinitialisation is possible if the DMS names are replaced by BMO names in the PPD file.

The PPD file then appears as follows:

```
[Header]
Tag1=A1,D,%s,BMO:StartStop:Text1

Tag2=A2,T,%s,Start:
Tag3=B2,T,%d.%m.%Y %H:%M:%S,$Start

Tag4=A3,T,%s,End:
Tag5=B3,T,%d.%m.%Y %H:%M:%S,$End

Tag6=A5,T,%s,ProportionalBand
Tag7=B5,D,%3.1f,BMO:StartStop:Param1
Tag8=C5,T,%s,s

Tag9=A6,T,%s,ResetTime
Tag10=B6,D,%3.1f,BMO:StartStop:Param2
Tag11=C6,T,%s,s

Tag12=A7,T,%s,MinValue
Tag13=B7,D,%3.1f,BMO:StartStop:Param3
Tag14=C7,T,%s,KLux

Tag15=A8,T,%s,MaxValue
Tag16=B8,D,%3.1f,BMO:StartStop:Param4
Tag17=C8,T,%s,KLux

Tag18=A9,T,%s,TargetValue
Tag19=B9,D,%3.1f,BMO:StartStop:Param5
Tag20=C9,T,%s,‹C

BeginData=A12
ReplaceBMO=Trace01
```

```
[StartEnd]
StartDate=ALG:StartStop:StartDate
StartTime=ALG:StartStop:StartTime
EndDate=ALG:StartStop:StopDate
EndTime=ALG:StartStop:EndTime

[File]
Export= ALG:StartStop:Export
```

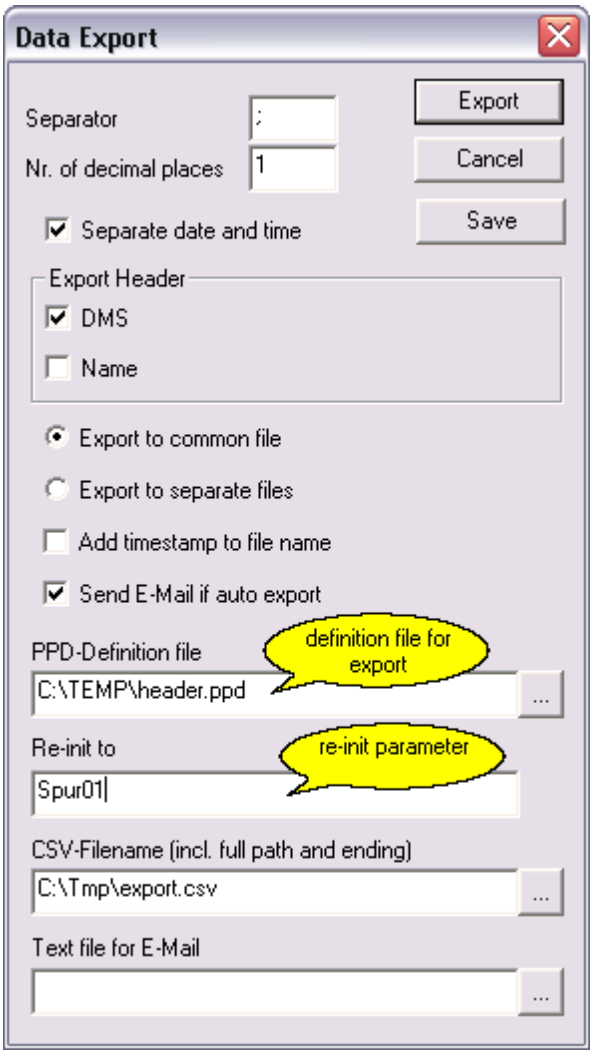
When writing the header, the BMO is replaced by the value in pChart (in this case by Trace01-Parameter ReplaceBMO), so that the DMS names are based on the derived objects.

10.8.8.3 Configuring pChart

The pChart data export dialog contains two input fields for defining structured data exports:

PPD Definition File:
The PPD definition file that should be used for the header is defined here.

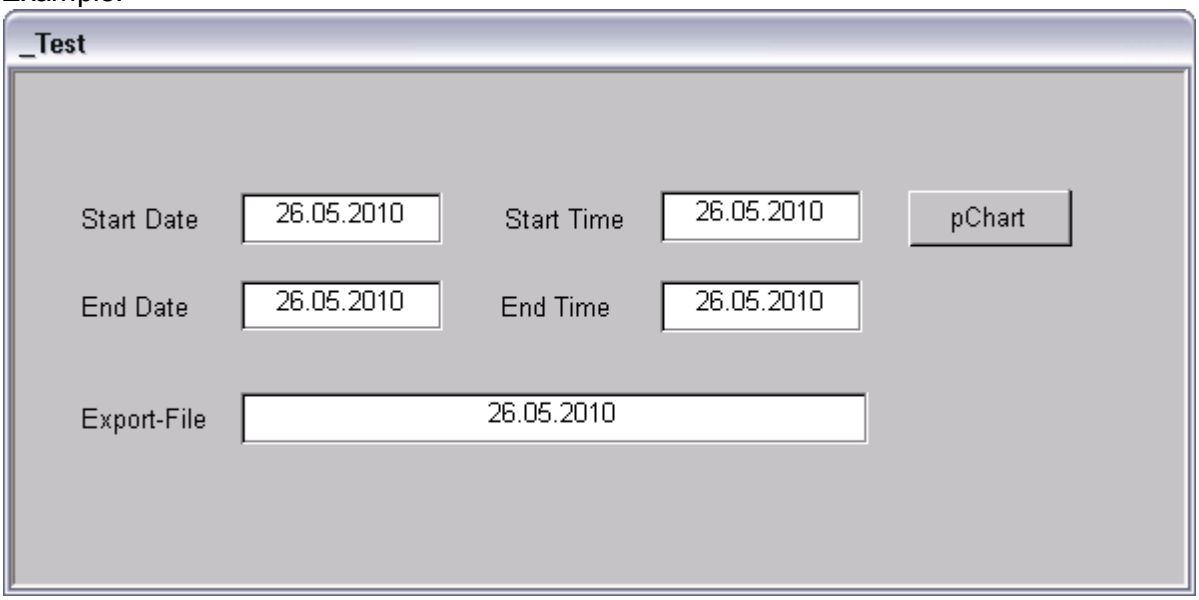
Reinitialise To:
Here the object that should be reinitialised to is indicated. This is entered in the PPD file in the section [Header] under
Replace BMO.



10.8.8.4 Operation

Data export normally takes place through an operator page in the graphical system.

Example:

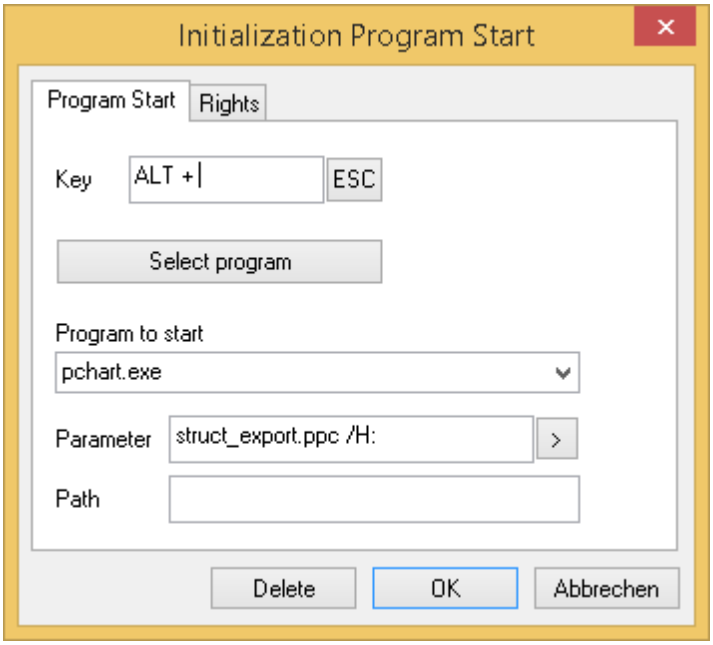


The design of the operator page can be freely determined (in the GE).

In accordance with the specifications from the example PPD file, the edit fields have been initialised to the following values:
Start date: ALG:StartStop:StartDate
Start time: ALG:StartStop:StartTime
Stop date: ALG:StartStop:StartDate
Stop time: ALG:StartStop:EndTime
Exportfile: ALG:StartStop:Export

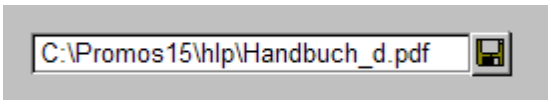
According to the input options in the GE, the starting and stopping times are separated into date and time, meaning a total of 4 DMS data points are required to hold the evaluation times (start/stop). An edit field can also be used for inputting the export filename. The use of edit fields is beneficial because the values can be entered directly from the keyboard.

pChart is started using a button, which must be initialised to Action > Program Start > Any Program.



The PPC file and the parameter /H: are given as parameters (do not forget apostrophe). The parameter /H: means that pChart will start autoexport in the background and read all necessary data from the PPD file or indirectly from the DMS (such as the evaluation times).

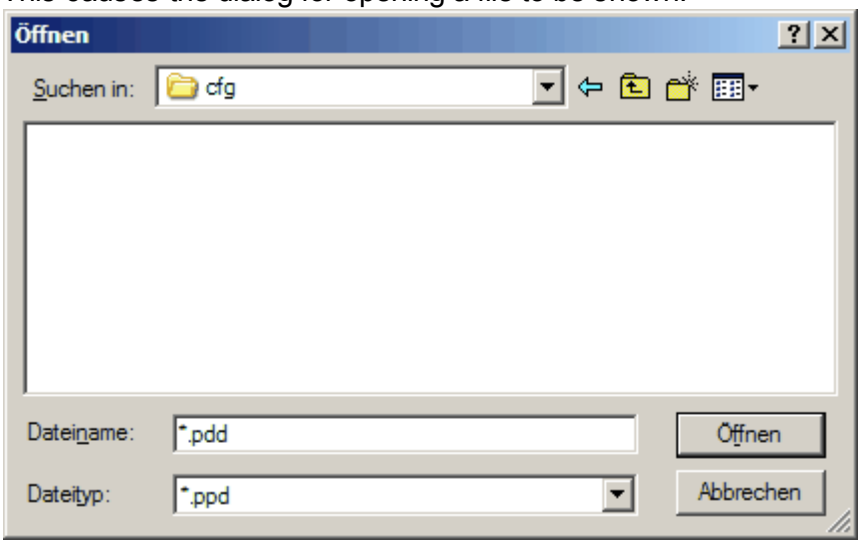
Tip:
A file can also be selected with the program Pmosfilepicker.



The button is initialised as follows:

```
c:\Visi.Plus\bin\pmosfilepicker LOAD System:Filename *.ppd
```

This causes the dialog for opening a file to be shown:



The selected file, including its path, is saved to the DMS data point "System:Filename". This

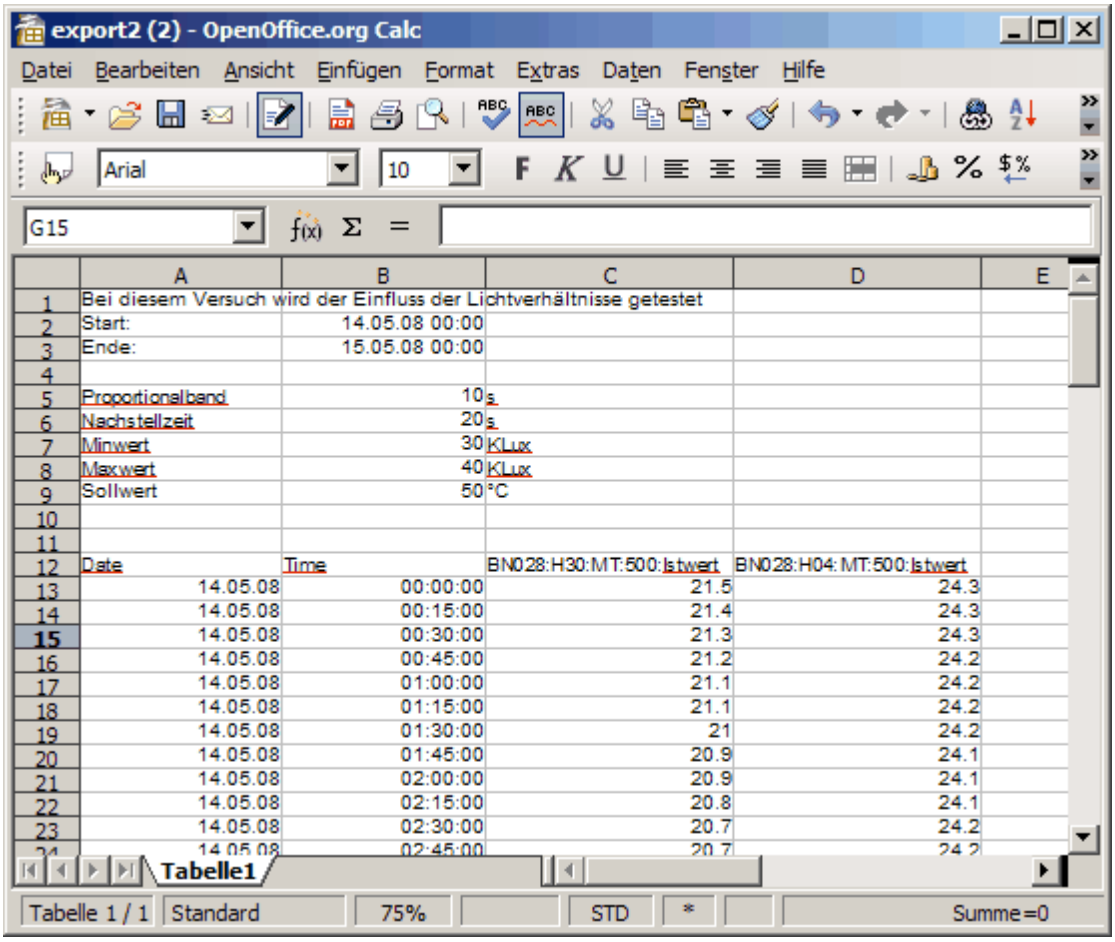
DMS name can be freely defined.

10.8.8.5 Output to File

An example output (can be opened as a CSV file in Excel, OpenCalc, etc.)

```
This test measures the influence of lighting conditions
Start:;14.05.2008 00:00:00
End:;15.05.08 00:00:00
;
ProportionalBand;10.0;s
ResetTime;20.0;s
MinValue;30.0;KLux
MaxValue;40.0;Klux
TargetValue;50.0;°C
;
;
Date;Time;BN028:H30:MT:500:ActualValue;BN028:H04:MT:500:ActualValue
14.05.08;00:00:00;21.5;24.3
14.05.08;00:15:00;21.4;24.3
14.05.08;00:30:00;21.3;24.3
14.05.08;00:45:00;21.2;24.2
14.05.08;01:00:00;21.1;24.2
14.05.08;01:15:00;21.1;24.2
14.05.08;01:30:00;21.0;24.2
14.05.08;01:45:00;20.9;24.1
14.05.08;02:00:00;20.9;24.1
14.05.08;02:15:00;20.8;24.1
14.05.08;02:30:00;20.7;24.2
```

This CSV file can be opened in Excel (the delimiter in the example is a semicolon):



export2 (2) - OpenOffice.org Calc

Datei Bearbeiten Ansicht Einfügen Format Extras Daten Fenster Hilfe

Arial 10

G15

	A	B	C	D	E
1	Bei diesem Versuch wird der Einfluss der Lichtverhältnisse getestet				
2	Start:	14.05.08 00:00			
3	Ende:	15.05.08 00:00			
4					
5	Proportionalband	10	s		
6	Nachstellzeit	20	s		
7	Minwert	30	KLux		
8	Maxwert	40	KLux		
9	Sollwert	50	°C		
10					
11					
12	Date	Time	BN028:H30:MT:500:Istwert	BN028:H04:MT:500:Istwert	
13	14.05.08	00:00:00	21.5	24.3	
14	14.05.08	00:15:00	21.4	24.3	
15	14.05.08	00:30:00	21.3	24.3	
16	14.05.08	00:45:00	21.2	24.2	
17	14.05.08	01:00:00	21.1	24.2	
18	14.05.08	01:15:00	21.1	24.2	
19	14.05.08	01:30:00	21	24.2	
20	14.05.08	01:45:00	20.9	24.1	
21	14.05.08	02:00:00	20.9	24.1	
22	14.05.08	02:15:00	20.8	24.1	
23	14.05.08	02:30:00	20.7	24.2	
24	14.05.08	02:45:00	20.7	24.2	

Tabelle1

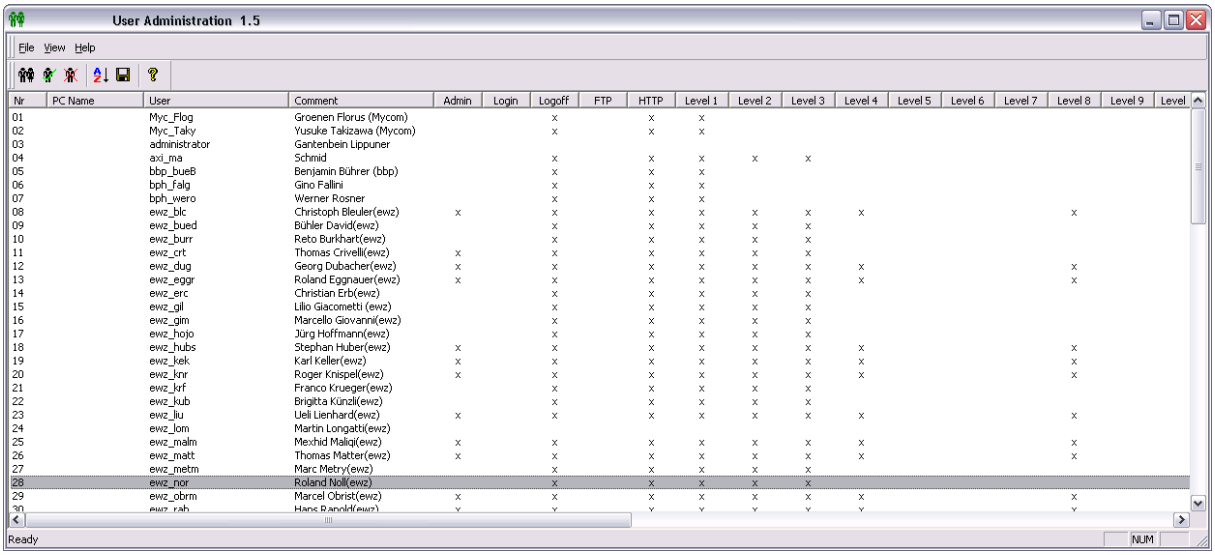
Tabelle 1 / 1 Standard 75% STD * Summe=0

10.9 User Management (pUser.exe)

With the user management module it is possible to create so-called **users** and grant them access rights.

With the assignment of access rights it is possible:

- to approve control elements or input fields in the graphical editor (GE) for specific employees and to lock them for others.
- to display only the alarms or log entries of parts of facilities for which the employee is responsible in the Alarm Viewer or Protocol Viewer.



Any number of users can be defined. The users are stored in the file <proj>\cfg\user.pms. The file is encrypted with a 156-bit key and cannot be edited with a text editor (e.g. TextPad). A defective file must be deleted.

If no file named user.pms is available, the user "Admin" is automatically activated, because no changes would be possible otherwise.

10.9.1 Starting the program

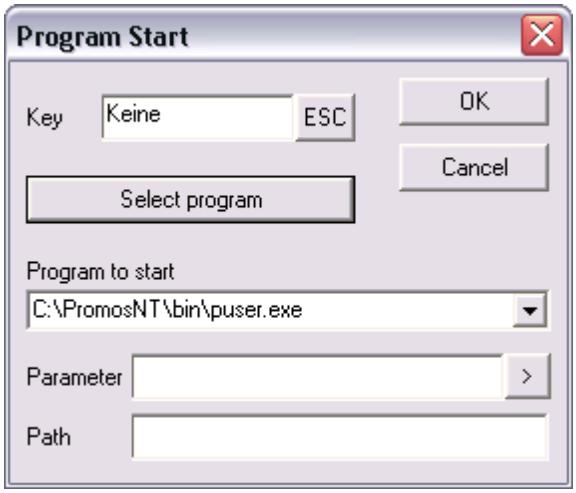
Requirement: The DMS must be running!

The program is started through the Start menu.
"**Start > Programs > Saia Visi.Plus > Tools > User Management**".

Other ways to start the program:
The program **pUser.exe** can be started from the directory **c:\Visi.Plus\bin** using a file management program such Windows Explorer, or via an initialised "Button" in the Graphical Editor.

Brief instructions for initialising pUser to a button:

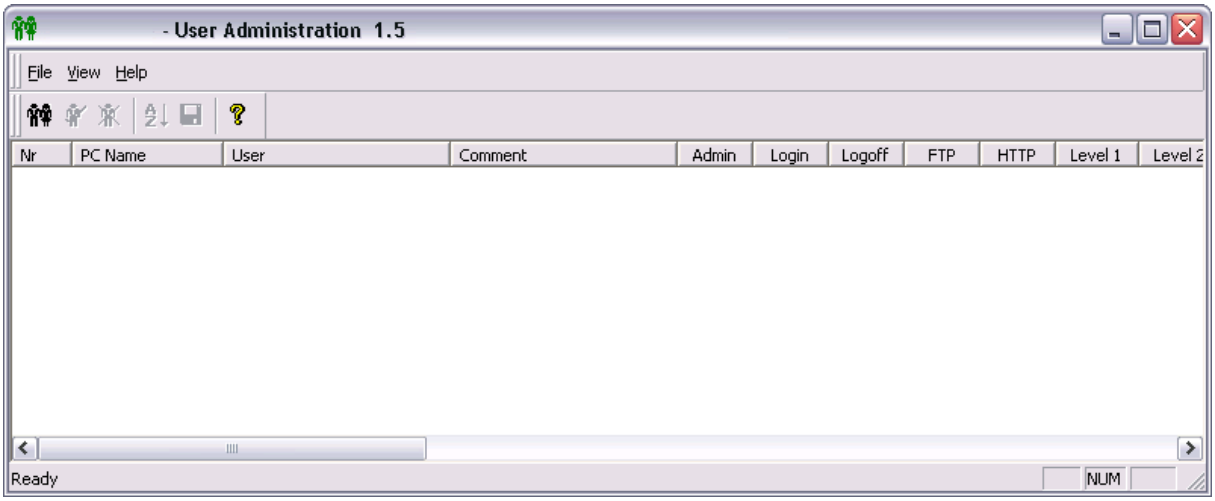
- *After a "**Button**" graphical object has been placed in the process diagram, it must be clicked on once with the mouse in order to be selected.*
- *Then open the context menu with the right mouse button and select "**Attributes**".*
- *In the open Attributes window under "**Action**" in the right column (Initialisation column), select "**Program Start > Any Program**" and enter in the dialog the path for pUser.exe under "Program to be Started":*



- start ***pUser*** *in edit mode in the GE with* ***<CTRL>*** *+* ***<Button>****.*

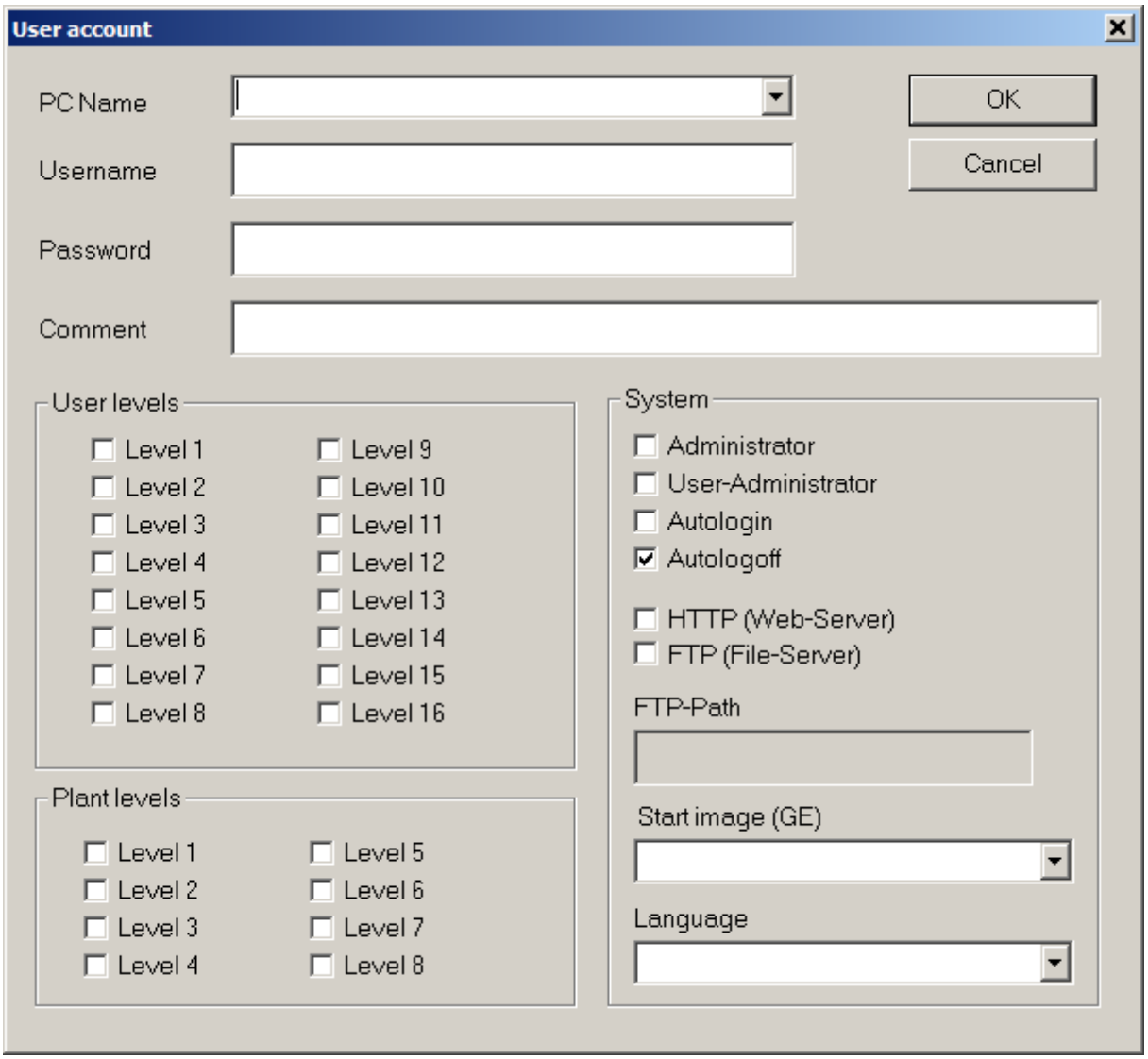
The program can only be started by a user with administrator rights! If the user does not have the necessary rights, the Visi.Plus Login window appears.

If no users have been defined yet, the program window appears as follows after startup:



10.9.2 Creating a new user

or **<CTRL+N>** or the menu command "**File > New User**"



The following attributes can be specified with each user in Visi.Plus:

PC Name
A user can be created dependent on a PC name. In doing so, the user may only log in from the indicated PC. If no PC name is entered, the created user can log in from any remote station. Clicking on the dropdown menu displays the current PC name.

User Name
Here the desired user name is entered.

Password
The password for the user can be defined here. The passwords are saved in encrypted form.

Comment
Any comment can be entered for the user (e.g. user name written out in full form)

Access Levels

Up to 16 access levels can be assigned for each user. They can be selected in any combination. Buttons, for example, in the GE can then be locked. If, for example, a button has been initialised in the GE so that it may only be operated with access level 10, a user must have this access level to be able to press the button.

Plant Levels
In order to explain the meaning of these access rights, it must be discussed in greater detail. Each alarm and log can be assigned a system group in the PET. The alarm Viewer and Log Viewer can then filter according to this group. The idea in doing so is that the logged-in user is only shown alarms and log entries of system parts for which they are responsible. The visibility of alarms and log entries is regulated by their plant levels. If an alarm or log was assigned to System Group 2, it is only shown to the logged-in user if plant level 2 is available to that user.

8 plant levels are available. They can be selected in any combination.

System section

Administrator
Any user can be given **administrator rights**. If users have **administrator rights**, they can switch to **edit mode** in the Graphical Editor by pressing the **<e>** key and carry out changes anywhere. Users without administrator rights cannot switch to edit mode in Visi.Plus.

Auto-Login
If the checkbox **"Auto-Login"** is marked by a user, they are automatically logged in if another user has logged in beforehand. Auto-Login can only be assigned to one user.

*Auto-Login should **only** be assigned with systems where only one person has access.*

Auto-Logoff after 5 Minutes
If this checkbox is marked, the user is automatically logged off if no activity has been detected by the system for 5 minutes (mouse movements, keypresses).

HTTP (Web Server)
If this checkbox is marked, the user can operate the system via Web Server (if licenced).

FTP (File Server)
If this checkbox is marked, the user can exchange data over the Visi.Plus FTP Server. By marking the checkbox the input field **"FTP Path"** can be edited.

FTP Path
Here a specific folder can be enabled for the user. The user can only copy, delete or move data in the enabled folder.

Languages
The language for the user can be selected here. When the respective user logs in, the corresponding system texts are then loaded. The texts are available in German, French and English.

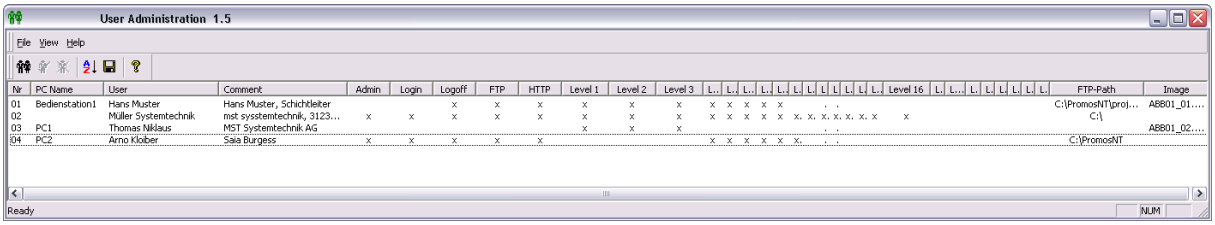
User Administrator

A user with these rights may create new users and change their permissions, except for the Admin option and the rights they themselves have.

Example:
If the folder "**C:\Visi.Plus\proj\<project-name>**" is shared for a user, the user may access all data in the project-folder as well as all sub-folders. The user cannot access higher-level folders.

Image
If a specific process startup diagram should be shown in the Graphical Editor when a user logs in, it can be indicated here. The selection is made from a list of all process diagrams in the project, which appears by clicking on the drop-down menu.

If users are defined, the rights available to the user can be seen from an **X** in the User Management window:



10.9.3 Edit User

In order to edit a user in the **User Definitions** window, proceed as follows:

Select the appropriate entry in the User Management window with the mouse (line is then highlighted in blue) and then click on  , press the **<ENTER>** key, select the menu command "**File > Edit User**" or **double-click on the line with the mouse.**

10.9.4 Delete User

To delete a user, select the relevant entry in the User Management window with the mouse (line is then highlighted in blue) and then click on  , press the **<DEL>** key or select the menu command "**File > Delete User**".

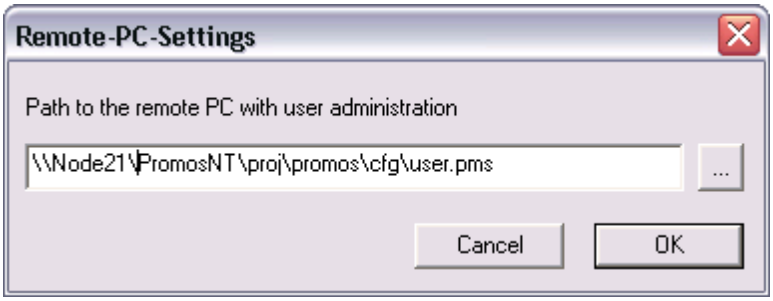
10.9.5 Configuring users for a remote DMS

Users are always created on the PC that the DMS runs on.

In order to make users available on remote stations (workstations, also known as control stations), the user file just needs to be entered on the workstation in the program **pUser**.

When creating users, the user data is saved in the default installation directory **C:\Visi.Plus\proj\ProjectName\cfg** in the user file **user.pms**.

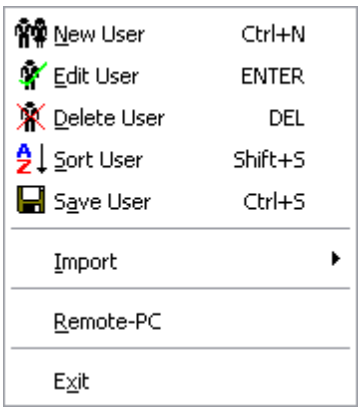
The network path and the corresponding user file are entered via **File > Remote DMS** in the menu.



The path can be selected by clicking on the button to the right of the input field.

10.9.6 User Management menus

10.9.6.1 File menu



New User <CTRL+N>
Creates a new user (see chapter Creating a new user).

Edit User <ENTER>
Allows you to edit a user (see chapter Edit Users).

Delete User <DEL>
Deletes a user (see chapter Delete Users).

Sort Users <SHIFT+S>
Sorts the user data. The users are first sorted according to the PC name and then based on the **User** column.

Save User <CTRL+S>
Saves the user data in the file "**user.pms**" in the current project under **C:\Visi.Plus\proj\ProjectName\cfg**.

Remote DMS
Makes the users available to remote stations (see chapter Configuring users for a remote DMS).

Exit
Exits the program "**pUser.exe**".

10.9.6.2 View menu

In the "**View**" menu, the **toolbar** and/or **status bar** can be switched on or off.

Toolbar
Switches the toolbar on or off.

Status bar
Switches the status bar on or off.

10.9.6.3 Help menu

The "**?**" menu contains the Visi.Plus online help and current information for the **pUser** version.

Help or the icon or the <F1> key
Visi.Plus online help is called up.

About pUser...
Displays the current version of **pUser.exe**. The version must always be given in case of any support queries!

Should support queries become necessary, the internet forum is preferential to contact via telephone or e-mail. This offers two advantages: First, help is available more quickly; and secondly, other users also benefit from the entries.

10.10 The MAlm Configurator (MalmCfg.exe)

The **MAlm** Configurator is use to create alarm groups and users. Furthermore, the alarms are configured with the following notification types:

- E-mail
- SMS: SMS-UCP protocol for Switzerland and SMS-TAP protocol for Austria
- Pager
- Cityruf (paging service in Germany)

An alarm group may, for instance, be a group people (users) who are to be informed if a specified alarm occurs, e.g. fitters or technicians.

MALM configuration procedure:

Create remote alarm groups.
Create remote alarm users.
Assign users to the group.
Create remote alarm format.
Create a remote alarm in the Pet.
Configure remote alarm. (Mail, SMS, ESPA, Pager, City Call, Voice)
Configure priorities.
Configure scheduler priorities.


Malm Manager processes all alarms in parallel. This means that in the Malm Configurator each alarm type must have it own interface. It is not possible to respond to, for example, a pager and a Citycall via the same interface. This means that two modems are required.


In Version 1.5 and higher the following improvements are included:

1. New notification types:
   - ESPA external systems, e.g. staff paging systems, telephone exchanges, etc.
   - Voice: Telephone voice response with the possibility of acknowledging the alarm immediately
2. Time program for activating/deactivating triggering by alarm priority and notification type.
3. Adjustable, time-delayed triggering depending on alarm priority and notification type
4. (Return) acknowledgment (alarms recognised) by telephone with code and entry of the caller details in the Alarm History
5. Alarm send repetitions after an adjustable delay, until the alarm is acknowledged/ recognised
6. When an alarm has been acknowledged:
   Option for each notification type as to whether a confirmation message should be sent to the relevant participants, to whome the alarm was previously sent ("Alarm recognised by xxx", xxx=user name)


With email notifications, it is important to be aware that if too many emails are sent

within a certain period of time, mail servers can jam and will no longer be able to send out notifications. Please note the information given by the email provider. If several users in a Malm group are sent notifications, a new email will be sent for each user.

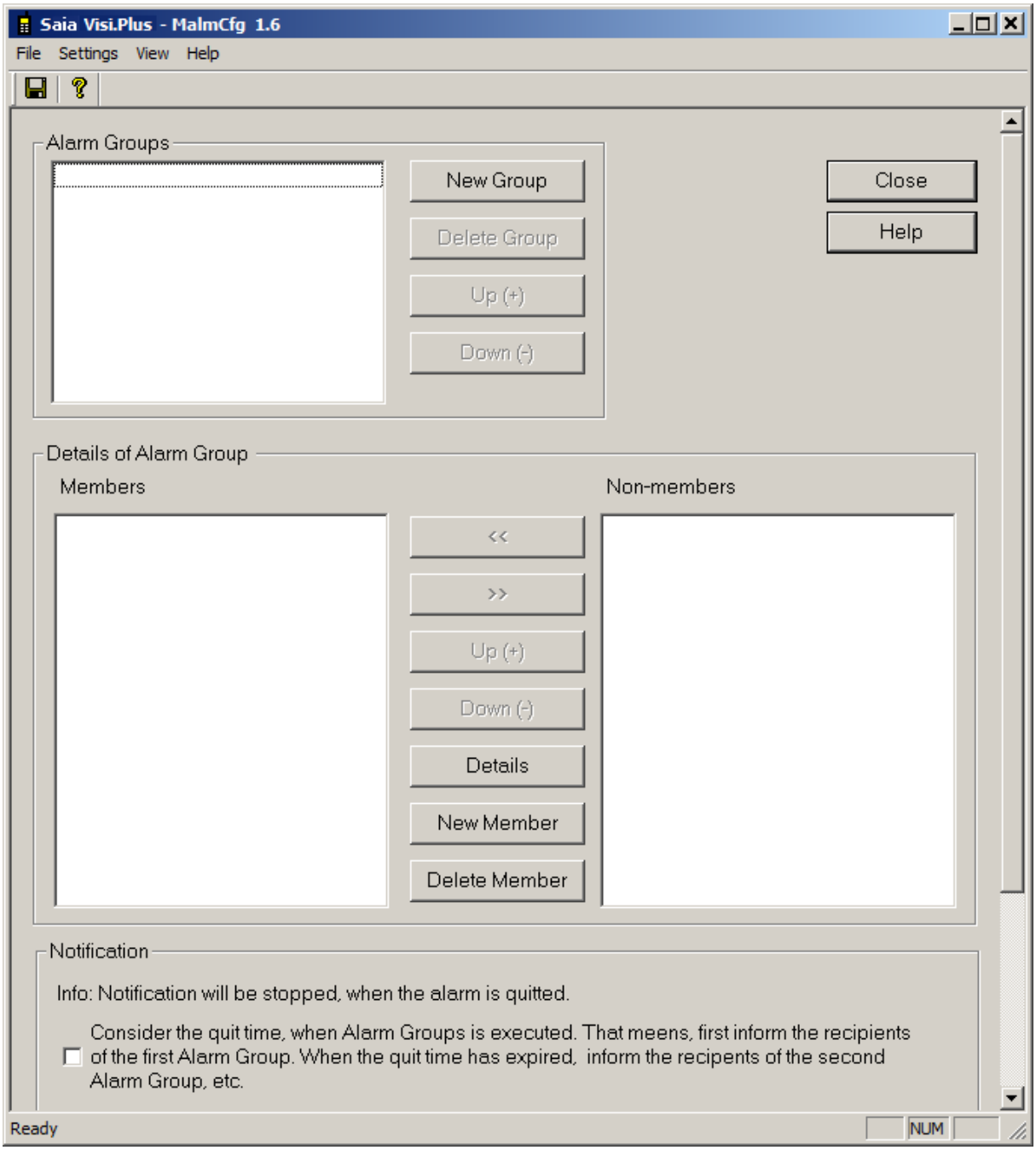
10.10.1 Starting the MAlm Configurator

In the **PET** call up the menu command "**Options**" > "**Remote Alarms**". Or proceed conventionally and start MalmCfg.exe from the standard installation directory **c:/Visi.Plus/bin** with a double click.

10.10.2 The MAlm Configurator main window

In the main window of MalmCfg.exe the following settings can be made:

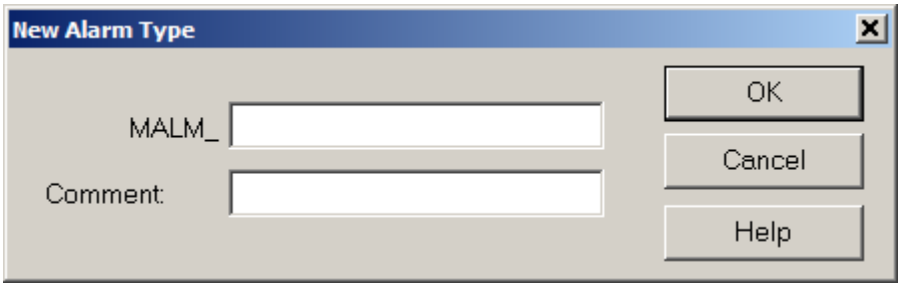
- Definition of alarm groups
- Definition of users to whom an alarm will be sent
- Assigning users to alarm groups



10.10.3 Creating remote alarm groups

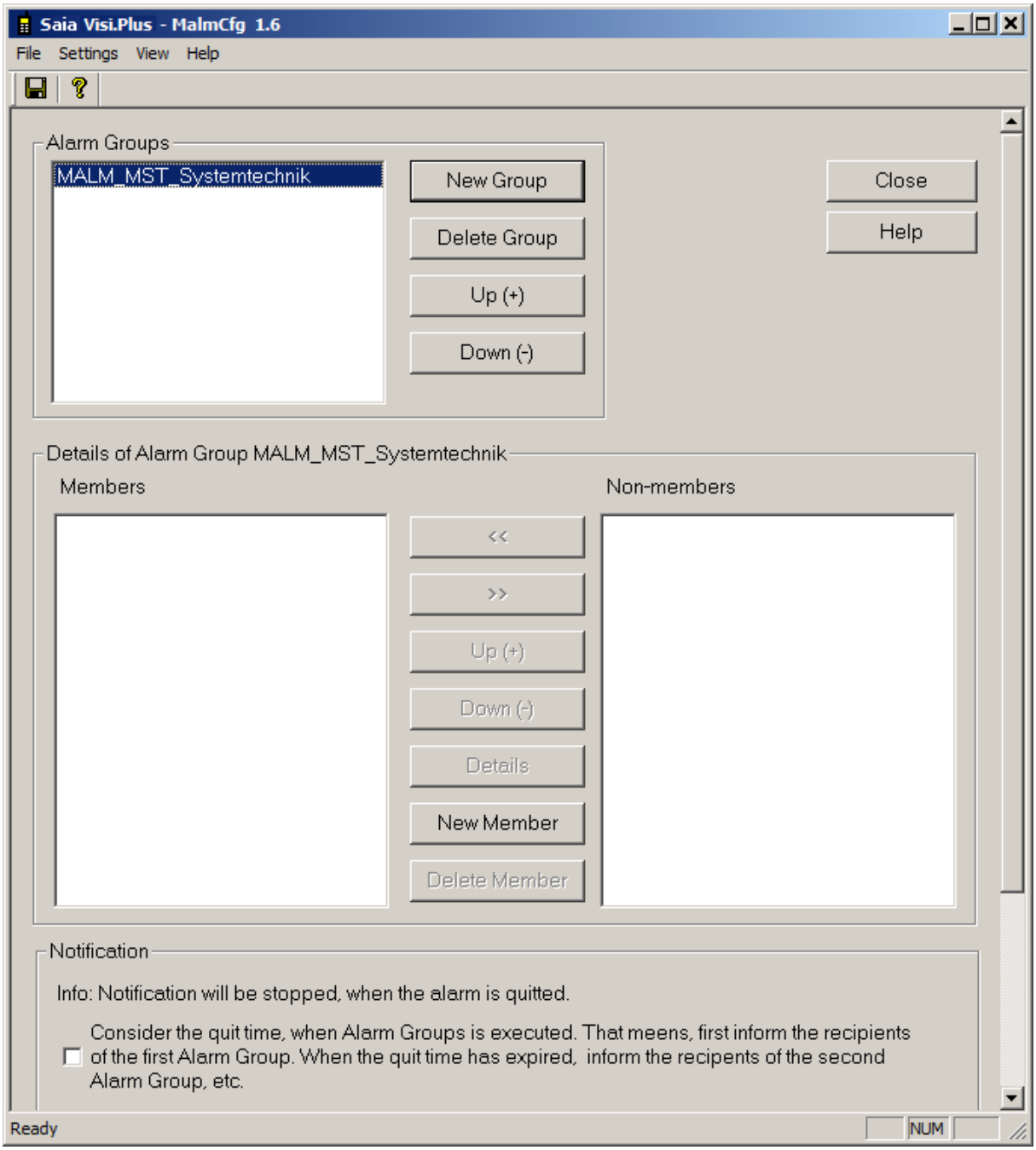
The first step is to create an alarm group. The alarm group is assigned to the desired data point later in the PET on creation of the remote alarm. In this way, **MalmMng** knows which alarm group/person must be informed on occurrence of an alarm. The number of alarm groups is unlimited.

