Operating Manual

catman®
Professional 5.0

HBM Software

B0528-71 en

FAX

To	HBM	Attention	Jens Boersch
Dept.	M-M	Fax No.	+49 6151 - 8039243

Hottinger Baldwin Messtechnik GmbH
Im Tiefen See 45
64293 Darmstadt
Germany
www.hbm.com

Hottinger Baldwin Messtechnik GmbH
Dept. M-M
Mr. Jens Boersch
64293 Darmstadt
Germany

Software Registration

Device:

❑ MGCplus
❑ MGC
❑ Spider8
❑ PME
❑ Scout
❑ MVD
❑ DMP40
❑ AED
❑ K800
❑ UPM100/Centipede
❑ other __________

Interface:

❑ RS 232 C
❑ RS 485
❑ IEEE 488.2
❑ Ethernet
❑ USB
❑ CAN
❑ Profibus-DP
❑ other __________

Program: ____________________

Lizense-No.: ____________________

Purchase date: ____________________

Address

Name: ____________________

First name: ____________________

Company: ____________________

Dept.: ____________________

Street address: ____________________

P/O Box: ____________________

Phone: ____________________

Fax: ____________________

City: ____________________

Postal code: ____________________

E-mail: ____________________

Our company agrees that HBM uses the company’s name as a reference **without** details like company’s address or type of application.

Date: ____________________

Signature: ____________________

Why register?

If, as a registered user, you have a question regarding the software you have acquired, we would be happy to answer it. See Chapter A, Section 5, *Technical support*, page A-38, for phone and fax numbers. So take the opportunity to fill in the registration form and send it or simply fax it directly to us. The form fits into a window envelope for easy posting.

Thank you for taking the time.

General supply and license conditions for software

(Extract from standard terms and conditions of sale from HBM)

Special Rules concerning Software

1. If the Goods or any portion thereof includes software, which was developed by third parties, then the scope of the rights and entitlements granted to the customer will be determined in accordance with the third party's license terms, which we will include in the delivery and which we will send in advance upon demand. The aforementioned applies specifically to software such as operating systems and comparable components of the delivered system. We will provide suitable advance notice to the customer in the event that software from third parties will be included in the delivery (e. g., by naming the third party manufacturer in the bid documentation).

2. If the Goods we deliver include software developed by us (either as a component of the devices or as an independent product), then the following terms shall apply:

2.1 Unless otherwise specifically agreed to in writing in the individual case, the transfer of software for purposes of use in return for a onetime payment constitutes a sale at law.

2.2 We grant the customer a non-exclusive, transferable right to use the software developed by us on a single computer system for an indefinite period of time. In the absence of written approval, the software may be used for the customer's own purposes only. The software may not leased, either expressly or implicitly, without our prior written consent.

2.3 A transfer of the license to a third party requires that the software be completely deleted from the customer's system and that the media (including all documentation related thereto), which had been provided by us, be transferred to the purchaser of the software.

2.4 Subject to any other written terms, the customer shall be responsible for installing the software.

2.5 The customer is prohibited: a) from in any manner disclosing to third parties the software or any related material without our prior written consent (except for a complete transfer in accordance with section 2.3); b) from modifying the software without our prior written consent; c) from creating works derived from the software or copying the written materials; or d) from translating or modifying such software or preparing any derivative works based on the written materials. The foregoing provision shall not apply if the user is entitled, either by law or expressly by contract, to engage in any of the individual acts.

2.6 We reserve all rights to the software developed by us and any related material as well as to any modifications we performed. The software and related materials must be used and stored in such a manner that they are reasonably safe from any use, reproduction or disclosure in violation of the agreement.

2.7 One copy may be made for back-up purposes. A reference to our copyrights must be made or applied to the back-up copy. If a copyright and/or registration number appears in the software, this information may not be removed.

2.8 The software is subject to a warranty inasmuch as the delivered software materially achieves the main functions, meets the recognized standard of technology and does not contain any errors, which will materially reduce or eliminate its value or its fitness for ordinary or contractually stipulated use.

2.9 Unless expressly agreed otherwise, the software supplied by us is not error tolerant and was not developed or produced for purposes of use in a dangerous environment, in which a failure-free operation is absolutely necessary, such as in nuclear installations, aircraft navigation or communication systems, in aviation safety, in life-support machines or in weapons systems in which the failure in technology would directly cause death, personal injury or severe damages to property or the environment.

2.10 Any errors or bugs in the software will be removed, to the extent such action would be fair to the customer, exclusively by providing the customer with new program versions as part of the continued product development and care. In enforcing the warranty, the customer is required to reasonably support us—in the event any program failures are identified—by sending us the error reports and any other requisite information at our request. A delivery of any new program version will not generally cause the warranty period to commence to run anew.

2.11 Otherwise, with respect to the software, the general provisions of the agreement and our Standard Terms of Sale (specifically those terms relating to warranty and our liability) govern.

catman® Professional V5.0

Contents

A Introduction

C Customizing catman® Professional

D Defining Devices and Channels

E Measuring with catman® Professional

F catModules and Add-Ins

G Online Document

J The Auto Command List

K The Script Development System

L Index

A Introduction

1 Getting started

We are delighted that you have decided on catman® Professional, the universal measurement technology software from HBM. We are sure you will enjoy problem-free working with catman® Professional, having spent a short time getting used to it, and will soon find you are unable to do without the many facilities it offers.

This manual assumes that you:

- are familiar with your operating system
- know how to find and open files
- know how to use online Help files

☞ Information about Windows is always available from the “Start” menu in Windows, just choose “Help”.

1.1 Typographical conventions

For the sake of clear, unambiguous labeling and better legibility, we use the following conventions in this manual:

⚠ We have flagged important paragraphs with this caution symbol.

☞ This symbol points to a significant detail or specific characteristic.

This symbol marks a tip or an interesting feature.

We use italics to draw particular attention to a section of text or a word and for marking cross-links to other chapters.

`APP.critical` This typeface marks commands for the programming language of catman.

ESC — we use for descriptions of keys, in this case the ESCAPE key.

“File → Open” — All menus and menu commands are shown in quotation marks (in this case “File” is a menu and “Open” is a sub-menu). The same applies to register tabs and frames in dialog windows. In the case of menus, the title *Main menu* or *Window menu* distinguishes between the menu for the catman window (main menu) and the menus for the worksheets, which each have a window of their own. A *context menu* is a menu that pops up when you place the mouse pointer over an object or specific part of a window, e.g. a table, and press the *right* mouse button.

“*Start*” — Italics together with double quotation marks are used for the caption of push buttons, check boxes, input fields etc. and for user input.

Radio button — For marking Online Document objects this font is used. These objects are found especially in Chapter G, *Online Document* and Chapter K, *The Script Development System*.

This symbol means more information is available in the online Help.

1.2 Using the manual

If you are already familiar with an older version of catman® Professional, you should firstly read Section 3, *Installation*, page A-9 and the corresponding sections about the changes, e.g. Section 4, *What’s different from previous versions*, page A-19, in this chapter.

If you are not familiar with catman® Professional at all, we recommend reading the Chapter B, *Overview*, after installation. Then look for the chapters of most interest to you and read them through. You do not need to have read every chapter before you can work with catman; just select topics according to the requirements of your specific measurement task.

The separate chapters have a section called *Introduction* summarizing the main points about how to use the catman components concerned and giving a short description of what the chapter contains.

If you need more detailed information, such as get a list of an object's properties, use the context-sensitive online Help.

If you still make no headway with your problem, Section 5, *Technical support*, page A-38, shows you how to get help directly from HBM.

Please also note the support and the seminars offered by the HBM representatives in your country.

1.3 How this manual is structured

So that you can handle this manual easily (and so that you can also *hold* the manual in *one hand*), it deals primarily with the underlying *concepts* of catman® Professional. It gives an overview of the functionalities and use of catman, divided into different *operating levels* (degrees of difficulty), and shows how you can make the most productive use of catman. The manual contains no explanations of the type *How do I change the font and color of a text*, since this is a standard Windows procedure, nor does it contain any reference to individual script commands, since you can look these up more easily in the online Help (also standard Windows procedure). It is also faster and more convenient to follow up any cross-references using online Help. You can find an introduction to the structure of the Help system in Chapter B, *Overview*, in Section 6, *The catman® Professional online Help system*, page B-31.

The manual is divided into eleven chapters:

Quick Start

After this introduction (Chapter A, *Introduction*) you can get to know catman's structure and components, and how to work with them, in Chapter B, *Overview*. By way of illustration, we also provide a section containing a typical measurement example: Section 5, *Getting started (Quick Start)*, page B-22.

Configuration In Chapter C, *Customizing catman® Professional*, you will be able to find out which of catman's defaults you need to customize to suit your own requirements and what options are available in the user administration.

Chapter D, *Defining Devices and Channels*, explains how to set up a measurement system, from defining a device right through to computing in real time, outputting analog or digital values and saving your measured values to a file.

Chapter E, *Measuring with catman® Professional*, deals with such matters as setting up a sample rate and triggering, as well as the tricky question of simultaneous measurement using several devices. In the following the Event monitor and special topics like the Remote Data Server and the DAQ background process for long-term measurements are covered.

Level 1: In Chapter F, *catModules and Add-Ins*, we explain the catModules and Add-Ins available in catman and how you can use them to carry out measurement tasks. This is Level 1 which is the simplest and quickest to learn of all the catman levels.

Level 2: Chapter G, *Online Document*, shows you how you can create output pages yourself for the screen and/or printer.

Chapter H, *The Measurement Wizard*, describes how you can carry out measurement tasks using your own output pages (see Chapter G) and the Measurement Wizard. This is catman operating Level 2. Although there are many choices at this level, it only takes a small effort to become familiar with it.

Chapter I, *Database and Post-processing*, explains how you handle the Database, data import and export, as well as post-process mathematics.

In Chapter J, *The Auto Command List*, we show you a way of automating certain procedures, e.g. performing mathematical computations after each test period, and using them at operating Levels 1 and 2 without having to use the Script development system.

Level 3: Chapter K, *The Script Development System*, describes how you can define your own programs for purposes of measurement, testing or control. This is catman operating Level 3. For this you should have some elementary knowledge of a programming language such as Basic or similar, and be prepared to spend at least two days on familiarization, or perhaps even several days if the application you are planning is particularly complex.

2 Operating requirements

In order to operate version 5.0 of catman® Professional, you need a PC with the following *minimum* requirements:

- Intel Pentium 600 MHz or equivalent processor
- Windows® 2000 or Windows® XP
- Microsoft Internet Explorer version 5.0 or higher
- 128 megabytes of main memory (RAM)
- Graphics card with a resolution of 1,024 x 768 pixel
- 100 megabytes free space on your hard disk
- Microsoft or 100% compatible mouse
- Default printer configured
- One of the following interfaces for the connection of measurement devices:
  - Ethernet (10/100 MBit), USB, RS-232 (COM1, COM2), RS-485, printer port (LPT1, LPT2), IEEE 488.2 (GPIB), CAN bus or Profibus. For supported interface cards see Section 3.5, page A-14 and Section 3.8, page A-18.