In order to create an alarm group, click on the **<New Group>** button.



The preset name "**MALM_**" next to the input field in the above example is completed with text of your chosing - in this example with "**MST_Systemtechnik_AG**". This produces the full name of the alarm group. Confirm with **<OK>**.

The newly created alarm group now appears in the Alarm Group window:



The created alarm group was prefixed with the text **MALM_**. The text **MALM_** serves for identification in the DMS.

An alarm group can be deleted by selecting the group in the **Alarm Groups** window and then pressing the button
<Delete Group>.

If the checkbox here is marked, the respective members of this alarm **group are notified in the indicated sequence**. The notifications are stopped as soon as the alarm is acknowledged.

Cyclical notification is only maintained for such groups that have alarms of the same

type of notification, e.g. only SMS or only voice, etc. If different alarm types appear in a group, these members are notified at the same time.

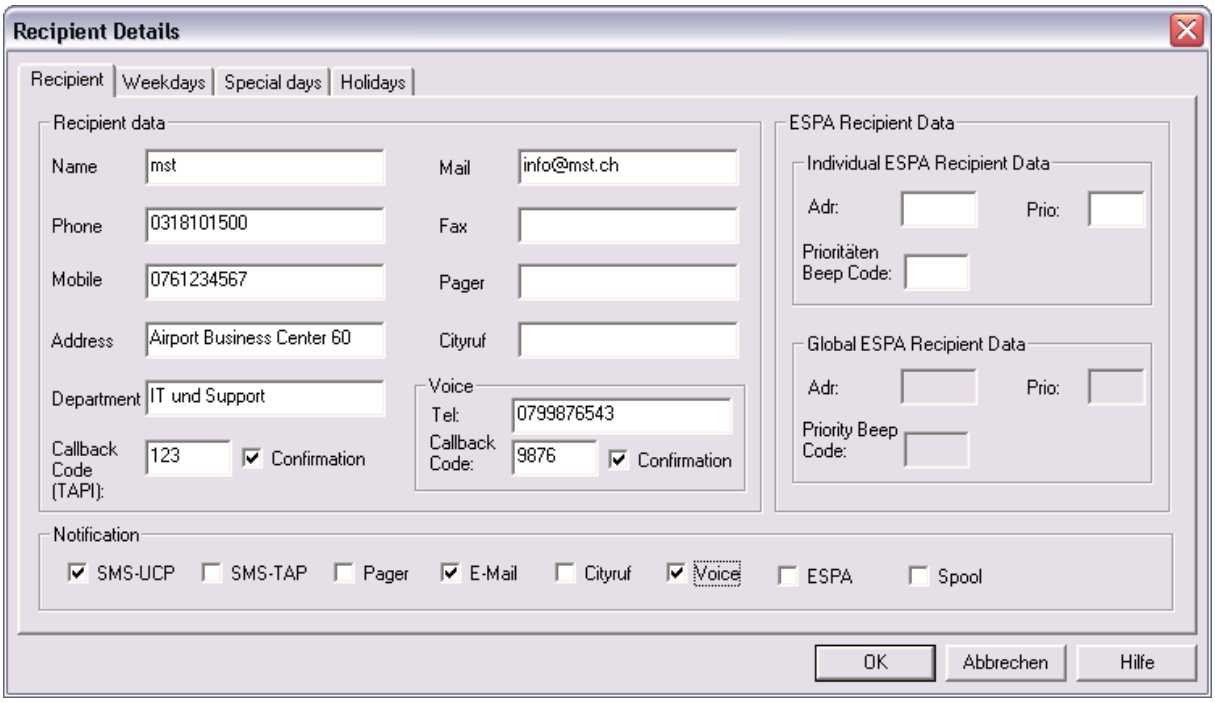
The **time for acknowledgment (in minutes)**, indicates how much time should be waited before the next user in the group is notified if the the first user has not acknowledged an alarm in the prescribed time.

The number of cycles indicates how many times the group members are cyclically notified until the alarm is acknowledged.

10.10.4 Creating a remote alarm user

In order to create a new user, click the button **<New User>**. The following dialog window is displayed:

Recipient tab:



The user's personal data is to be entered here. Under **Messaging By** the desired service for the user upon occurrence of a remote alarm can be selected. Multiple messaging methods can also be selected simultaneously.

The following names are reserved and may not be used as recipient names:

- "Activated"
- "Ack"
- "AckNo"
- "Sequencing"

Recipient data

TAPI call-back code see the next chapter

Voice
Tel: Telephone number of the user who is called by the voice output in the event of an alarm.
Call-back code see the next chapter

ESPA recipient data see also Global ESPA recipient data
Individual (personal) ESPA recipient data
Address (Call address): 0000..9999
Priority**Beep Code**: 0..9
Priority 0..9
The personal ESPA recipient data has higher priority than the global data. That means that the personal ESPA settings are sent, if they are defined, otherwise the global settings are sent. However, the information from the ESPA formats has the highest priority (see ESPA format definitions): if an ESPA format is present in the alarm format of the current alarm, then it is sent.

Global ESPA recipient data
This is only shown here for overview purposes, as a reference. Entry is only possible in Global ESPA recipient data configuration.

Messaging By...
The following services (messaging types) are available:

- **SMS-UCP** Mobile protocol for Switzerland (Germany is not supported by all providers under certain circumstances
- **SMS-TAP** Mobile protocol for Austria
- **Pager**
- **E-mail**
- **Cityruf** (Paging service in Germany)
- **Voice**: Voice output of an alarm audio file via telephone with the possibility for alarm acknowledgment (after user-code verification) using the telephone keys.
- **ESPA**: Alarms via an external system e.g. light call system, fire alarm system, staff paging system and telephone exchange, etc.
- **Spool**: Each alarm is spooled to the file system as a separate ASCII file. The spool file that is generated is written to a specific spool directory. A suitable server spool program must check the spool directory at regular intervals and forward any messages that are present.

ESPA, voice and spool are only available in Version 1.5 or higher.

UCP and TAP are different SMS protocols. If an SMS is sent through a Swisscom exchange, the UCP protocol must be used. The SMS message centre operator responsible in your particular location can be asked as to which SMS protocol to use.

*The pager number is entered **without** a prefix, because the pager prefix is processed directly in the pager exchange centre. If the pager number is entered with a prefix, the centre*

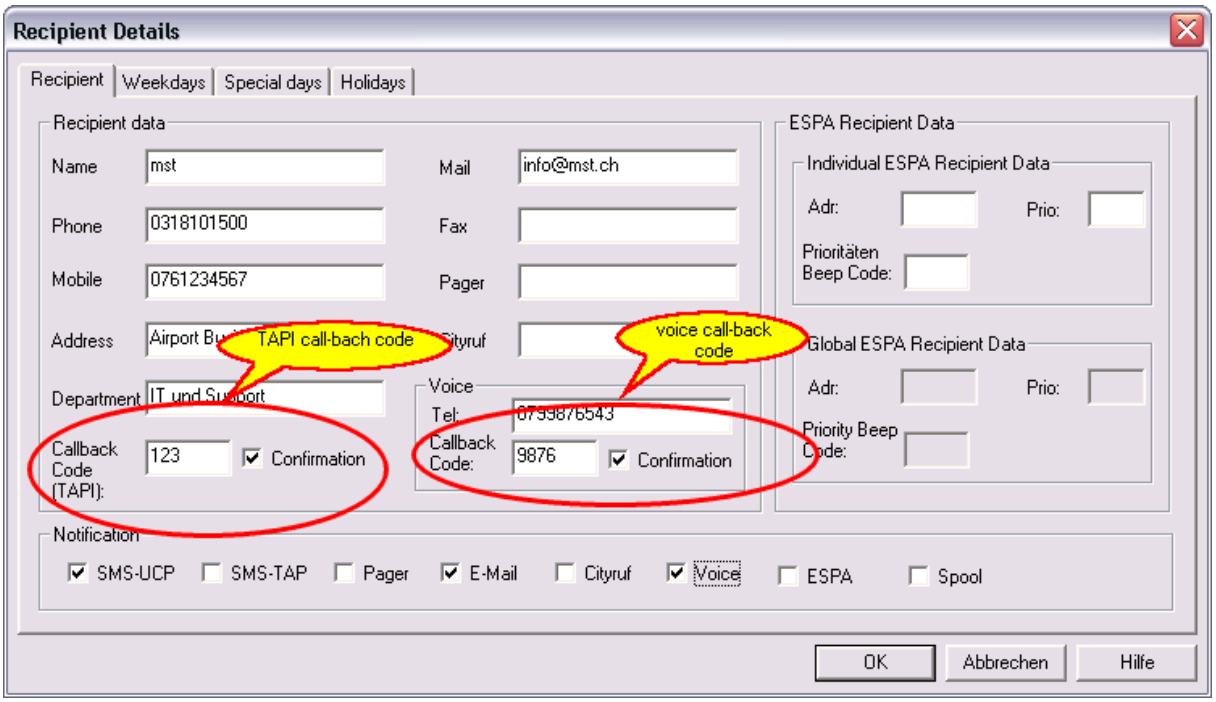
always reports Error 106 (invalid number).
The mobile or fax number may not be separated by a blank space. Otherwise, in the case of an alarm via SMS, the telephone exchange will not be able to process the mobile number correctly.

10.10.4.1 Call-back code

See also Alarm acknowledgment.

The two "**Call-back codes**" for entries (2 to 4-digits recommended) in the **Recipient Details** dialog are used for user verification for return acknowledgment of the alarms by telephone. Remote acknowledgement of an alarm is performed as follows:

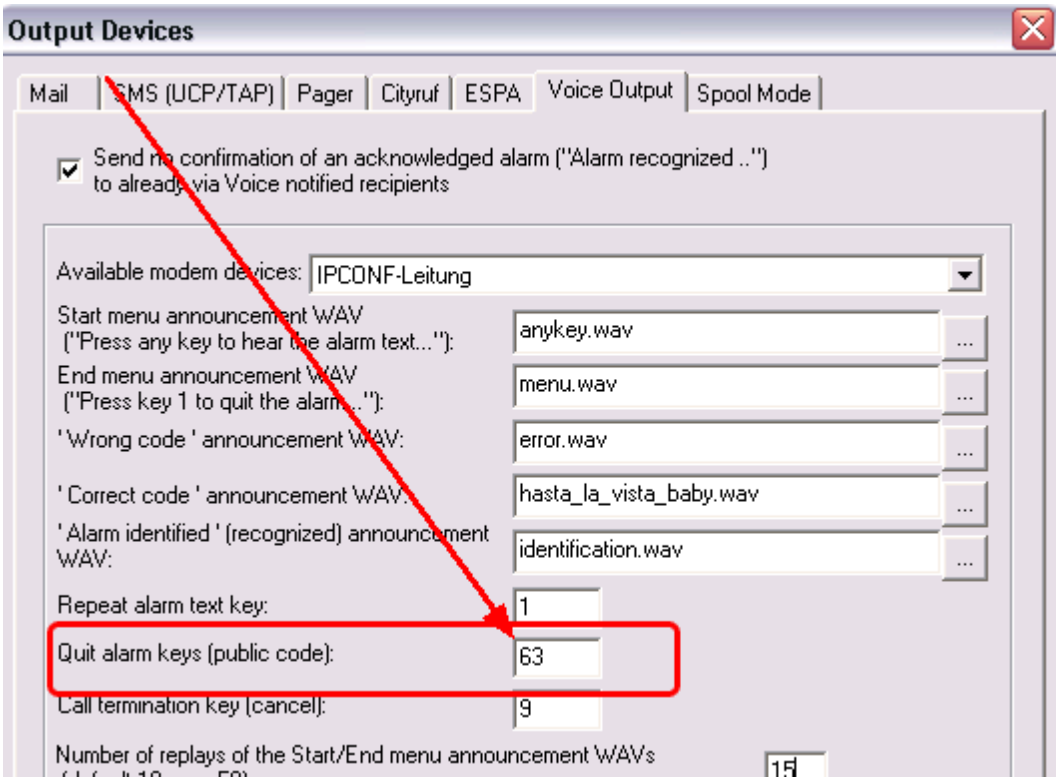
- directly from the voice telephone (voice output message) via telephone keys: after entry of the **voice call-back code** the alarm is acknowledged
- indirectly by telephone call via TAPI driver: after entry of the **TAPI call-back code**, **all** currently pending alarms are acknowledged (and therefore not processed further)



Confirmation
If selected, it becomes mandatory to enter the code on the telephone. If not, just the acknowledgment key (by default '#') suffices for the TAPI driver; for voice, the global code (see below).

The two call-back codes may be identical, however independent from one another.

Global VOICE call-back code
For VOICE alarms, it is still possible to define a global call-back code (Settings menu, General Settings, VOICE Alarm tab). A "Public key" is determined for all users that have not defined a private, personal call-back code in their Recipient Details settings.



The personal call-back code has a higher priority than the global one.

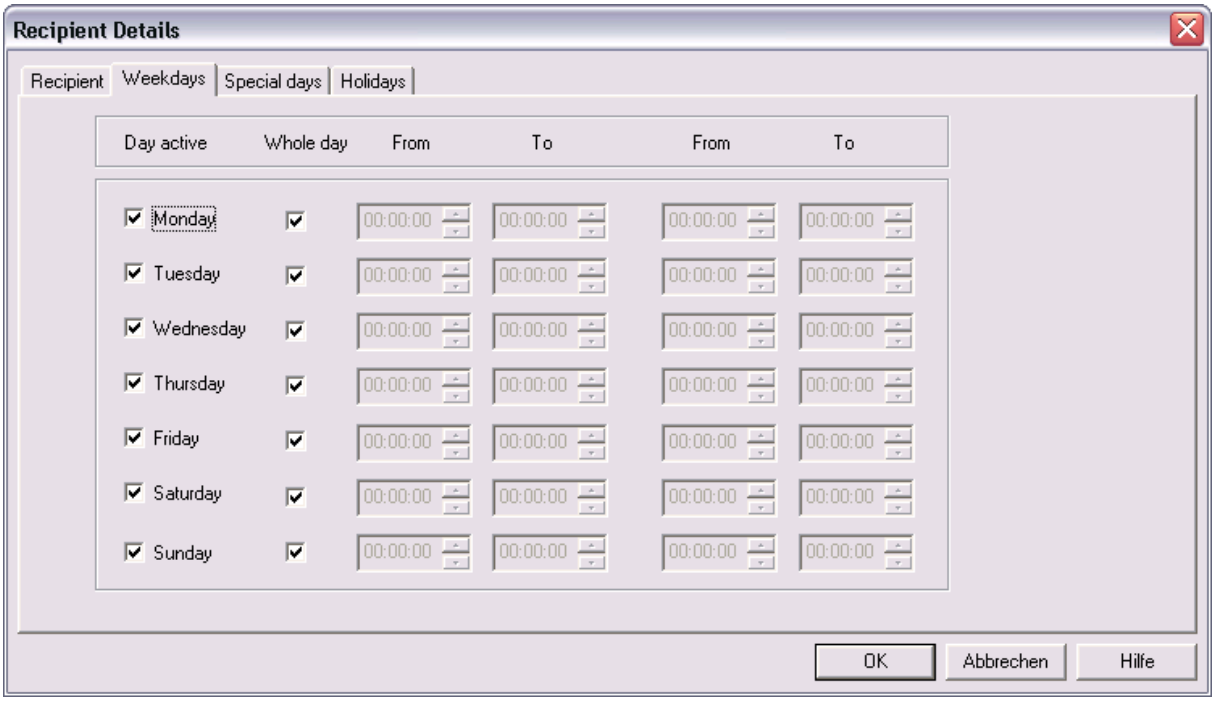
10.10.4.2 Time Program for Users

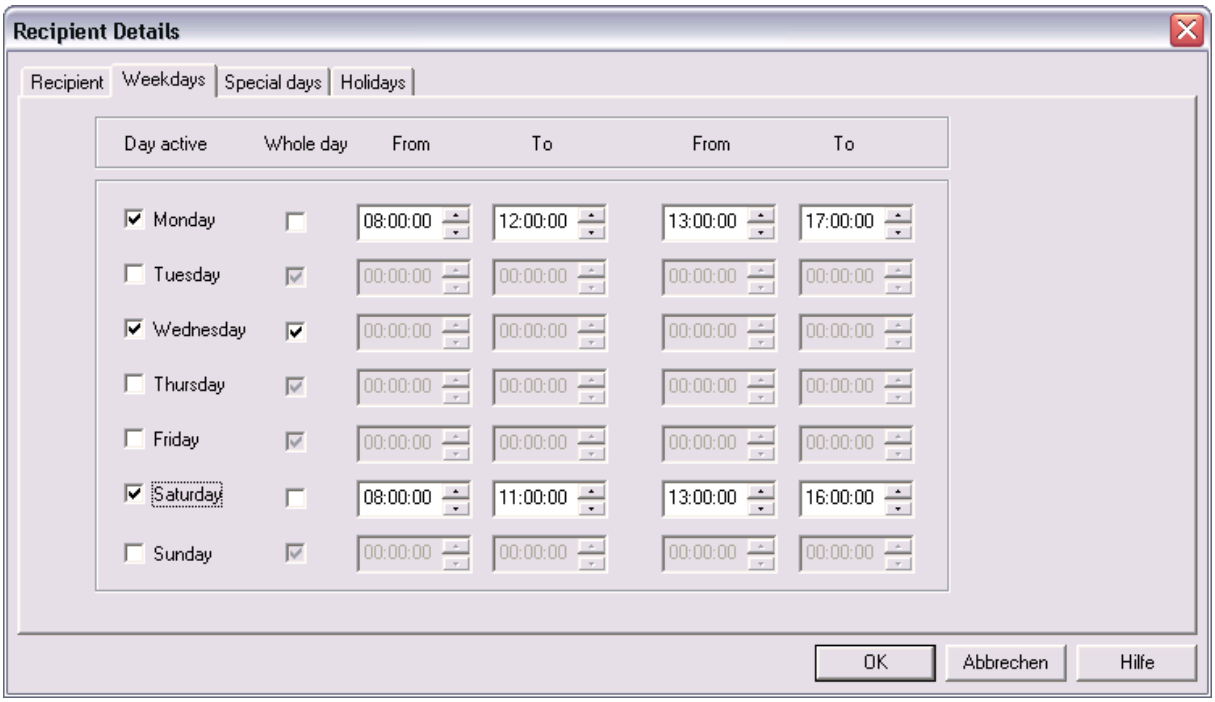
(See also Time Program for Priorities)

A Time Program enables the transmission of alarms to be activated/deactivated for each individual user.

The **Days of the Week** tab:

Allows you to define on which days of the week and at what times the user should be notified. ***No*** setting is predefined for days of the week; you must define this at your own discretion and according to your own requirements.





If the checkboxes next to the corresponding days of the week are marked, the user is notified on these days. In the preceding image the user is **not** notified on Tuesday, Friday or Sunday.

If the checkbox for "**Entire Day**" is marked on each day, the user can be notified at any time (the respective day of the week must also be selected).

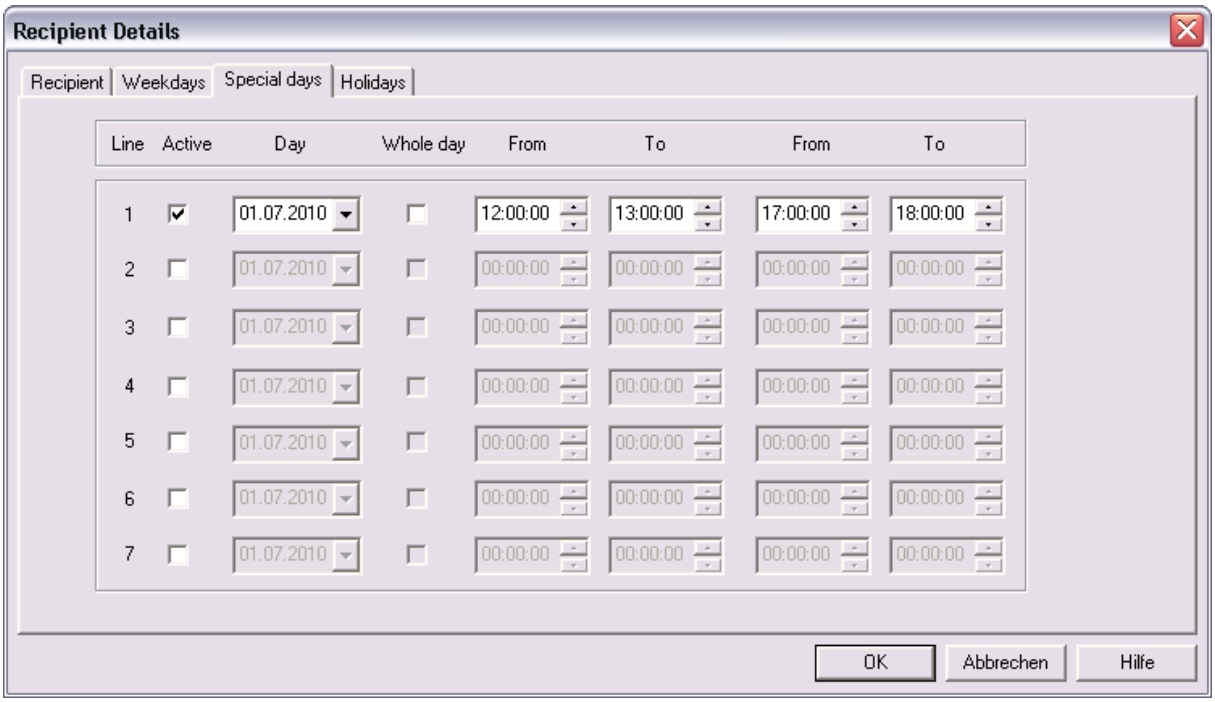
In the preceding image, the user is notified at the defined times on Monday and Saturday, as well as at all times on Wednesday and Thursday.

*By default, the **MAlm** Configurator sets **all** checkboxes to **Entire Day** on creation of a new user. Here you must configure which on days and at what times the user is active (able to receive messages).*

It must be ensured that one or more users are available at all times. If a remote alarm occurs at a time when no user is available, the remote alarm will not be reported!
A second user, who is available during the absence of the first defined user (Tuesday, Friday and Sunday), must be defined.

The **Special Days** tab:

Allows you to define special days where the user has special times of availability. In the "**Day**" column, the respective date and the corresponding times are entered in the time fields "**From/ Until**".
No setting is predefined for special days; you must define this at your own discretion and according to your own requirements.

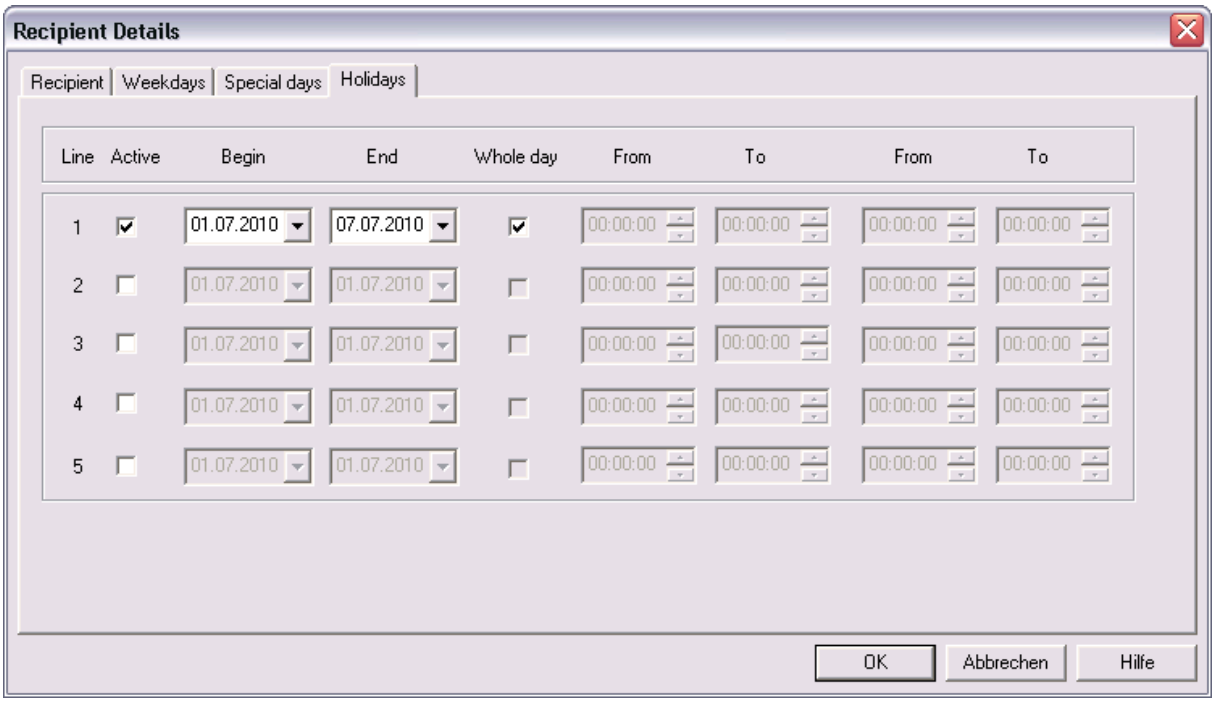


*The special days that are entered have a higher priority than the data entered in the **Days of the Week** tab!*

*For example, if 22.05.2008 is a Thursday, the times indicated in the "Recipient Details" screen are superfluous. The user is notified at the times indicated here on Thursday, 22.05.2008. If the checkbox in the Special Days tab for **Entire Day** is marked, the recipient is notified for the entire day on 22.05.2008.*

The **Holidays** tab:

Here the user's holidays can be entered. The settings defined in the **Days of the Week** and **Special Days** tabs are not taken into consideration during the indicated holiday periods. The data in the **Holidays** tab, in turn, has a higher priority than that in the tabs **Days of the Week** and **Special Days.**



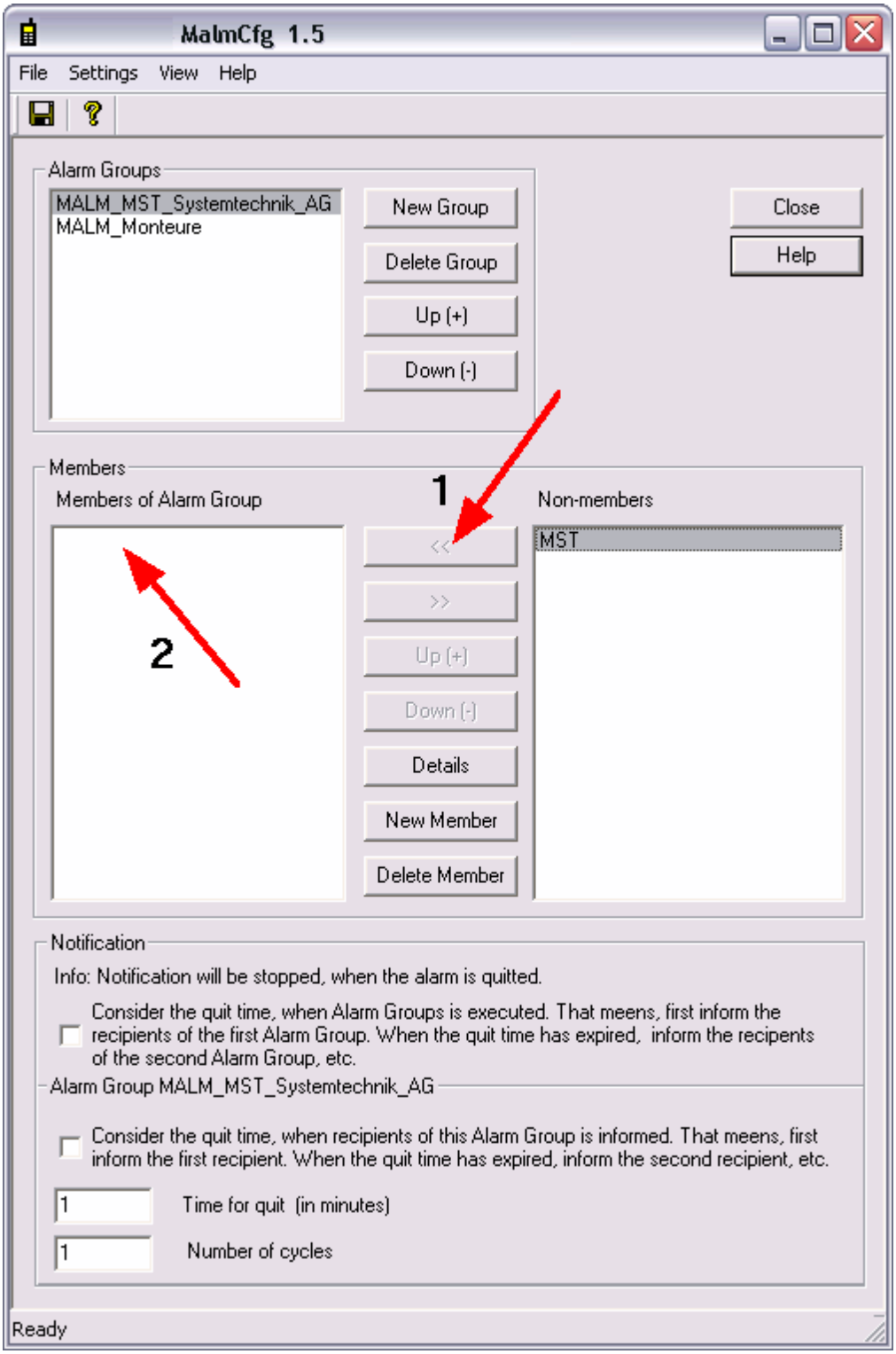
In the preceding image the user is not notified in the indicated holiday period from 19.06.2008 to 23.06.2008, regardless of what is entered in the **Days of the Week** and **Special Days** tabs.

*The data entered in the **Holidays** tab has a higher priority than the data in the **Special Days** tab. The data in the **Special Days** tab, in turn, has a higher priority than that in the **Days of the Week** tab.*

10.10.5 Assigning users to alarm groups

In order to assign users to an alarm group, the desired group must first be marked in the window **Alarm Groups**.

Select the desired user in the field "**Defined User**" and click on the button << in order to assign users to alarm groups or remove users from the alarm group with >>.



In the preceding image the user **mst** has become a member of the alarm group "MALM_MST_Systemtechnik AG". By pressing the **<< key** the selected user is moved to Area 2.

*User data can only be edited in the right field "**Defined User**" with the <**Details**> button. User data cannot be changed or deleted in the display field "**Alarm Group***

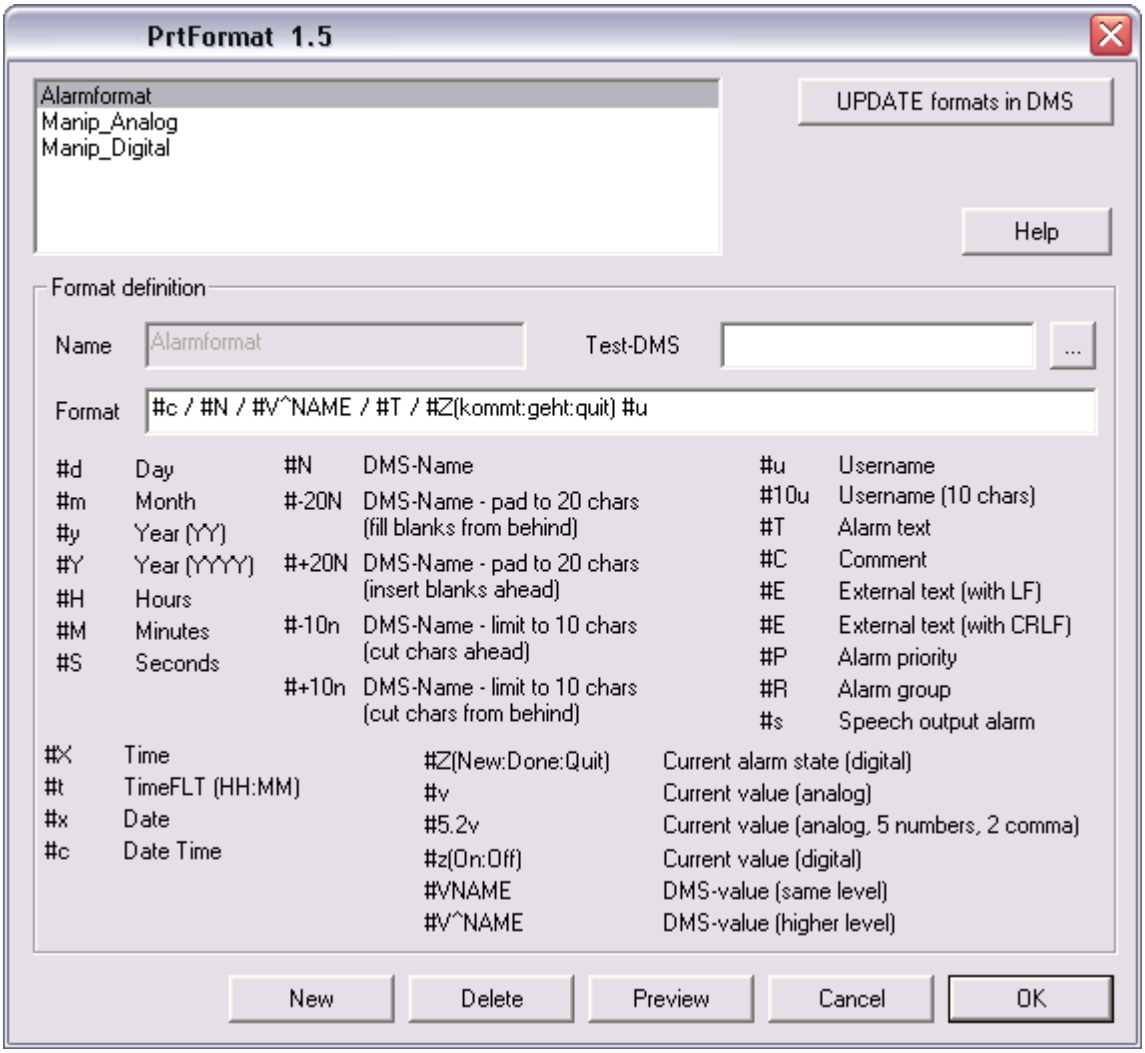
***Members**". To edit a user, they can also be removed from the alarm group with* >> *and reassigned to the alarm group with* << *after the change.*

10.10.6 Creating the remote alarm format

Remote alarms are assigned to data points in the PET. To do so, log formats must be defined first.

For this purpose, in the PET module select the menu item "**Options**" **>** "**Log and format definitions**".
Click on the **<Define Formats>** button in the displayed selection.
The window shown in the following image appears.
With this window, the format definition for the remote alarms can be composed, among other things (for instructions on usage, see the chapter on auxiliary programs under **prtformat**).

If no formats are defined, the default Visi.Plus format template is used. Of course, existing formats can also be adopted.



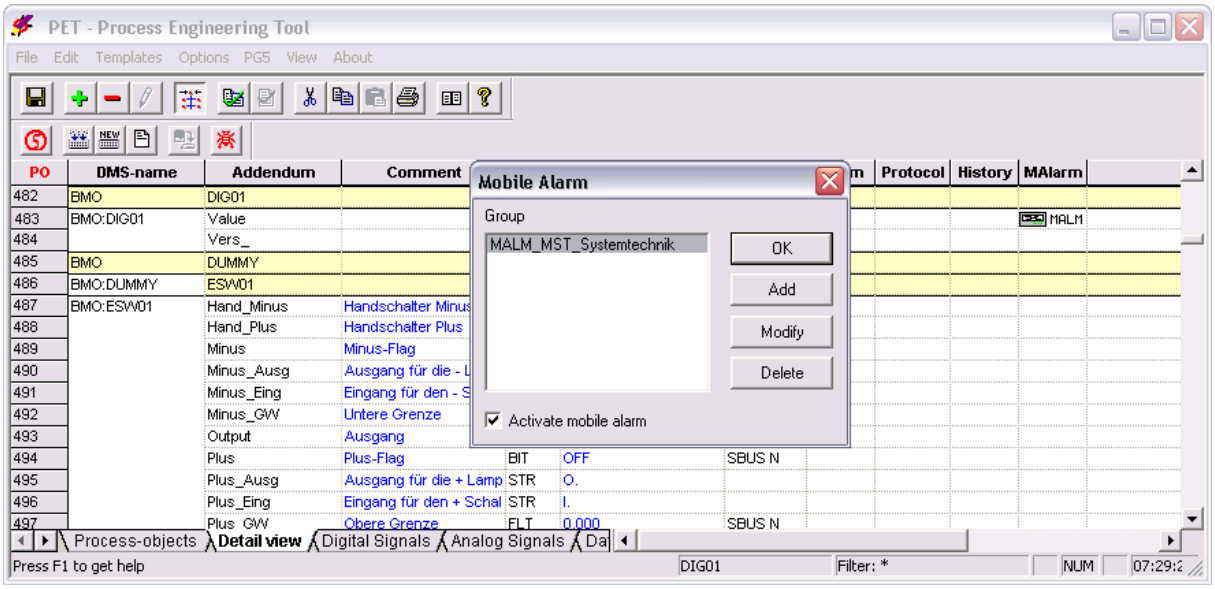
More information about the creation of formats is described in this manual in the chapter on auxiliary programs under **prtformat**.

*Special characters such as the forward or backward slash (/ or \) and square brackets ([or]) are **not** permitted in the format. The SMS and pager centres cannot process special characters. Non-permitted characters result in an error message, e.g. with the warning "Syntax error".*

10.10.7 Inserting remote alarms in the PET

After a format definition has been created, a remote alarm can be inserted in the **PET** module in the column **MAlarm** at any data point.

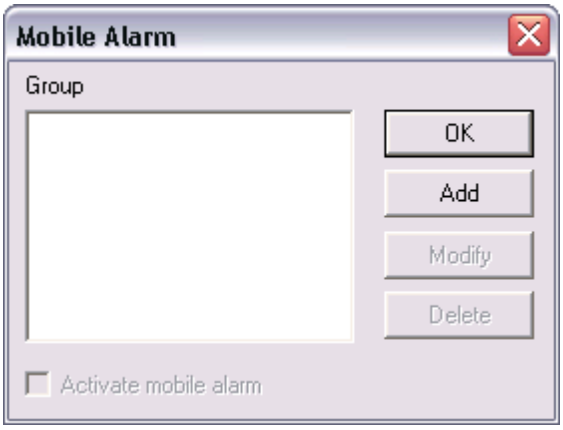
If the **MAlarm** column is not visible, it must first be activated through the **View menu**.



*It does **not make sense** to insert a remote alarm directly into a system object. We recommend inserting the remote alarm directly into the template object.*

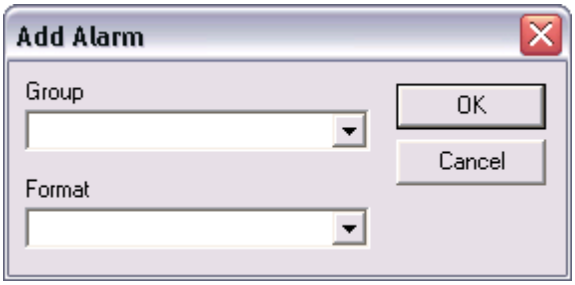
The PET must be open for the following steps and Detail View must be selected.

Next click on the desired data point in the column **MAlarm**. The Remote Alarms dialog window opens.



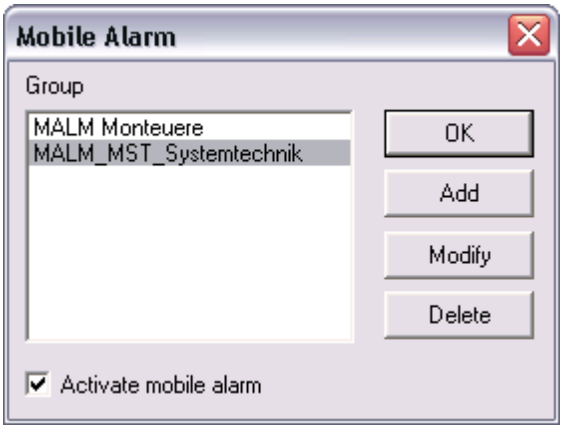
In order to insert defined alarm groups into the MAlm Configurator, click the **<Add>** button.

By clicking on the respective dropdown menu the alarm group and the alarm format can be selected.



Adopt the settings by clicking the **<OK>** button.

Multiple alarm groups can be assigned to a data point.



By clicking on a group's **Remote Alarms Active** checkbox, it can be switched on or off.

If changes are made to a system and no remote alarms are desired, it is possible to deactivate all remote alarms.
*In order to do so, set the value to logic 0 (ZERO) in the **DMS** in the **Object View** under "**System** > **MAlmMng** > **General** > **Enable**".*

*If the initialisation of the data point "**System** > **MAlmMng** > **General** > **Enable**" is defined for a button, the remote alarms can be operated from the visual display. More information about initialisations can be read in the chapter on operating programs under **Graphical Editor**.*

10.10.8 Acknowledging alarms

See also Call-back code.

Remote acknowledgement of an alarm can take place as follows:

- directly from the voice telephone (voice output message) via telephone keys: after entry of the **voice call-back code** the alarm is acknowledged
- indirectly by telephone call via **TAPI driver**: after entry of the **TAPI call-back code**, **all** currently pending alarms are acknowledged (and therefore not processed further): When a call is placed to the control system (received by the TAPI driver), **all** relevant, pending and sent alarm(s) can be acknowledged by entering a valid code.
  Alarms which were not set are not acknowledged and are obviously handled in accordance with the applicable rules! (remain pending and are sent as soon as the Time Program becomes active again.)

If the respective alarm(s) is (are) acknowledged, the following actions result:

1. The user name (according to the code) is entered in the alarm history for the purpose of traceability.
2. If a notification type has a respective setting (see below), a confirmation message - "Alarms recognised by MALM-xxx" (xxx=UserName:AlarmType-Tel.No.) - is sent to the participants that the alarm(s) was (were) previously sent to. The notification types for the confirmation message are identical to those of the respective alarms. This means, for example, that if an alarm was sent by ESPA (but not yet by SMS), the acknowledgment confirmation is also only output to ESPA. If, however, the alarm was sent by ESPA and also by SMS and pager, the acknowledgement confirmation is also output in all 3 message types.
   If acknowledgement occurs directly on the control system (not by a telephone call-back), a confirmation message is also output in the same way with the name of the "acknowledger".

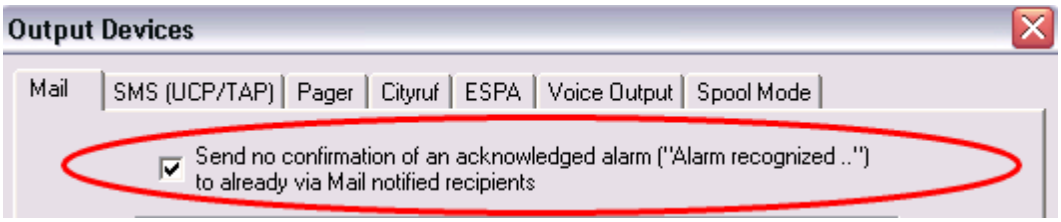
 Alarms can, of course, also always be acknowledged through **AlarmView** (local or remote) of the **control system** (see AlarmViewer)

10.10.9 Confirmation message of acknowledged alarm

Every acknowledged alarm can trigger a confirmation message. However, this is also sent to any participants that were previously notified of this alarm.

Every alarm type (e-mail, SMS, pager, voice, etc.) has its own option as to whether a confirmation message should be sent to the participants that the alarm(s) was (were) are sent to.

This **option, "Do not send a confirmation message of acknowledged alarm ..."** is advantageous if the frequent call-backs of acknowledged alarms would become disruptive, because they would overlap real alarm calls, such as with the voice telephone alarm type:



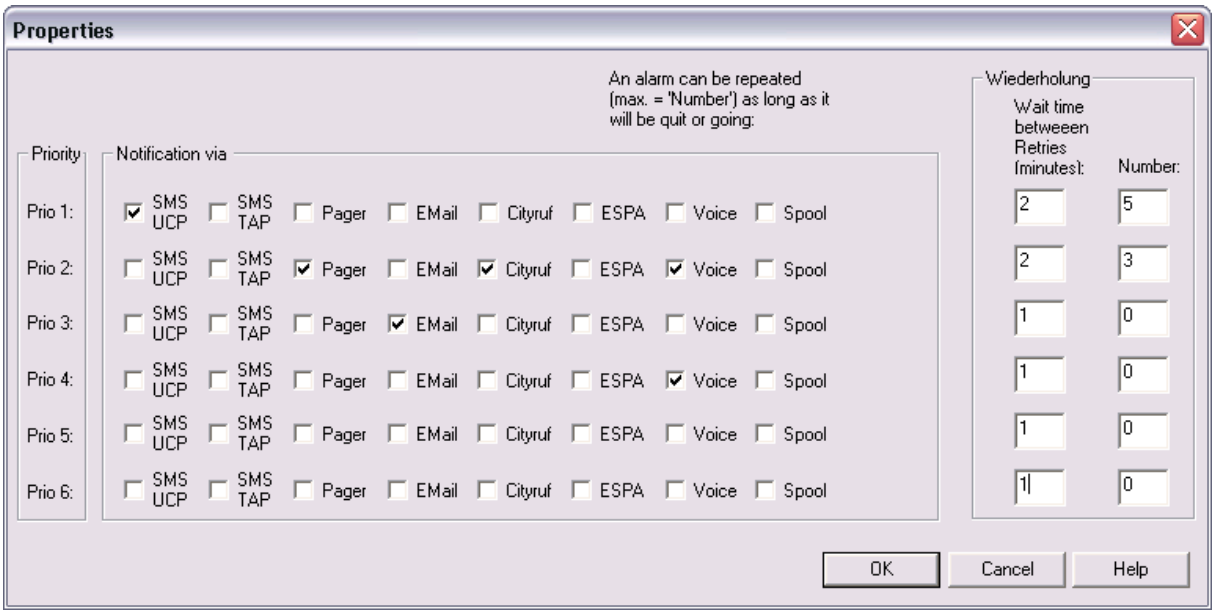
The confirmation message has the following format:
"Alarms recognised by MALM-xxx:aaa-Tel.No."
wherein xxx=user name of the "acknowledging party", aaa=alarm type (SMS, VOICE, etc.), Tel.No.= number the acknowledgment came from.

Example: `"Alarms recognised :MALM-C.Mueller:VOICE-00798295022"`

10.10.1 Configuring priorities

Through the menu **Configuration** > **General Settings** in the **Priorities** tab, the configuration of priorities for the notification methods can be carried out.

If the settings are made as shown in the image, Prio1 alarms are sent to SMS and Prio2 alarms to pager, while Prio3 alarms are sent to e-mail. Prio3 to 5 alarms are also sent to Voice media (tel. voice output).



Repetition

A repetition factor and a wait time (in minutes) can be defined for each priority. Then the sent alarm is resent if it is still pending and has not been acknowledged in the meantime. Time=0 means no repetition.

*By default, **no** priorities are activated.*

Visi.Plus supports only Prio 1 to Prio 5 as standard.
Higher priorities are not allowed for and make the entire concept with maintenance logs impossible.
No Prio higher than 5 may be used!
Please note the concept of the maintenance alarms.

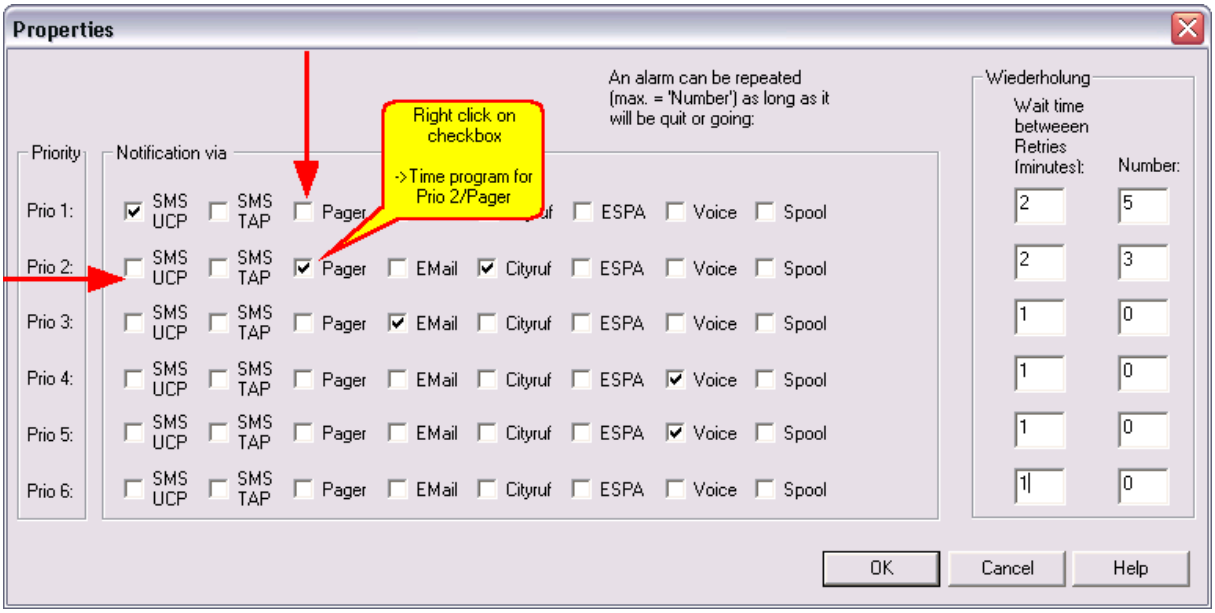
10.10.10. Time Program for Priorities

(See also Time Program for Users)

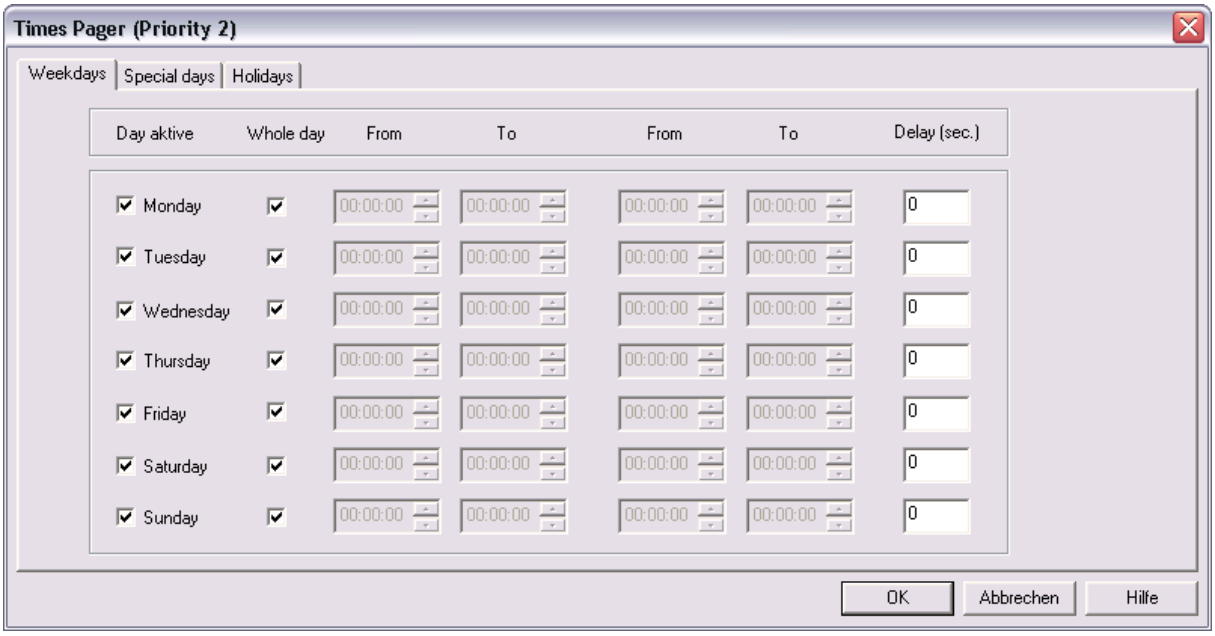
The Priorities Time Program allows you to make the following settings:

- The transmission of alarms can be activated/deactivated for each individual priority and notification type
- Adjustable, time-delayed transmission depending on alarm priority and notification type

The Time Program in which details/settings can be specified for each notification type and for each priority, can be activated with a button click on the corresponding output device or a right mouse click on the respective priority line/notification type column:



The Time Program for **Pager/Prio 2**, on the **Days of the Week** tab appears:

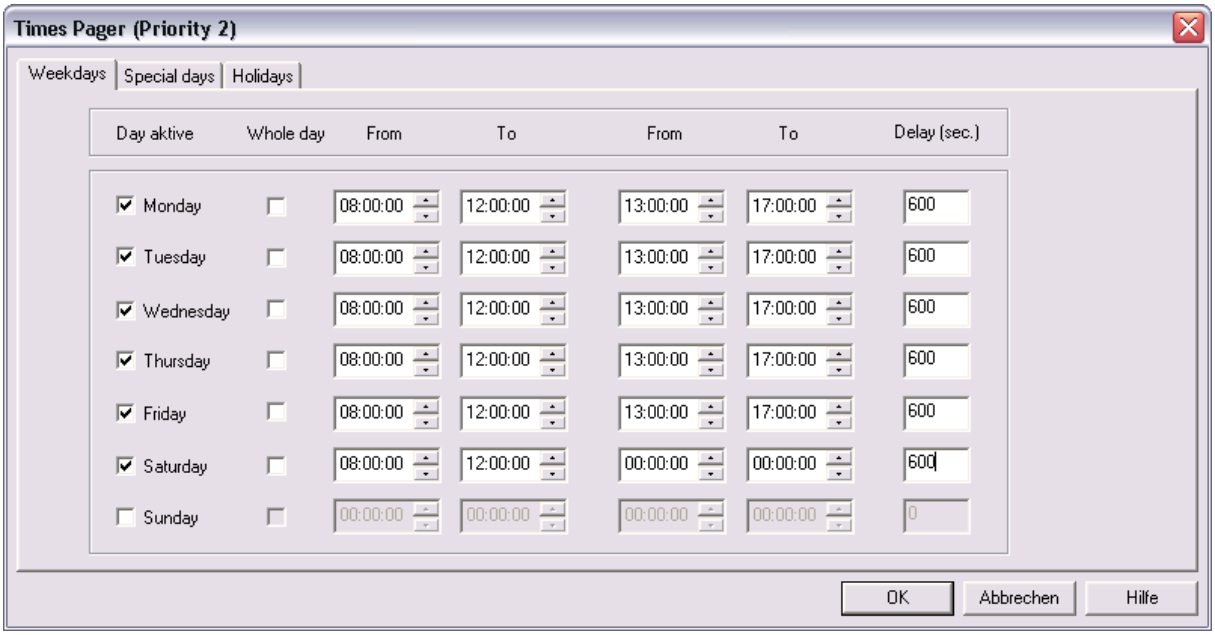


It allows you to define on which days of the week and at what times the alarm should be sent. ***No*** setting is predefined for days of the week; you must define this at your own discretion and according to your own requirements.

The functions are, in principle, the same as for the recipients (cf. Time Program for Users); alarms are only transmitted on the days/times/data that are selected here. Alarms that were not transmitted, on the basis of the day/time/date, are sent as soon as the Time Program is reactivated. For example, you can ensure that Prio 3 alarms are sent on the following work day, e.g. at 8:00, rather than in the middle of the night.

An adjustable **delay time** can also be included (separately for each day, also in the Special Days and Holidays tabs). That means that, on the occurrence of an alarm, the message can be sent to different notification types graded according to the delay time.

Example:
An alarm Prio 1 is sent after 0 sec. (immediately) to ESPA, after 120 sec. to a pager, after 5 min. (300 sec) via SMS (insofar as it was not acknowledged in the meantime or the alarm is no longer pending). A different alarm Prio 2 is first sent after 30 sec. to ESPA and after 600 sec. to a pager. This allows enough time, for instance, for acknowledgement on the control system before the alarm is sent "across the world".



*By default, the **MAlm** Configurator sets **all** checkboxes to **Entire Day** on creation of a new user. Here you must configure on which days and at what times the alarm is active (a message is sent).*

The **Special Days** tab: (see the same procedure in Time Program for Users)
Allows you to define special days where the alarm has special times of availability. In the "**Day**" column, the respective date and the corresponding times are entered in the time fields "**From/Until**".
No setting is predefined for special days; you must define this at your own discretion and according to your own requirements.

*The special days that are entered have a higher priority than the data entered in the **Days of the Week** tab!*

The **Holidays** tab: (see the same procedure in Time Program for Users.)
Here the user's holidays can be entered. The settings defined in the **Days of the Week** and **Special Days** tabs are not taken into consideration during the indicated holiday periods. The data in the **Holidays** tab, in turn, has a higher priority than that in the tabs **Days of the Week** and **Special Days.**

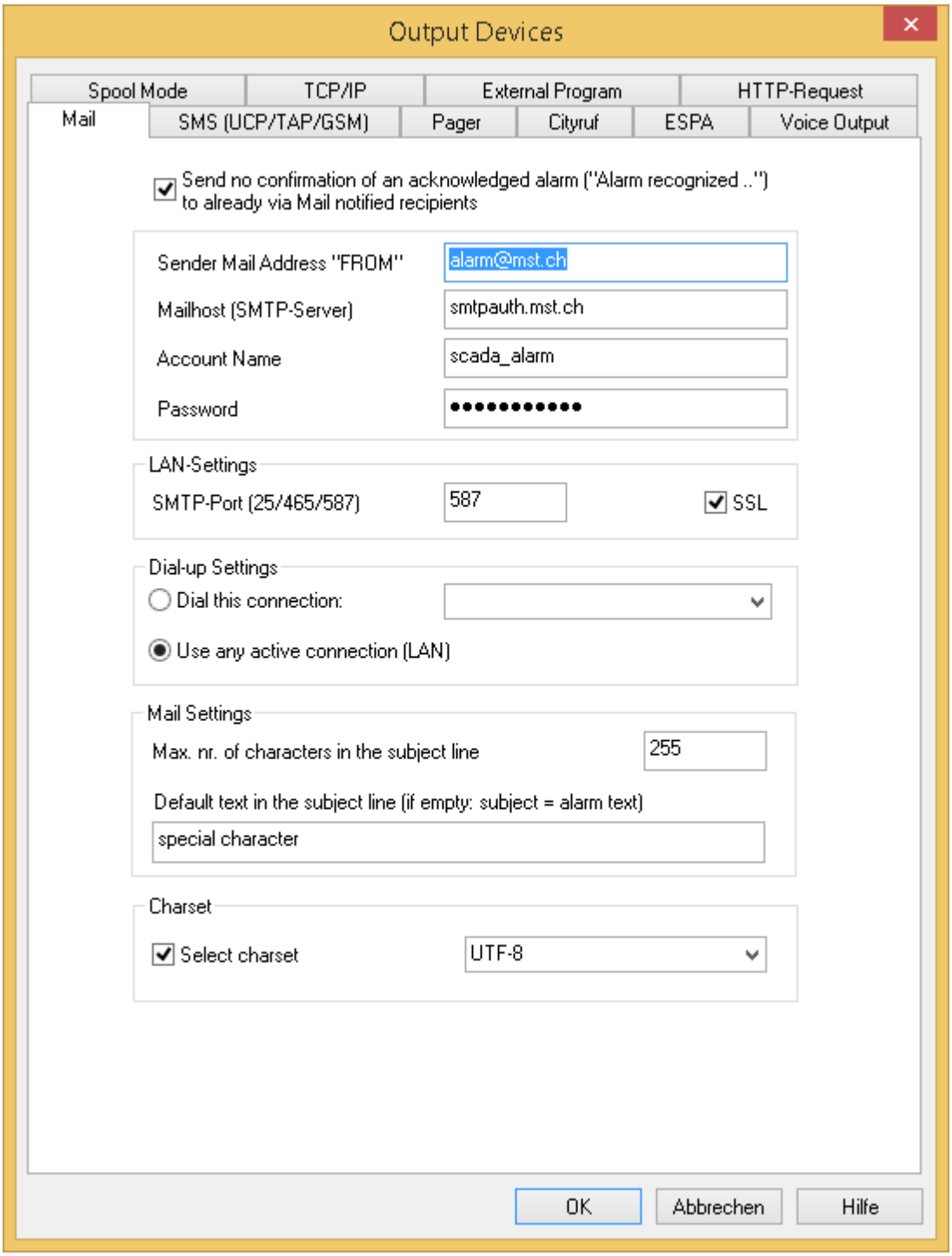
*The data entered in the **Holidays** tab has a higher priority than the data in the **Special Days** tab. The data in the **Special Days** tab, in turn, has a higher priority than that in the **Days of the Week** tab.*

If a message is triggered at a time when priority is not active, the debug window will display: "Device is sleeping".

10.10.1 MAlm configuration via e-mail

In order to set off a remote alarm via e-mail, the mail account must be configured in **MalmCfg**.

Configuration of the mail account can be carried out through the menu **Settings** > **Output Devices,** in the **Mail** tab.



No confirmation message of acknowledged alarm
This option is checked by default, meaning that no confirmation message of the acknowledged alarm is sent.

Sender e-mail Address
Holds the sender e-mail address.

Mail Host (SMTP Server)
Here the mail server for outgoing post (SMTP) is entered. If this information is unknown, check with your internet service provider.

Account Name and Password
An e-mail account user and password can be indicated.

SMTP port (25)
Defines the mail server port (SMTP). This is normally port 25. Check with the internet provider if this information is unknown.
If the SSL checkbox is activated, the mails are sent via SSL. Otherwise they are sent via TLS. Please pay attention to the information from the internet provider.

Remote Transmission Settings
In this section the connection settings are configured.

Coding
Here it is possible to use an appropriate coding if an e-mail is required that uses non-West-European characters.

*The information in the preceding image **cannot** be used as displayed. The SMTP server and its port depend on the internet service provider. The ending of the **sender e-mail address** (e.g. ...@mst.ch) must be registered with the indicated SMTP server.*

10.10.11. Configuring the layout of the e-mail

By default, the subject line of MAlmMng is left blank. The content of the respective data point is sent as the body text of the e-mail.

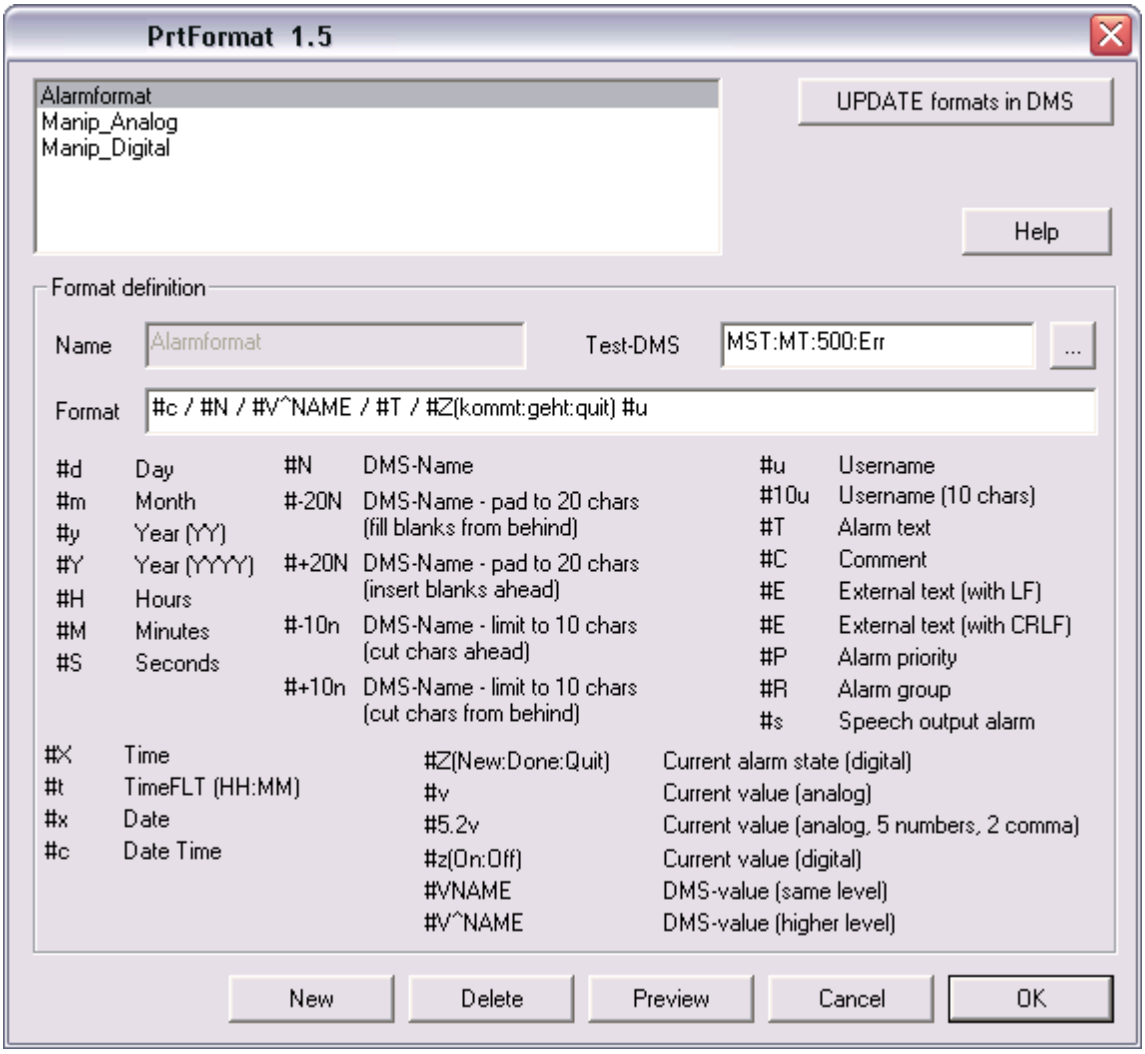
In order to define a subject line, the parameter **#E** must be used in the remote alarm format. The parameter **#E** calls up a text file in the corresponding project directory in the folder **\txt**. The name of the text file must correspond to the DMS name (separator: set to "_" instead of "**:**") of the respective data point. Alternatively, a file with the name **default.txt** can be created. This is called up if no text file with the appropriate DMS name is found.

Example:

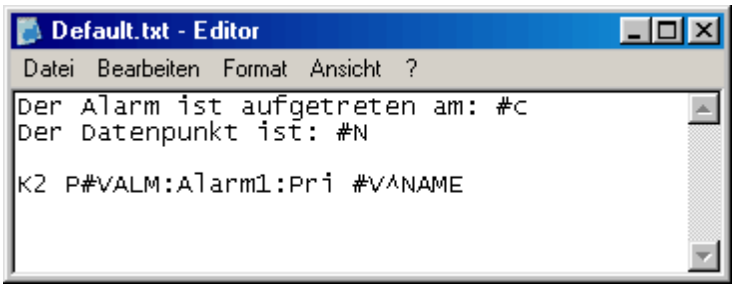
DMS Names	**Text File Names**
S01:AT:001	S01_AT_001_txt

The contents of the text file are sent as the text of the mail. By default, the contents of the respective data point are placed in the subject line.

Example:
The following remote alarm format is defined in the project:
The external text file is called up with the parameter #E.

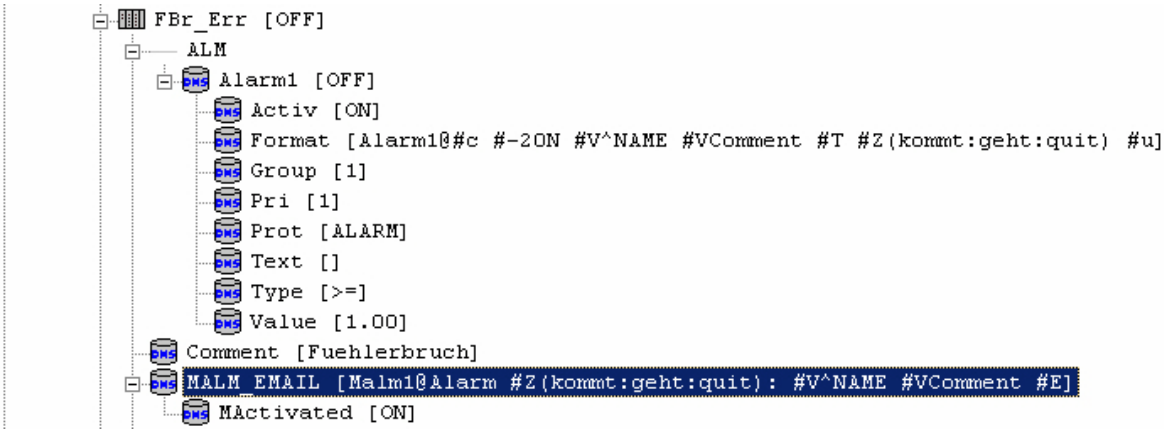


All parameters made available by prtformat can be used in the text file (see also the chapter Auxiliary programs).



The file default.txt must be saved in the project in the subdirectory **\txt**.

For illustrative purposes, look at how it was entered in the DMS here:
A remote alarm was added for the attribute **FBr_Err**.

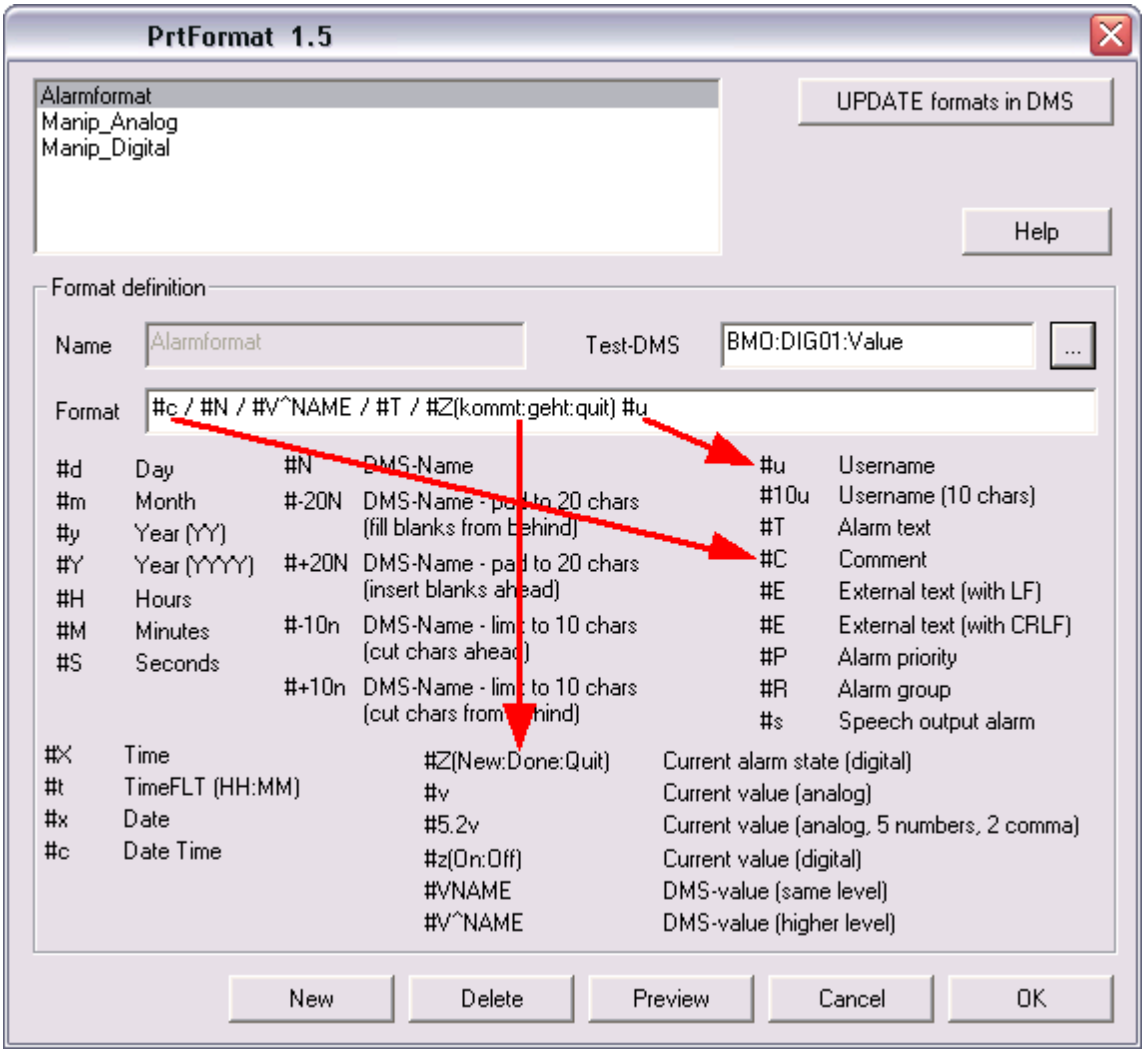


A remote alarm was added to the **attribute FBr_Err** in the DMS. The remote alarm format can be viewed in the data point **MALM_EMAIL**.

Example: defining the subject line of the e-mail manually:
The following configuration is an example of how the subject line of the e-mail to be sent is configured manually:

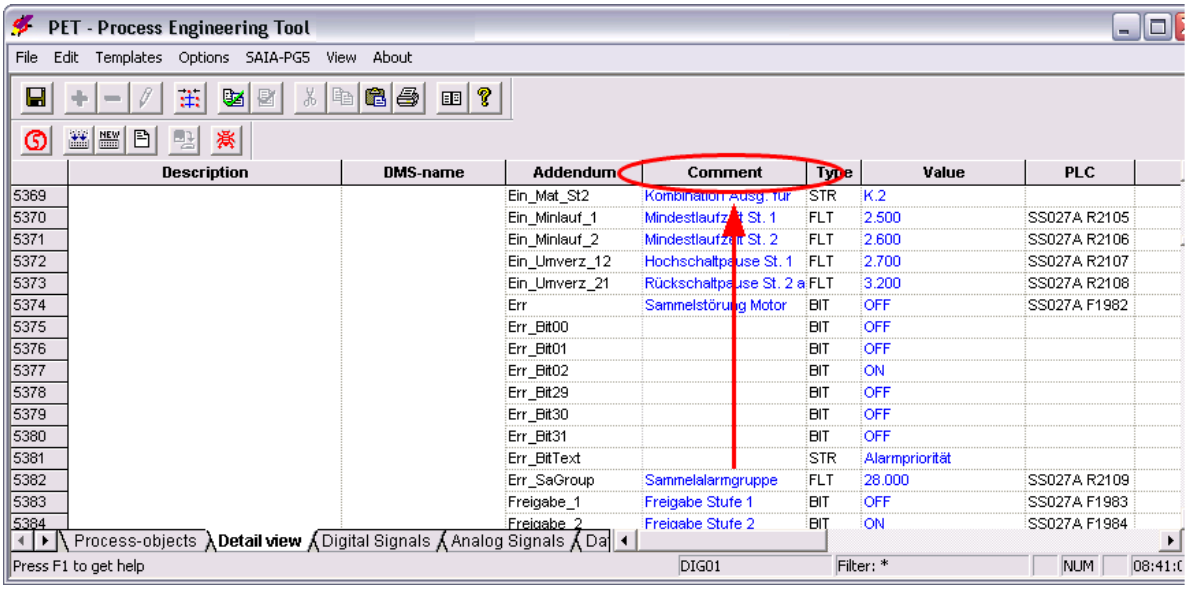
There are hardly any limits when it comes to defining formats!

1. Defining the alarm format for e-mails using prtformat.exe:



#C (Comment) refers to the editable "***Comment"*** column in the PET (see the image of PET below).
These comments are shown in the e-mail subject line as soon as you work with the style sheet ***#E (external text)***. This, like the text file defined in the preceding example, contains all the desired information for the e-mail text area.
In this example, the current alarm status must also be placed in the e-mail subject line. This would also be possible.

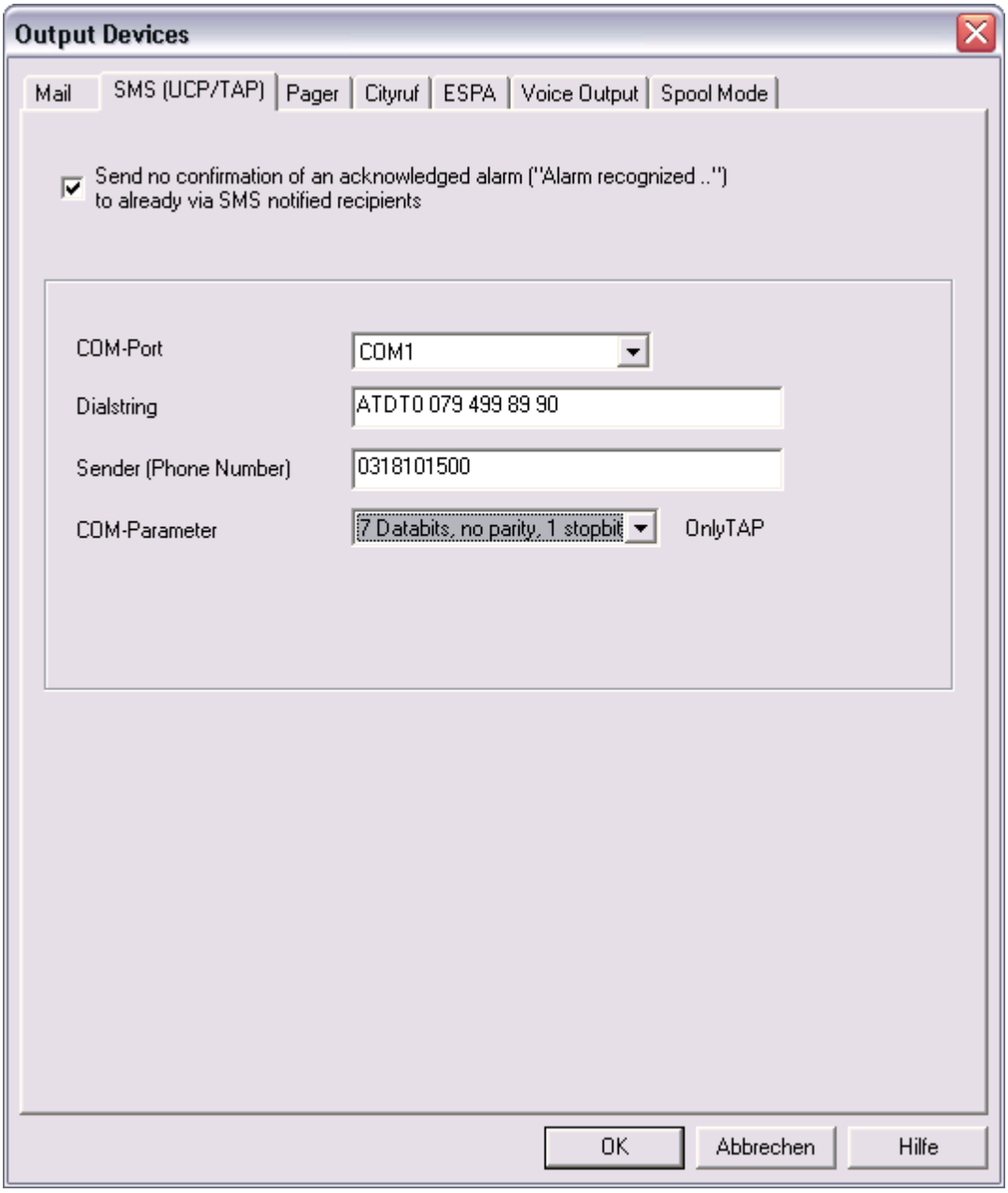
*Working with **#E** is practically mandatory once you have decided to manually define the subject line and that additional information should be shown in the e-mail.*



10.10.12 MAlm configuration via SMS

In order to set off a remote alarm via SMS, the settings of the SMS message centre must be adjusted in MalmCfg.

The SMS account can be configured through the menu **Settings** > **Output Devices,** in the **SMS(UCP/TAP)** tab.



No confirmation message of acknowledged alarm
This option is checked by default, meaning that no confirmation message of the acknowledged alarm is sent.

COM Port
Defines the port to which the modem is connected (COM1 to COM8). The port can be the same as the COM port entered in the register **Pager**.

Dial String (AT)
Adopts the number of the SMS message centre (the Swisscom centre in the example). The type of dialing process must be entered before the number, i.e. ATDT (Attention Dial Tone). The dial string depends on the modem used. Questions about the modem must be directed to the supplier or manufacturer.

Sender (Tel.No.)
Holds the telephone number of the SMS sender. This number is freely definable. The sender number may not contain any blank spaces!

COM Parameters
Makes different SMS-TAP protocols available for the interface:

7 data bits	no parity	1 stop bit
7 data bits	even parity	1 stop bit
7 data bits	odd parity	1 stop bit
8 data bits	no parity	1 stop bit
8 data bits	even parity	1 stop bit
8 data bits	odd parity	1 stop bit


In the preceding image a 0 (zero) was entered under dial string (AT) before the SMS message centre number, which is necessary with some telephone exchanges. In the example the SMS message centre of Swisscom is selected:
079 499 89 90** for **analogue modems,
0900 900 941** for **ISDN modems


SMS message centres cannot process blank spaces in the sender telephone number, so they must absolutely be avoided!


Only the COM port is relevant for the GSM modem.

10.10.12. SMS message centre numbers

Switzerland

Provider	**SMS message centre**
Swisscom	
Analogue:	079 499 89 90
ISDN:	0900 900 941

Italy

Provider	**SMS message centre**	**Hotline**
Telecom Italia Mobile I TELECOM / TIM	+39 33 5960 9600	+39 33 99119
Omnitel I OMNITEL	+39 34 9200 00	0(2|3)+39 34 9200 0190
Wind Telecomunicazioni I WIND	+39 32 058 58500	+39 32 0050 00158

Worldwide
The following links are useful:
http://www.nobbi.com/netw_mnc.htm
http://www.handycheats.de/seiten/smszentralen.html

10.10.12.2 SMS-GSM

MalmMng sends the alarm texts as an SMS using the AT commands if using a GSM modem. There is usually one set of commands for each modem.

Therefore, the type of modem is irrelevant. The only condition is that text mode is used instead of PDU mode. This can be set on most modems.

The Malm sends the message with the command <at>+<cmgs>:

```
<at>+<cmgs>= "phone number" <cr>
"text message" <ctrl>+<z>
```

The modem must therefore also support this command. Otherwise the MalmMng will not transmit any alarms. We recommend using a simple telnet program like Putty to test the system:
www.putty.org

Use this program to test the AT commands.

Some modems forget the configuration after a restart. You may therefore find that the modem returns to PDU mode after every restart.
To remedy this, create a configuration file with the designation gsm.cfg. Here you can store all the AT commands that are executed when starting the MalmMng or before each SMS is sent. This config file (normal text file must be renamed to gsm.cfg) must be saved in the corresponding project directory in the cfg folder, and has the following format:

```
<HEADER>

at-commands ; Define the at commands which are executed when MalmMng.exe is starting.

^<BODY>

at-commands ; Define the at-commands which are executed before an alarm is sending.
```

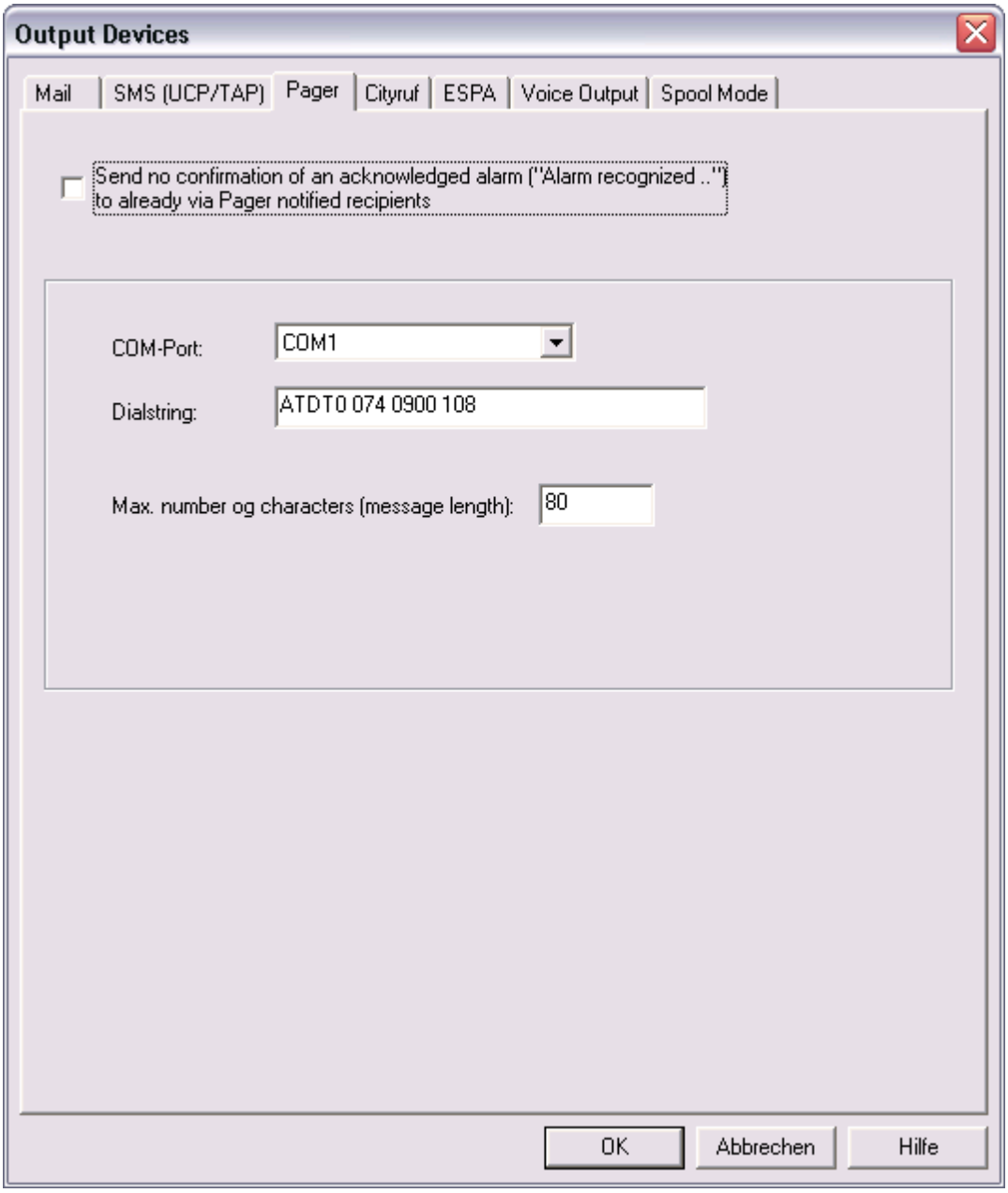
The AT commands must be written in lower case. The executed commands are displayed accordingly in the Malm Windows.