If your system does not yet have a printer configured please define a printer type, even if it is not connected or not available for printing. catman® Professional derives the page size for printing from the paper size of the default printer and will not start if no printer is installed.

The following fonts must be installed:

Arial (TT), Courier, MS Sans Serif, Small Fonts, Times New Roman (TT) and Wingdings, preferably also Tahoma.

In order to be able to save measurement data you should have further free space on your hard disk. Estimate 8 bytes per measured value here; with 1,000,000 values for 50 channels you will need 400 megabytes for a file.

Notes

- You must have power user access rights for working with catman. Additionally write access must be permitted for the catman directory and sub-directories.

- If you want to work with an interface other than Ethernet, USB, LPTx or RS-232, this must be already installed and functioning before you install catman® Professional. The same applies to NI-MIO data acquisition cards from National Instruments. See also Section 3.5, *Notes on the Ethernet interface*, page A-14 and Section 3.8, *Notes on the National Instruments boards NI-MIO*, page A-18.
- Windows installs the USB driver for each connection (socket) again. If you use several USB connections, each time a new connection is detected the driver will be installed.
- Please read the requirements needed for sending e-mails via Event monitor in Chapter E, Section 6.2.7, *Send e-mail*, page E-29, if you want to use this feature.
- For continuous working, a high number of measured values or high data transfer rates we recommend that the main memory is at least twice the size stated above. A fast hard disk and a fast graphics card will also improve system performance so that you can carry out real time computations even at high data transfer rates and achieve a delay-free display.

3 Installation

3.1 Installation from CD

☞ We strongly recommend to uninstall older versions of catman before installing a new one. Data files will not be deleted when uninstalling! However, if you modified the Sensor Database and did not change the file name, you must move this file to another directory and later import your settings into the new Sensor Database, see Chapter D, Section 5.3, *Changing/creating/importing a Sensor Database*, page D-57.

Put the CD in your CD-ROM drive. In standard configuration, Windows automatically opens the CD. The opening screen appears and you can specify which language you want to see in subsequent dialog boxes.

Figure 1: Opening screen after executing SETUP.EXE

☞ You can cancel the installation at any point.

If you have deactivated this self-starting Windows function, look for file SETUP.EXE in the main directory of the CD and double-click on its icon to get to the opening screen. Click on "*English/Englisch/Anglais*" to get the next screen in English language.

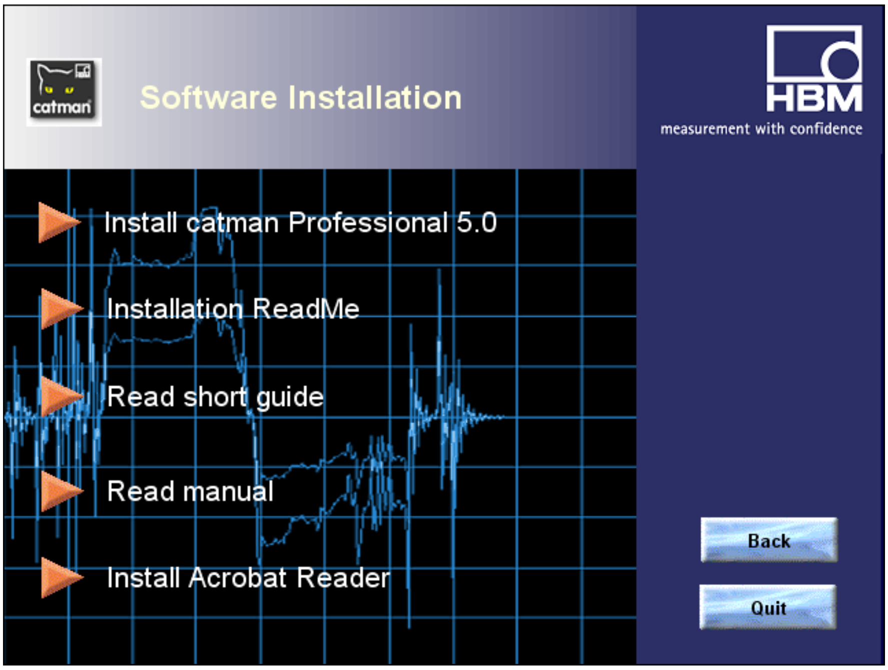


Figure 2: Installation screen of catman

☞ We recommend that you close all open programs.

To install catman® Professional, click on "*Install catman Professional 5.0*". Read through the license agreements and acknowledge with "*Yes*". If you have a runtime license, install this version of catman, otherwise click on "*catman Professional*" in the next window. Then enter the directory in which you want the program to be installed: either confirm the suggested target folder or click on "*Browse*" and choose which one you want. If necessary, SETUP.EXE creates the directory you have specified and then copies all the files to it. Please follow the rest of the instructions in the Setup program.

☞ The *first* time you start the program, you must enter your name and company and the license number shown on the CD. If you do not enter a license number, catman® Professional will start in evaluation mode (the program can be run 25 times).

A sub-directory called CATMOD\BACKUP is created during installation. If you ever accidentally overwrite a catModule or one of its associated Online Pages, just copy the files from this directory back into your catModules directory.

3.2 Notes on the installation

Installation requires a further system service to be started. Administrator rights are required for this.

You have two options for installing catman® Professional:

1. Obtain Administrator rights for installing catman and perform the installation. Then restart the PC.

or

2. First carry out installation under the name of the later user (catman® Professional license holder) and then ask your system administrator or a user with such rights to register the service required for operating catman® Professional. To do this use the program NT_IODRV.EXE which after installing catman will be in the WINNT\SYSTEM32 directory. Alternatively, you can also have the complete installation carried out in the same directory by the system administrator and the system service will then be entered and activated automatically. Then restart the PC.

☞ For Spider8 the driver DIRECTNT.SYS is needed. The driver is installed in the Windows directory SYSTEM32.

USB adapters under Windows® XP

A warning may be issued when installing the drivers for the USB adapter under Windows® XP that the driver concerned is not signed by Microsoft or a certified driver. Continue with the installation. The driver will only be activated for the HBM device and other system settings will not be affected. See also Section *Notes on the USB interface*, page A-15.

3.3 Notes for MGCplus users

If connecting the MGCplus via the printer port of the CP32 please ensure that you use either the cable supplied or an *Interlink* cable. A cable of this type can also be used to connect two PCs directly, e.g. for data transfer, and is available from specialist computer supplies retailers if you need a cable longer than that provided. With MGCplus approx. 15 kilobytes per second is transferred via the printer interface as well as via a USB adapter connected to this parallel port, i.e. almost 4,000 values per second in the format 4-byte.

☞ These figures are not valid for CPUs with built in USB interfaces such as CP22 or CP42. See also Section *Notes on the USB interface*, page A-15.

⚠ With the MGCplus, the maximum data rate is not reached via this interface. Use either IEEE488 or (Fast) Ethernet.

3.4 Notes for Spider8 users

☞ If the procedure described below does not run successfully or it is not possible to make the required settings you can also connect the Spider8 via USB. For this an adapter is required for the Spider8 which converts the interface to USB, see also Section *Notes on the USB interface*, page A-15. The plug can be obtained from HBM. Transmission rates like with the EPP mode are also possible with the USB interface.

Used with a modern PC, catman® Professional allows transmission rates of up to 80,000 measured values per second via the LPT printer port. However, since not all PCs have the same hardware there are different modes in which this interface can be operated. Unfortunately, this function is not only influenced by catman, but also by the BIOS setup of your PC and also by the operating system. For this reason when using the Spider8 you should always call up the Setup dialog for the interface (menu “Device → Setup interface” in the I/O Definition worksheet) to find the best possible setup.

Procedure

Try to find a functioning operating mode in the order *EPP—Byte Mode—Bit8 Mode—Nibble* Mode. To do this you will probably have to modify the BIOS of your PC. To do this it is generally necessary to restart your PC while holding down a particular key, e.g. DELETE or F1. This depends on your PC—please read the documentation supplied with your PC.

In the BIOS setup the printer interface can often be set to settings such as *AT, standard, compatible, bidirectional, PS/2, ECP* or *EPP*. Unfortunately, not all the terms used here have been standardized, so you will have to try out the different settings. ECP does not generally work since the operating system uses special commands here to test the printer from time to time and to check functioning. Try *EPP—PS/2—bidirectional—SPP—AT* in this order if these terms can be found in your PC-BIOS. Then check whether Windows also works in EPP mode:

- From the Windows Start menu call up the “System Management”.
- Double-click “System” and then click the “Device Manager” tab. In Windows® 2000 click first on tab “Hardware” and then on button “*Device manager*”.
- Select “*View devices by type*” and click on your printer interface for the COM and LPT ports; then click on “*Properties*”.

If “*ECP*” or “*ECP Printer Port*” is still entered here for your printer port, proceed as follows:

- Click the “Drivers” tab and then click “*Update driver*”.
- Do *not* let the system search for the driver automatically but instead click “*Select driver*”.

- Click “*Continue*” until you reach the wizard for device driver updates and there select “*View all devices*”.
- Now select “*Printer port*” instead of “*ECP Printer Port*” and complete the installation.

If you later define the Spider8 in the I/O Definition, you should always call up the setup dialog for the interface: “Setup interface” in menu “Settings”. Choose the mode you selected in the BIOS to reach the maximum possible transfer speed for your hardware configuration.

Provided you are not starting with an I/O Definition which has already been modified, each time you create an empty I/O Definition worksheet, the slowest (but still functioning) “*Nibble Mode*” setup is selected. You can however modify this from the “Options → Device” menu: enter the fastest possible operating mode under “*Spider8 operating mode”. T*he I/O Definition will then always use this type of connection as default.

Please use only high quality printer cables for connection to Spider8 (Premium Quality); normal cables are not suitable. Also, keep the length to a minimum.

3.5 Notes on the Ethernet interface

The Ethernet interface is a standard interface under Windows, you can use all cards supported by the operating system. The installation sequence of catman and card drivers is not relevant.

Regarding the installation procedure, please read the manual of the interface card.

☞ If you want to operate an HBM device via an office Ethernet, you should ask your system or network administrator whether this is possible. This concerns the loading of the network with the data transfer between the device and the PC. We recommend to use a separate network for measuring devices, i. e. to connect the devices to a second Ethernet card in the PC.

The transfer speed may be increased with larger TCP/IP buffers, see Chapter D, Section 2.4.1, *Bus-compliant interfaces*, page D-12.