The SIM lock must be deactivated to enable sending.

10.10.13 MAlm configuration via pager

In order to set off a remote alarm via pager, the pager service must be configured in MalmCfg.

The pager account can be configured through the menu **Settings** > **Output Devices,** in the **Pager** tab.



No confirmation message of acknowledged alarm
This option is checked by default, meaning that no confirmation message of the acknowledged alarm is sent.

COM Port
Defines the port to which the modem is connected (COM1 to COM8). The port can be the same as the COM port entered in the **SMS-UCP** tab.

Dial String (AT)
Requires the number of the pager provider. The type of dial process must be indicated before the number. In the image, ATDT (Attention Dial Tone) is selected. The dial string depends on the modem used. Questions about the modem must be directed to the supplier or manufacturer.

Max. Number of Letters (Message Length):
Determines the maximum number of characters that are sent.

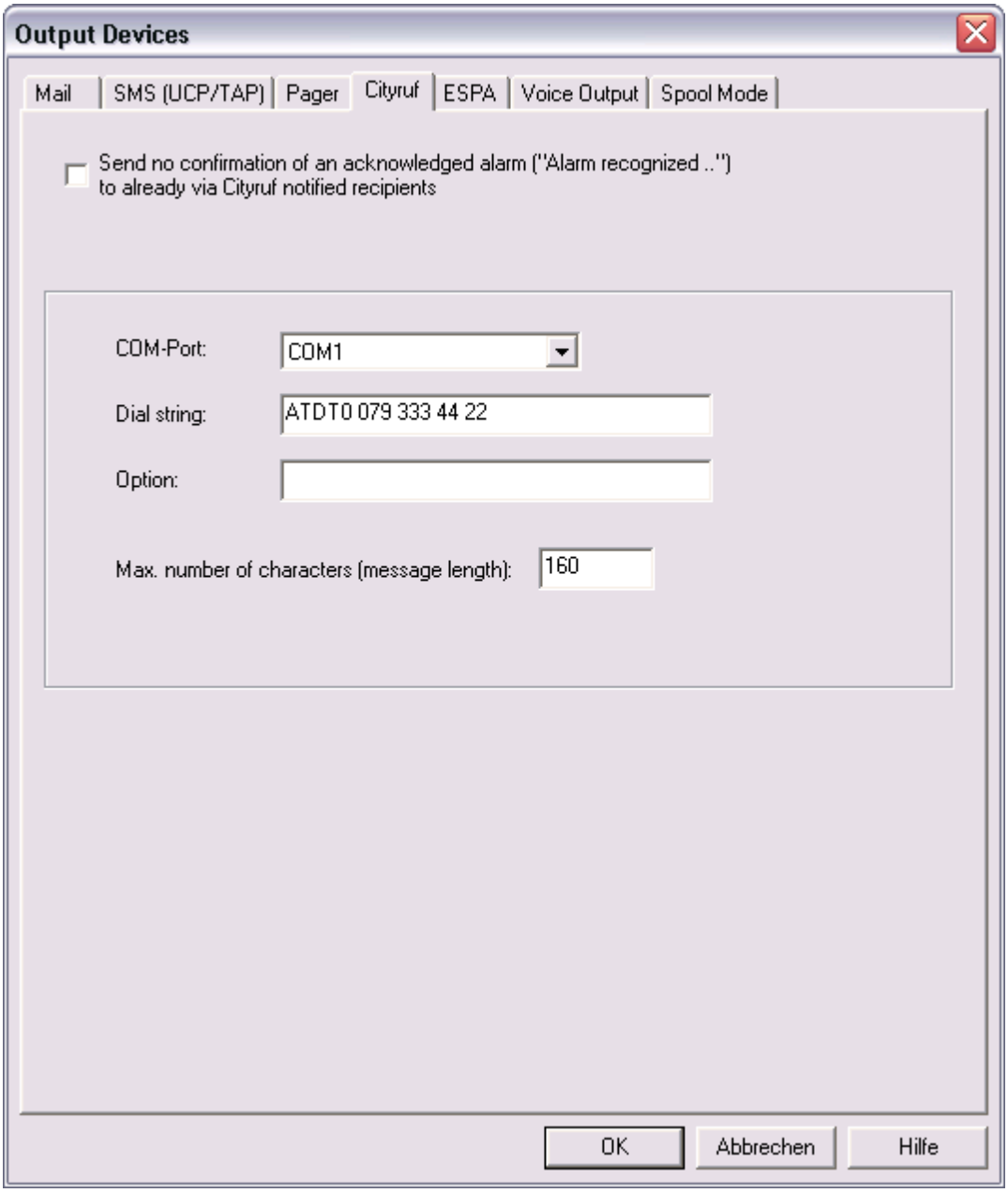
Pager centre numbers:
for ISDN and analogue:
074 0900 108 (up to 38 kBaud)
only analogue:
074 0900 003 (up to 1200 Baud, with 3Com – faster modems – it does not work, error message 11 appears)

*In the preceding image a 0 (zero) was preselected before the pager centre number. The pager centre in the example is for **All Wireless**.*
*The pager centre number depends on the pager subscriber. The number **074 0900 108** only works with pagers that have a subscription to "**SWISS**" (pager via ISDN).*
*If a user who should be informed by pager is defined, the prefix of the pager (074) may **not** be entered in the user information, because the pager prefix is processed directly at the pager centre. If the pager number is entered with a prefix, the centre reports error 106 (invalid number).*

10.10.14 MAlm configuration via Cityruf (paging service)

In order to set off a remote alarm via Cityruf, Cityruf must be configured in MalmCfg.

The Cityruf account can be configured via the menu **Settings** > **Output Devices,** in the **Cityruf** tab.



No confirmation message of acknowledged alarm
This option is checked by default, meaning that no confirmation message of the acknowledged alarm is sent.

COM Port
Defines the port to which the modem is connected (COM1 to COM8). The port can be the same as the COM port entered in the **SMS-UCP** tab.

Dial String (AT)
Requires the number of the pager provider. The type of dial process must be indicated before the number. In the image, ATDT (Attention Dial Tone) is selected. The dial string depends on the modem used. Questions about the modem must be directed to the supplier or manufacturer.

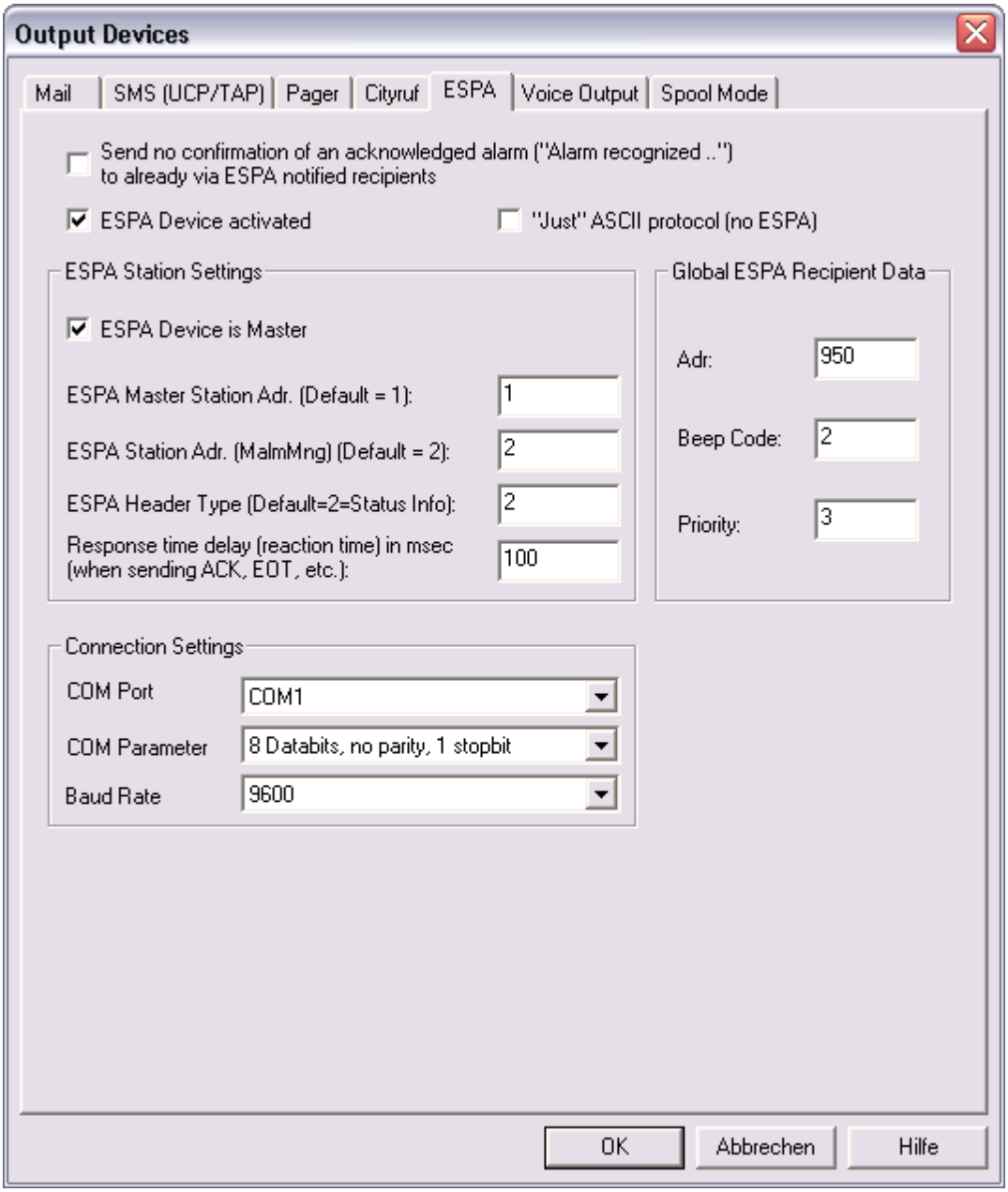
Max. Number of Letters (Message Length):
Determines the maximum number of characters that are sent.

10.10.1 MAlm configuration via ESPA

Licence
For alarm output via the ESPA interface, an additional ESPA licence is required.

In order to set off a remote alarm via ESPA, the settings of the ESPA centre must be adjusted in MalmCfg.

The ESPA account can be configured through the menu **Settings** > **Output Devices,** in the **ESPA** tab.



No confirmation message of acknowledged alarm

This option is checked by default, meaning that no confirmation message of the acknowledged alarm is sent.

Activate ESPA Device
Activate/deactivate the ESPA interface, all settings are retained.

"Only" ASCII Protocol (not ESPA)
When this option is selected, the alarms are sent through the same, configured COM interface, however not according to ESPA protocol, but according to a simple ASCII one-way protocol without confirmation or acknowledgment. This option is suitable for test, evaluation or trial arrangements (customer-specific interface).

The sent data has the following form:

```
<Adr/Alarmtext/Beepcode/Prio>CRLF          e.g.: "<950/Alarm Sensor Break
Incoming/2/3>CRLF"
```

ESPA station settings
The meaning and definition of the ESPA parameters are described in the following documents:

- ESPA 4.4.4.: Serial Data Interface For Paging Equipment (http://gscott.co.uk/ESPA.4.4.4/)
- Protocol Serial Data Interface S942SI (http://en-ssp.ericsson.net/publish/bct/productdocumentation/si_espa.pdf)
- Documentation for EPC444 ESPA Protocol (http://www.concept-training.biz/download/Espa444.pdf)

ESPA Device is Master
When selected, the MalmMng ESPA output type behaves like an ESPA control station, meaning operation as master, otherwise operation as slave.

ESPA Master Station Addr. (Default = 1)
Address of the ESPA Master Station. With master operation this is our own address = MalmMng as ESPA control station; with slave operation this is the address of the remote ESPA master station.

ESPA Station Addr. (MalmMng) (Default = 2)
Address of this MalmMng ESPA output type as ESPA station (must be entered regardless of whether MalmMng ESPA is operated as master or slave).

ESPA Header Type (Default=2=Status Info)
The alarm messages of MalmMng ESPA output types are logged by default as "Status Info" (Header Type = 2).
Other possible settings:
"1" Call to pager
"2" Status information
"3" Status request
"4" Call to subscriber line
"5" Other information

Global ESPA recipient data
The recipient is determined by the following data:
Address (Call address): 0000..9999
Beep Code "1" .. "9"

Priority "1" .. "9"

Along with the **general** (global) recipient data there is also **personally** (see Individual ESPA recipient data) and **ESPA format** defined recipient data (see ESPA format definitions).
They are also sent according to specific rules:
The personal ESPA recipient data has a higher priority than the global data, meaning a personal ESPA setting is sent if it is defined, otherwise the global setting is sent. However, the information from the ESPA formats has the highest priority: if information is present in the alarm format of the current alarm, then it is sent.

Connection Settings

COM Port
Defines the port to which the modem is connected (COM1 to COM8).

COM Parameters
Makes the following parameters available for the COM interface:

7 data bits	no parity	1 stop bit
7 data bits	even parity	1 stop bit
7 data bits	odd parity	1 stop bit
8 data bits	no parity	1 stop bit
8 data bits	even parity	1 stop bit
8 data bits	odd parity	1 stop bit

Baud Rate
Makes the following baud rates available for the COM interface:
1200
2400
9600
14400
19200
38400
56000
57600
115200

10.10.10 MAlm configuration via voice output

Licence
For alarm output through the voice output interface, an additional voice output licence is required.

A **requirement** for the alarm voice output is the presence of **audio wav files**.

- A corresponding audio file must be created in wav format for each alarm defined.Waveform Audio Format is a Microsoft/IBM Audio File Format standard for saving an audio bit stream.
- The Malm alarm format must include the term "**#s**" (format for voice output).

(see also Voice output format definition).

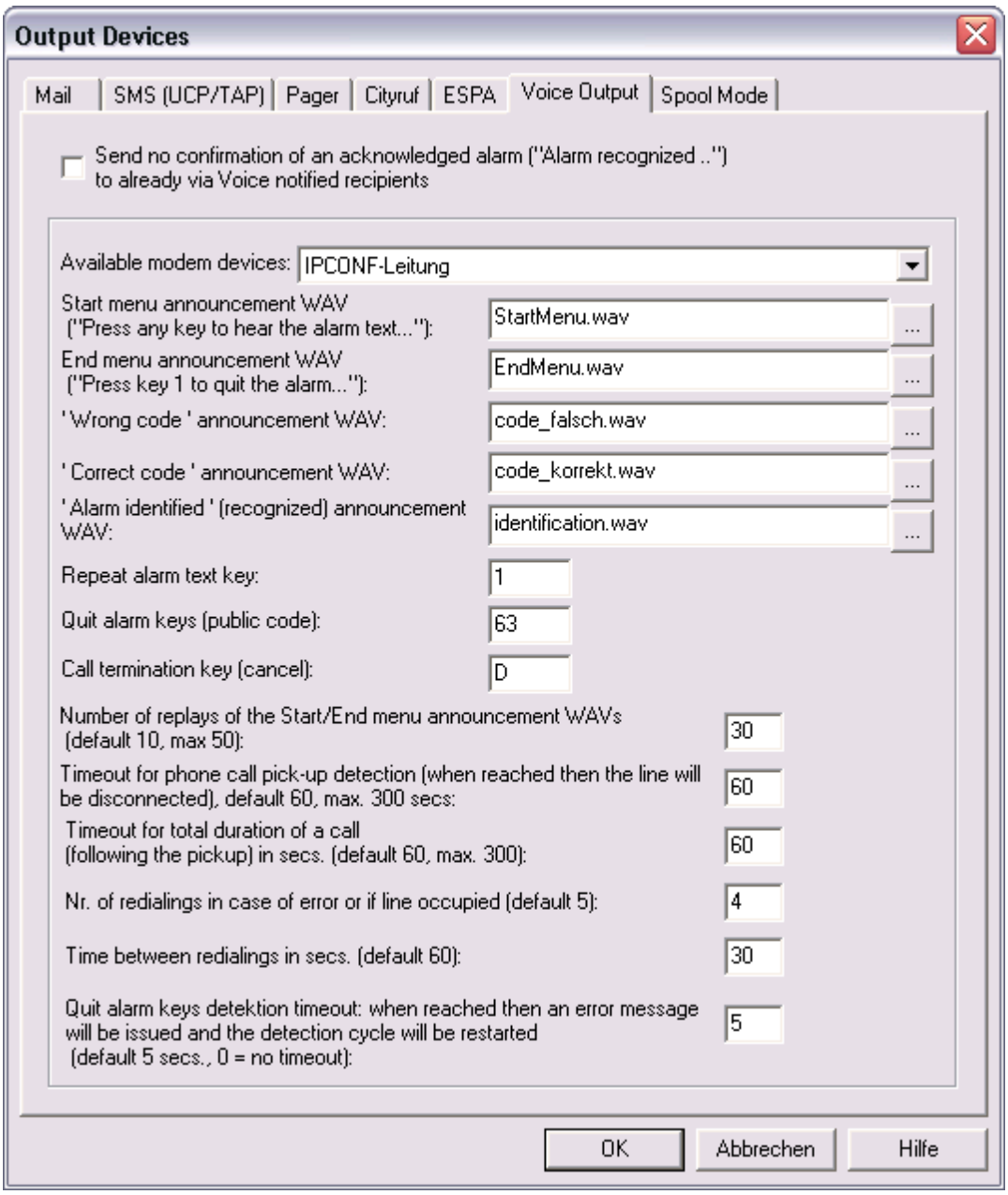
The respective wav audio file is determined from the associated signal data point, e.g. data point = **MST:MT:501:Err**, audio file = **MST_MT_501_Err.wav** or **MST_MT_501.wav**

The audio file must be located in the <proj>\wav directory.

If no audio file with the respective name exists, then the contents of a default file "**Default.wav**" is output, which must also be located in the directory <proj>\wav.

In order to set off a remote alarm via voice output, the voice output settings must be adjusted in MalmCfg.

The voice output account can be configured through the menu **Settings** > **Output Devices,** in the **Voice Output** tab.



No confirmation message of acknowledged alarm
This option is checked by default, meaning that no confirmation message of the acknowledged alarm is sent.

Available Modems
The voice modem is selected from this list.

Start menu announcement WAV ("Press any key to listen to the alarm text..."):
The start announcement audio file is entered in this input field. Alternatively, it can be selected by browsing to the default directory <proj>\wav. This announcement is played when the caller picks up.

End menu announcement WAV ("Press 1 to acknowledge the alarm...")
The end menu announcement audio file is entered in this input field. Alternatively, it can be selected by browsing to the default directory <proj>\wav. This announcement is played when the caller accepts the alarm call after picking up by pressing any key.

'Incorrect code' announcement WAV
The 'Incorrect code' announcement audio file is entered in this input field. Alternatively, it can be selected by browsing to the default directory <proj>\wav. This announcement is played when the caller does not enter the correct call-back code.

'Correct code' announcement WAV
The 'Correct code' announcement audio file is entered in this input field. Alternatively, it can be selected by browsing to the default directory <proj>\wav. This announcement is played when the caller enters the correct call-back code.

'Alarm recognised' announcement WAV
The 'Code recognised' announcement audio file is entered in this input field. Alternatively, it can be selected by browsing to the default directory <proj>\wav. This announcement is played when the current call is an alarm acknowledgment call-back (see Alarm acknowledgment).

Repeat alarm text key
The caller can have the alarm text repeated at any time by pressing the **Repeat alarm text key**.

Call-back code (alarm acknowledgment button) (public key)
(see also Global VOICE call-back code) By inputting these keys, all users who do not have a personal call-back code or confirmation option defined in the dialog "Recipient Details" can acknowledge the alarm call.

Call terminate key (cancel)
The caller can terminate the call at any time by pressing the **Terminate alarm text key**. As a result, the next group member is notified immediately (if this option is selected). If the call is terminated by simply hanging up, the alarm call is counted as having failed and is repeated (according to pre-configured wait time and number of attempts).

Number of repetitions of start/end menu announcements (default 10, max 50)...
The start/end menus are played in an announcement loop until the caller presses the respective menu key or the number of repetitions has been reached.

It must be ensured in the process that the start menu announcement is not played from the moment of the pick-up, but the initialisation of the call. The reason for this is that the voice modem cannot recognise the process of the pickup (it only recognises when the caller presses the key).

Max. time exceeded for tel. pick-up detection...
If the telephone is not picked up within this time, the call is cancelled (line hung up). The call is classified as a failed attempt; this means after the timeout has elapsed (time between redial attempts, see below) the call is repeated, as long as the number of redial attempts has not been reached.

Max. time exceeded for entire length of telephone conversation...
Entry of the longest possible call duration ("Duration of a call") after picking up. If this time is exceeded, meaning that the call is not terminated with a cancel key or acknowledge input, the alarm call is classed as having failed and is repeated (according to configured wait time and number of attempts).

Number of redial attempts in the event of error or busy line...
If a call cannot be successfully terminated (failed attempt), it is repeated the number of times entered here.

Time between redial attempts in sec...
If a call cannot be successfully terminated (failed attempt), a repetition is only attempted after the wait time prescribed here (as long as the number of redial attempts has not been reached).

Call-back code key detection time exceeded...
By default, the call-back code consists of multiple numbers and all individual key entries must be made within a time specified here. If this timeout for pressing a key is exceeded, an error announcement ('Incorrect code') is issued and the code entry cycle is restarted.

The reason for this time limitation is that it is the only way to determine "swallowed" or undetected key events. Based on past experience, there is a problem with the detection of the pressing of a key with some telephones (cordless DECT, in particular). In this manner the caller always receives a response and can repeat the code entry as necessary.

10.10.16. Alarm voice output function

The user to whom an alarm is to be sent is called using the telephone number entered in their "Voice Recipient Details".
The **call is menu-controlled**: the recipient hears an announcement (Start menu audio file) and can then trigger the following actions via guided key selection:

- accept the call and listen to the alarm message by pressing **any key**
- end the call with the **Terminate alarm key**. As a result, the next group member is notified immediately (if this option is selected). If the call is ended just by hanging up, it is taken to be a failed call and repeated after a timeout.
- the alarm text can be repeated at any time with the **Repeat alarm key**.
- the recipient is identified by entering a **Call-back code**. If the entered code matches the user's entered call-back code (see Recipient details: voice call-back code), the alarm is acknowledged, a confirmation statement is announced and the call is ended. Otherwise an error announcement is made and the whole cycle starts again from the top with the alarm text announcement.

If the user call-back code is missing (or the confirmation option), the public

call-back code is valid (see Voice output: call-back code settings).

An alarm call must always be ended with a key action (cancel or quit), otherwise the alarm call is counted as having failed and is repeated (after a preconfigured wait time and number of attempts).

10.10.16.1 Hardware recommendations

Based on our practical experience, we make the following recommendations:

Modem
This model has proven to be robust, reasonably-priced and to experience no installation problems (for the voice driver):
"**MicroLink 56k Fun USB s**" analog Modem by Devolo or ELSA.

All required voice drivers are usually installed automatically when the modem is connected via USB.
If desired, setup can also be performed from the supplied CD, or the most up-to-date trainer can be downloaded from http://www.devolo.de.

The voice/speech output is only ready for use when the following configuration is present in your PC hardware Device Manager:
Sound, video and game controllers: **Unimodem full-duplex audio device.** The Properties dialog must state: Audio devices, modem wave drivers.

This modem is not compatible with Windows 7, only with Windows XP.

This modem has been discontinued since the end of 2010!
The following alternative modems have been verified:

1. Manhattan USB modem (external)
Chipset: Connexant
Connection: USB
Modem name in the Device Manager: USB Data Fax Voice Modem
Additionally installed device: Unimodem Half Duplex Audio Device
Supported Operating Systems: Windows 98 SE/ME/2000/XP/Vista7 / Linux

MALM voice functionality on the following OSes:
Windows XP 32 bit: works with this modem
Windows XP 64 bit: not tested
Windows Vista 32 bit: tested and not working

Windows Vista 64 bit: tested and not working
Windows 7 32 bit: tested and not working
Windows 7 64 bit: tested and not working

Possible reason why this modem does not work with Vista and Win. 7 is that this modem has a Conexant chipset. Windows Vista and 7 were reported to have a bug keeping them from working together with this chipset correctly.

2. Trendnet TFM-PCIV92A (internal PCI soft modem)
Chipset: LSI
Connection: PCI
Modem name in the Device Manager: LSI PCI-SV92PP Soft Modem
Additionally installed device: Unimodem Full Duplex Audio Device
Supported Operating Systems: Windows 98 SE/ME/2000/2003 server/XP/Vista7 / Linux

MALM voice functionality on the following OSes:
Windows XP 32 bit: not tested
Windows XP 64 bit: not tested
Windows Vista 32 bit: not tested but supposed to work
Windows Vista 64 bit: not tested but supposed to work
Windows 7 32 bit: tested and working (good sound quality)
Windows 7 64 bit: not tested but supposed to work

(For some modems it was reported that a sound card is needed to work correctly. These modems were tested in a machine where a sound card was previously installed.)

3. Trendnet TFM-560x Hardware Version:YO (external serial modem)
Chipset: Agere
Connection: Serial
Modem name in the Device Manager: Agere OCM Serial Modem
Additionally installed device: Unimodem Half Duplex Audio Device
Supported Operating Systems: Windows 98 SE/ME/2000/2003 server/XP/Vista7 / Linux / MAC

MALM voice functionality on the following OSes:
Windows XP 32 bit: not tested
Windows XP 64 bit: not tested
Windows Vista 32 bit: not tested but supposed to work
Windows Vista 64 bit: not tested but supposed to work
Windows 7 32 bit: tested and working (sound quality not so good compared to PCI modem)
Windows 7 64 bit: tested and working (sound quality not so good compared to PCI modem)

All tests were carried out with the following type of wave audio files (xxxx.wav): PCM, 8000Hz, 16bit, mono

Server

Installation of the wave driver in **XP/Vista** passes off without any problems; in **Windows Server 2003** it is sometimes necessary to activate the sound (Start, Administrative Tools, Services).


Until now MalmVoice modem has been tested only on Windows XP and Server 2003. There is no guarantee that the driver of the listed modem runs on other operating systems.

Telephone devices

No problems have been detected with any mobile or fixed-line telephones. Only when using **DECT wireless handsets** are frequent problems experienced with DTMF (key detection). If the voice software is unable to detect a key pressed from a DECT device, then it can also not be used to directly acknowledge an alarm. It is usually possible to adjust this accordingly at the telephone exchange.

10.10.17 MAlm configuration via spool mode

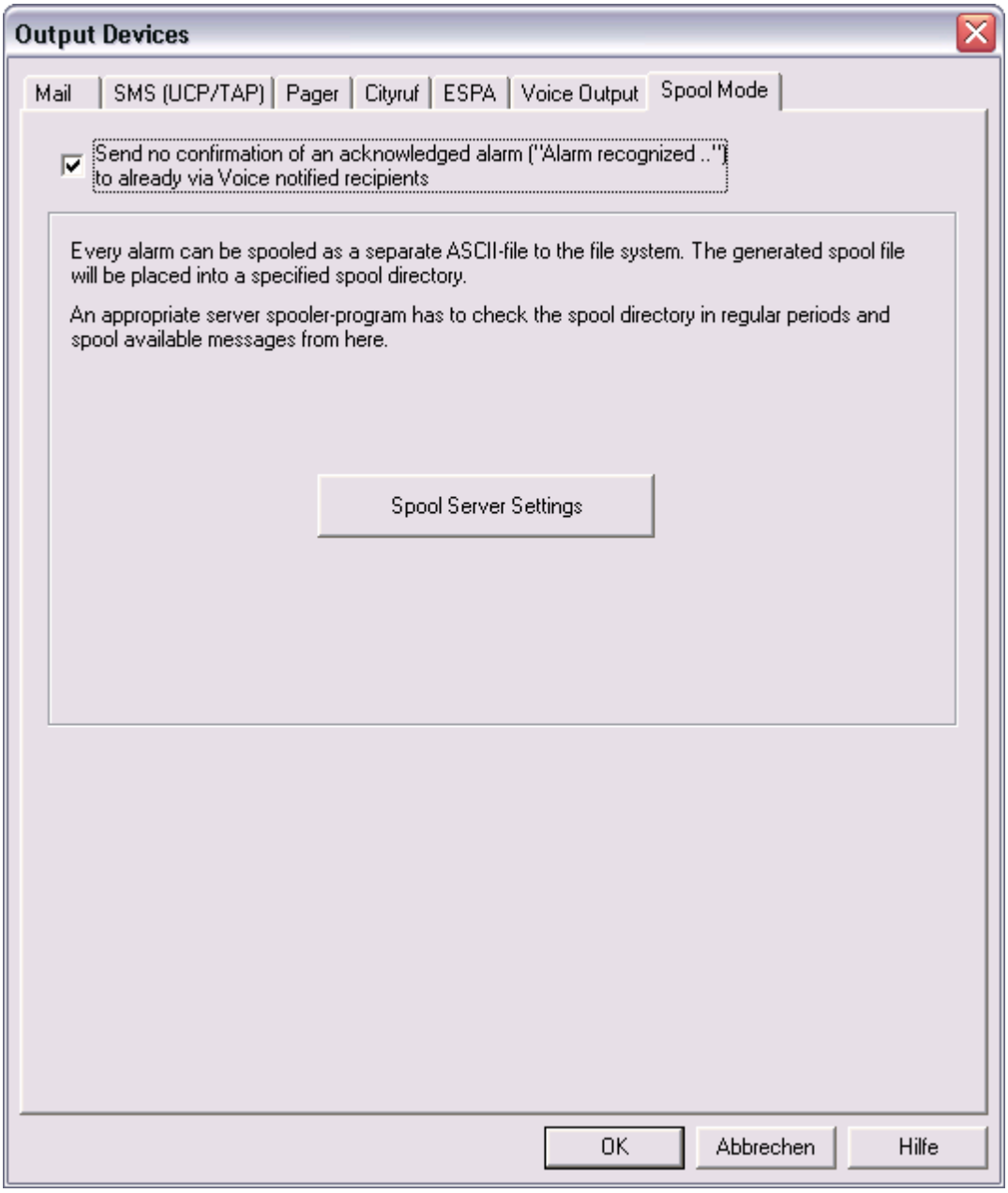
Using the alarm spool mode output requires a **spooler program**, which regularly checks a spool directory and forwards any alarm spool files from here.

Spool file
A spool file is a copy of a spool file template with the following modifications in the 'Provider' section, as per the example below = [SwissCom]:

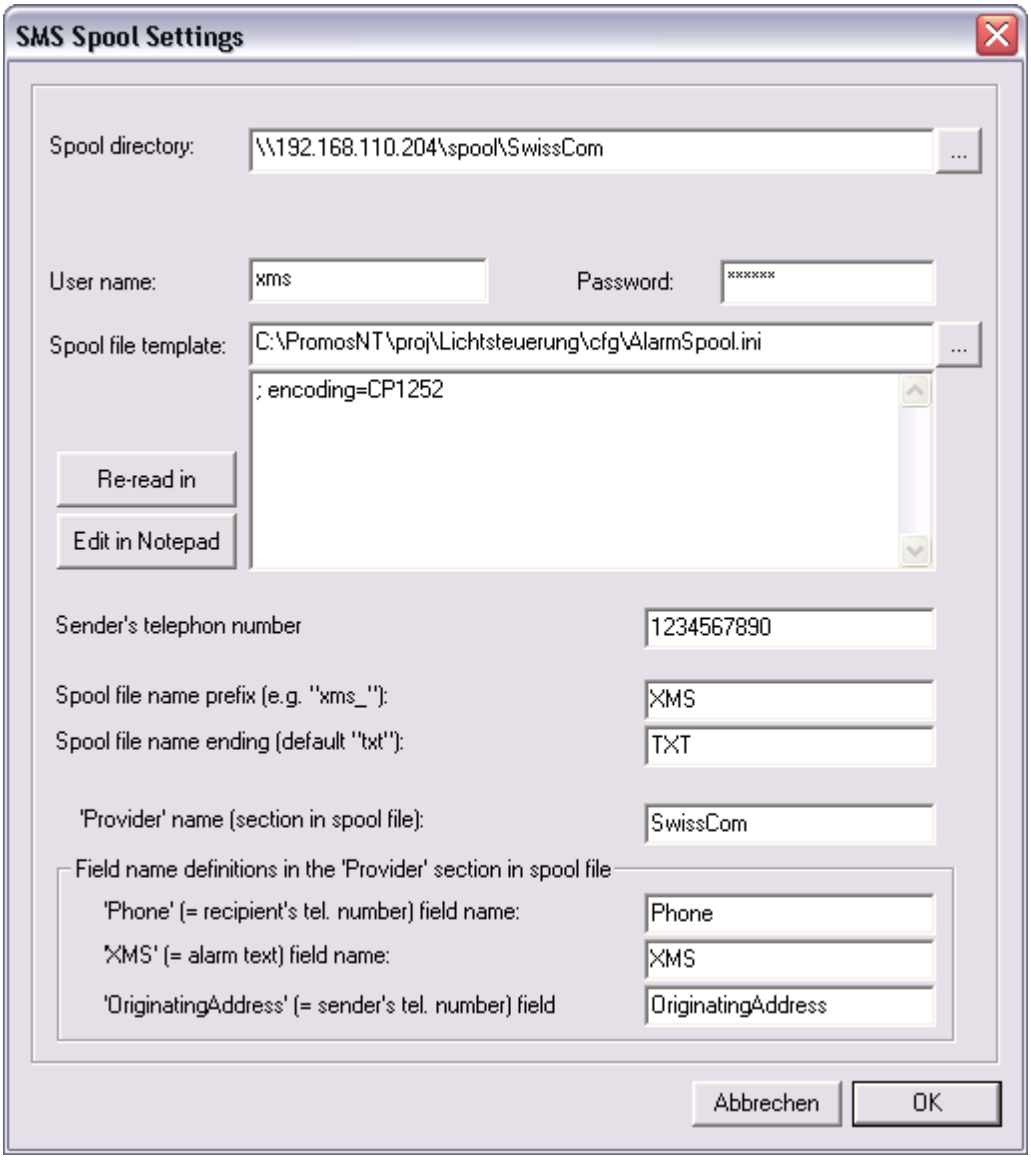
- The telephone number of the current alarm is written in the 'Phone' field
- The current alarm text is written in the 'XMS' field

The spool file name is comprised of the spool file name prefix, a date/time string dd.mm.yyyy_hh.mm.ss (e.g. 11.08.2008_14.25.51) and a spool file name ending.

The spool interface can be configured via the **"Spool server settings"**. These can be found in the menu **Settings** > **Output devices** in the **Spool mode** tab:



No confirmation message of acknowledged alarm
This option is activated as default, meaning that no confirmation message is sent from an acknowledged alarm as standard.



Spool Directory
The spool directory is entered in this field. Alternatively, it can be selected by browsing.

User Name, Password
Mandatory access data for the spool directory (so that the spool directory can connect to a network drive).

Spool File Template
The spool file template file is entered in this input field. Alternatively, it can be selected by browsing to the default directory <proj>\cfg. The content of this file is shown in the field below.

Reimport
If the spool file template was modified in an external editor it is reimported and then displayed.

Edit in Notepad
The spool file template is loaded into Notepad for editing.

Sender Number (Tel.No.)
The telephone number of the SMS sender, which is shown to the recipient..

Spool File Name Prefix
Prefix of the spool file name, e.g. "**MSTsms_**11.08.2008_14.25.51.TXT".

Observe capital and lower-case letters for the prefix (case sensitive)!

Spool File Extension
Extension of the spool file, e.g.: "MSTsms_11.08.2008_14.25.51.**TXT**".

Provider name (section in spool file)
Designation of the section (e.g.: [SwissCom]) in the spool file template containing the necessary alarm information.

Section: Field name definitions in the 'Provider' section in the spool file

'Phone' (=receiver number) field name
Designation of the field (e.g.: [Phone]) in the spool file template containing the telephone number.

'XMS' (=alarm text) field name
Designation of the field (e.g.: [XMS]) in the spool file template containing the alarm text to be transmitted.

'OriginatingAddress' (=sender number) field name
Designation of the field (e.g.: [OriginatingAddress]) in the spool file template containing the telephone number of the SMS sender.

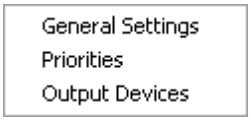
10.10.18 MAlm configuration via TCP/IP

It is possible to forward alarms via TCP/IP. Here, MalmMng works in exactly the same way as forwarding from AlmMng (see section on alarm forwarding (sockets, scripts)).

The TCP/IP connection can be configures via the **"TCP/IP settings..."** button. This is located in the menu **Settings** > **Output devices** in the **TCP/IP** tab:

10.10.19 The MAlm Configurator menus

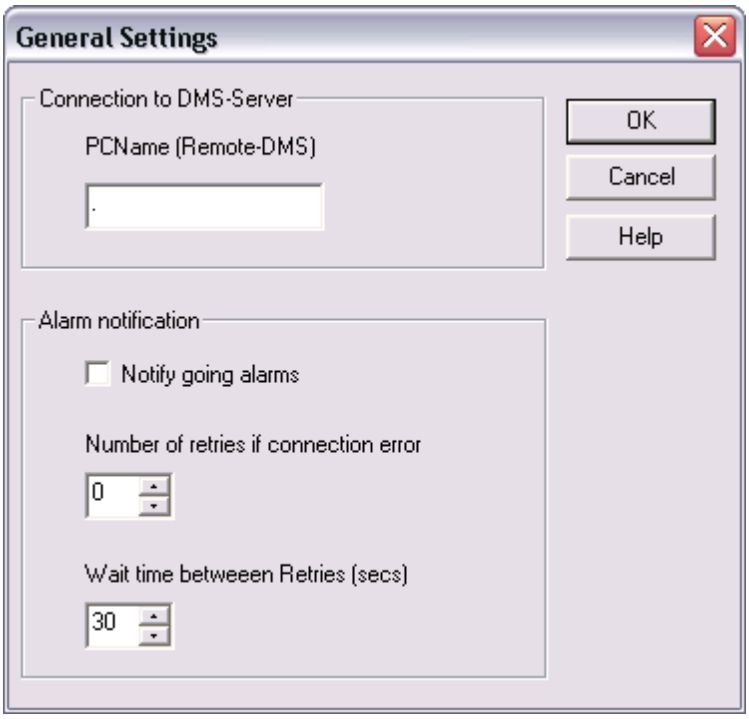
10.10.19. Settings menu



The following options for configuring **MalmCfg** are available under the "**Settings**" menu:

- **General settings** for DMS connection and alarm messages
- Time program for **priorities** (cf. the chapter Configuration of priorities)
- **Output devices** such as **e-mail**, **SMS, pager, Cityruf, ESPA, voice output and spool mode**.

General Settings dialog: General settings for the remote alarms.



PC Name (Remote DMS)
Contains the name of the computer that the DMS is running on. The suggested default setting is a dot (.). The dot means that the DMS is running on the current computer, otherwise the network address of the computer that the DMS is installed on must be entered here.

Thus it is possible to "add" a computer only running the Graphical Editor, which retrieves values from or transmits values to the remote DMS system.

Remote DMS:
The computer that will access the DMS must have access to the server PC, otherwise no connection can be established (the drive must be connected).

Also report outgoing alarms
When this is activated, Visi.Plus sets off a remote alarm for both incoming and outgoing alarms (falling edge).

*Remote alarms are disable for a certain amount of time during the startup of MAlm Manager. If the checkbox **Also report outgoing alarms** is marked, every alarm is displayed in **MAlm** Manager, but not forwarded. If a large number of remote alarms are defined, the disablement time may not suffice to stave off all alarms! For this reason, this function should be used with caution.*

Redial on connection error
When this is activated, Visi.Plus carries out the defined number of redials when a connection error is present.

Wait time between redial attempts in sec.
Visi.Plus will wait the length of time indicated in seconds until a redial attempt is started.

Priorities dialog
Here the priorities of various MAlarms can be defined. See the chapter Configuring priorities.

Output Devices dialog

The **Mail** tab
Configure the e-mail account. More information about this topic can be found under MAlm configuration via email.

The **SMS (UCP/TAP)** tab
Configure the SMS message centre. More information about this topic can be found under MAlm configuration via SMS.

The **Pager** tab
Configure the pager service. More information about this topic can be found under MAlm configuration via pager.

The **Cityruf** tab
Settings for Cityruf alarms (only Germany).More information about this topic can be found under MAlm configuration via Cityruf.

The **ESPA** tab
Settings for ESPA alarms. More information about this topic can be found under MAlm configuration via ESPA.

The **Voice Output** tab
Settings for voice output alarms.More information about this topic can be found under MAlm configuration via voice output.

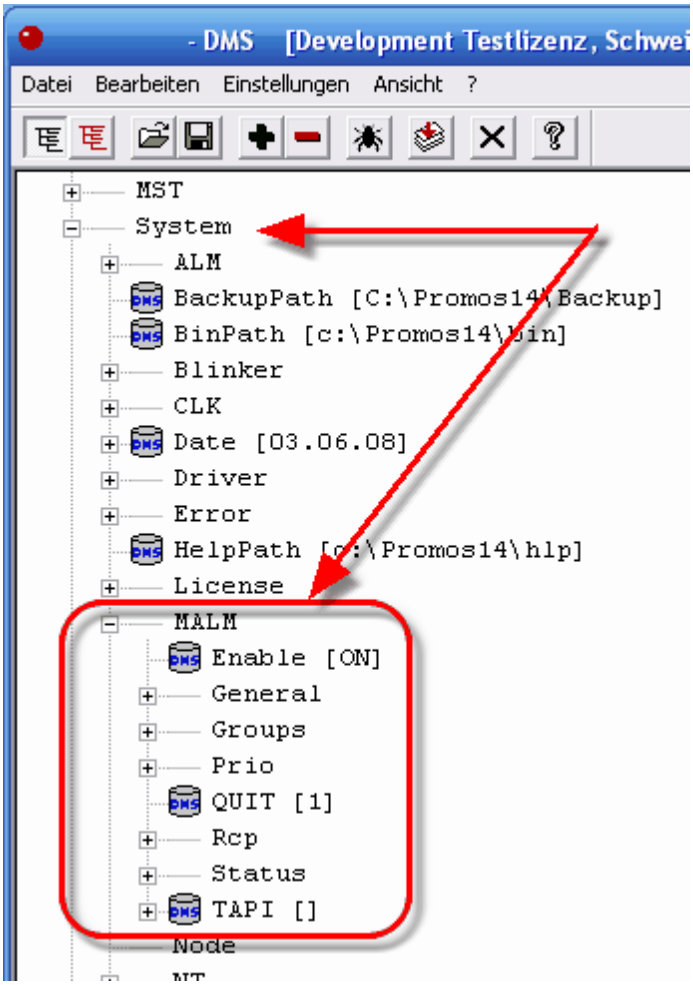
The **Spool Mode** tab
Settings for spool mode alarms. More information about this topic can be found under MAlm configuration via spool mode.

10.10.19.1.1 File menu

The "**File**" menu of the **MAlm** Configurator contains the menu items "**Save**", "**Import**", "**Export**" and "**Exit**".

Save <CTRL+S> or 
The "**Save**" command is used for saving the current **MAlmMng** configurations.

*The command saves the entire DMS, because all configurations of the MAlm Configurator are stored directly in the DMS under the data point "**System:MALM**" (cf. DMS system data points):*



Import
The command reads all configurations of MAlm (= sub-tree "System:MALM") from the Malm.cfg file in the project \cfg folder, writes them to the DMS and then refreshes the MAlmCfg display. The previous configuration is deleted in the process (also from the DMS).

Export
The commands writes all configurations of MAlm (= sub-tree "System:MALM") to the ASCII file Malm.cfg in the project \cfg folder.