3.6 Notes on the USB interface

The USB interface is already implemented in Windows. However, each device also requires a driver for this interface. The adapter will be recognized after connecting to your PC (Plug&Play) and Windows requests the corresponding files. Insert the disk or CD which has been supplied with the adapter or the CPU (MGCplus) and enter the path to where the files can be copied. The Windows Hardware Assistant will then install the driver.

Windows installs the USB driver for each connection (socket) again. If you use several USB connections, each time a new connection is detected the driver will be installed. The CD is however only required the first time.

If the original disk or CD is not available, you will find a version on the catman CD (this may however be an older version): directory USB_HBM. The catman CD only contains the specific drivers for the Spider8 and MGCplus, not those for the USB support in Windows. These are on the Windows installation CD or supplied with the USB Hardware.

If you want to switch over between a USB adapter for the CAN bus and a PC-CAN card or a Parallel-to-CAN adapter, you must restart catman, so that the appropriate drivers can be loaded.

Installing under Windows® XP

A warning may be issued when installing the drivers for the USB adapter under Windows® XP that the driver concerned is not signed by Microsoft or a certified driver. Continue with the installation. The driver will only be activated for the HBM device and other system settings will not be affected.

Notes concerning USB 1.1-Adapter (High Speed 12 Mbaud) from HBM

- LED1 must light up green (connection established with PC) and blink when data is being transferred.
- LED2 must light up yellow (connection established with HBM device); during the initialization (looking for connection) the LED blinks red.
- The current consumption for the USB interface adapter is approx. 140 mA at 5 V.

☞ The maximum possible speed of the USB 1.1 adapter from HBM depends on the measuring devices used. For MGCplus approx. 15 kilobytes per seconds is reached (this corresponds to about 4,000 measurements per second in 4-byte format), for Spider8 approx. 300 kilobytes per second are reached (this corresponds to approx. 150,000 measurements per second).

3.7 Notes on the interfaces IEEE 488.2, CAN bus and Profibus DP

3.7.1 Supported interface adapters

The following interface adapters are supported by catman® Professional. Regarding the installation procedure, please read the instructions for the interface adapter.

IEEE 488.2 (GPIB)

- GPIB plug-in cards from National Instruments. We recommend the use of types PCI-GPIB and AT-GPIB/TNT and the PC card PCMCIA-GPIB for laptops. We do not advise use of the (older) cards PC2 and PC2A.

CAN bus

- Passive Parallel-to-CAN adapter from the company Peak Service GmbH, Darmstadt, Germany.
- Passive PC-CAN card from the company Peak Service GmbH, Darmstadt, Germany.
- Passive USB-to-CAN adapter from the company Peak Service GmbH, Darmstadt, Germany.

If you want to switch over between a USB adapter for the CAN bus and a PC-CAN card or a Parallel-to-CAN adapter, you must restart catman, so that the appropriate drivers can be loaded.

Profibus

- Hilscher CIF30 DPS and CIF50 DPS Profibus DP Slave card.
- Hilscher CIF30PB, CIF30DPM, CIF50PB and CIF60PB Profibus DP Master cards.
- Siemens CP5613/14 Profibus DP Master cards.

  ☞ The Siemens CP5412 A2 Profibus DP card is no longer supported as no Windows® 2000 drivers exist.

catman® Professional supports the operation of devices via the Profibus in the operating mode as master and has read, write and diagnosis functions. When the Hilscher CIF50 DPS Profibus DP slave card is used, catman supports operation as slave. However, in this case, the catman script language must be used, it is not possible to use it via the I/O Definition. The bus system must already have been configured as this cannot be carried out by catman. Up to 32 slaves can be connected to I/O channels.

3.7.2 Installation sequence

GPIB-plug-in cards from National Instruments

National Instruments driver software, either 32-Bit Compatibility Kit or native 32-Bit driver has to be installed beforehand.

Passive Parallel-to-CAN adapter from Peak Service GmbH

The installation sequence of catman® Professional and interface is not relevant.

Passive PC-CAN card from Peak Service GmbH

The installation sequence of catman® Professional and interface is not relevant.

Hilscher Slave and Master cards

The installation sequence of catman® Professional and interface is not relevant.

Siemens CP5613/14 Profibus DP Master card

The installation sequence of catman® Professional and interface is not relevant.

3.8 Notes on the National Instruments boards NI-MIO

At present catman® Professional supports the National Instruments cards in the MIO-E series. For digital I/Os and analog outputs, the following are also supported:

- AT-AO-6
- AT-AO-10
- DIO-PC
- PCI-DIO

When using AD converter cards from National Instruments the driver software NIDAQ must be installed before installing catman® Professional.

4 What's different from previous versions

We strongly recommend to uninstall older versions of catman before installing a new one. Data files will not be deleted when uninstalling!

You can find more notes on what's new in catman® Professional's online Help files *Introduction* and *Script language*.

4.1 What's new in version 5.0?

This summary provides information about the new functions in catman® Professional version 5.0 for users who have already a good knowledge of catman® Professional. In addition to many detailed improvements, the modifications described in the following have been made.

4.1.1 General

Online Documents created with release 3 cannot be loaded into older releases of catman 5.0.

You must have power user access rights for working with catman. Additionally write access must be permitted for the catman directory and sub-directories.

- From the "Options → Projects" main menu, you can now assign the worksheets or settings, which should be transferred as a file reference into a *Project file*.
- Additionally you may now create a *packed* project: in this case, all files are compressed into one.

- From the "Options → System colors" main menu, you can also specify the colors that should be used for the buttons, text input fields and dialog windows in catman.
- The multiple start option of catman 4.5 (starting catman twice on the same PC) is no longer available.

4.1.2 Test Manager and Data Explorer

catman now offers you the possibility of assigning *test parameters* via the Test Manager. This allows you to provide information such as tester, department or test sample for the different measurements, and to manage these measurements in a view. Up to 128 files, which were created during a test, can also be combined into a list, thus allowing quick, easy access. The Data Explorer (previously Test Explorer) can now be called from the Database worksheet or the Project window. You also have a quick overview of these parameters and the data files generated during the measurement. See also Chapter E, Section 5, *Test Manager*, page E-21, and Chapter I, Section 2.5.1, *The Data Explorer*, page I-15.

☞ Starting with Release 3 the Data Explorer is again closed if data has been imported. Deactivate "*Close Explorer after import*" if you want to import several files

4.1.3 Dialog Measurement settings (I/O Definition, Measurement Wizard and catModule)

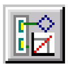
To simplify the operation, the dialog initially displays only the most important settings (sample rates and timing). You can access settings that are used less frequently by clicking on "*Advanced view*".

☞ The sample rate groups 2 and 3 of the MGCplus can only be accessed in the advanced view.

4.1.4 Changes in the I/O Definition

The menu structure has been changed in the I/O Definition. The number of columns has been reduced in order to provide a better overview:

- The "Setting", "Scaling", "Target" and "Diagnosis" menus have been removed. Their submenus can now be found under the new "Device" and "I/O channel" menus.
- All *device-related* settings can be found under the "Device" menu.
- All *channel-related* settings can be found under the "I/O channel" menu, e.g. "Scaling", "Zero balance", "Online Export (target)".
- Diagnosis functions, which are seldom used, have been combined into a submenu under "Device". The "Test device connection" and "Test link resources" menu commands, which are more frequently used, are under the "Device" menu. The current measured signal can be displayed via the "I/O channel" menu.
- The unit is displayed in the "Scaling" column and this can only be changed here.
- The "Slot" column has been removed, the slot used is shown in the "Connection" column.
- The handling of TEDS- and T-ID information has been improved.
- The Sensor Database has been extended. It can now contain different linearizations and correction data.



- As default, a user scaling is no longer allowed for a channel, if a sensor from the Sensor Database has been assigned. You may however change the specified scaling (“Sensor” column, “Modify sensor adaptation” or ). This enables you, for example, to set gage and bridge factor for a strain gage channel as well as the temperature compensation method. The data in the Sensor Database is not affected by this change.
- The time channel types NTP and IRIG-B for the CP42 (MGCplus) are supported.
- Third-party devices can also be linked via a DLL, see Chapter K, Section 6.13, *DLL Drivers for “unknown devices”*, page K-84. This means that there are no longer restrictions like with Script Drivers and time-synchronized collection of data is also possible if the device is able to deliver the necessary time data.
- The “*Add Sensor ID (from TEDS or T-ID) automatically into Sensor Database*” option has been removed. This can now be called from the context menu, “Add Sensor ID to Sensor Database” command.
- The zeroing can now also be carried out via catman (“Options”). Instead of carrying out a single measurement, several measured values are averaged here.
- The function for scanning the Ethernet interface has been changed in catman 5.0 R2 as in Windows XP several tries to address non-existent Ethernet addresses are seen as a potential threat (Virus) and are blocked. A ping is now sent and the system waits for an answer. The disadvantage here is that this process requires more time, as there must be usually be a wait of approx. 200 ms. You can however shorten the waiting times in fast networks (“Options”).
- In order to be able to identify later in the measured values when a channel was overloaded, you can specify in “Options → Channel” that a certain value (that is outside the normal measuring range) should be displayed.

4.1.5 Changes in the Online Documents

- Data source drag&drop has been revised. You can now create Digital indicators or Post-Process graphs by dragging&dropping a data source onto a free position on the Online Document.
- The Scientific graphs now have their own toolbar, thus allowing quick access to the most important configuration options.

  The new Scientific Graph toolbar will also be displayed when old Online Documents are imported. However if the graph is positioned on the top edge of the window, only part of the toolbar may be visible. In this case, either move the graph down or hide the toolbar ("Interaction" tab, "*Internal dialogs*").
- You can now save a certain Scientific graph configuration as *default layout* ("General" tab). This layout will then be used when a new graph is created.
- The Scientific graphs can now be synchronized, not only with a video but also with other graph objects so that these either also move the cursor or—for Digital indicators or Spreadsheets—display the value of the channels at the same point in time.
- QuickView and Scientific graphs can also use values with double floating point accuracy for the y axis.
- For the Scientific graphs, a series of new options are available, e.g. shadows for text and bars. Interruption of curves at Null values can also now be switched off.

4.1.6 Changes in the Database

- The dynamic format used for the catman® Easy Database can now also be used in catman. If you are using less than 512 channels, you can use a dynamically increasing Database. This is only limited by the amount of free space on the hard disk. The disadvantage is that writing values to the Database, at least for the first writing operation, takes longer than with the standard formats, as the space required must first be reserved on the hard disk.

- Further sensor information (e.g. for CANHEAD) is now included in the traceability data (channel information).
- When exporting data in the nSoft® time series format (*.DAC), the file name is formed from the name you have entered, followed by an underscore and the channel name, then the file extension DAC.
- ASAM-ODS has now been added as a new export format.
- See also Chapter I, Section 2.5.1, *The Data Explorer*, page I-15.