Example of a Malm.cfg file:

```
.........................
MALM:Groups:MALM_PWR_KELLER:Pikett PWK Käserei;WOU;0;RW
MALM:Groups:MALM_PWR_KELLER;STR;;RW
MALM:Groups:MALM_PC_CONVENIENCE:Activated;BIT;0;RW
MALM:Groups:MALM_PC_CONVENIENCE:Pikett ELW/MEW;WOU;0;RW
MALM:Groups:MALM_PC_CONVENIENCE;STR;;RW
MALM:Groups:MALM_TEST_AL:Activated;BIT;0;RW
MALM:Groups:MALM_TEST_AL:Wingeier Stefan;STR;;RW
MALM:Groups:MALM_TEST_AL;STR;;RW
MALM:Groups:MALM_QF:Activated;BIT;0;RW
MALM:Groups:MALM_QF;STR;;RW
MALM:Rcp:MEW 601:Dept;STR;;RW
MALM:Rcp:MEW 601:Fax;STR;;RW
MALM:Rcp:MEW 601:Mobile;STR;;RW
MALM:Rcp:MEW 601:Address;STR;;RW
MALM:Rcp:MEW 601:Pager;STR;;RW
MALM:Rcp:MEW 601:Phone;STR;;RW
MALM:Rcp:MEW 601:Mail;STR;;RW
MALM:Rcp:MEW 601:Cityruf;STR;;RW
MALM:Rcp:MEW 601:Voice;STR;601;RW
MALM:Rcp:MEW 601:VoiceCode;STR;01;RW
MALM:Rcp:MEW 601:VoiceConfirm;BIT;1;RW
MALM:Rcp:MEW 601:ESPAadr;STR;;RW
MALM:Rcp:MEW 601:ESPAbeep;STR;;RW
MALM:Rcp:MEW 601:ESPAprio;STR;;RW
MALM:Rcp:MEW 601:Code;STR;;RW
MALM:Rcp:MEW 601:Confirmation;BIT;0;RW
MALM:Rcp:MEW 601:Notify:Device1;WOU;6;RW
MALM:Rcp:MEW 601:Cal:WD:Mo:Active;BIT;1;RW
MALM:Rcp:MEW 601:Cal:WD:Mo:All day;BIT;1;RW
MALM:Rcp:MEW 601:Cal:WD:Tu:Active;BIT;1;RW
.........................
```

 *Malm.cfg is an ASCII file, but we advise against modifying the file directly in an editor (e.g. Notepad)!*

Exit
Exits the **MalmCfg** module.

10.10.19.2 View menu

The "**View**" menu offers different possibilities for modifying the appearance of the **MAlm** Configurator.

Toolbar

Switches the toolbar on or off.

Status bar
Switches the status bar on or off.

10.10.19.3 Help menu

The "**Help**" menu contains the Visi.Plus online help and current information for the MAlm Configurator.

Help
Opens the help system of the MAlm Configurator.
The Visi.Plus online help is accessed through the **"Help"** menu, via the corresponding **<Help>** button or by pressing the **<F1>** key.

About MalmConfig
Displays the current version of **MAlm Configurator**. The version must always be given in case of any support queries!

Should support queries become necessary, the internet forum is preferential to contact via telephone or email. This offers two advantages: First, help is available more quickly; and secondly, other users also benefit from the entries.

10.11 Change Password (ChangePWD.exe)

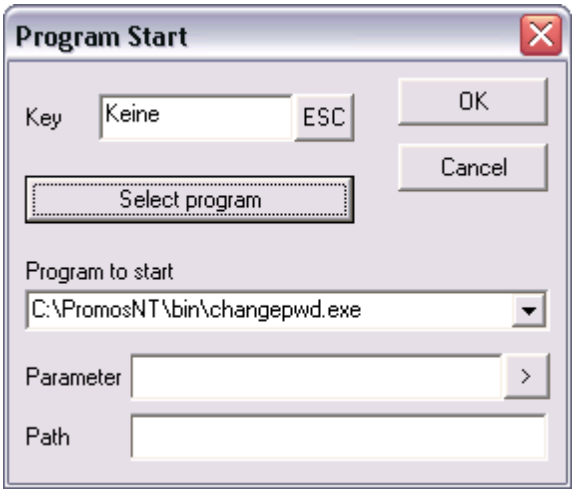
*The passwords of individual users can be changed with the program **ChangePWD.exe** (ChangePassword) without requiring the user to have administrator rights.*

10.11.1 Starting the program

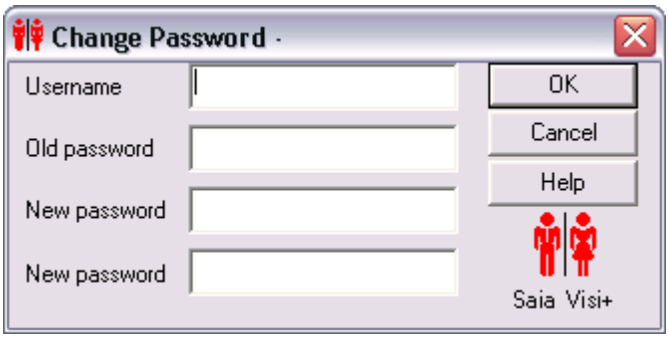
ChangePassword is normally started in the Graphical Editor through an initialised button or right from the default installation directory **c:/Visi.Plus/bin** in the conventional manner.

To do so, proceed as follows:

- Draw a button and call up the "**Action**" attribute.
- Select the "**Start Any Program**" command and fill in the **Program Startup** dialog window accordingly.



10.11.2 The ChangePWD user interface



User Name	Name for identification
Old Password	For approval of the change
New Password	Enter new password
New Password	Enter new password again
<OK>	Confirms the entries and closes the window
<Cancel>	Cancels the change and closes the window

If the user with the indicated information was not found, the following message is shown:

If the user has not correctly entered the new password twice, the following message is shown:

If the new passwords are not identical, the old password remains. Either try again or close the ChangePwd window with **<Cancel>**.

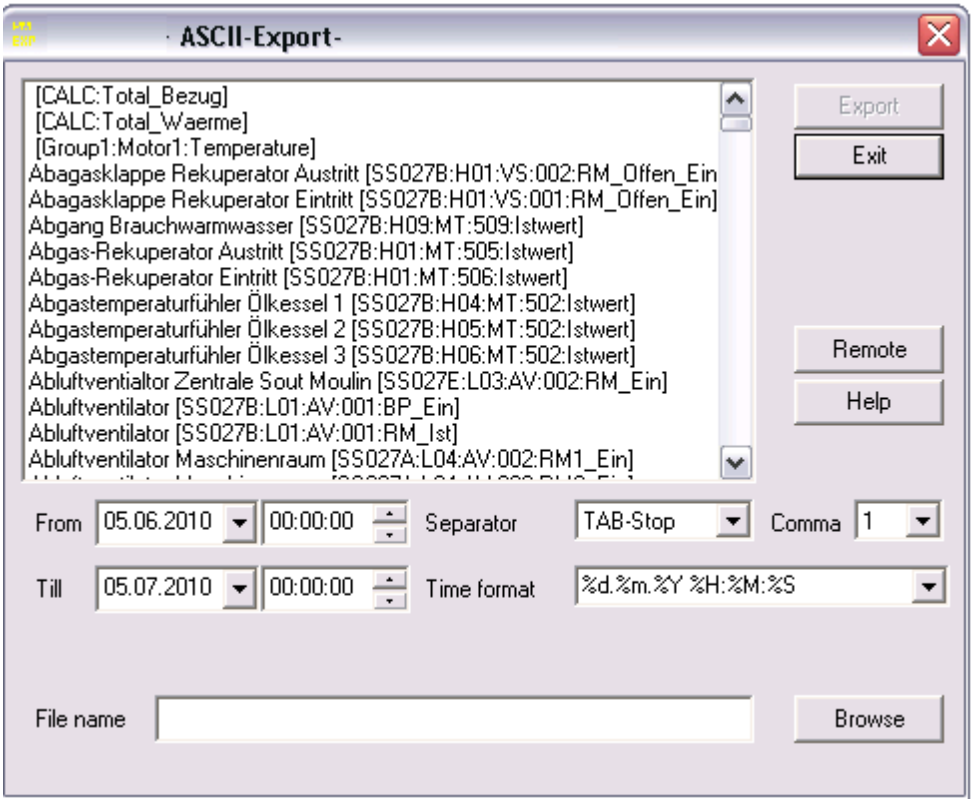
10.11.3 Changing the password in the WebServer

Since no EXE files can be started in Internet Explorer on security grounds, this function is implemented in the Login window in the web server.

This option can be defined in the Web Server under "Settings":

10.12 Creating text files with AsciiExport.exe

The program allows you to perform a data export of a single data point. The data is exported in CSV format (Comma Separated Value) as per the trend data stored in the database.



In this window, all data points for which trend acquisition is activated are listed. After selecting a data point and defining the start (from) and end (until) times, all data from this time period is exported in CSV format (ASCII) to a *.csv file by pressing the **<Export>** button.

By pressing the **<Exit>** button, the ASCII Export program is closed.

By pressing the **<Remote>** button, a connection to a database on another computer on the network can be established. The data is adopted by the remote PC.

By pressing the **<Select>** button, a filename can be selected. The data is then exported to this file.

From	The data is exported from this time onwards. ***From** must be an earlier date than **Until**, otherwise an error message that notifies you of this is displayed.*
Until	The data is exported until this point in time.
Separator	The fields **Date**, **Time** and **Value** are separated by the separator indicated here.
Time format	Any time format can be selected or defined.

%d	Day
%m	Month
%y	Year (2 digits)
%Y	Year (4 digits)
%H	Hours
%M	Minutes
%S	Seconds

Decimal point	Number of decimal points for the exported data and values.

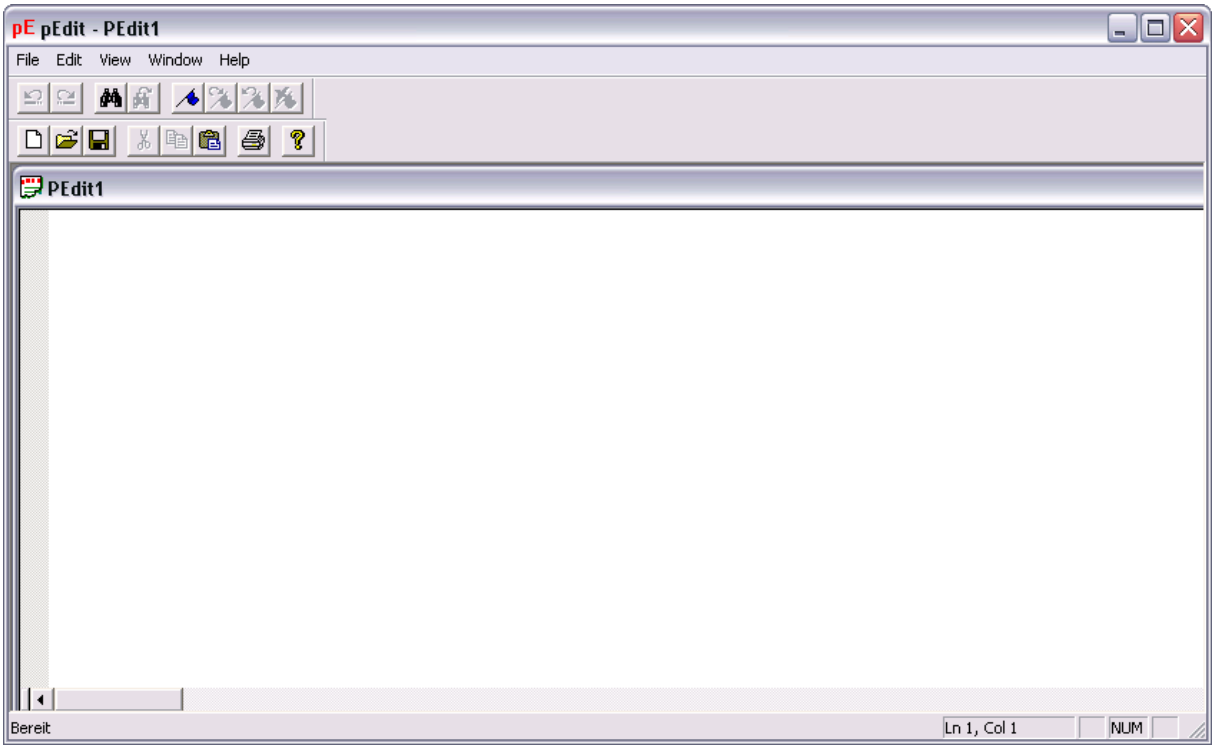
Attention: All data sets in the database will be exported. This should not be performed regularly. More or less data per time unit is logged, depending on the size of the deviations. This makes analysis in a spreadsheet program difficult. The program pChart provides a workaround that permits a CSV export according to preset time intervals.

The exported information (*.txt file) can now be viewed with ease in a Windows text editor (e.g. pEdit.exe or TextPad.exe) and processed further as necessary.

For extensive analysis, we recommend the program pChart (which offers the option for synchronous export).

10.13 Text Editor (pEdit.exe)

The program pEdit is a simple ASCII editor which allows the editing of text files. Instead of this program, it is also possible to use the Windows programs Wordpad or Notepad.
The advantage of pEditor is that this program is somewhat more comprehensive than the Notepad program, and on the other hand it does not permit formatting functions that are not supported in a configuration file, as is the case with Wordpad.
The program can be started from the default installation directory **c:/Visi.Plus/bin**, or via an initialised button.



In addition to pure text editing, the program offers the following functions:

- Find
- Find/Replace
- Copy, Cut

- Bookmarks for easily finding text passages again
- Undo (CTRL+Z)
- Redo (CTRL+Y)
- Print

These functions will not be addressed any further at this point, because they are standard Windows functions.

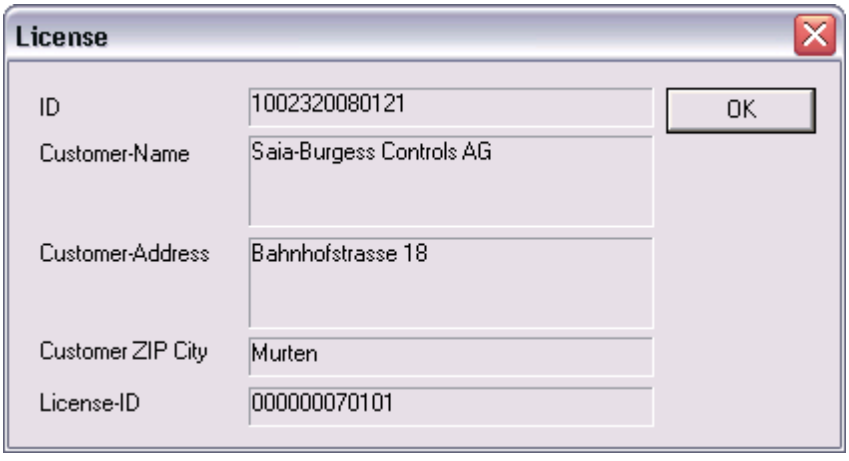
10.14 Logging in with pLogin.exe

A user can use pLogin to log into the system. When doing so, it does not matter which computer the user logs in from. The login always occurs with the DMS (Data Management System). All programs, like the graphics system, the Alarm Viewer, etc., check the access rights through the DMS. Each user is entered individually in the DMS in **System:User** (incl. the PC where login was called up).

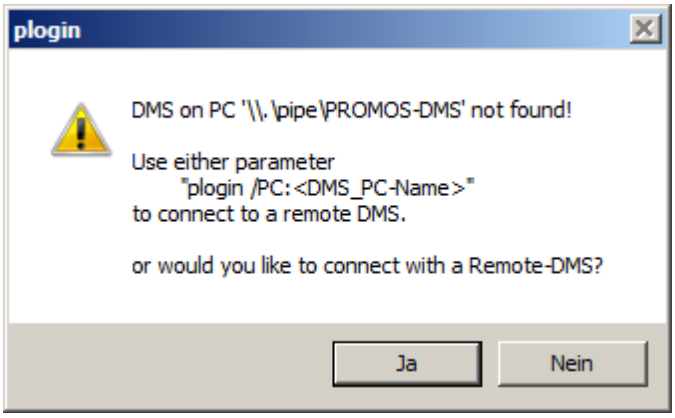
pLogin is usually started from the default installation directory **c:/Visi.Plus/bin**, or via an initialised button in the GE.

The licence data of the currently set system is shown in the Login window:

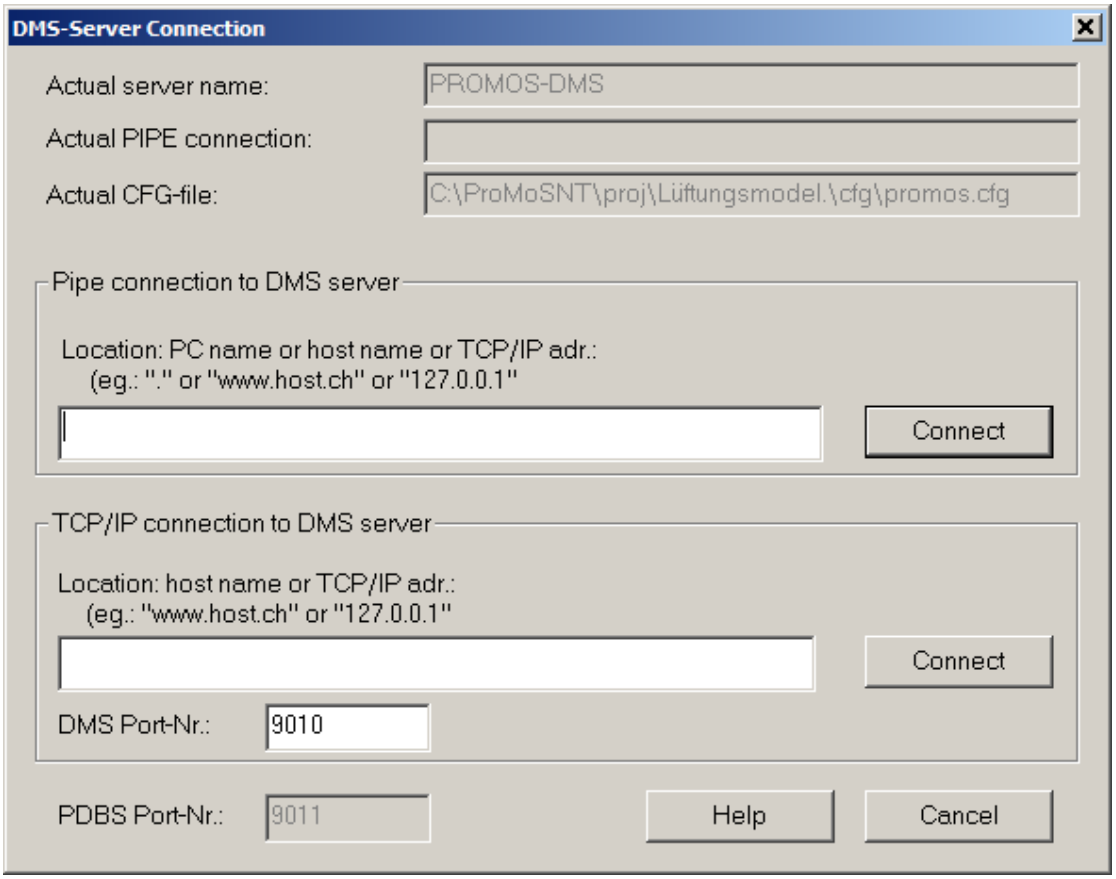
If the **<Licence>** button is pressed, the following dialog window opens, where precise information regarding the installed licence is shown



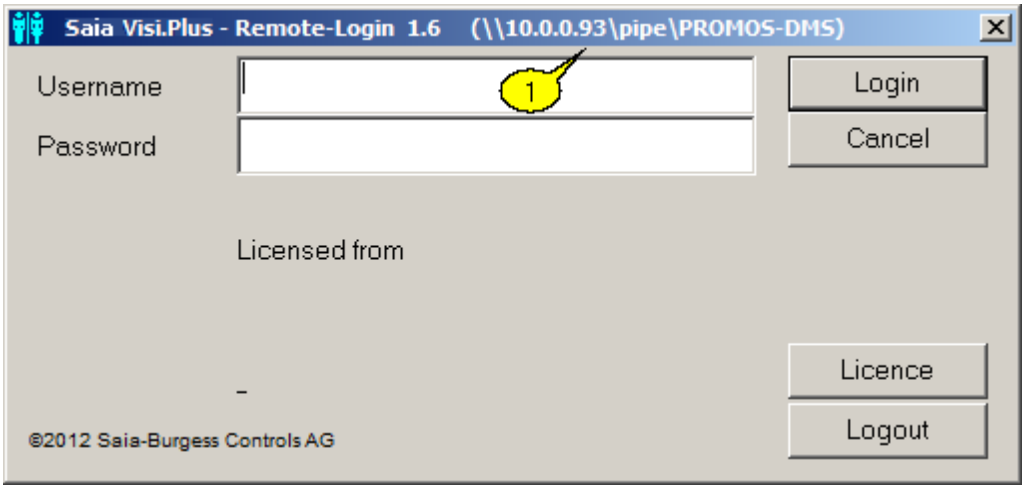
If no DMS is present on the local PC and pLogin starts, the following message will appear:



Use the **<Yes>** button to open the settings window:

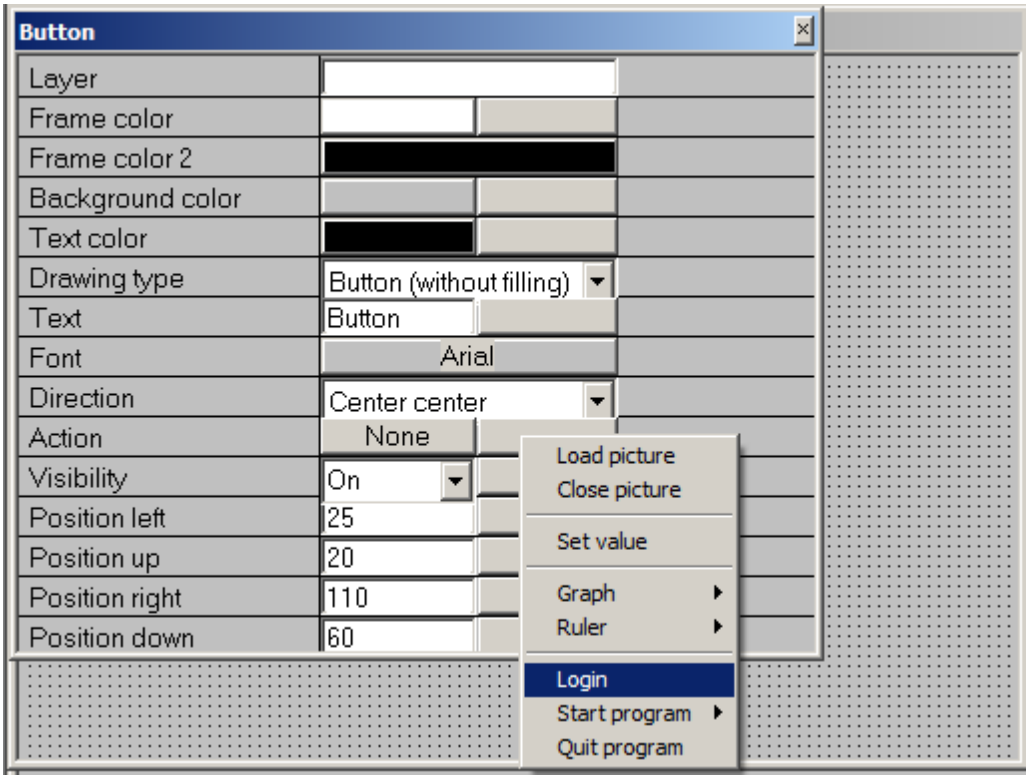


The settings with pipe or TCP/IP are the same as for a remote connection of the graphical editor. For more information, see the chapter Starting a Remote Graphical Editor

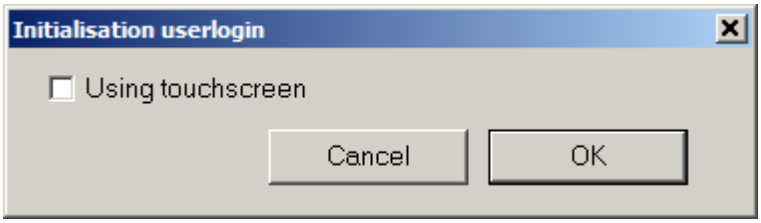


1): Once the remote connection has been established, the connection address appears in the header.

If pLogin is to be started via a button from the GE, a button must first be drawn.
Then open the attributes of the button and make an initialisation to System Login under **Action**.

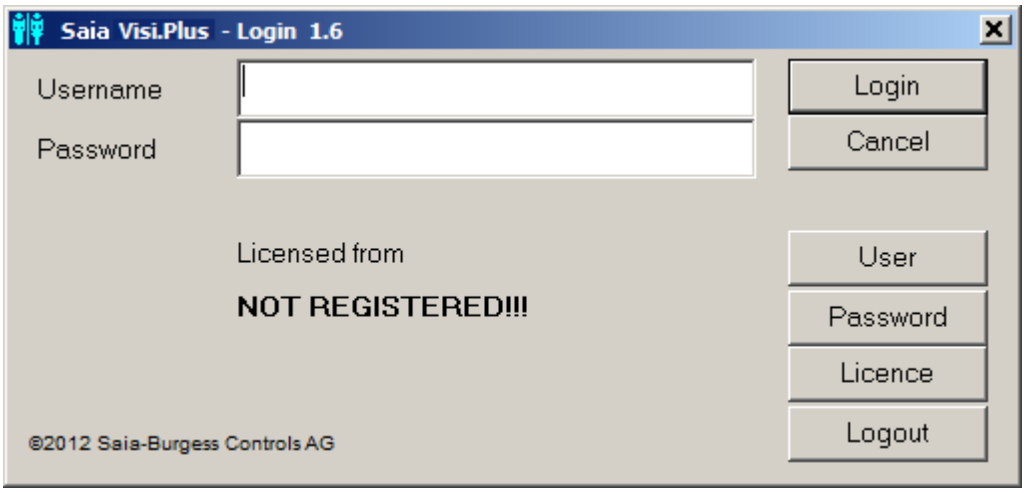


The following dialog opens:



If you want to make entries on a touchscreen, mark this checkbox.

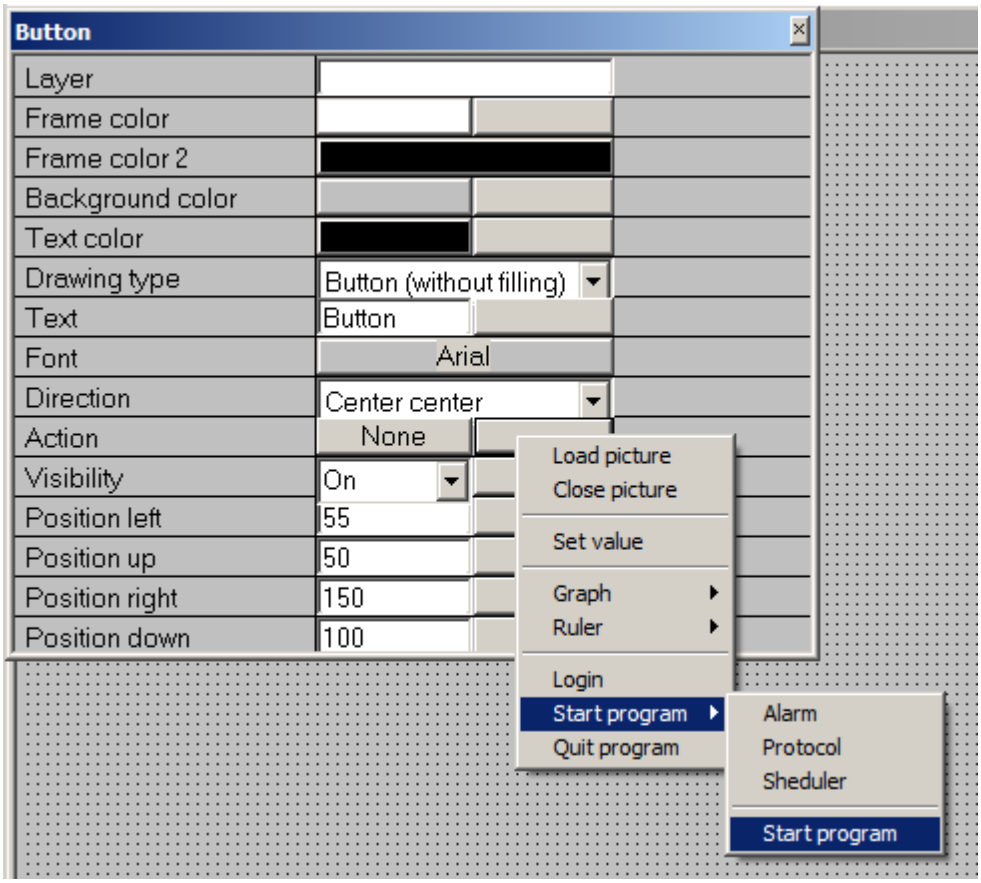
With the checkbox activated, two additional new buttons then appear in the Login window: ***<User>*** and ***<Password>***:

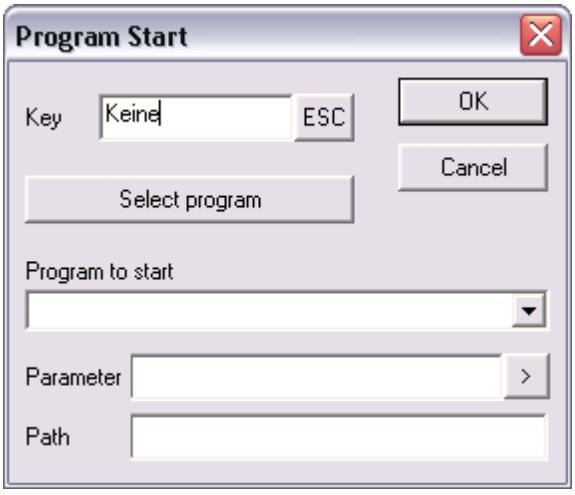


If the **<User>** or **<Password>** button is pressed, the following window opens with a virtual keyboard:

Here the necessary entries can be made with the mouse pointer via the touchscreen and confirmed with **<OK>**.

pLogin can also be called up directly with parameters, but a button must also be created for this purpose in the GE and initialised to ***Program Start > Any Program****:*





Select the program to be started here from the appropriate directory.

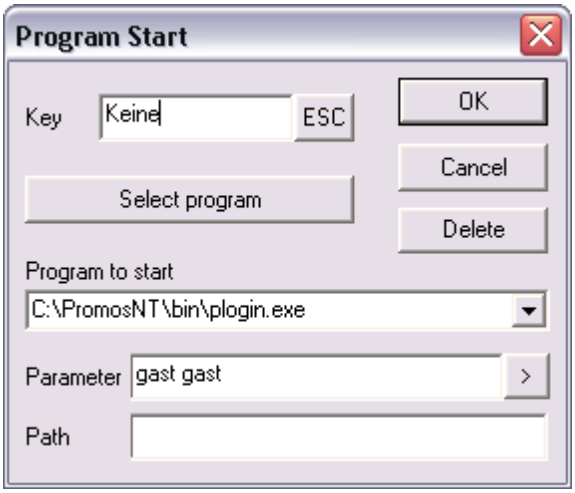
10.14.1 Program parameters

Possible parameters:

Username
Username Password
PC=Name of the PC (Remote)
Username Password PC=Name of the PC

Example: guest guest PC=198.168.0.45
(Here the user was logged in as a "guest" with the password "guest" on the PC with the IP address)
-->Visi.Plus login

Example: guest guest
With these parameters the user "guest" is logged in with the password "guest" when the program starts.



Example: guest
If only the parameter **"guest"** is entered, the Login window opens and the user name **"guest"** is already entered.

The program pLogin.exe can also be started from other applications in order to ensure a user access to the functions of Visi.Plus. The login authorises the PC from which the program was started.

10.14.2 Additional command line parameters

If the program pLogin.exe is started from other applications, the following command line parameters are offered (they can be entered in any combination):

Virtual keyboard (touchscreen)
"**/T**" or "**/t**": two new additional buttons appear in the Login window: ***<User>*** and ***<Password>***; when clicked, a window with a virtual keyboard opens: see Input via touchscreen.

Logout
"**LOGOUT:1**": if available, a logout is executed, meaning login with User="" and Password="".

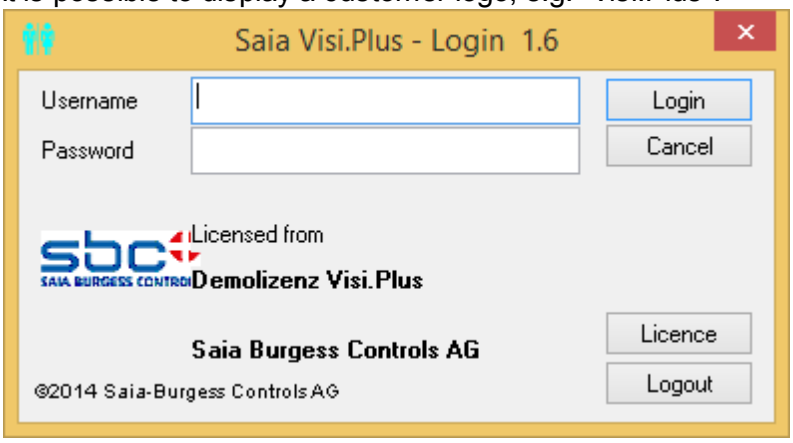
PC name
"**/PC:pcname**"

User Name
"**/USER:username**"

Password
"**/PASS:password**"

10.14.3 Displaying a customer logo

It is possible to display a customer logo, e.g. "Visi.Plus":



The logo file must be in the BMP or JPG formats.

The settings can be defined in pLogin.cfg (folder ...\proj\cfg):

```
[Settings]
Logo = logo.bmp
;Logo = c:\Visi.Plus\bin\logo.bmp
;(if the full path is not specified, the logo is searched for in the folder proj\xyz\bmp)


OrignSize = 1
;or OrignSize = 0
;(1 = original size , 0 = resized, Default = 0).
```

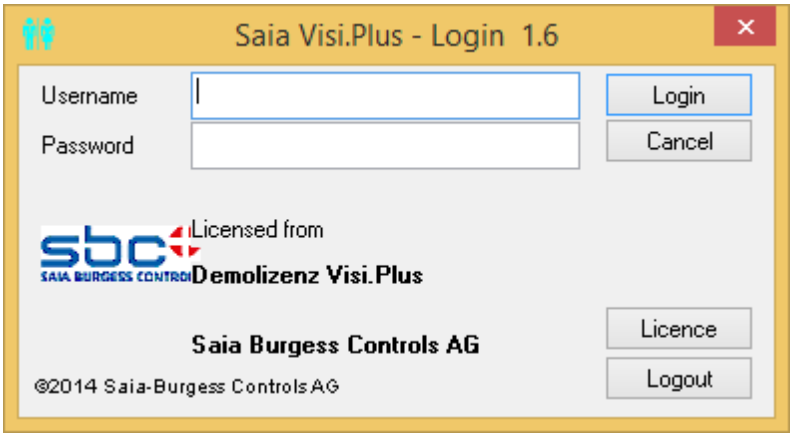
If the logo exceeds a size of 64 x 52 pixels, it is resized automatically.

The size and position of the logo can also be adapted through the following settings:

```
[Settings]
LogoPosX=20        ;Position, x-axis default= 11
LogoPosY=120       ;Position, y-axis default= 80
LogoMaxHeight=50   ;Height of the diagram  Default= 100
LogoMaxWidth=40    ;Width of the diagram, default= 80
```

10.14.4 Display the remote login button

If desired, a remote login button can be inserted. The button appears directly below **<Cancel>**.



The settings can be defined in pLogin.cfg (folder ...\proj\cfg):

```
[Remote Settings]
RemoteLoginON=1    ;1= button visible, 0= button hidden default=0
```

Saia Visi.Plus

Auxiliary programs

Chapter 11

11 Auxiliary programs

11.1 Summary

This chapter describes how the Visi.Plus **help programs** are used and operated.

Visi.Plus has the following **help programs**:

Recipe Management
(PMosFilePicker.exe)
Save and process recipe data

Format Definitions
(prtformat.exe)
Generate output formats

Modify data point content
(SetDMSValue.exe)
Read and process DMS values

Image conversion
(bmp2jpg.exe)
Convert images

Drawing program
(pPaint.exe)
Create drawings

Version overview
(ShowVersion.exe)
Overview of all program module versions

Data backup
(pBackup.exe)
How data can be backed up

11.2 Recipe Management (PMosFilePicker.exe)

If values from an **ASCII file** are written to the **DMS** by pressing a button ("button" graphical object) or the reverse, this program is applicable.

The **PMosFilePicker** has two primary functions:

- Loading DMS values from an ASCII file
- Saving DMS values to an ASCII file

The **PMosFilePicker** can be called up from Visi.Plus using an initialised "**button**" graphical object in the **GE** (interactive mode) or directly from the command line (silent mode) - suitable for calls from individual customer programs.

Preview

- **Prepare DMS for data exchange**
  The next chapter describes how to prepare the DMS so that PmosFilePicker can be used.

- **Structure of ASCII files**
  Conditions for editing ASCII files and two sub-chapters on their structure for "**Loading**" and "**Saving**" data points

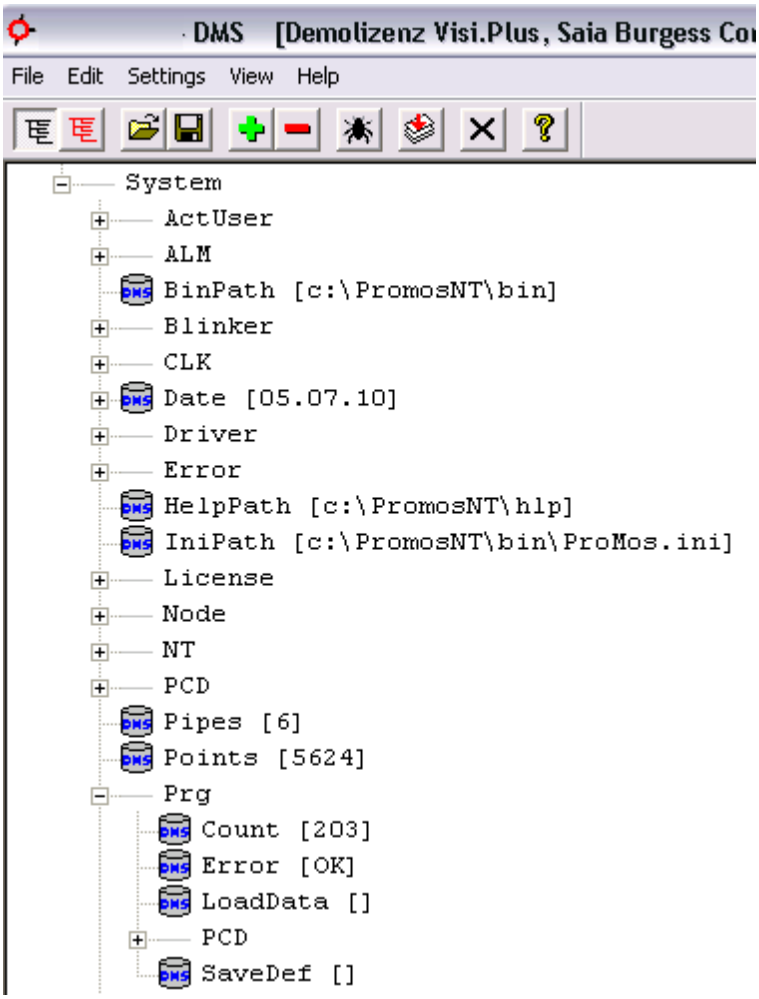
- **Examples**
  At the end of this chapter there are two examples with "step-by-step" instructions. They demonstrate very simply the two options - "**Loading**" and "**Saving**" - with the "**promos**" project included in the scope of delivery.

- **Further examples**
  Of **PMosFilePicker** being called up using the command line

11.2.1 Preparing the DMS for data exchange

At present, the following data points must still be partly entered in the DMS manually so that **PMosFilePicker** can be used (for information on editing the DMS, see **Chapter 7.2.4**):

System:Prg:LoadData
(is already present)
System:Prg:SaveDef
Type **STR** (must be entered manually)



After making entries in the DMS, do not forget to save!

The following three system variables (the last three lines in the upper image) are used by **PMosFilePicker** as described:

System:Prg:LoadData
Shows the current filename, including the directory path from which the data is gathered (see "**File structure for loading DMS values**").

System:Prg:SaveDef
Indication of the filename including directory path.
The file contains a list with DMS names that indicate the data points that should be saved to an ASCII file (see top image).

*Of course the contents of the variables can be displayed in the process diagram by a "**Text**" graphical object in the GE when necessary.*

11.2.2 File optimisation

If you have a lot of small DMS data exchange files (recipes), they can be bundled into a single file. For this purpose, use the standard method of file data classification according to sections.

The file is broken down into sections, each of which extends to the beginning of the next section or to the end of the file in the case of the last section. Sections are indicated in square brackets.

The data is then read from a section, which was indicated in the data point **System:Prg:LoadData** in addition to the filename (see below).

Example language switching text file:

```
[Texts_D]
System:Text:Buttons:Beenden;Beenden
System:Text:Buttons:Einrichten;Einrichten
System:Text:Buttons:Heizkurve;Heizkurve
System:Text:Buttons:Hilfe;Hilfe
System:Text:Buttons:Inbetriebnahme;Inbetriebnahme

[Texts_NL]
System:Text:Buttons:Beenden;Beenden;beëindigen
System:Text:Buttons:Einrichten;Einrichten;inrichten
System:Text:Buttons:Heizkurve;Heizkurve;stooklijn
System:Text:Buttons:Hilfe;Hilfe;help
System:Text:Buttons:Inbetriebnahme;Inbetriebnahme;in bedrijfname
```

The data point **System:Prg:LoadData** must then also contain the corresponding section name after the filename (with a **";"** as a separator):

"System:Prg:LoadData" = "c:\Visi.Plus\proj\abcd\language.txt;Texts_D"

or:

"System:Prg:LoadData" = "c:\Visi.Plus\proj\abcd\language.txt;Texts_NL"

 Entries are case-sensitive, however, blank spaces before and after **";"** are disregarded.

11.2.3 Structure of ASCII files

All files that are used by the program **PMosFilePicker** or by the **DMS** may only be created or edited with a normal ASCII editor!

 *The files may only be created with an ASCII editor. The simple "**Notepad**" included with the operating system could be used. It is usually found under "**Start > Programs > Accessories**".*
*"**Wordpad**" and similar editors do not work, because they insert control characters (e.g. for bold type) in the file!*

11.2.3.1 File structure: loading values into the DMS

When loading DMS values from a file into the DMS, the file must have the following format:

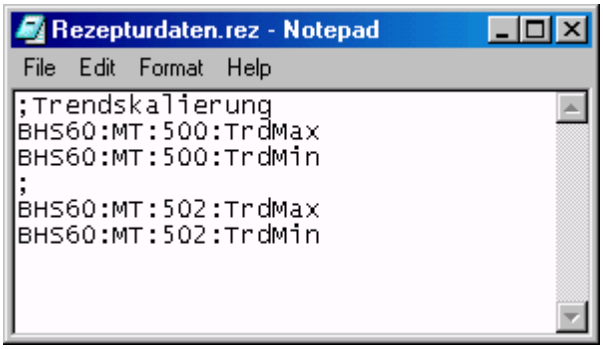
<DMS Name>;<Value>;Comment
<DMS Name>;<Value>etc.

One line per value!

A semicolon ";" MUST be placed between the DMS name and value as a separator.

Binary values are defined as **0** and **1**.

Comment lines begin with a semicolon **";"**.

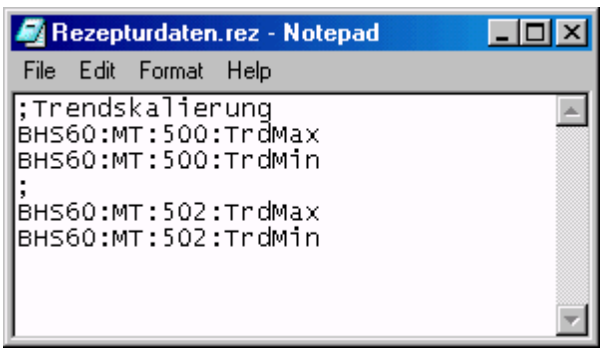


***Attention**:*
The indicated DMS names must be present in the DMS!

11.2.3.2 File structure: saving DMS values

In order for **PMosFilePicker** to know which data points should be exported from the DMS, **PMosFilePicker** searches in the **DMS** for a data point with the name **SaveDef**, as soon as it starts up. A filename with extension and directory path must be entered in this data point, which, in turn, should include a list of the data to be saved. See the following image.