4.1.7 Changes in the Script development system

The changes and new commands are described fully in Chapter K, *The Script Development System*.

4.2 What was new in version 4.5?

General changes

- A user administration has been introduced. You can use it both to speed up the catman start (Startup options) and the restrict access to catman for different users.
- Event monitoring is used to check the entry of events in all measuring modes (catModule, Measurement Wizard, DAQ background process or with a script) and execute different actions.

  ☞ The Event monitor settings are saved with the I/O Definition.

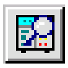
- With the post-process mathematics and the configuration of graphical objects, as long as there can be no confusion, the identification IO and DB have been removed from the channel list names. As a rule, the channel type is derived from the type of graph, the post-process mathematics can only work with the Database channels.
- catman has a Log file, which opens when the program starts. Other events can be logged via the Event monitor or script commands.

Changes in the Project view

- The sequence of the project files has been changed. The new setting is now more in line with the sequence followed when working with catman.
- The groups “Add-Ins” with the program DataView and VBScript are new.

Changes in the I/O Definition

- There are several important changes in the I/O Definition which make it more comfortable to configure measuring chains with MGCplus and Spider8:



- The Device scan  searches through different interfaces for the HBM devices MGCplus, Spider8 and DMC*plus*.
- The Sensor Database, which allows quick assignment of transducer characteristics to channels and also saves these settings in the I/O Definition (not in the device as previously). With MGCplus and connection plates with the identification i, for example AP815i, The transducer identification (T-ID) can also be used for configuring.
- With MGCplus and Spider8 the band width and Filter settings for a channel can also be done via the I/O Definition. A setting can also be selected which sets up a corresponding filter depending on the sample rate in order to prevent alias effects. In this case the cut-off frequency which shall be applied can be pre-set via the Options menu.
- The columns “Sensor setting” and “Filter setting” have been added in order to display these new possibilities.
- An additional column (“B”) allows Description files of any type to be assigned to I/O channels, e.g. in order to link pictures or explanatory texts to the channels.
- Integral, Derivative and IF-conditions are now also available for the Algebraic computations.
- The channel names with Rosette computations can now be assigned according to different patterns.

- Trigger conditions: The error when using negative user scaling where the trigger mode had to be set to “*Less than or equal*” to for triggering on exceeding the value (and vice versa), has been fixed.

Changes in the Database

- The Test Explorer allows you to see the comments for all files in a directory that contain measuring data. Binary catman files, files which were created by the MGCplus and files from DIAdem® are recognized.
- When Importing, the file names can be appended to the channel names. In this way, you can differentiate between the channels coming from different files but which use the same channel name.
- You can transfer data from a PC Card memory in the MGCplus to the PC via “File → Transfer MGCplus data file”, the MGCplus Assistant is not required.
- The Database channels can be write-protected through the user administration. In this way, you can prevent measurement values from being edited.
- The new computation change/convert sample rate converts data which were acquired with one sample rate in the required number of values in order to be able to compare them with a measuring series of a higher or lower sample rate.
- The designation of the catman binary standard format has been changed: It is now simply called the “*catman*” format.

Changes in the catModules

- Visualization of measurement data can now be done using an Add-In: DataView. You can load with the new completely reworked module not only MGCplus files but also compress and display larger catman files.
- The display speed of the Scientific graphs can be considerably increased. If required, min/max pairs can be compressed (“*Data compression*”), so that the amount of data to be processed is reduced by a factor of more than 30.

Changes in the Measurement Wizard

- The extended status window now contains information about the transmission speed during the measurement. You can also read whether the display is still being done in real-time or whether only the measuring device buffer is being read. The latter would lead to a delay between display and measurement.

- For the Data logger you can now choose between different types of dynamic creation of file names when restarting. You can also enter a comment before each new measurement starts.
- The display speed of the Scientific graphs can be considerably increased. If required, min/max pairs can be compressed (data compression), so that the amount of data to be processed is reduced by a factor of more than 30.
- When scrolling through pages, real-time Scientific graphs can be refreshed using the data already in the Database (“*Update on page change*”). In this case the last seconds are displayed again after browsing to a new page.
- For MGCplus with CP42 there is a new operating mode which allows data to be recorded onto the internal PC Card memory in the MGCplus.

Changes in the Online Documents

- The display speed of the Scientific graphs can be considerably increased. For post-process graphs, there is a completely transparent compression of min/max pairs (“*Post-process compression: Yes*”), which is even carried out when necessary when zooming. With real-time graphs only the min/max pairs are displayed, a zoom also displays only these and not the original data. You can however display the original data from the Database at any time in a post-process graph.
- When scrolling through pages real-time Scientific graphs can be refreshed with the data available in the Database (“*Update on page change*”). In this case the last seconds are displayed again after browsing to a new page.
- You can now display segments of curves in graphs.
- You can select different legends for the Scientific graphs: single line, within the graph, within the scaling level, etc.
- The cursor now displays the x and y values of all curves in Scientific graphs. There are also different options for the display positions.
- The background of the Scientific graphs can be configured via different style templates or manually and, for example, pictures can also be displayed.

- The plot methods for Scientific graphs have been removed: You can now select a different display type for each curve.