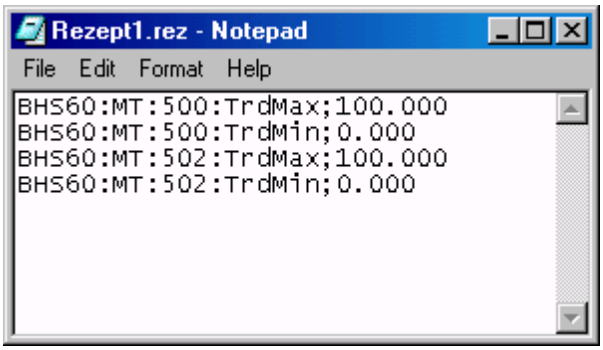
The structure and the rules of the file contents, in other words the list, correspond to those described on the preceding page, but **without** value entries!



If an appropriately initialised "**Button**" graphical object is pressed, a "**Save As...**" window opens.

The desired name of the new recipe file is entered as the filename or, if recipe data already exists, one of these is selected to be overwritten.
The file extension is automatically assigned for a new name.
After pressing **<Save>**, a copy of the contents of the file indicated under **System:Prg:SaveDef** in the **DMS** is saved to the specified recipe file. The current value is entered after each DMS name followed by a semicolon **";"**.

Result:



11.2.4 Starting the program

Requirement:

- The DMS must be running!
- The program **PMosFilePicker** is always called up using parameters. It would not make sense to start the program without the use of parameters. It is preferable to start the program using an initialised "**Button**" graphical object in the **GE** (for the procedure, see examples 1 and 2, section 11.2.4).

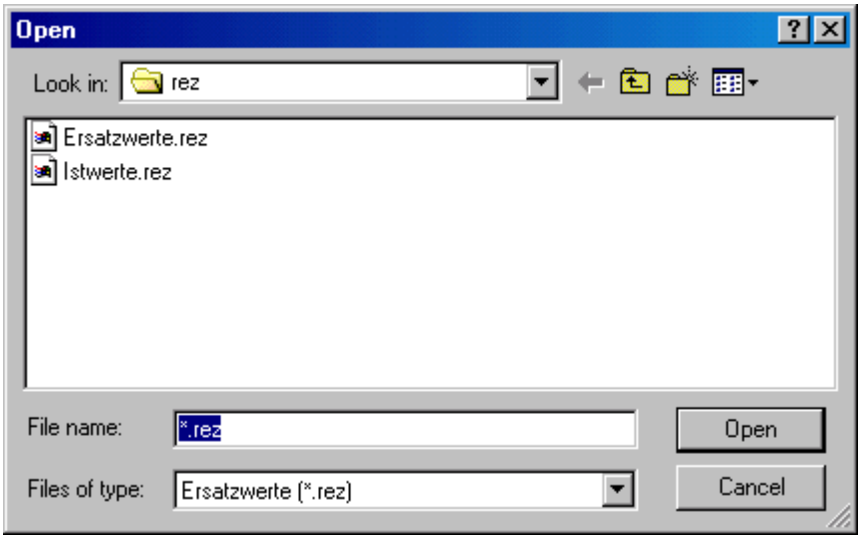
PmosFilePicker program start parameters:

The entry of DMS names (e.g. "System:Prg:LoadData") is case-sensitive.

Three start **parameters must** be included:

LOAD or **SAVE**
Loads data into the DMS or **saves** data from the DMS.

DMS path
"**System:Prg:LoadData**" File specification
Type of file contents, e.g. "default values" data and the prescribed file extension
Example: "**default_values (*.rez)|*.rez**" see the following image:



As a result, the following program call or initialisation comes from a button graphical object "Action > Program Start > Any Program"

PMosFilePicker LOAD System:Prg:LoadData "(.rez)|*.rez"*
------- 1 --------- --- 2 -- ----------- 3 ------------- ------------- 4 ------------------

-1- Program call
-2- Action: **LOAD** or **SAVE**
-3- DMS path specifies where the file will be saved if LOAD is selected.
-4- Description of the file contents (Filter)| File extension

The separator between the text *"default_values(*.rez)* and the template file **.rez"*is made with the "|" character (key combination **<AltGr> + <7>**).

The **file specifications** must be in quotation marks, because the individual parameters are separated by blank spaces and blank spaces or other special characters can usually also be included in the file specification.

11.2.4.1 Remote operation

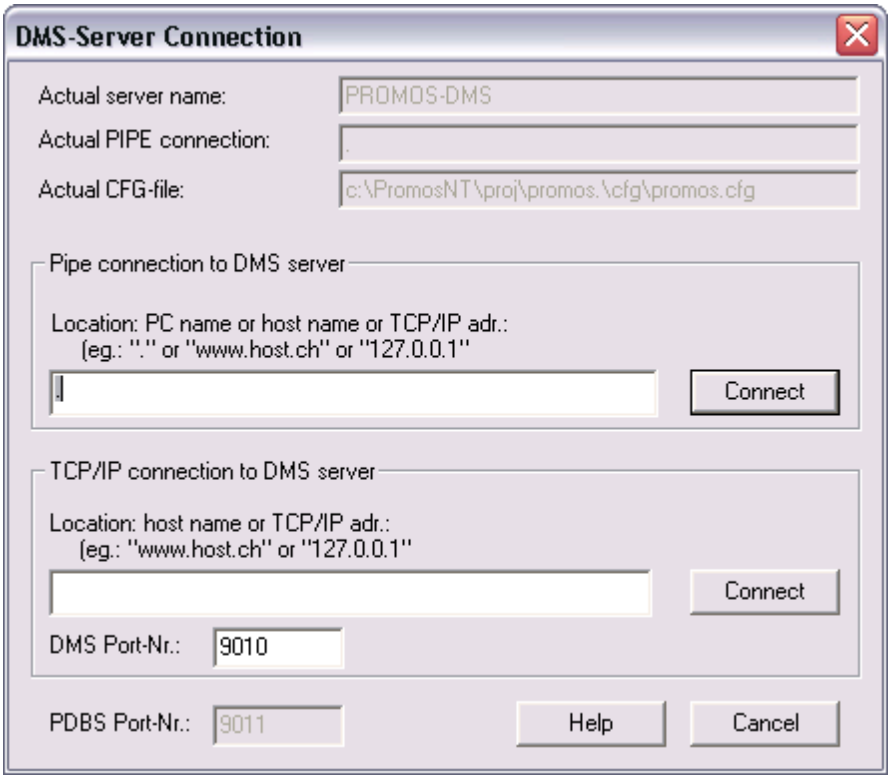
PMosFilePicker can be operated fully remotely, meaning that it can be installed on one or multiple PCs and access DMS data over the network.

PMosFilePicker notes the last-used connection setting and attempts to reestablish this automatically after startup.

Using the command line parameter "**/ASK**" deletes the stored connection setting, meaning that the user is forced to reconfigure the connection setting each time the program is started.

If a DMS is not running on the local PC, i.e. no connection can be established to the DMS, the following dialog is displayed:

Click on "Yes" and the connection dialog appears displaying the Settings and Connection Configuration for the current DMS Server and allowing you to connect to another DMS Server:



The connection can be established via pipe or TCP/IP (sockets):

Pipe: offers a faster connection if SetDMSValue is used within the private DMS network.

TCP/IP: simpler connection configuration; if SetDMSValue accesses the DMS server over the internet (use outside of the private DMS network) and must communicate through firewall(s). In this case, only the DMS port (9010 by default) must be opened in the firewalls.

11.2.5 Example 1: Loading data into the DMS via a button

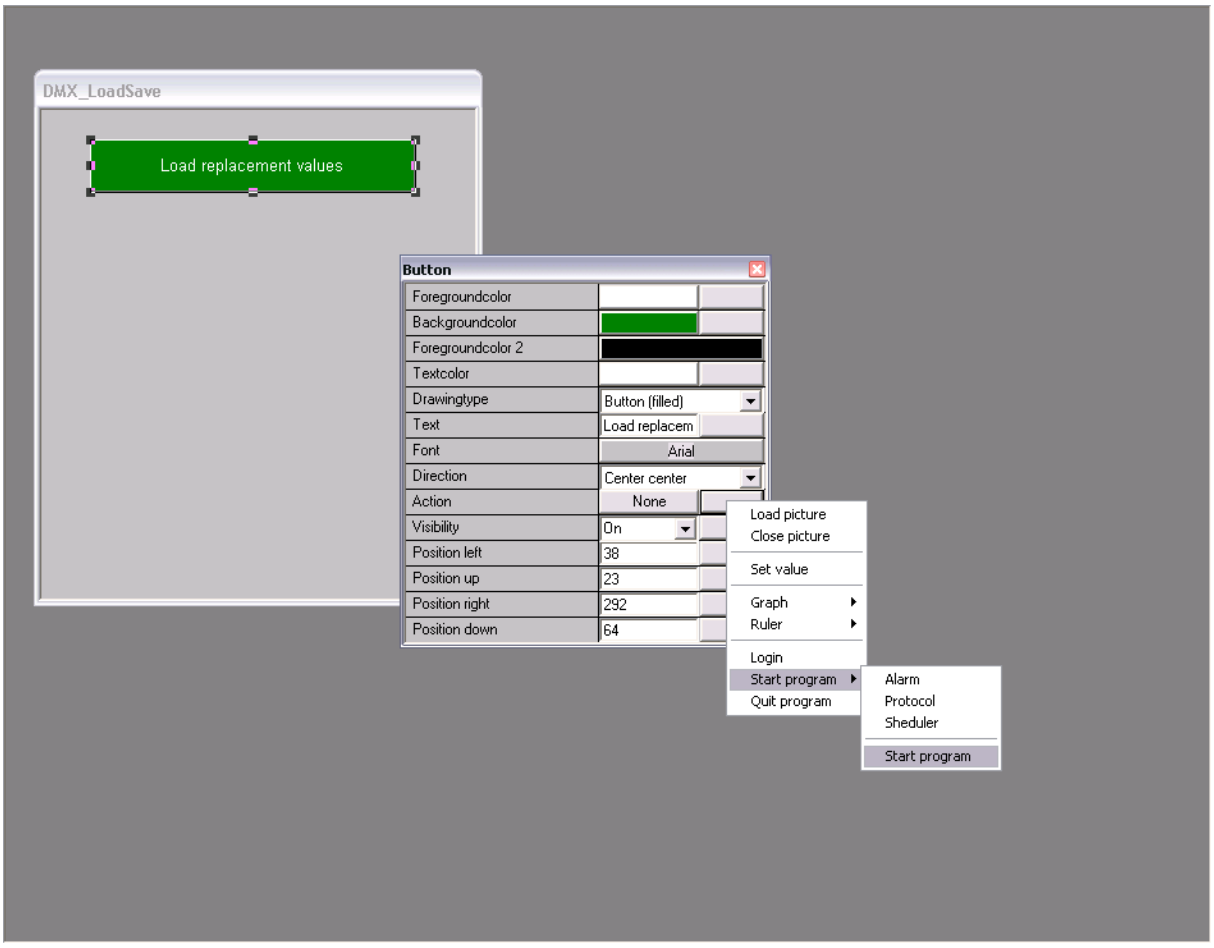
Principle:
Values in data points "BHS60:MT:500:DefaultValue" and **"BHS60:MT:502:DefaultValue"** are to receive new values through a "default value" file.
The "default value file" or recipe files should be easy to locate. Therefore a subdirectory **\rez** should be set up in the current project.
Therefore the complete path will be **"c:\Visi.Plus\proj\<project-name>\rez\default_values.rez"**.

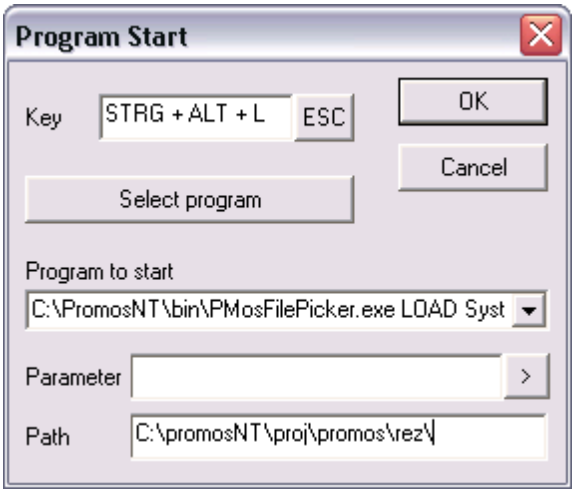
This should be triggered by pressing the button "**Load Default Values**".

To enable monitoring of the value changes, the data point contents must be displayed (see following image).

Procedure:
Create a new process diagram in the **GE** called "**Load Default Values**". Create a "**Button**" with the name "**Load Default Values**" and initialise the switch's "**Action**" attribute.

*If questions arise in regard to the following initialisations, consulting **Chapter 10.2 Graphical Editor (GE)** is recommended.*





In the field below "Program to be Started" the following has been entered, with the underscore "_" used to represent a blank space:

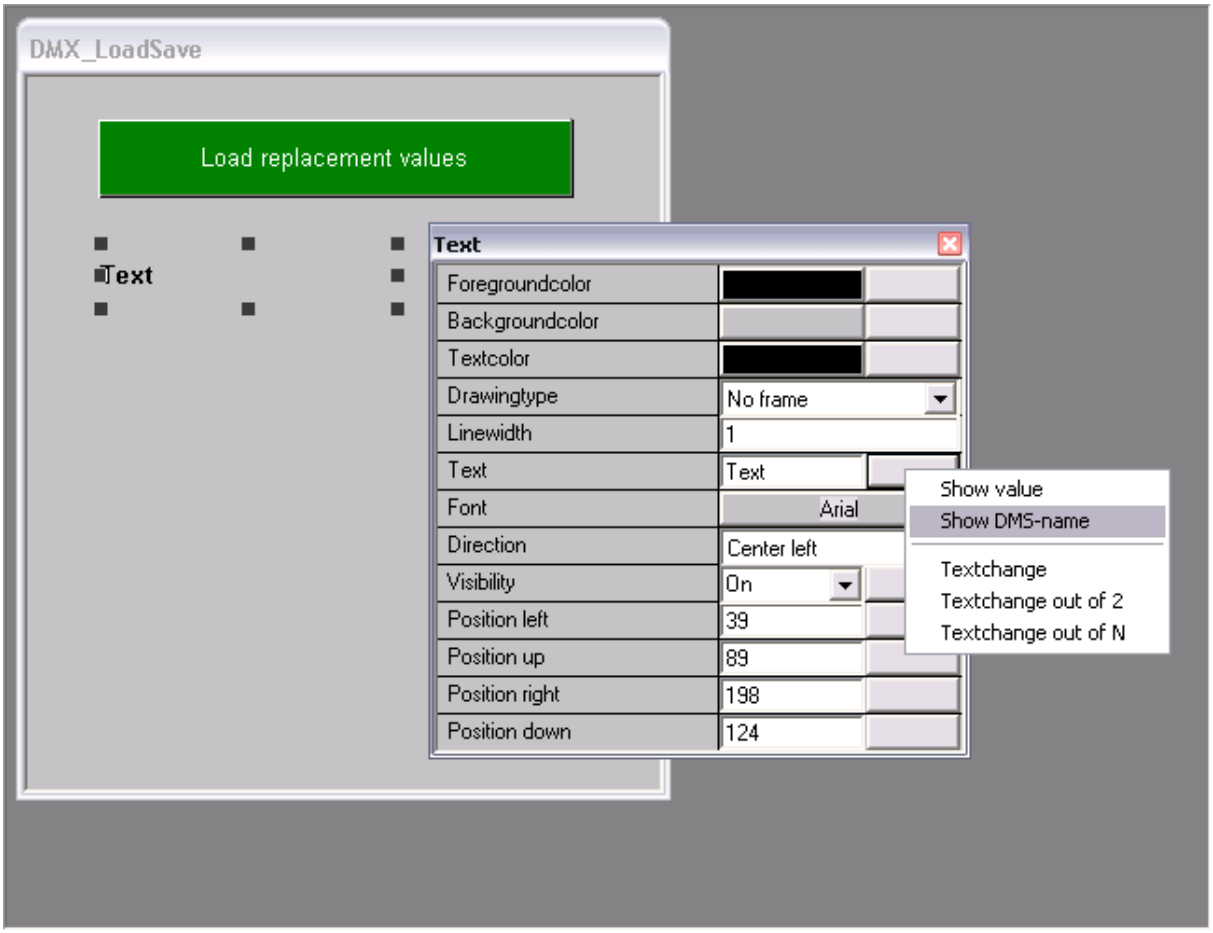
C:\Visi.Plus\bin\PMosFilePicker.exe_LOAD_System:Prg:LoadData_ "default_values_(.rez)|*.rez"*

Fill in the remaining fields as shown and confirm the entries with **<OK>**.

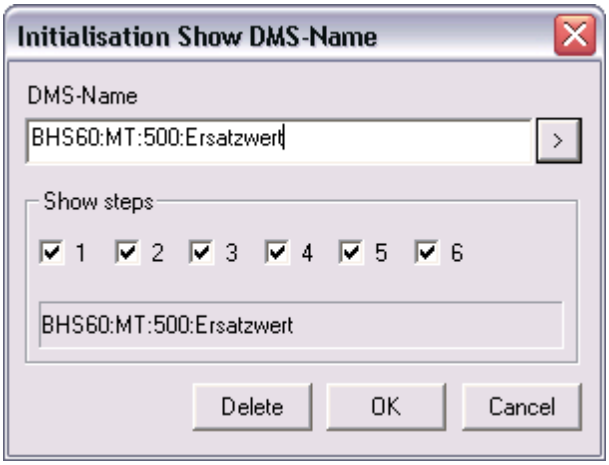
Save the process diagram with  !

Below the button "**Load Default Values**" the DMS data point names "**BHS60:MT:500:DefaultValue**" and "**BHS60:MT:502:DefaultValue**" should be displayed with the help of "Text" graphical objects.

This is done by drawing the "Text" graphical object and then initialising its "**Text**" attribute as follows:

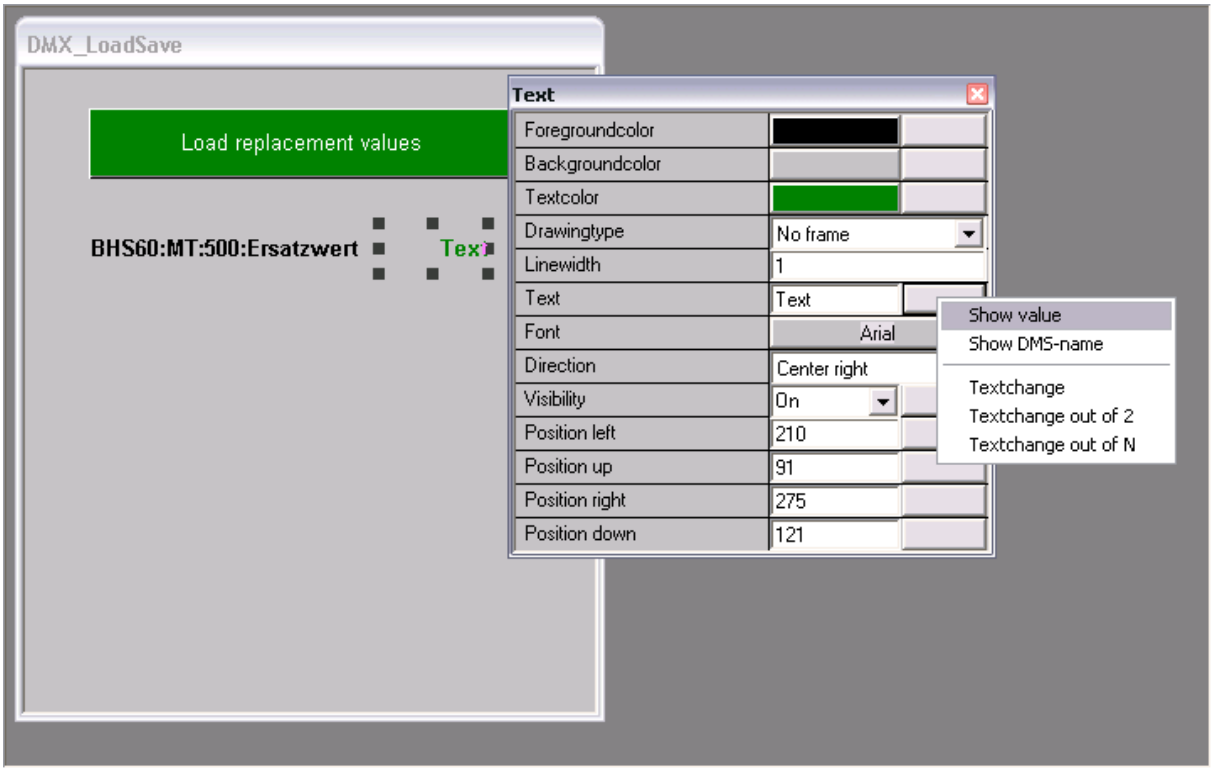


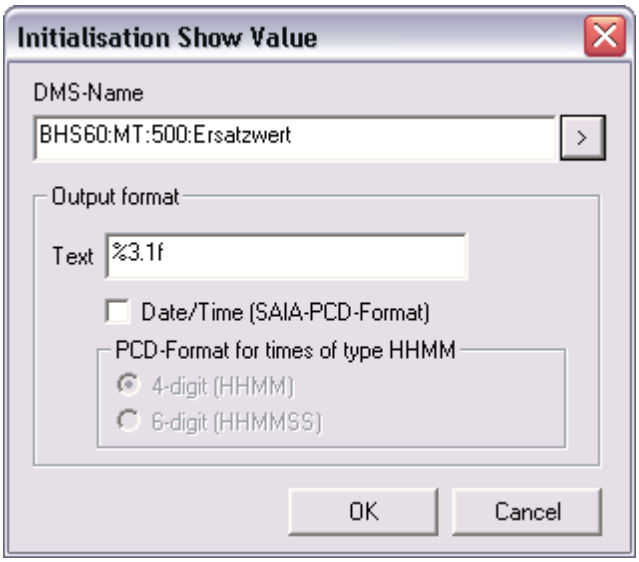
and ...



The contents of the data point should also be displayed with a "Text" graphical object, as already recommended. This is done by initialising the "**Text**" attribute.

and ...

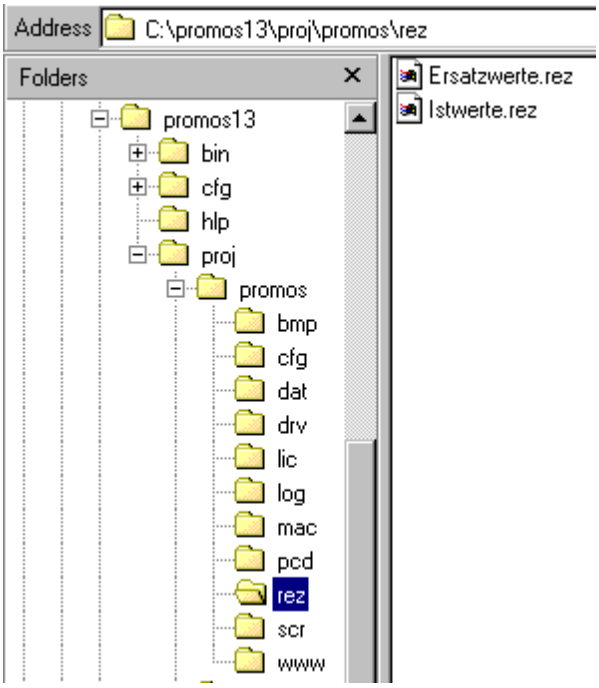




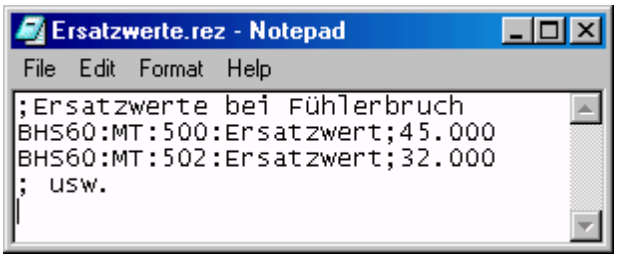
Save the process diagram with  !

The second line with the data point "**BHS60:MT:502:DefaultValue**" should be created in a similar way.

Finally, the file with the data for **loading** into the DMS should be created. In order for the "**default value**" file and all future recipes to be found more easily, a new directory with the name **rez** should be set up in the current project directory (see following image).



In this directory, a file with the name "**default_values.rez**" is created with an ASCII editor or with Notepad. The contents should appear as shown in the following image:



>> Work complete <<

Test follows.

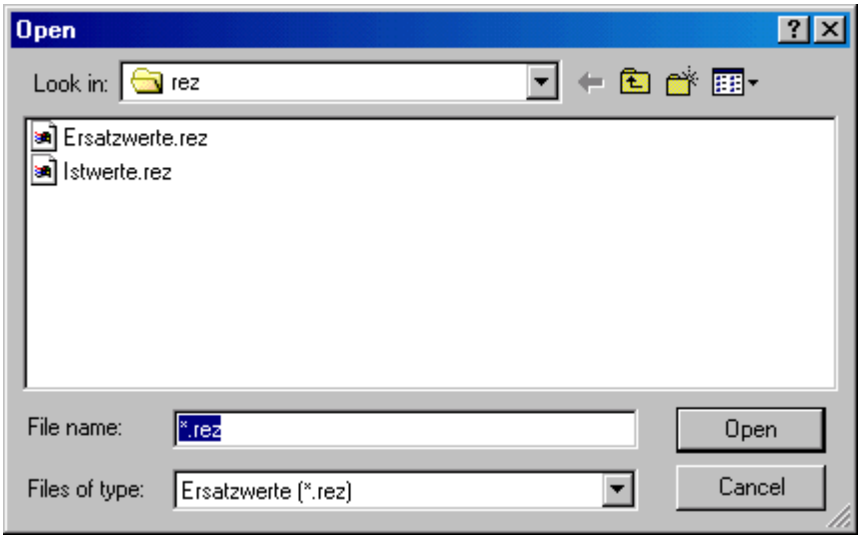
Test of Example 1:

Return to the **GE:** the button "**Load Default Values**" should be tested.

If the **GE** is not already in runtime mode, use the **<e> key** on the computer keyboard to switch to runtime mode.

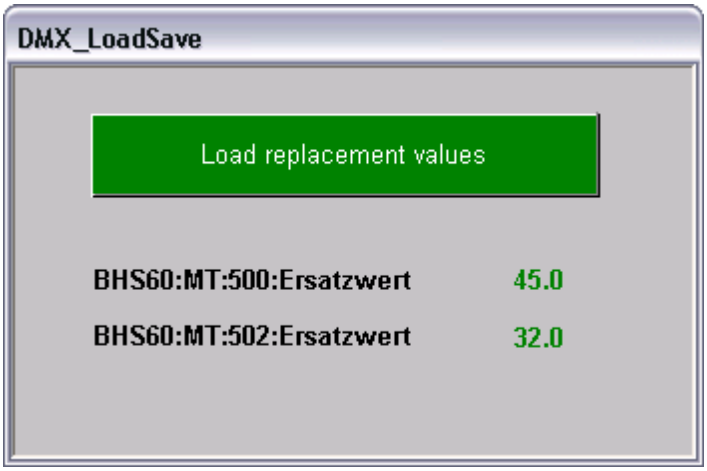
Currently, the contents of the data points "**BHS60:MT:500:DefaultValue**" and "**BHS60:MT:502:DefaultValue**" still show "**0.0**" (or another value from a previous test).

By clicking on the button "**Load Default Values**" the file selection window "**Open**" appears.



After selecting the file "**default_values.rez**" and **<Open>** the data is loaded into the DMS.

The following values should now be displayed:

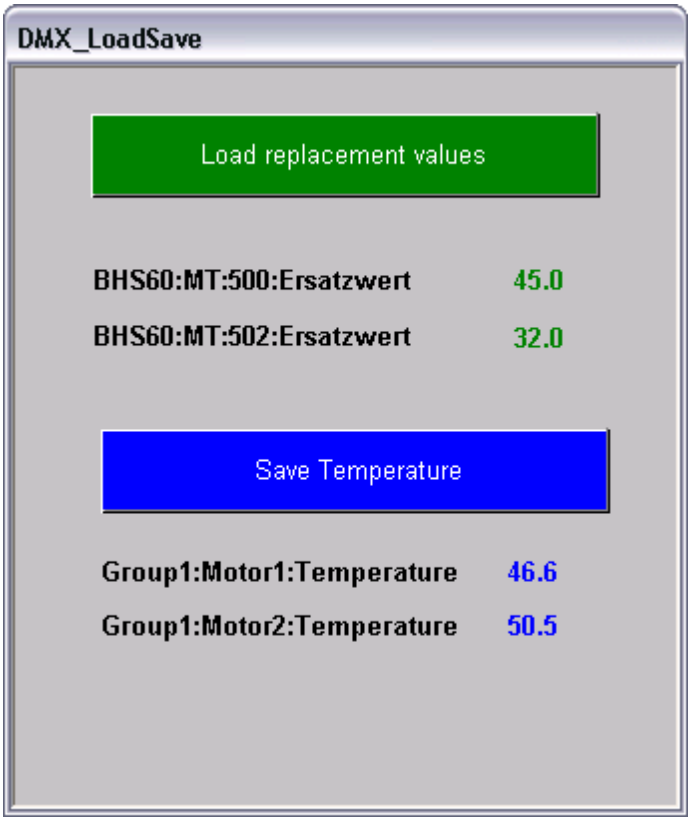


11.2.6 Example 2: Saving data from the DMS

Principle:
Extend the process diagram from Example 1 as follows.

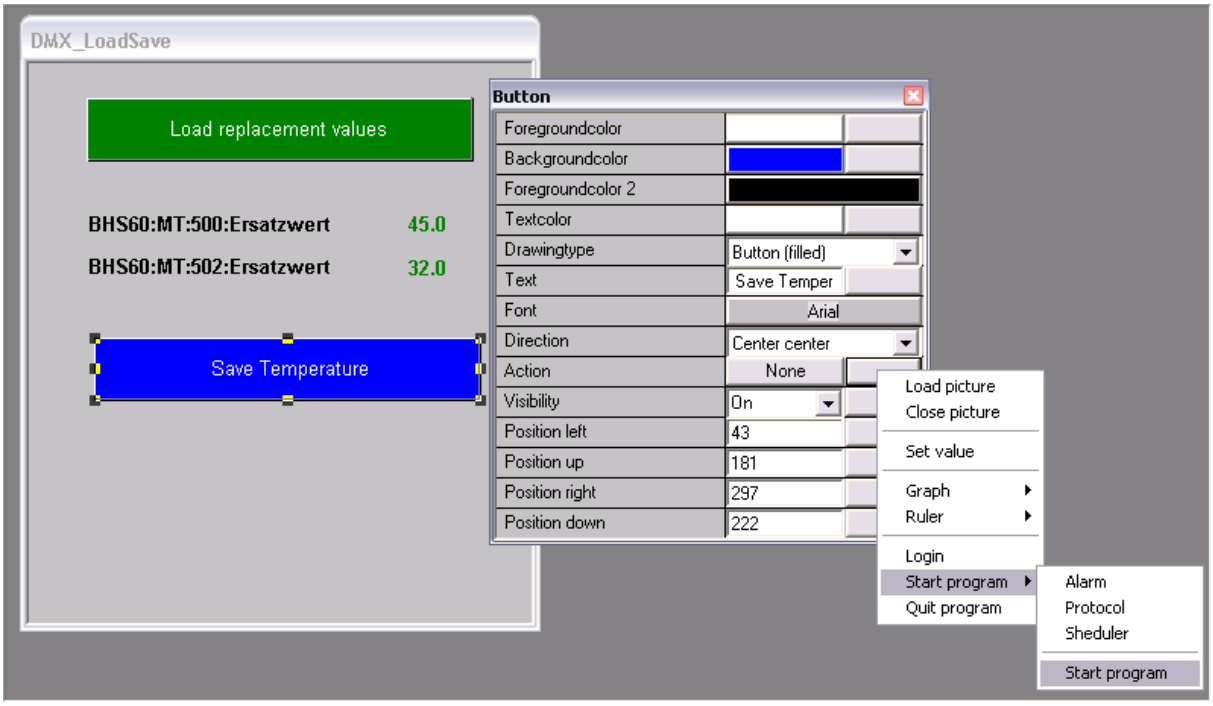
With the "**Save Temperatures**" button, the current values of the data points "**Group1:Motor1:Temperature**" and "**Group2:Motor2:Temperature**" are saved.

For information about the "Group 1:..." simulation data points, see **Chapter** Value simulation (Group1).

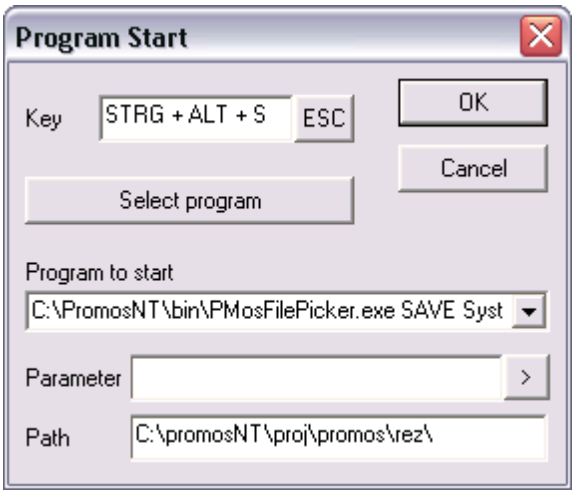


Procedure:
Create a "**Button**" graphical object with the name "**Save Temperatures**" and initialise the button's "**Action**" attribute:

*If questions arise in regard to the following initialisations, consulting **Chapter 10.2 Graphical Editor (GE)** is recommended.*



and ...



In the field below "Program to be Started" the following has been entered, with the underscore "_" used to represent a blank space:

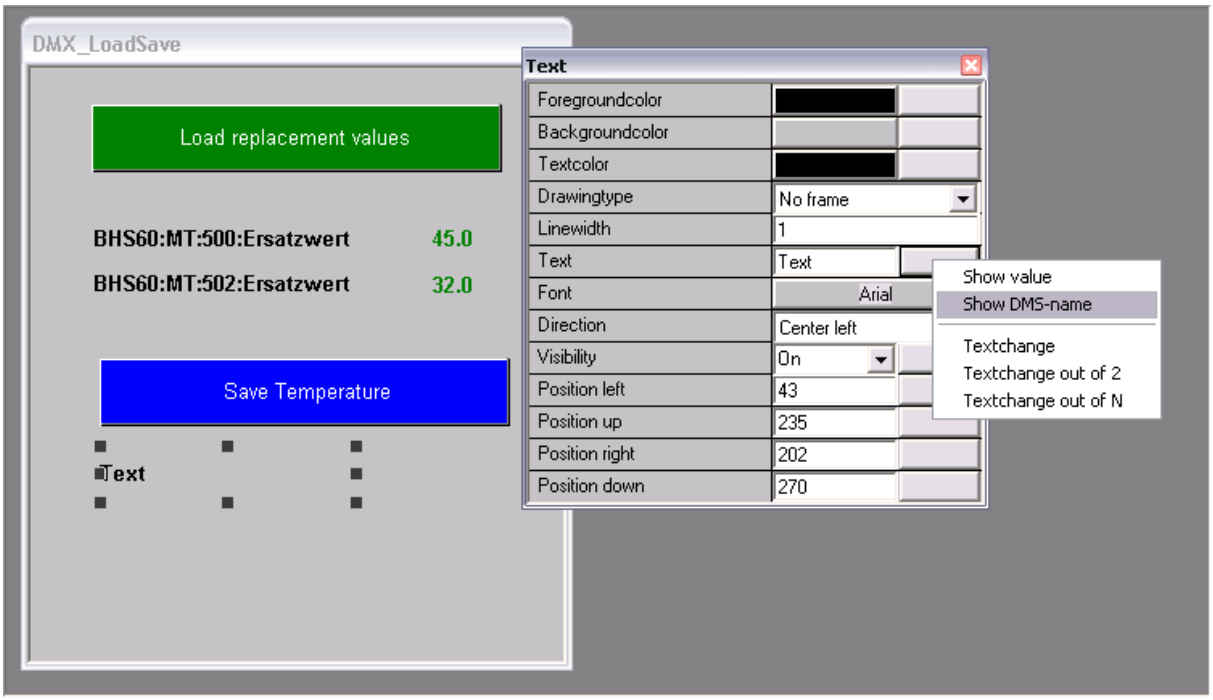
C:\Visi.Plus\bin\PMosFilePicker.exe_SAVE_System:Prg:SaveData_ "actual_values_(.rez)|*.rez"*

Fill in the remaining fields as shown and confirm the entries with **<OK>**.

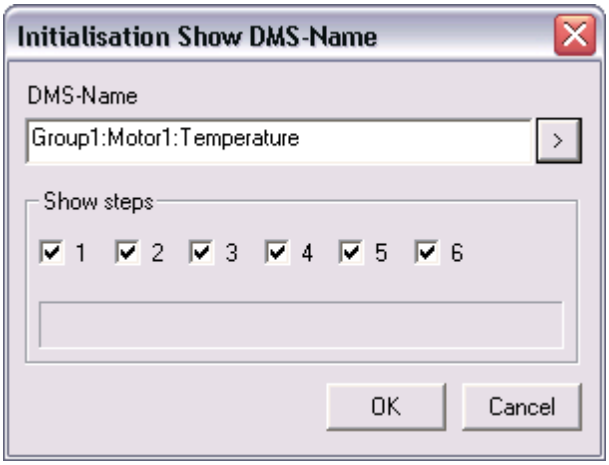
Save the process diagram with  !

Below the "**Save Temperatures**" button, the DMS data point names "**Group1:Motor1:Temperature**" and "**Group2:Motor2:Temperature**" are to be displayed with the help of "Text" graphical objects.

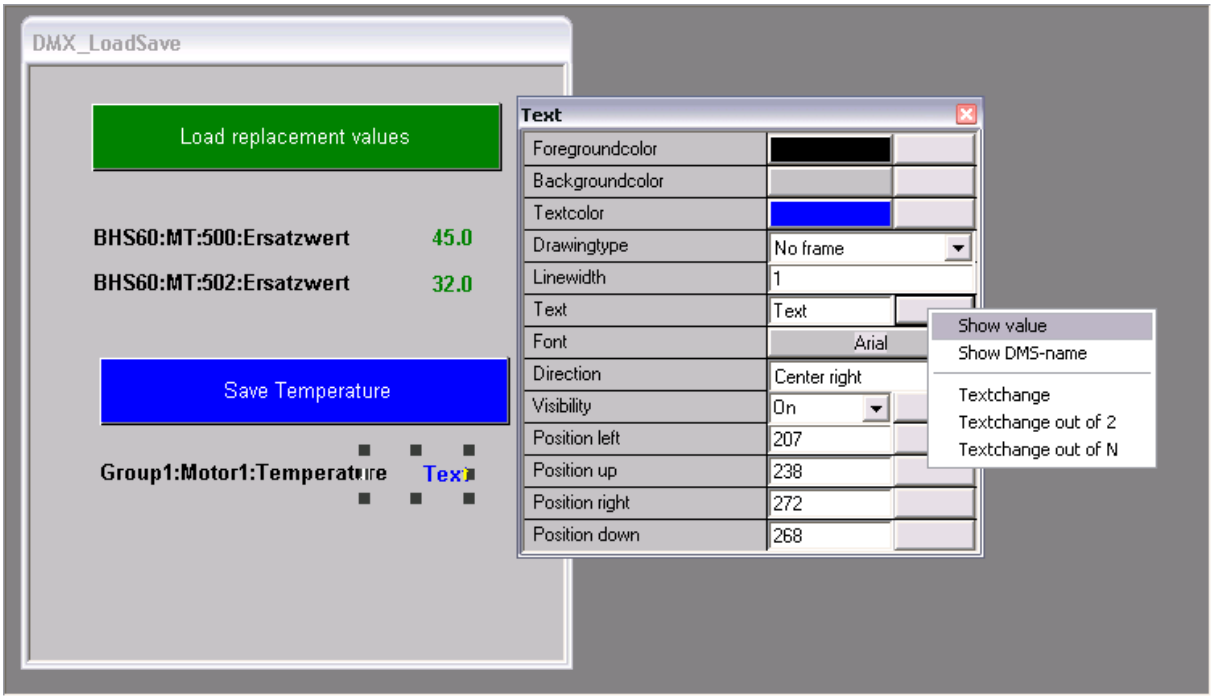
This is done by drawing the "Text" graphical object and then initialising its "**Text**" attribute as follows:

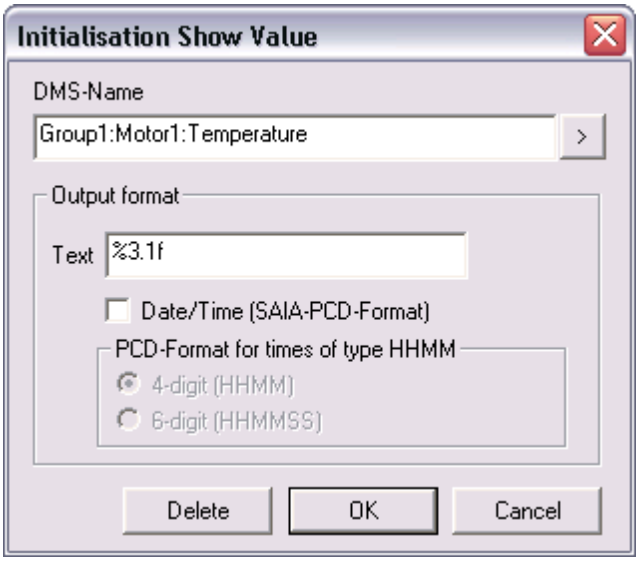


and ...



The contents of the data point should also be displayed with a "Text" graphical object. This is done by initialising the "**Text**" attribute.



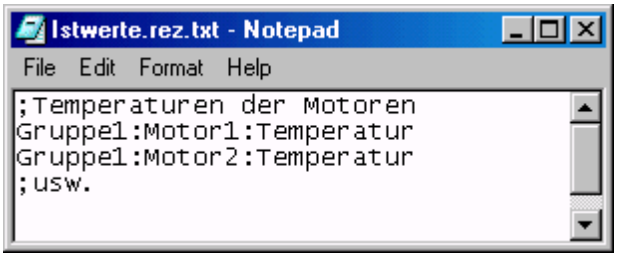


The second column with the data point "**Group1:Motor2:Temperature**" should also be extended in the same manner.

Save the process diagram with  !

Finally, the file with the name "**actual_values.rez**" is to be created with the data point list for **saving from the DMS to a new file**. The save location of the files is in turn the directory **\rez** in the current project directory.

The "**current_values.rez**" file should contain the following data point list:



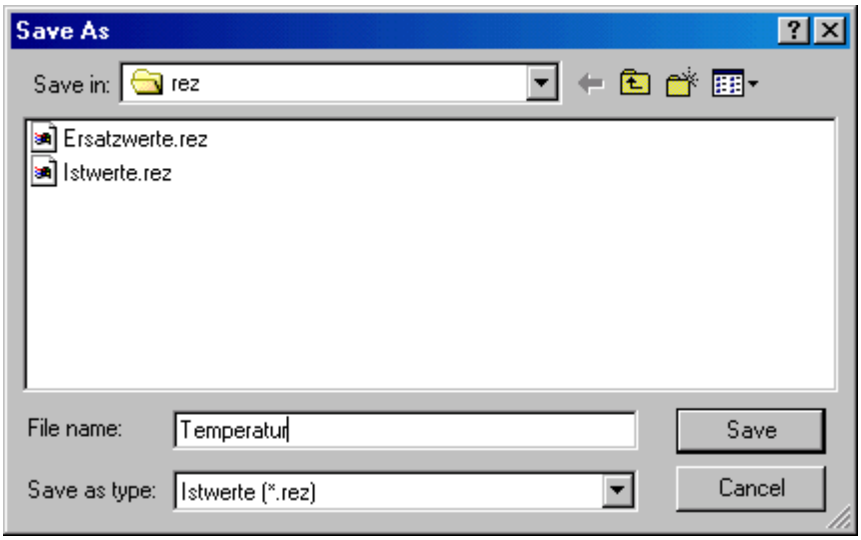
>> Work complete <<

Test of Example 2:

Return to the **GE:** the "**Save Temperatures**" button should be tested.

If the **GE** is not already in runtime mode, use the **<e> key** on the computer keyboard to switch to runtime mode.

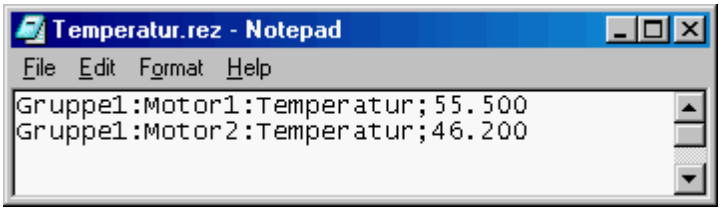
By clicking on the "**Save Temperatures**" button, the file selection window "**Save**" opens.



After entering a filename of your choosing, e.g. "**Temperature**" and pressing **<Save>**, the data is obtained from the DMS and saved to the file "**temperature.rez**".

*The file to be saved **must** have a different name than the file where the DMS names are specified.*

If the file **"temperature.rez"** is now viewed with an editor, the values from the moment in which the "Save Temperatures" button was clicked will appear after a semicolon ";" after the DMS name.



11.2.7 Example 3: Saving data via a command line call from DMS

With the following parameter (WRITE), the **PMosFilePicker** can also be called directly from the command line in **silent mode** (e.g. from individual client programs).

In this case **silent mode** means that no dialogs are opened (for saving or loading files); PMosFilePicker runs in the background.

Command line parameter:
"WRITE <[<DMS-Path>] <Filename>"

Example: "WRITE Text.D" or "WRITE c:\tmp\Text.D".

The path specifications mean either file names or DMS DP string variable, which contain the filename.

<DMS path>: This file contains a list of the DMS names, which specifies the data points which are written to an ASCII file (= <file path>).
<file path>: ASCII file with the DMS DP names and their values.

If a data point from <DMS path> DMS DP list does not exist in actual DMS, then an according message in log file is made (PmosFilePicker.log).

11.2.8 Example 4: Importing data to the DMS via command line call

With the following parameter (READ), the **PMosFilePicker** can also be called directly from the command line in **silent mode** e.g. from individual client programs.

In this case, **silent mode** means that no dialogs are opened (for saving or loading files); PMosFilePicker runs in the background.

Command line parameter:
"READ <DMS path> <file path>"

The path specifications mean either file names or DMS DP string variable, which contain the filename.

<DMS path>: This file contains a list of the DMS names, which specifies the data points which are to be imported from an ASCII file (= <file path>).
<file path>: ASCII file with the DMS DP names and their values.

Example: "C:\Visi.Plus\bin\PMosFilePicker.exe READ xxx:yyy:zzz:Prg:SaveDef c:\Visi.Plus \proj\rugo\rez\addresses.rez"

If a data point from <DMS path> DMS DP list does not exist in actual DMS, then an according message in log file is made (PmosFilePicker.log).

11.3 Format definitions (PRTFormat.exe)

The output for logging and alarm messages is line-by-line.

The content of an output line is defined with the program **PRTFormat**.

Up to 50 different formats (including logs) are possible.

The output texts are implemented by combining various placeholders. Placeholders stand for actual values that are to be inserted later by Visi.Plus.
Thus the placeholder "**#c**" means that Visi.Plus should insert the current **date** and the **time** in the message during output, e.g. to a printer (in the format DD.MM.YY.HH:MM:ss, i.e. 23.09.04 14:15:44).
Possible placeholders are described below.

The following logs and formats are defined as standard:

Logs
Alarms
Manipulations

Formats
Alarms
#c / #N / #V^NAME / #T / #z (On:Off)
Manipulation format
#c / #N / #V^NAME / #z (On:Off) / #u

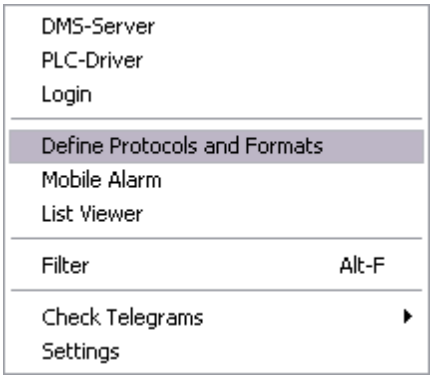
*All **alarms** are saved in the same file "**Alarm.pdb**" in the project directory **\dat**. The file is placed there by Visi.Plus.*

*For each DMS attribute that should be logged, an individual log file "Name.pdb" is placed in the **...\dat** folder of the current project (for information on the creation of a new log, see the description in the chapter "Create new log").*

11.3.1 Starting the program

Requirement:
The DMS must be running!

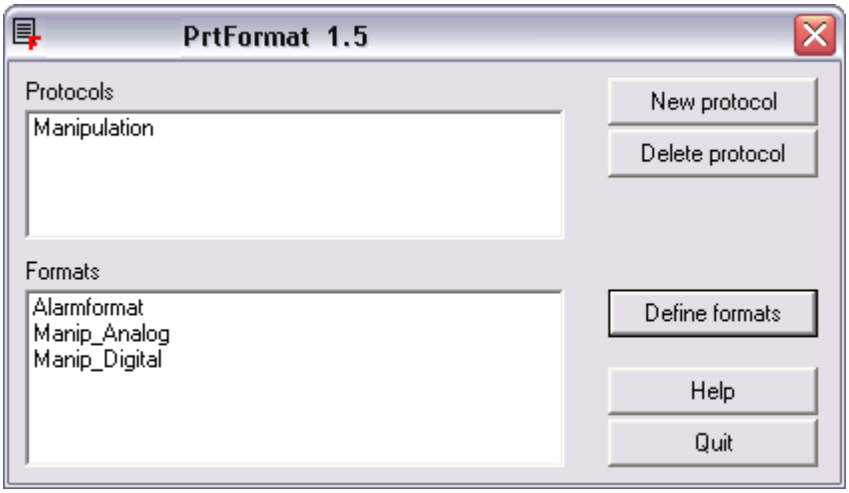
PRTFormat is started in the **PET** via the menu command "**Options > Log and Format Definitions** " .



The program can also be started from the main menu (under Tools).

11.3.2 The PRTFormat user interface

After calling up PRTFormat, the following selection menu appears:

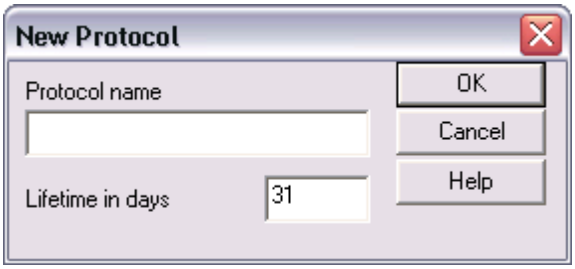


<Exit>
Exits the program **PRTFormat**.

The rest of the button functions are explained in the following pages.

11.3.2.1 Creating a new log

<New Log> opens the following window:



Log Name
Enter the desired name for the log.

Lifetime in Days
This is the duration for which the log should be retained. Older data is deleted from the current file (ring buffer).

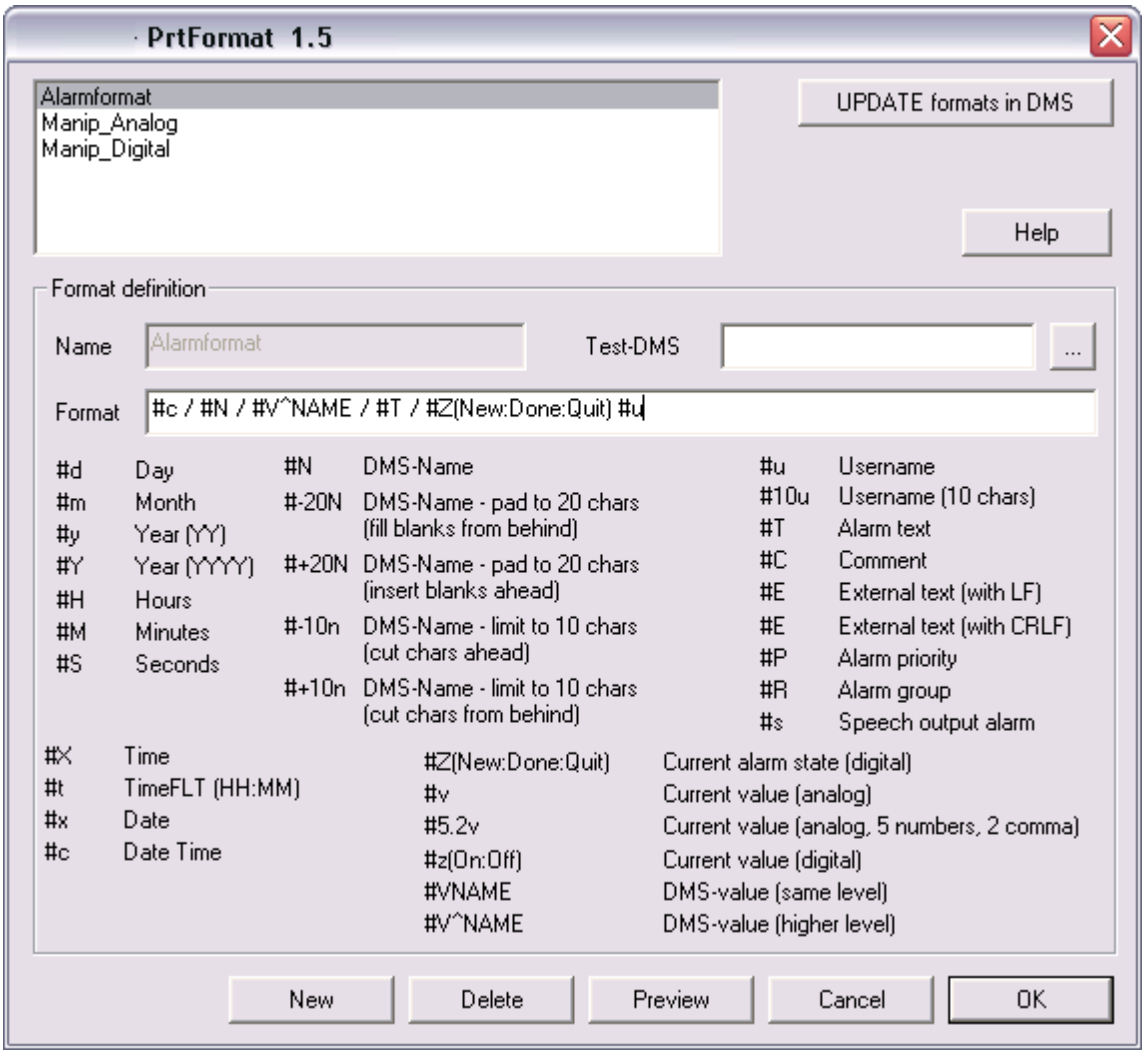
The log name entered in the image above is also the filename under which the log lines are stored, e.g. " **ActualValue.pdb**".
The storage location for all log files is the directory "**..\dat**" of the current project.

A lifetime is defined for the trend data and the logs in the PET (normally 31 days). After expiration of the lifetime, the data is overwritten (function of a ring buffer)! If you require the data to be permanently saved, the backup function in the PDBS is recommended (chapter Backup).

*All occurring alarms are saved by default in the file **Alarm.pdb** in the directory **"..\dat"** of the current project (this cannot be changed).*

11.3.2.2 Define Format

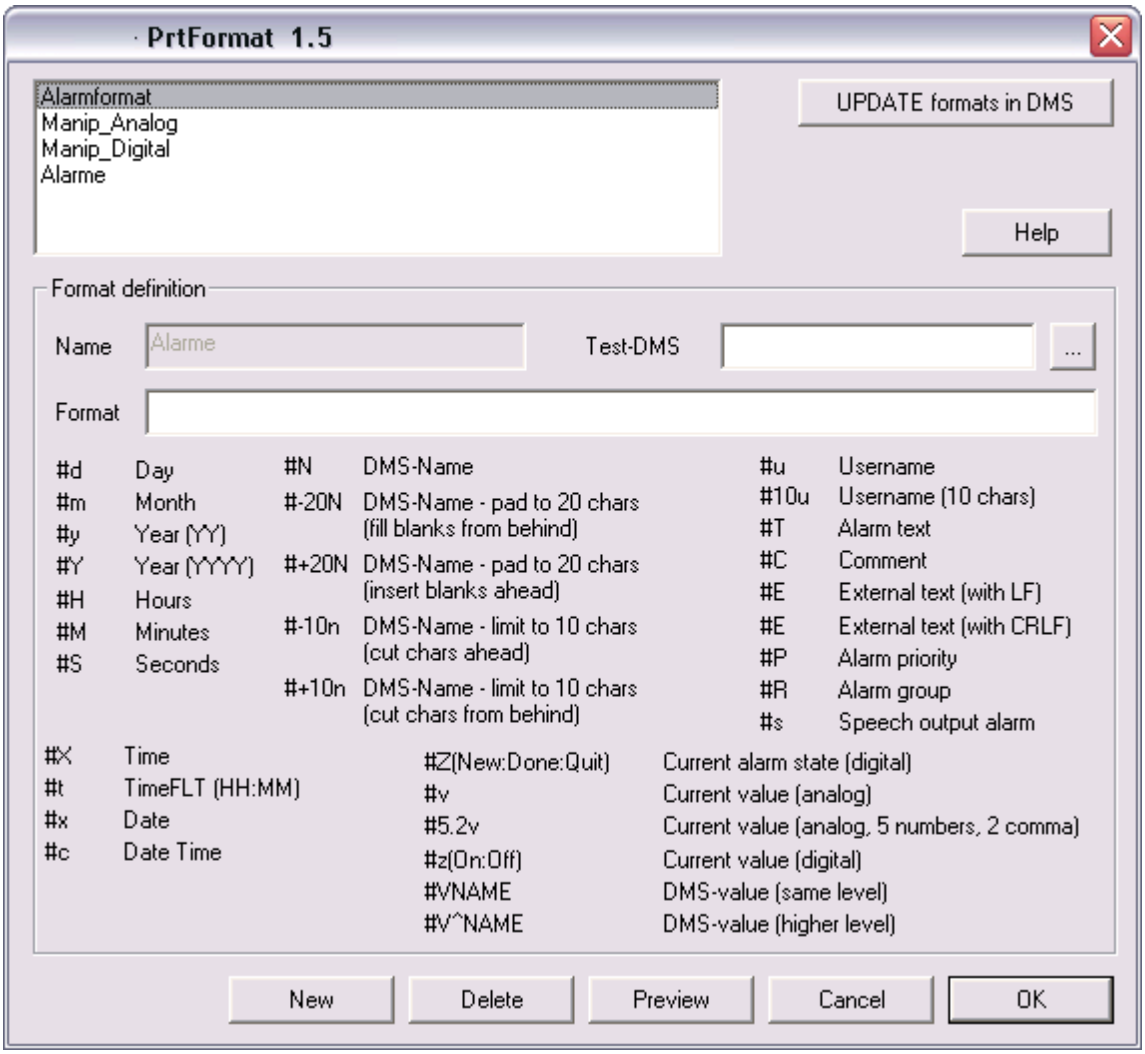
In order to define a format, click on the **<Define Format>** button. The following dialog window opens:



As you can see, a format with the name **Alarm Format** has already been created.

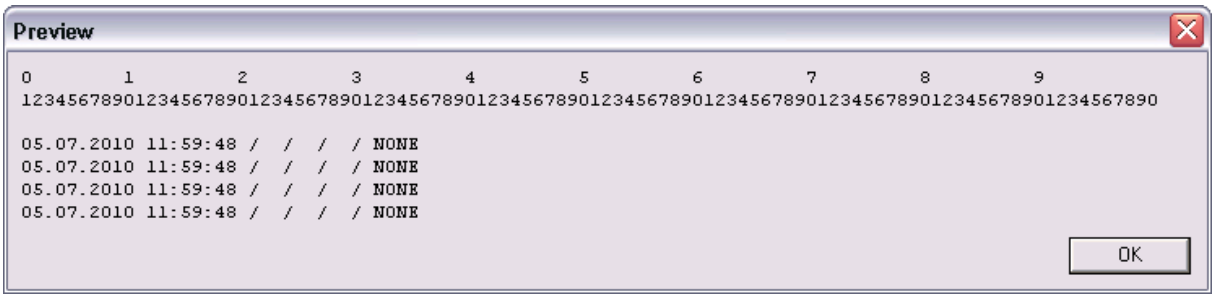
<New>
Creates a new format. In the process, a dialog window appears in which a format name must be entered (see following image).

The dialog window "New Format" displays the name already entered, "**Alarms**".



<Delete>
Deletes a highlighted format definition.

<Preview>
Shows a preview of the definition defined in the input field, based on the address entered under **<Test DMS>**.
e.g. for Test DMS = Test1:CLK:DAY1, Format = #c / #N / #V^NAME / #Z(in:out:ack) #u:



<Cancel>
Closes the "Log Formats" window without saving the current format.

<OK>
Closes the "Log Formats" window and saves the currently entered format.

Name
Displays the selected format against a grey background.

Format
The input window displays the current format and is used for entering changes. In order to assign a format, the format name must be selected (large, white field at the top left)!

Test DMS
see <Preview>

<UPDATE Format in DMS>
Click this button in order to adopt a format change of a format of the same name, already in use by the DMS. The changed format is adopted on the next DMS startup.

A format line can be defined as desired. It consists of **static text** and **placeholders for dynamic values** from the DMS.

In addition to all text characters except "**#**", the following placeholders (preceded by this "**#**" character) are available for format definition:

Format Definitions

Format	Designation	Value Range
#d	Day	01...31
#m	Months as Number	01...12
#y	Year (2 digits)	99,00,01,02,03,...
#Y	Year (4 digits)	2001,2002,2003,...
#H	Hours	00...23
#M	Minutes	00...59
#S	Seconds	00...59
#X	Time (format HH:MM:SS)	14:15:44
#x	Date (format DD.MM.YY)	23.09.03
#t	Time FLT (format HH:MM)	14:15
#c	Date and Time (format DD.MM.YY HH:MM:SS)	23.09.03 14:15:44

Format	Designation	Value Range
#N	DMS name	"System:Group1"
#-20N	DMS name in 20 characters (fill in with empty spaces after name)	"System:Group1 "

#+20N	DMS name in 20 characters (fill in with empty spaces before name)	"System:Group1"
#-10n	Limit the DMS name to 10 characters (truncate beginning)	"tem:Group1"
#+10n	Limit the DMS name to 10 characters (truncate end)	"System:Gro"

Format	Designation	Value Range
#Z(in:out:ack)	An additional short announcement can be issued with an alarm output, which indicates whether the alarm is pending or no longer pending or was acknowledged.	In -> fault received Out -> fault remedied Ack -> fault acknowledged
#v	Current Analogue Value	100
#5.2v	Current Analogue Value 5 digits before decimal point / 2 digits after decimal point	12345.67
#z(On:Off)	Current Digital Value The text that is to be displayed should be indicated in brackets.	Logic 1 -> Text On Logic 0 -> Text Off
#VNAME	Name stands for a DMS extension. The program then displays the DMS content instead of the placeholder. The DMS data point must be located on the same level as the data point that triggered the message or fault.	The content of the Actual Value template object attribute is output.
#V^NAME	As above, only the DMS data point must be on a higher level.	The content of the Actual Value template object attribute is output.

Format	Designation	Value Range
#u	User Name	"Müller Systemtechnik"
#10u	User Name 10 characters	"Müller Sys"
#P	Alarm Priority	3
#R	Alarm Group	4

ESPA format definitions (see also Malm configuration ESPA alarm output)
If a signal is relayed via **Malm** to an ESPA device, the following ESPA-specific data can be defined:

Format	Designation	Value Range

#300=esp_A	Address (=300)	0..9999
#3=esp_B	Beep Code	0..9
#1=esp_P	Priority	0..9

Format definition for the voice output (see also Malm Voice output configuration)
#s
If a signal is conveyed via **Malm** to a voice device, the corresponding wav audio file from the respective signal data point determines, e.g.

data point = **MST:MT:501:Err**, audio file = **MST_MT_501_Err.wav** or **MST_MT_501.wav**

The audio file must be located in the <proj>\wav directory.

If no audio file with the respective name exists, then the contents of a default file "**Default.wav**" is output, which must also be located in the directory <proj>\wav.

Format definition for external text (with LF vertical spacing)
#E
The content of a text file is output or displayed. The text file must be located in the directory <proj>\txt and must have a name based on the the name of the data point in the following manner, e.g.

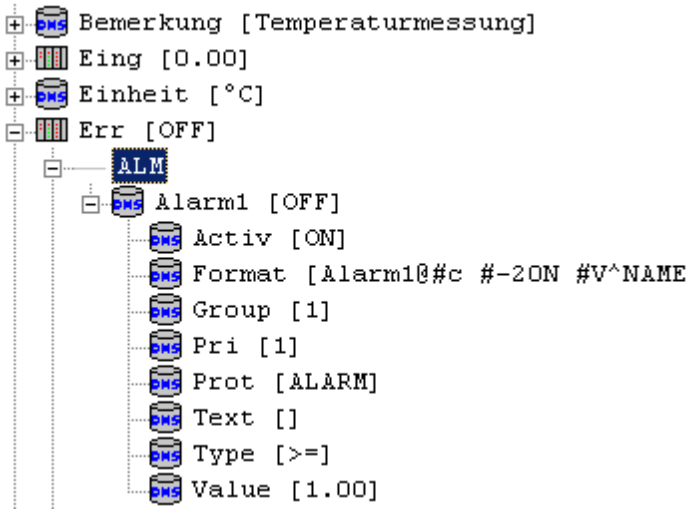
data point = **MST:MT:501:Err**, text file = **MST_MT_501_Err.txt** or **MST_MT_501.txt**

If no text file with the respective name exists, then the contents of a default file "**Default.txt**" is output, which must also be located in the directory <proj>\txt.

Format definition for external text (with CRLF vertical spacing)
#+E
This format is used for special applications, which require vertical spacing with CRLF, e.g. if the alarm is relayed as an e-mail via MalmMng and is then automatically processed by other applications at the client's location.

Otherwise, as above.

DMS Data Point Name
If the data point is not on the same DMS level as the alarm, a **<^>** character can be placed in front of the template object attribute.



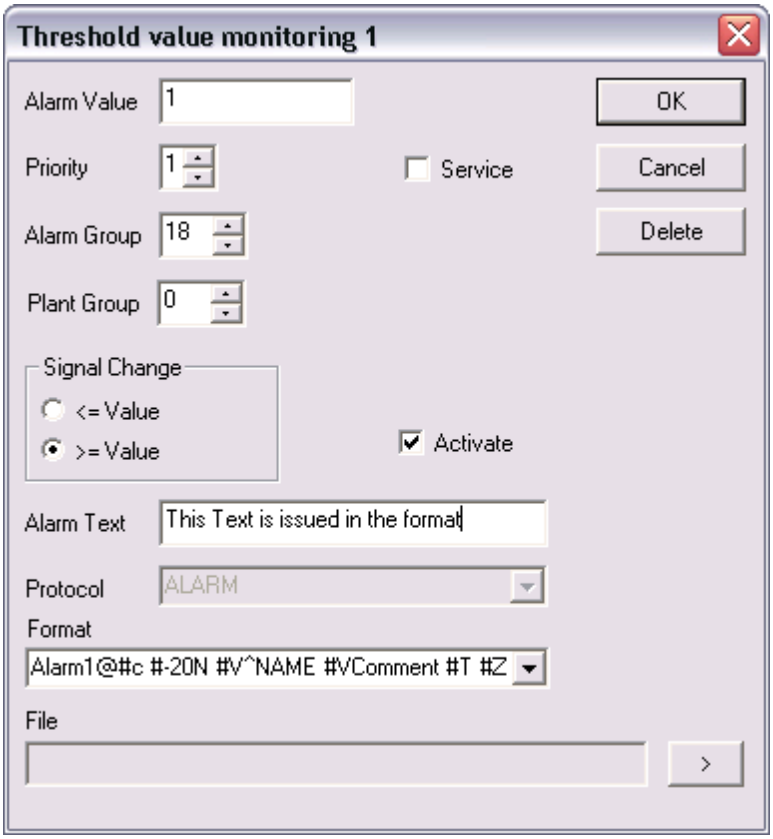
The **Comment** template object attribute is not on the same DMS level as the installed alarm (marked blue).

In order to output the **Comment** template object attribute, the following command must be entered in the format:

#V^Comment
The content of the **Comment** template object attribute is output (temperature measurement).

--- Display of alarm text ---
#T
The text entered in the PET under **Alarm Text** is output.
The stated **Alarm Text** can be found at the following location:
In the PET in the Detail View, click at the level of the attribute in question in the **Alarm** column.
The dialog window "**Limit Value Monitoring X**" appears.
Along with the various alarm settings, the **Alarm Text** input field is to be located and appropriate text should be entered there (following image).

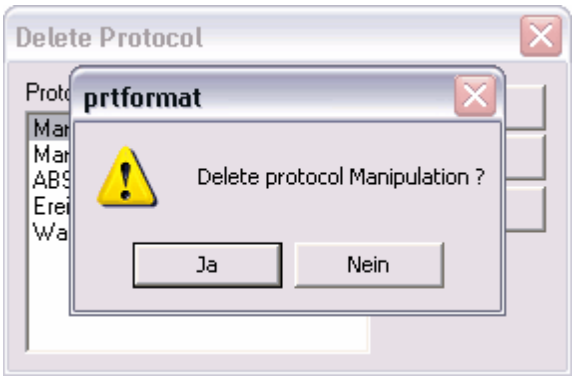
--- Display of comments ---
#C
The text entered in the PET in the Comments column is output:



11.3.2.3 Deleting a log

Select the log and click the <Delete Log> button.

Answer confirmation request:



If logs that have already been used are deleted, they will still continue to be output, because the log is still entered in the DMS.
*Therefore, this means that the entry must also have been deleted in the template object (**VLO**) with the corresponding **attribute** in the **PET**.*

*If logging should only be temporarily disabled, the **Logging Active** checkbox in the*

PET can be switched off.

11.3.3 Assigning a log to a signal

Requirements:
The DMS and PET must be running.

A **log** is assigned to a signal in the **PET** in the **Detail View**

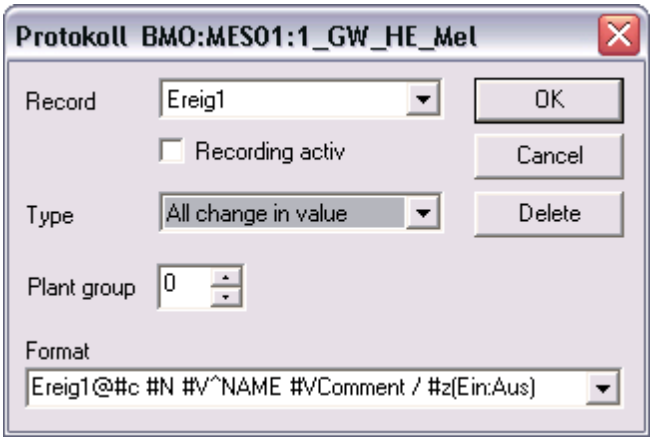

with **VLO View**  switched on.
The advantage in doing so is that: all system objects that are already created are also assigned this log entry.

ATTENTION: *You could actually enter a log from right in the* ***Detail View*** *(all the actual data points are visible,* ***VLO***  *switched off).* ***However, we strongly advise against this!***
If logs are not assigned in the VLO, the log at the assigned attribute can be lost with any change to the VLO.

The **Detail View**, therefore, is switched to **VLO** View!

*If the "**Log**" column is not visible, it can be switched on in the menu under "**View > Log Column**".*

Click in the "**Log**" column on the desired **attribute**. The **Log** control window opens:



The DMS name of the selected data point is displayed in the window title bar.

Log
Which log should be used?

Logging Active
Must be selected in order for the data point to be logged.

Type

For selecting the logging type. The following logging types can be selected from:

- All changes in value
- 0 to 1 changeover (positive flank)
- 1 to 0 changeover (negative flank)

Format
For selecting which format generated in PRTFormat should be used.

<Cancel>
Settings are not adopted

<OK>
Adopts the settings

The symbol 

is then shown in the "**Log**" column in the PET. All template object attributes with the **Log** symbol are logged when their value changes.

*In order for a log to be generated (log file), the program **PrtMng.exe (see this chapter)** must also be started. If the program has not been started, any events which occur are not logged!*

11.3.4 Assigning an alarm to a signal

Requirements:
The DMS and PET must be running.

An **alarm** is assigned to a signal in the **PET** in the **Detail View**

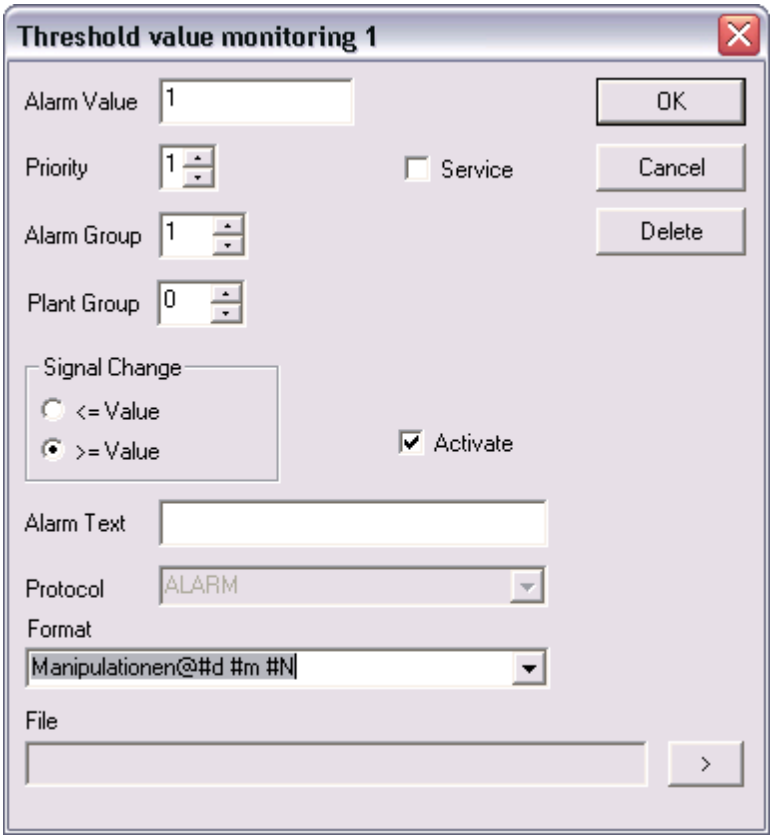

with **VLO View**  switched on.
The advantage in doing so is that: all system objects that have already been created are also assigned this alarm entry.

ATTENTION:** You could actually enter an alarm from right in the **Detail View** (all the actual data points are visible, **VLO**  switched off). **However, we strongly advise against this!
If alarms are not assigned in the VLO, the alarm at the assigned attribute can be lost with any change to the VLO.

The **Detail View**, therefore, is switched to **VLO** View!

*If the "**Alarm**" column is not visible, it can be switched on in the menu under "**View > Alarm Column**".*

Click in the "**Alarm**" column on the desired **attribute**. The **Threshold Monitoring 1** control window opens:



Limit value
At what threshold should the alarm be issued?

Signal transmission
Defines whether an alarm should be issued when the threshold is exceeded or not reached.

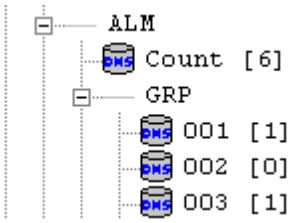
Activated
In order for the alarms to be logged, this checkbox must be marked.

Priority
A priority can be assigned for each data point with an alarm. The priorities have no function in regard to sequence, etc.
These priorities 1-99 are only groupings.
The interpretation of the priorities is therefore to be determined by the programmer.

Example:
Priority 1 = Service technician should come immediately
Priority 2 = Check during the next service.
Priority 3 = No priority
etc.

Alarm group
Has the same function as priority, except that a GRP entry with a group number is made in the DMS for any group number that is assigned (see the following image).

If multiple data points are assigned to the same alarm group, if their alarms are active, they are counted by group and the number of alarms per group is displayed in square brackets after the respective group number in the DMS.



Groups 1 and 3 each display one alarm.
Group 2 has no alarm.

With the option of alarm groups, so-called collective alarms can be generated for each group by initialising, for example, a graphical object accordingly in the GE to "System:ALM:GRP:001" (binary signal).
*The alarm groups are saved in the DMS in the path **System:ALM:GRP**.*

Alarm text
Any text can be entered here. This text can be output in an output format by the parameter "#T" (see PRTFormat).

Log
Always inactive. By default, PET saves all alarms in the log file "**ALARM.pdb**" in the current project directory "**..\dat**".

Format
By clicking on the dropdown menu, all generated format definitions available in PRTFormat are displayed. Select the desired format for the alarm.

<Cancel>
Settings are not adopted.

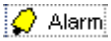
<OK>
Confirms the settings and closes the dialog window.

The symbol Alarm is then shown in the "**Alarm**" column. All template object attributes with the symbol Alarm will trigger an alarm if the threshold is exceeded or not reached, provided that the "**Activated**" checkbox is marked and the Alarm Manager (AlmMng.exe) is loaded.

*In order to transmit an alarm, the program **AlmMng.exe (see this chapter)** must be loaded and/or started up. If the program has not been started, any alarms which occur are not processed!*

*Alarms can be disabled by manually switching the **Active** signal of the respective data point in the DMS to "**0**". All disabled alarms are listed by the program **Revalm**.*

11.3.5 Assigning multiple alarms

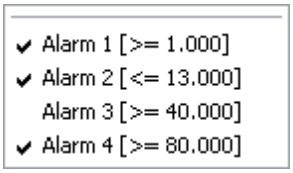
Up to four different alarms can be included with a data point. In order to include an additional alarm with a data point, the existing alarm  must be clicked on in the "**Alarm**" column in the PET. The following dialog window is displayed:

Insert Alarm
Opens the Alarm Configuration dialog.

Alarm 1 [>=1,000]
Selecting this existing alarm also opens the configuration dialog in order to carry out modifications.

After modifying an alarm, the Alarm Manager "AlmMng.exe" must be restarted so that it can import the modified definitions.

Four different alarms have been set up for the data point shown.



The checkmark indicates whether the respective alarm is active or not.
In this example, alarms 1, 2 and 4 are activated.

The defined signal transmission is shown after the alarm number in square brackets.

- Alarms 1 and 4 are triggered when the threshold value is exceeded. Alarm 2 is triggered when the threshold value is not reached.
- Alarm 3 is not triggered, because it is inactive.

11.4 Modify data point content (SetDMSValue.exe)

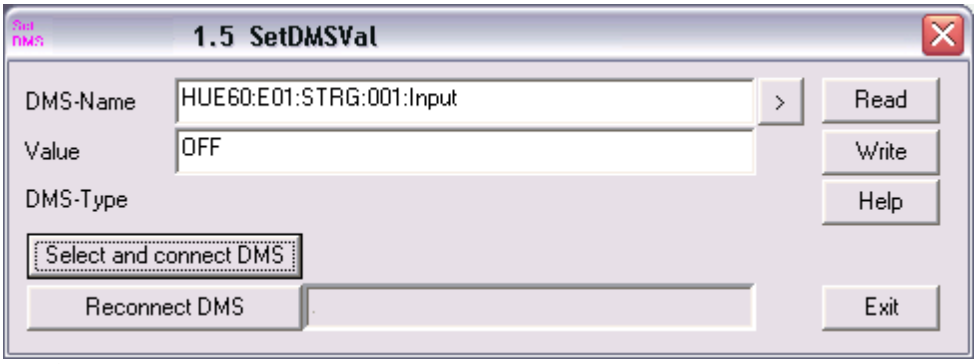
With the program **SetDMSValue**, any DMS value can be read and modified.

TIP: Multiple instances of SetDMSValues can be started in order to have multiple values ready for editing.
Initialise a button in the Graphical Editor for starting this program.

SetDMSValue.exe is located in the default installation directory **c:/Visi.Plus/bin** and can be run from there.

Caution: If a PLC controller is connected, the values are also changed in the PLC when the data point is communicated.

11.4.1 The SetDMSValue user interface



DMS name

For entering or selecting the DMS name with > that should be read or processed.

Value
By clicking on the **<Read>** button, displays the contents of the DMS data point.
By overwriting and then clicking the **<Write>** button, the value can be written to the DMS.

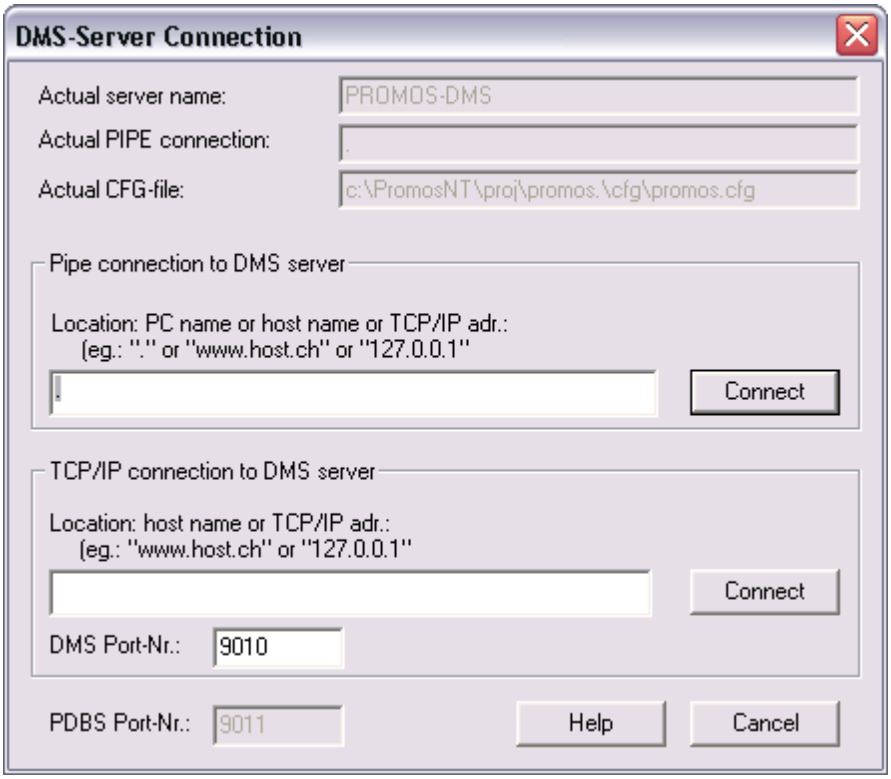
DMS Type
Indicates the data type of the selected DMS data point. If the indicated DMS name is of the BIT type, the <-> button appears.
By pressing this button, the value is changed or reversed from logic 0 to logic 1.

<Read>
Reads the value of the data point from the DMS and displays it in the **Value** field.

<Write>
Writes the value entered in the **Value** field to the selected DMS data point.

<Exit>
Exits the program SetDMSValue.

<Select and Connect DMS>
Shows settings and connection configuration for the current DMS server and enables connection to another DMS server:



The connection can be established via pipe or TCP/IP (sockets):

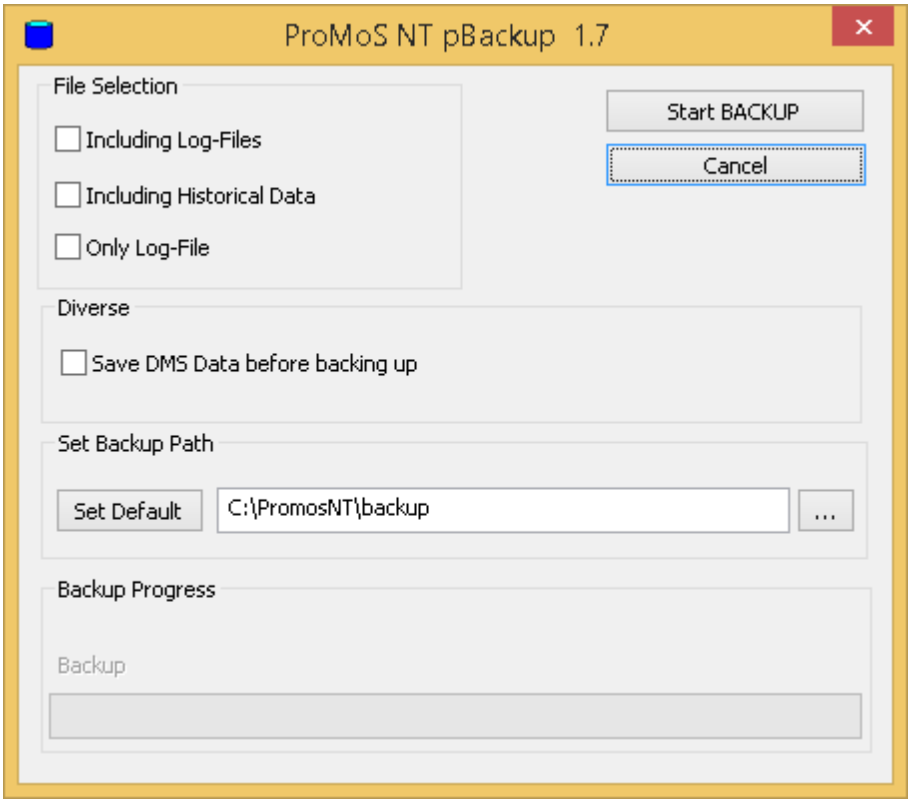
Pipe: offers a faster connection if SetDMSValue is used within the private DMS network.
TCP/IP: simpler connection configuration; if SetDMSValue accesses the DMS server over the internet (use outside of the private DMS network) and must communicate through firewall(s). In this case, only the DMS port (9010 by default) must be opened in the firewalls.

<Reconnect DMS>
In case of an interruption in connection or DMS restart, SetDMSVal can be reconnected.

11.5 Backup (pBackup.exe)

Running this application creates a ZIP file in the standard installation directory **"c:/Visi.Plus/backup"**, which contains the data of the current project and the defined settings of the **<data backup>** in the project manager (see images below).

The program is opened via the standard installation directory "**c:/Visi.Plus/bin**" or a dedicated shortcut.



The required files are saved (as a ZIP archive) after starting the backup.

The following options are available:

- Including log files: A project backup is created that includes the log files that are located in the project (log folder).
- Including historic files: A project backup is created that includes historic files, i.e. all files that are in the dat folder of the project directory. It does not include archived data, i.e. all historic data that is more than one month old.
- Only log files: A backup is created with the log files only. This may be helpful for troubleshooting by Support, for example.

- DMS memories first: The DMS is a database constructed in the RAM. Saving also stores the current data on the hard drive, meaning it is included in the backup. In a current project, we recommend saving the DMS before a data backup.

Any manually created sub-folders and files are also added to the backup.

Path setting:
Enter the path for the data backup here. Click "Default path" to create the backup under

<Installation path>/backup.

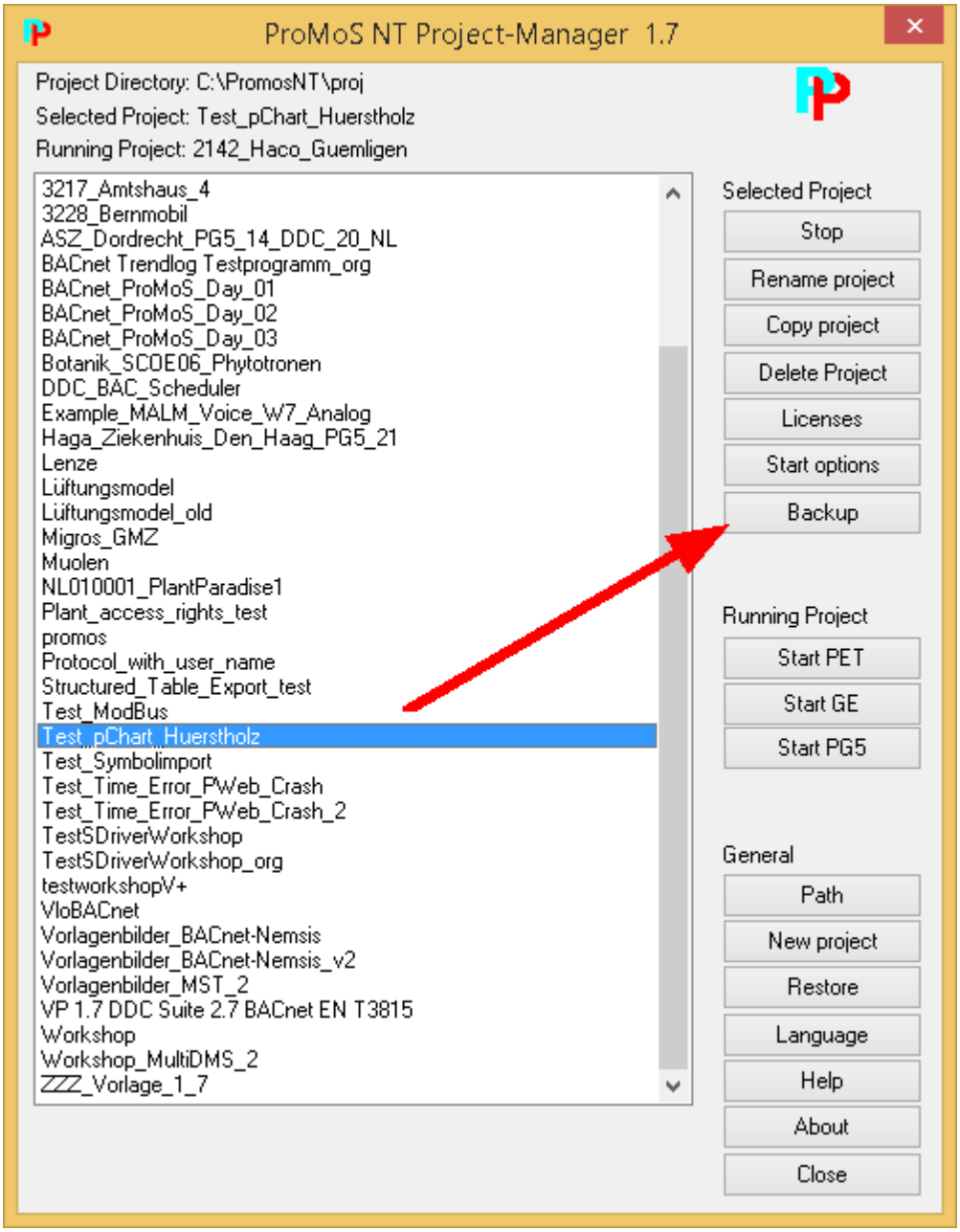
Start backup:
Press this button to start the backup with the entered settings. The backup duration depends on the size of the project, and may take several minutes.

It is possible to add one or more program parameters (arguments):

/LOG	Automatically activates the option for saving the log files
/DAT	Automatically activates the option for saving the historic data
/NOASK	Starts the backup immediately when the program starts (without user input)

If there is no project active or in operation, a backup is created of the last project in operation, together with its backup settings.

The data backup menu can also be opened directly via the project manager:



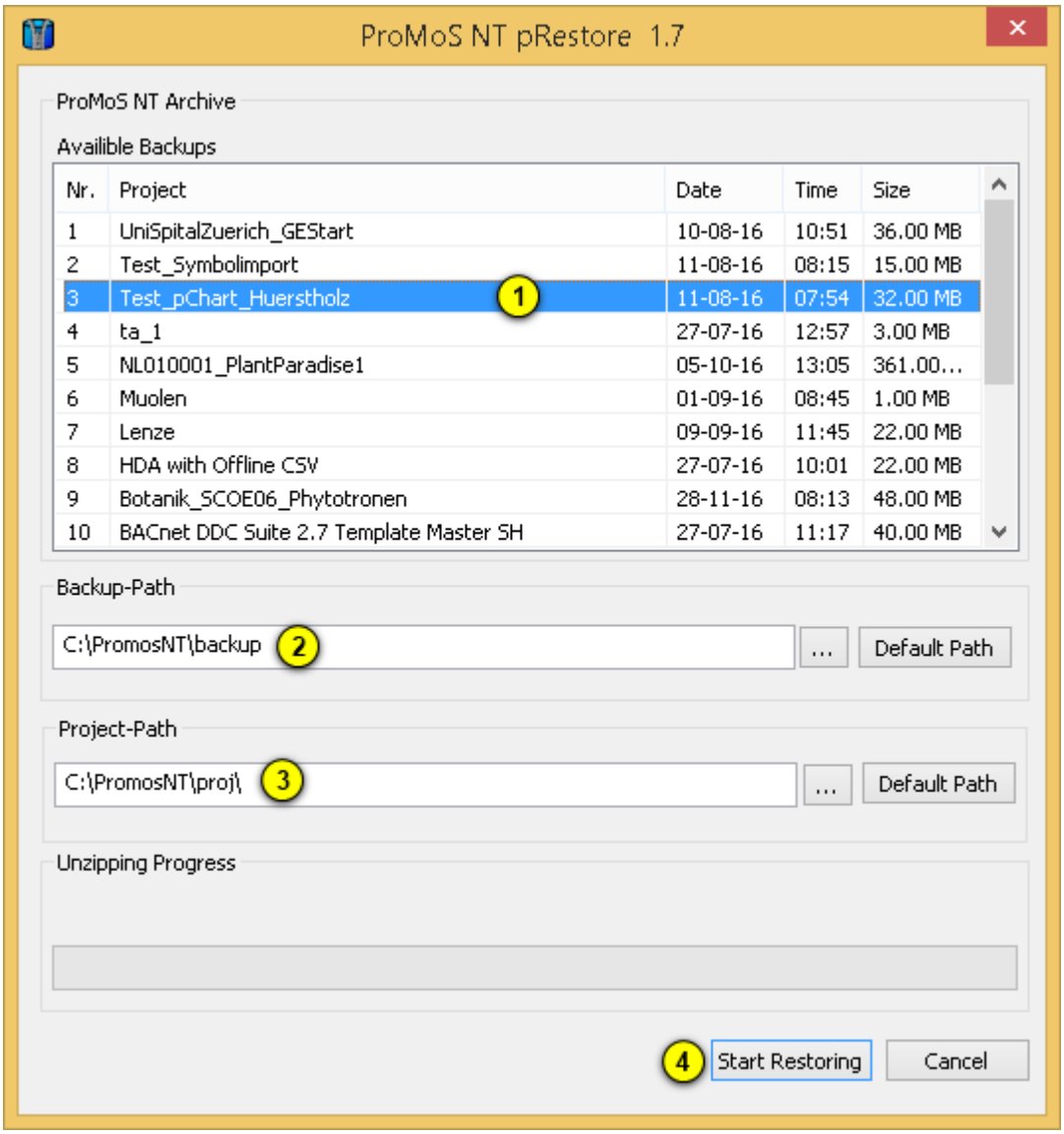
1 Opens the pBackup. See the image above.

The backup is created for the selected project. This does not have to be the current project!

11.6 Restor (pRestore.exe)

Restore (pRestore.exe)

This application unpacks a data backup.



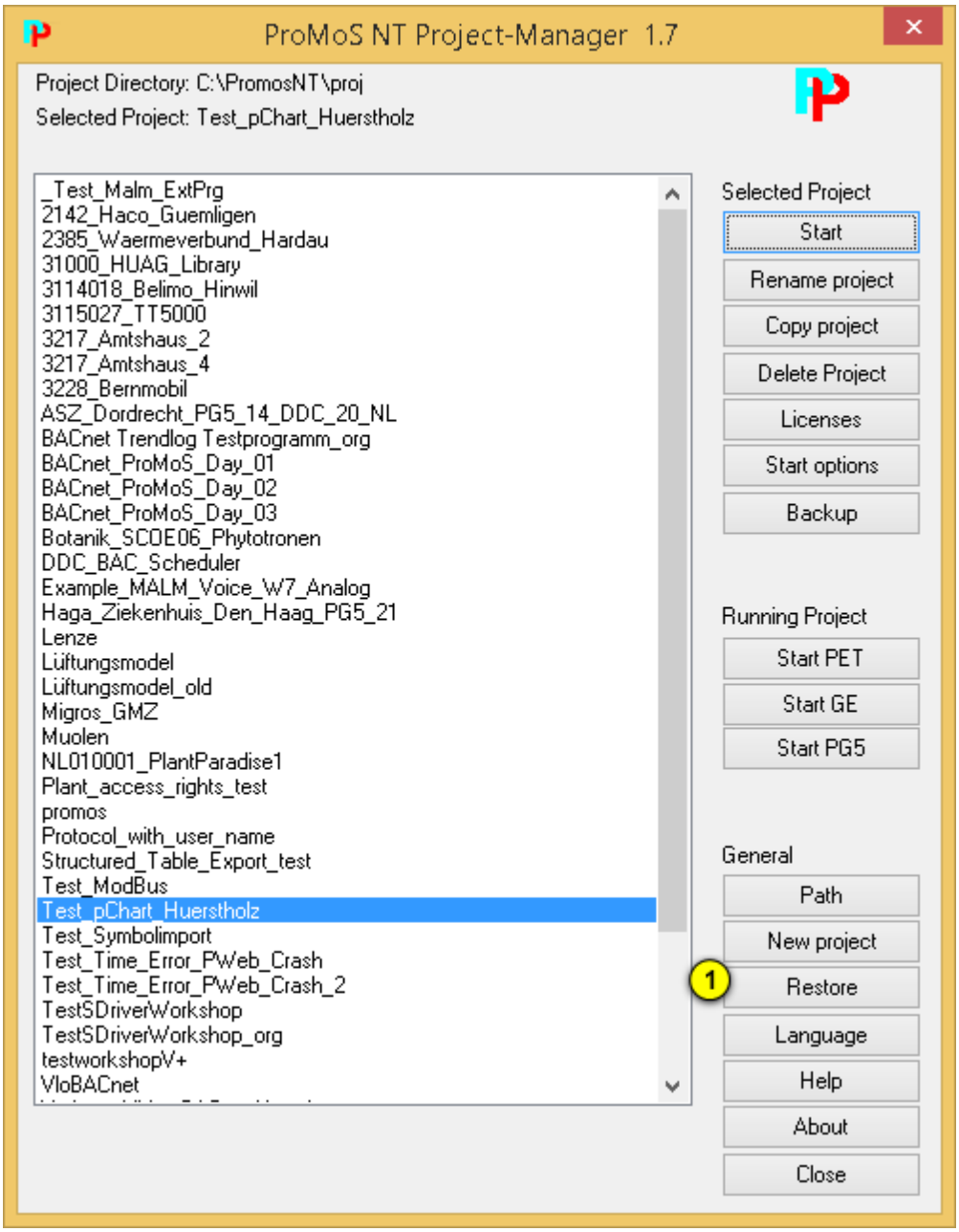
1 Displays all backups in the backup directory. The currently selected backup is highlighted in blue.

2 Backup directory path. Here you can set the location for the backup folder.

3 Project path. Enter the project path here. Unpacking will load the backup in this directory.

4 Button for starting the restore process. This unpacks the backup in the project directory.

pRestore.exe can be started from both the /bin directory of the installation path or directly from the ProjectCfg:



1 Click "Restore" to open pRestore.exe.

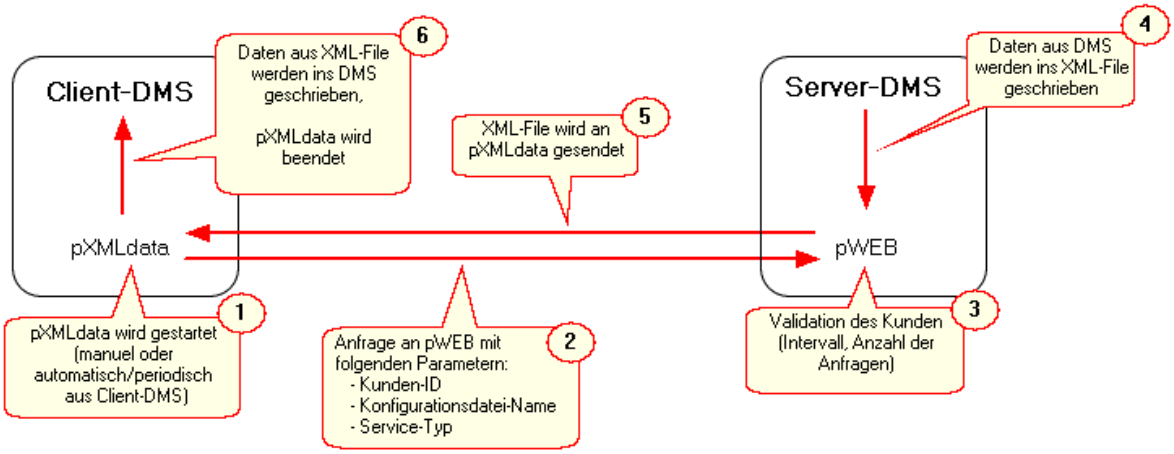
11.7 DMS Data Import/Export (pXMLdata.exe)

pXMLdata is a *command-line program* with no graphic user interface. All settings and control of the data transfer are performed by entering the parameters via the command line.

pXMLdata must be started on the DMS server (= ***client DMS***) which requests data from the remote DMS server (= ***server DMS***) and subsequently performs the **data transfer** between these DMS/Visi.Plus systems. PWEBWebServer must be running on the remote system (***server DMS***).

Function

pXMLdata establishes a connection to the WebServer (PWEB.exe) of the ***server DMS*** and transmits the command and the parameters to it using HTTP protocol.
The same connection is used to immediately return the data in the form of an XML file, which pXMLdata receives and incorporates in the ***client DMS*** and then the connection is terminated.



Parameters

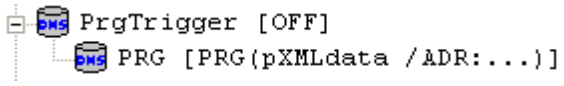
pXMLdata must be started with the following command parameters (pay attention to capitalisation!):

- ***/ADR:10.0.0.39*** = DMS server IP
- ***/PORT:80*** = DMS server port
- ***/PARM:DMSXMLGET.HTM?cust=3D78AE66?srvc=forecast?path=pxmldata.cfg*** = HTTP command string and parameters to the server PWEB
  where:
  cust = 3D78AE68 = Customer ID
  srvc forecast = Type of query: if present, it is used as a sub-folder in the \dat directory.
  path = pxmldata.cfg = Customer configuration file (see below) in the <proj> \cfg folder: Defines the requested DMS data points of the XML client

Client DMS

Starts the data transfer at the program start of pXMLdata.exe + the corresponding command line parameters:

- Manual start of pXMLdata from GE: pXMLdata can be started via the GE program start button, e.g.:
  "pXMLData /ADR:10.0.0.39 /PORT:80 /PARM:DMSXMLGET.HTM?cust=3D78AE66? srvc=forecast?path=pxmldata.cfg"
- Manual start of pXMLdata via the Windows command line; is recommended during the design and debug phases.
- Automatic/periodic start of pXMLdata:
  Can be programmed with a DMS control function, e.g. with EQT on a time counter and trigger on PRG:
  EQT(PrgTrigger, System:Blinker:Blink600) sets the DP PrgTrigger every 10 minutes, which then starts pXMLdata



Server DMS

There must be a configuration file (customer definitions) in the <proj>\cfg folder, which contains the following specifications of the client DMS:

- Client Identification (***cust*** - parameters, see above): every customer (Client) has their own section, e.g.: **`[3D78AE66]`**
  incl. additional customer information (optional):
  `Customer`, **`Location`**.
- DMS data points definitions which are transmitted to the client in the form **`<DMS-Server DP-Name>=<DMS-Client DP-Name>`**
- Time specification of the last successful data transfer (date/time stamp):
  `LastUpdate`
- Time specification indicating how often the client can request the data, in minutes (1440 = 1x daily): **`Interval`**
- Total number of successful data transfers (for statistical purposes, billing):
  `Count`

Example of a customer section (Customer ID = 3D78AE66) in the configuration file pxmldata.cfg:

```
.....
[3D78AE66]
Customer=MST Systemtechnik AG
Location=3123 belp
Interval=1440
LastUpdate=25.03.10 11:41:24
Count=123
Error=OK
Weather:3D78AE66:Forecast:DAY_1:Radiation=Weather:DAY_1:Radiation
Weather:3D78AE66:Forecast:DAY_2:Rain=Weather:DAY_2:Rain
Weather:3D78AE66:Forecast:DAY_3:Text=Weather:DAY_3:Text
.....
```

11.7.1 Sample Data Transfer Procedure Between two DMS Servers_2

1. The client requests data from the server:

via GE button, program start e.g.:

"pXMLData /ADR:10.0.0.39 /PORT:80 /PARM:DMSXMLGET.HTM?cust=3D78AE66?srvc=forecast?path=pxmldata.cfg?port=80"

where:

ADR: 10.0.0.39 = Server IP
PORT: 80 = Server Port
cust: 3D78AE68 = Customer ID
srvc: forecast = Type of query; if present, it is used as a sub-folder in the \dat directory.
path: pxmldata.cfg = Customer definition file (see below) in <proj>\cfg folder, contains DMS definitions of XML clients
port: 80 = XML Client Port

2. From this, the server determines the cfg file <path> and customer section <cust> contained there.

3. From this customer section, the server reads all key lines with the following specifications:

- Interval=1440 ; in minutes, only one access per day here
- Count ; number of correct client accesses = number of data deliveries
- Error ; last error message, e.g.:
  "XMLcust[3D78AE66] ERROR: data not sent,interval not passed min=[1440], still remaining=[1421]"
- <DMSServer>=<DMSclient> ; DMS definitions

4. The server checks whether the last access occurred longer ago than the minutes specified in the interval.

If not, then an XML file is created without DMS data but with the corresponding status:
<status>XMLcust[3D78AE66] ERROR: data not sent,interval not passed min=[1440], still remaining=[1439]</status>

5. The server reads the requested DMS data points

writes them to an XML file, incl. DMS client names and DMS type and a time stamp.
In addition, a <deliverydate> field with the current time is created as well as <nextdelivery> = current time + interval.
The XML file is saved under the following name in <proj>\dat\<srvc>\<cust>.xml before being sent.
If the folder \dat\<srvc> does not exist, it is created.

6. The XML file is sent to the client

at the same IP address from which the query came, port = <PORT> (from the command line of the query).
If the transmission is OK, then the count is incremented in the customer section of the <path> (pxmldata.cfg) file and the date/time stamp LastUpdate is refreshed.

7. The client detects the command line

and extracts the necessary information:
cust: 3D78AE68 = Customer ID
srvc: FORECAST = Type of query; if present, it is used as a sub-folder in the \dat directory.
path: 3D78AE68.xml = The XML file to be received with the requested DMS data of the server.

8. Subsequently, the client receives the XML file from the server

and writes it to <proj>\dat\<srvc>\<cust>.xml.

9. The client parses the XML file

The following DMS data are determined: DMS name, type, value and date stamp.
This information is encoded in the XML file in the following manner:
Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<data>version="1.5.1" link="http://www.mst.ch"
<status>OK</status>
<deliverydate>26.03.2010 12:39:14</deliverydate>
<nextdelivery>27.03.2010 12:38:57</nextdelivery>
<set>
      <parameter name="MeteoForecast:DAY_1:Radiation">
            <value datetime="2010-03-12 00:00:00" type="FLT">4094.000000</value>
      </parameter>
      <parameter name="MeteoForecast:DAY_3:Text">
```

```
                <value datetime="2010-03-14 00:00:00" type="STR">SnowShowers</value>
        </parameter>
        <parameter name="MeteoForecast:DAY_2:Radiation">
                <value datetime="2010-03-13 00:00:00" type="FLT">3817.000000</value>
        </parameter>
</set>
</data>
```

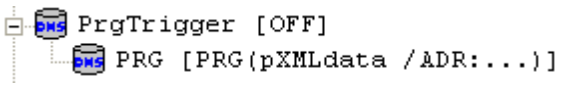
All DMS data points are created new corresponding to the type and the value is saved.

All data transfers are carried out by the web servers on the server and client sides. All XML events are logged by the the web server and are also displayed in the PWEB Debug window. There are also additional log options in the PWEB menu, log settings Dlg.: Log Send/ Receive XML data details.

11.7.2 Automatic/periodic start of pXMLdata

Automatic/periodic start of pXMLdata
Can be programmed with a DMS control function, e.g. with EQT on a time counter and trigger on PRG:

EQT(PrgTrigger, System:Blinker:Blink600) sets the DP PrgTrigger every 10 minutes, which then starts pXMLdata



Manual start of pXMLdata via the Windows command line
This is recommended during the design and debug phases.

11.7.3 DMS data import from XML file

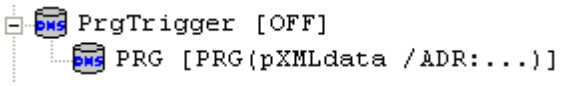
pXMLData can read XML files and write them to the DMS. It can be called via the command line using the "READ" command:

Example: "**pXMLData /READ:c:\Visi.Plus\test.xml**".

The XML file format follows the same guidelines as described in the previous chapter:
Example:If test.xml has the following content:

```
<?xml version="1.0" encoding="UTF-8"?>
<data>
      <set>
            <parameter name=”F909_R1:R01:PROTEL:TARGV">
                  <value type="FLT">2</value>
            </parameter>
            <parameter name="F909_R1:R01:PROTEL:ROOM">
                  <value type="STR">F001</value>
            </parameter>
            <parameter name=”F909_R1:R02:PROTEL:TARGV">
                  <value type="FLT">0</value>
            </parameter>
.....
      </set>
</data>
```

Then the result is:

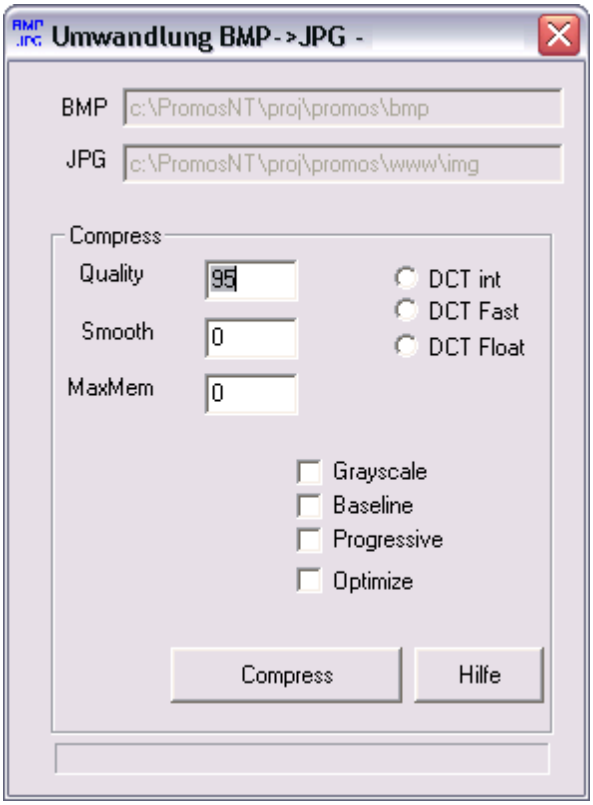


11.8 Image Conversion (bmp2jpg.exe)

With bmp2jpg.exe, entire directories of bitmaps can be converted to JPG images. This might be done to prepare system images for display on the web or for other purposes.
The program can be opened from the default installation directory "**c:/Visi.Plus/bin**" or via your own shortcut.

Initialise a button in the Graphical Editor for starting this program.
For further information in regard to conversion of images, please consult a specialist website or the appropriate literature.

When the **<Compress>** button is pressed, the conversion is executed and the window is closed.



BMP:
Stands for the directory in which the bitmap images to be converted are located (input).

JPG:
Stands for the directory in which the finished, converted JPG images are stored (output).

These two directories are predetermined and cannot be adjusted or changed. The currently set project is adopted by default.

11.8.1 Available settings

11.8.1.1 Quality

The adjustment of quality is a compromise between file size and quality: The higher the quality, the larger the file.

Normally a setting is chosen with which no noticeable reduction in quality is recognisable. Values between 50 and 95 are well suited for this: Begin with the base setting of 75 and change it in increments of 5 or 10 until you are satisfied with the results.
The optimal adjustment, however, can vary from one image type to another. Settings above 95 are not recommended for normal use, because the file size increases drastically with only minimal improvement in quality.

Settings below 50 create very small files, combined with poor quality.
Settings between 5 and 10 are only suitable for overview images and the like.

Please observe that settings with quality below 25 generate a 2-byte quantification table that cannot be read by all JPG decoders (in order to avoid this, also select the baseline format).
We recommend that only experienced users use this option.

11.8.1.2 Smooth

If required, you can smooth the image in order to suppress noise. With a reduction in colour, errors are inevitable.
You have the possibility of adjusting the smoothing from 0 to 100, where 0 (default setting) indicates **no smoothing**.

Smoothing is often helpful when saving 256-colour images.
A factor between 10 and 50 generates smaller and better images.
A factor that is too large, however, will make the image blurry.
We recommend that only experienced users use this option.

11.8.1.3 MaxMem

Here you have the possibility of setting the limit of the maximum memory to use, which may be necessary for larger images.

We recommend that only experienced users use this option.
For further information in regard to conversion of images, please consult a technical website or the appropriate literature.

11.8.1.4 DCT methods

There are multiple methods available for calculation of the **DCT** (**D**iscrete **C**osine **T**ransformation): The **floating** DCT method is much more precise than the **integer** method,

but also much slower, provided that your computer does not have a very fast **floating** point processor.

Please also observe that the **floating** DCT method can lead to different results on different types of computer, whereas the **integer** method should have the same result on all computer systems. The **fast integer** method is much less precise than the other two methods.

We recommend that only experienced users use this option.
For further information in regard to conversion of images, please consult a technical website or the appropriate literature.

11.8.1.5 Grayscale

Generates monochrome images typically from colour images. If you select the **grayscale** checkbox, you receive a smaller image in terms of the amount of data, which in turn demands less time for processing.

11.8.1.6 Baseline

Generates a JPG file in **BaseLine** format. That means the image is represented sequentially.

11.8.1.7 Progressive

Generates a **progressive** image file, which means that the image is contained in the file as multiple individual images of increasing quality. This can be beneficial if the file is to be transferred through a slow connection. Therefore the recipients receives an image at an early stage, even if it is in poor quality, which improves as the transfer progresses. The finished file contains the image in the same quality as a **non-progressive** file and the size of the file is approximately the same – the **progressive** file is usually even somewhat smaller.

*Please note, however, that not every JPG decoder can process **progressive** files.*
We recommend that only experienced users use this option.
For additional information regarding conversion of images please consult a specialist website.

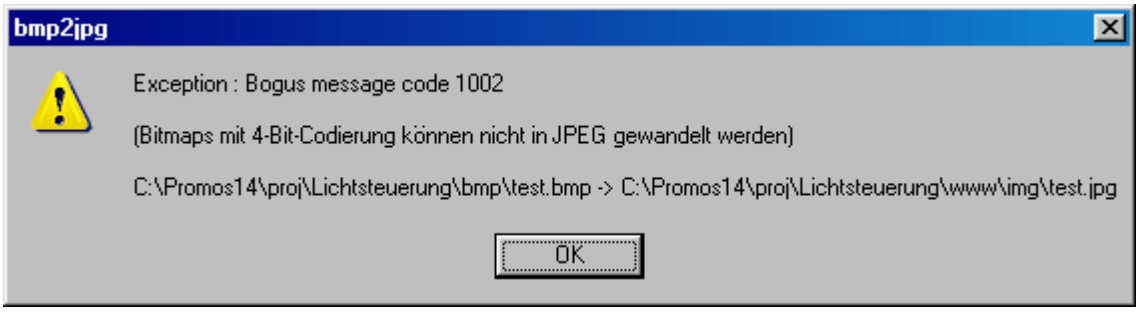
11.8.1.8 Optimise

Performs optimisation of the **entropy encoding** parameters. If these settings are not present, the preset parameters are used. Optimisation of the parameters requires somewhat more time for the generated image to become smaller.

We recommend that only experienced users use this option.
For additional information regarding conversion of images please consult a specialist website.

11.8.2 Error messages

"Bitmaps with 4-bit encoding cannot be converted to JPG"



If this error message is displayed, the corresponding bmp file must be converted from 4-bit to 8-bit or higher encoding with a graphics or conversion program (e.g. **Irfanview**).

11.9 Version overview (ShowVersion.exe)

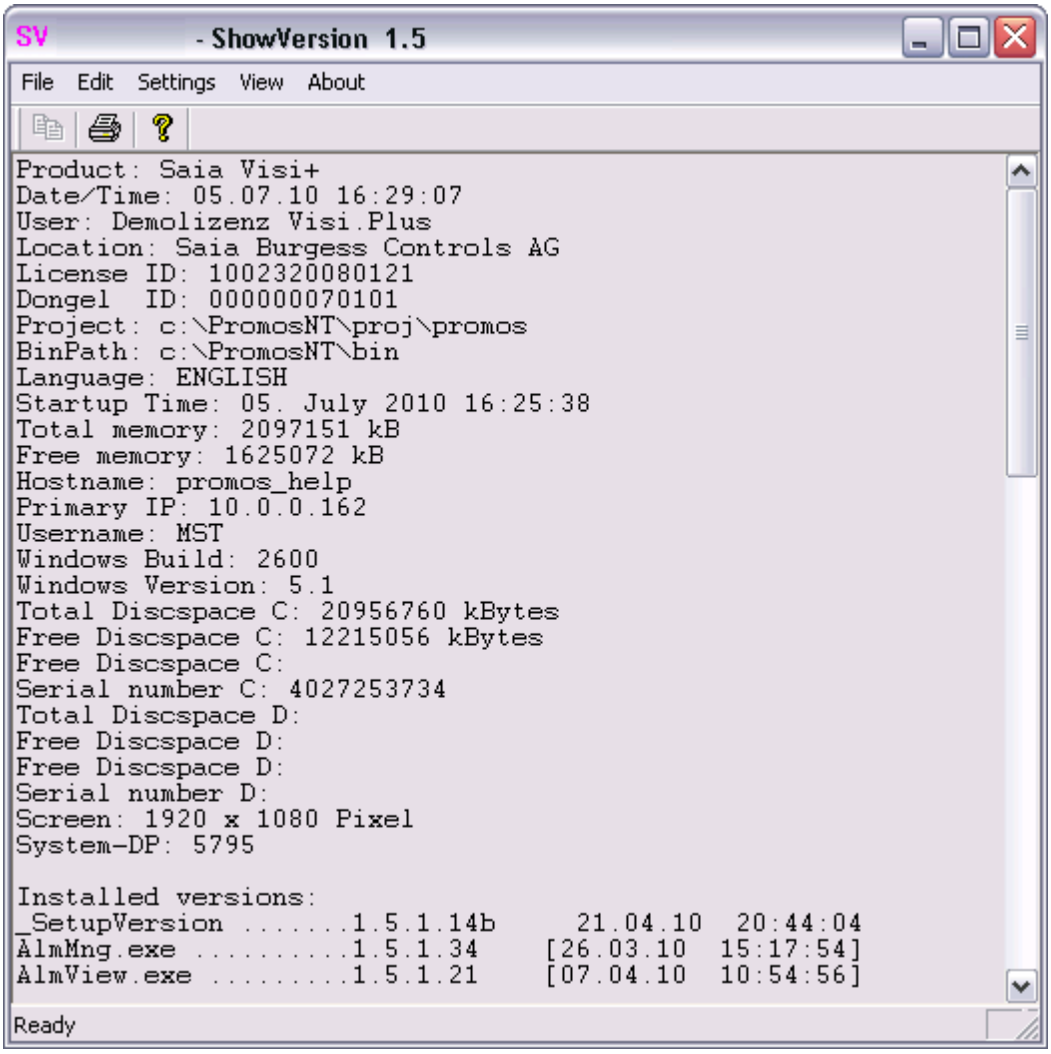
The program **ShowVersion** lists all versions of the Visi.Plus modules (programs) that are being used, as well as the last installed setup version, and displays them on the screen. The list of the currently used versions can be printed or saved to a file.

With **ShowVersion.exe** the Visi.Plus versions installed on your system and other specific information about your computer can be easily displayed.

The program can be opened from the default installation directory "**c:/Visi.Plus/bin**" or via your own shortcut.

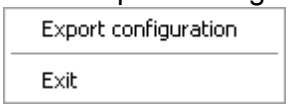
Initialise a button in the Graphical Editor for starting this program.

When the program is started, the following window appears on your computer screen:



To export the information, proceed as follows:

The complete configuration can be exported to a text file:



This can be useful for the support department.

Alternatively, the marked information can be copied with the key combination **CTRL+C** and then reinserted in the desired program with **<Paste>** or the key combination **CTRL+V**.

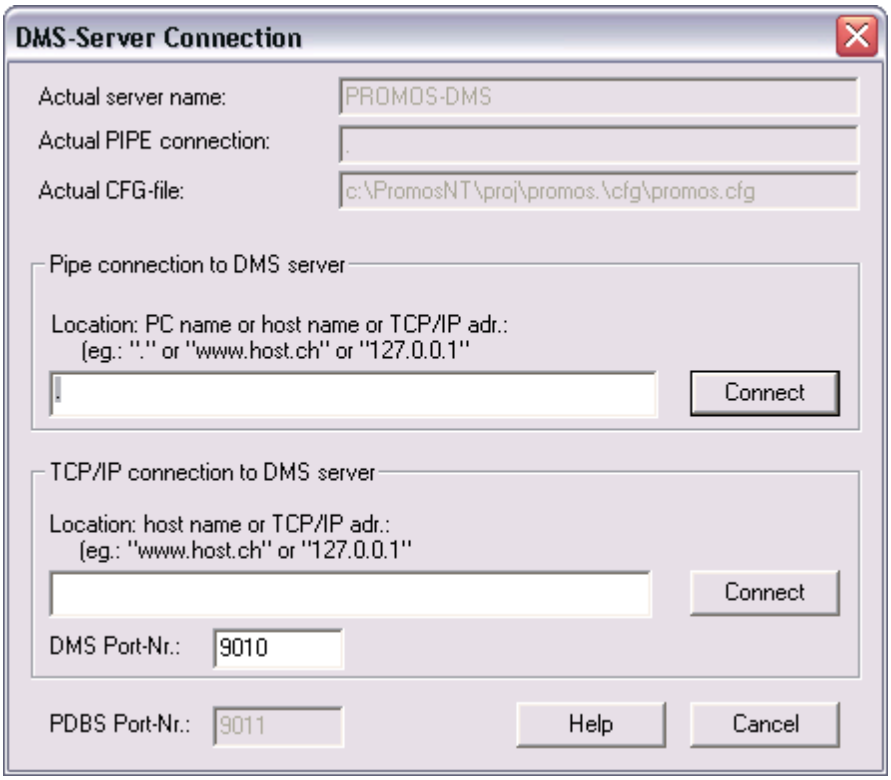
Clicking the **<Print>** button opens the Print Properties window and all displayed information can be printed immediately.

11.9.1 Starting the program

As an "exe" file, **ShowVersion** is located in the directory **"c:\Visi.Plus\bin"**. Therefore the file can be started from a file management application like Windows Explorer or using an initialised **"Button"** in the Graphical Editor.

If the DMS is not running on startup of ShowVersion, the following dialog appears:

<Yes>
A connection dialog appears. Now you can start the DMS or connect to a running DMS:



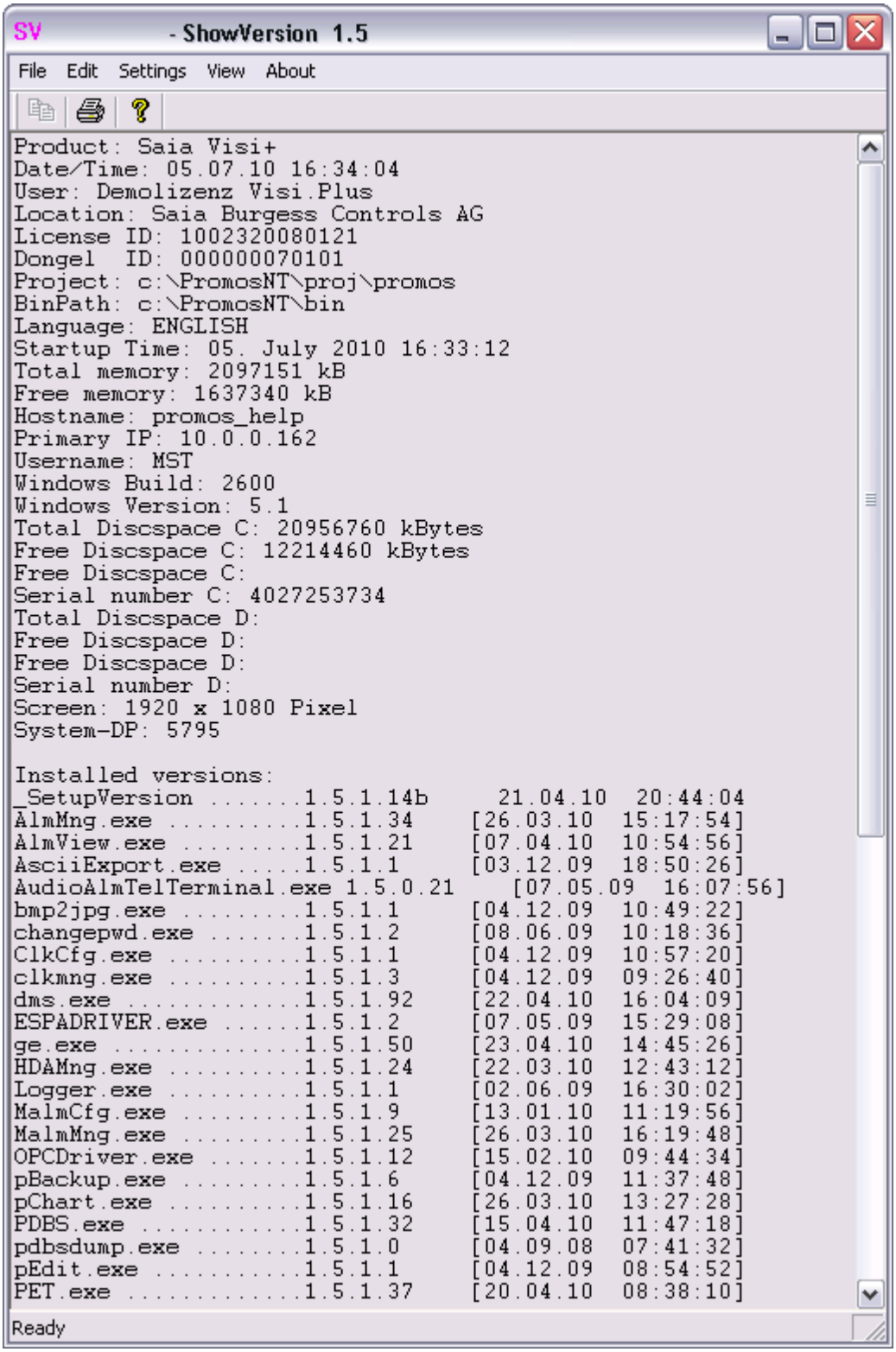
The connection can be established via pipe or TCP/IP (sockets):

Pipe: offers a faster connection if SetDMSValue is used within the private DMS network.
TCP/IP: simpler connection configuration; if SetDMSValue accesses the DMS server over the internet (use outside of the private DMS network) and must communicate through firewall(s). In this case, only the DMS port (9010 by default) must be opened in the firewalls.

<Reconnect DMS>
With an interruption in connection or DMS restart, SetDMSVal can be reconnected.

<No>
The ShowVersion program is exited.

11.9.2 The ShowVersion user interface



Information about the Saia PCD® controllers is also listed (only if the control system has already been online).

Example:

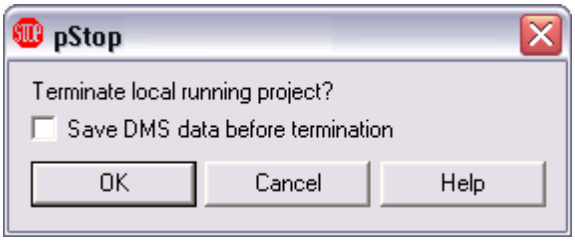
```
*** SS027E ***
PLC-Firmware: D3M3302B0
PLC-Channel: SDriver:SS027E:Station005
PLC-CodeFree:
PLC-CodeUsed:
PLC-Modified:
PLC-ProgName:
PLC-TextFree:
Online-AreaCode: 0
Online-AreaStation: 1
Online-Channel: SS027E
Online-CountryCode: 0
Online-Cpu: 0
Online-DeviceName:
Online-DontClose: 0
Online-IPAddress: 192.168.252.85
Online-Location:
Online-PhoneNumber:
Online-Retry: 3
Online-Station: 5
Online-UseDialing: 0
Online-Version: 1
IPModule-IPAddress: 192.168.252.85
IPModule-IPClientFlags: 1
IPModule-IPCpu: 0
IPModule-IPGateway: 192.168.252.81
IPModule-IPModified: 0
IPModule-IPNode: 5
IPModule-IPServerflags: 1
IPModule-IPSlot: 0
```

This version list must always be included in any e-mail support queries!

11.10 Exiting the whole system (pStop.exe)

The program pStop.exe is a small auxiliary program for exiting the system. The program sets the following internal data point in the DMS:**System:NT:Quit**

When the data point is set, all modules are informed that the system is exiting and the modules are then closed.



Save DMS Data Before Exit
If this option is selected, before exit the following data point is set inside the DMS:**System:NT:SaveDMS** and the DMS is saved.

Command line start options (no differentiation between lower/upper case)
Command line parameters can be combined as desired.
If the program is initiated via a button in the GE, the following parameters can be included:

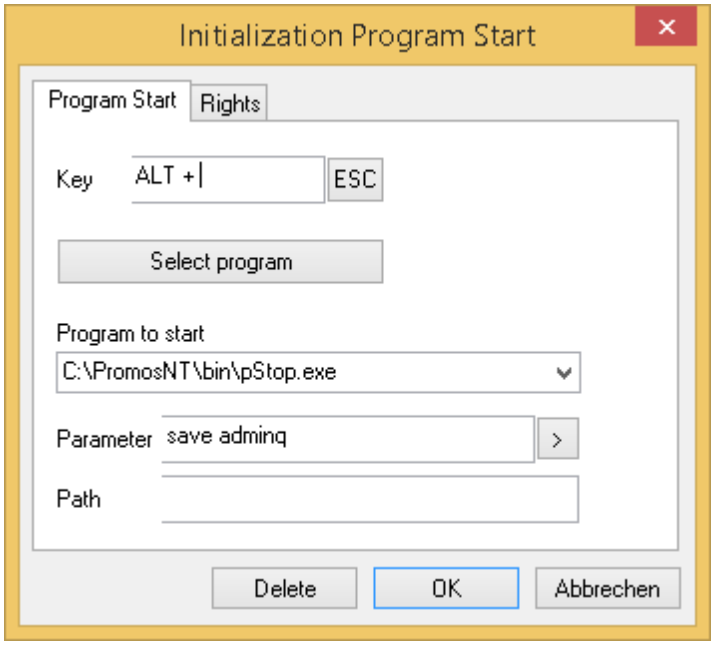
NOASK
It does not have to be confirmed with <OK>; the system exits without displaying the prompt if user got admin right.
Example: `pstop NOASK` or `pstop noask`

SAVE
The option "**Save DMS Data Before Exit**" is selected.
Example: `pstop SAVE` or `pstop save`

or `pstop SAVE NOASK`

ADMINQ
It does not have to be confirmed and no user with admin rights is needed.

Example:



Save the DMS and close the System without confirmation.

Index