4.3 What was new in version 4.0?

Modification of the project management

- The project management can now also contain files with QuickView window configurations.
- When creating a project, the script project files are only transferred into the project if the Script editor has been opened once.
- The device setup files for Spider8 and MGCplus can now be created directly in the project window: context menu “Add → Current device settings“. It is no longer necessary to call the device setup module.

Changes in the I/O Definition

- The different views have been removed, all (reworked) columns can be seen at the same time.
- Two new symbols have been introduced to perform a single measurement or activate a continuous measurement of all active channels.
- A new function for zeroing has been introduced which can also be used for different devices. This special zero adjustment is also used in the other catman components, e.g. the catModules or the Measurement Wizard. This replaces the previous taring function.
- There is now an icon in the I/O Definition to display the channel information (traceability data or acquisition parameters).
- Real-time computations are no longer created from their own menu in the column “Settings” but must be specified from a submenu in the “Connection” column when defining a computation.

- The Spider8 digital channel is no longer created when channels are automatically connected.
- In the “Options” menu, you can specify that the most favorable interface connection should always be used for the Spider8 (this means not “*Nibble mode*”), as soon as a new I/O Definition is created.
- A new “Device” has been introduced: the Remote Data Server. This means that a catman program can work as server and other catman programs process measured data from the server.

Changes to the Database

- The channel name used in the I/O Definition is no longer displayed here. Only the Database channel name is displayed. There is a command to transfer the name from the I/O Definition.
- There is now an icon to display the channel information (traceability data).
- The starting time of the measurement is also included in the channel information (date-time format).
- catman now supports a maximum of 10,000 channels. After the installation up to 1,000 channels may be created so that not too much storage capacity is occupied. This limit can however be increased to a maximum of 10,000 channels via the menu “Options → Startup options“. The number of actual channels used is then set in the Database configuration.
- When making computations, the formula is inserted as plain text in the comment line.
- When data is being exported, a suitable file extension is automatically suggested. This extension depends on the format selected and is set as a filter in the file selection dialog.
- When exporting in ASCII format, all traceability data (acquisition parameters) can now be saved. When saving this channel information, identification of the channels for catman is normally appended at the end. If required, this can however be dropped now.
- It is now possible to export straight into an Excel *file*.
- RPC III (MTS) has now been added as a new export format.

Changes in the catModules and the Measurement Wizard

- All output windows can now be sized as required. This allows you to display other programs which are running during a measurement in the background or next to the catman window. The graphs in the windows change their size according to the window.
- The taring which only works for devices having this function has been replaced with the new zeroing function. This allows zero adjustment of the channels, also with user scaling, even when different devices are used.
- New icons are included in the toolbar to save or delete individual values in the Database.
- Data can now be exported straight into an Excel file, for example, for periodic measurements after each period.

Remote Data Server

- catman can also be used as data server, i.e. the measurement can be carried out as normal via catman and one or more devices. This data can be collected from catman “clients” on one or more PCs and processed. The clients are normal versions of catman which are connected to the Remote Data Server via an Ethernet network and TCP/IP.

Changes to the Online documents

- Objects can be locked and the Design toolbox can be hidden/revealed from the toolbar. So now all important actions can be accessed via the toolbar.
- As the starting time of the measurement is now part of the channel information, the date and time can now be displayed on the x-axis in the Scientific graphs and QuickView windows, even if no absolute time channel has been recorded.
- For Scientific graphs, there is a new Real-time bargraph indicator which can display many channels as bars.
- The printer can now be selected when printing individual Scientific graphs.
- So-called *Control curves* were introduced for displaying limit values or set points for Scientific graphs.
- The Indicator panel can now show up to 250 channels.
- The object On / off switch and LED can in the configuration as LED also monitor the level of an I/O channel.

- The Video replay has been improved: Synchronous to the displayed picture, a cursor can be displayed in the 2D Scientific graphs and Post-process cursor graph. This is used to mark the measured data according to the picture.

4.4 What was new in version 3.1?

New start dialog

catman can be started:

1. in *Professional* mode, this is the normal routine for starting catman.
2. in *Express* mode, this corresponds to the catman Express version, and allows access to Level 1 and only limited access to Level 2.

General modifications

- Improved project management: when a new project is created, all necessary and existing member files will be set up automatically. These do not need to be saved beforehand.
- The traceability data in the channel information now also includes the catman internal scalings.
- Data in the Database can now be displayed quickly using the QuickView windows.
- The MGCplus support has been expanded. All new plug-in modules are also supported.
- An asynchronous measuring mode has been introduced for long-term measurements: *DAQ background process*. Online Pages can be edited and other tasks carried out while the measuring function is running in the background.

Changes in the I/O Definition

- New devices can be defined and existing ones modified in a dialog window.
- The “Status” and “Unit” columns can also be seen in the Standard view.
- Data can be exported online into separate files. One file is created for each channel.

- The “Show measured signal” function is now also available for synchronized MGCplus. (Depending on the firmware version of the CP32 there might be an update necessary.)
- The request for automatic connection and the number of time channels to be set up when connecting to a MGCplus can be defined from the “Options” menu.

Changes for Online Documents

There have been a number of changes in this version, but many of them are not particularly obvious: As a result of feedback from users we have enhanced the performance of some objects, for example when texts ran together, too much free space was used up etc. We have also improved user-friendliness by making changes to the way the Design Toolbox looks and behaves.

- Design Toolbox: It is now configurable, and can be called up with different defaults. This enables the various user groups to display just the objects they need and to hide any objects superfluous to their requirements. The current configuration is saved and still applies next time catman is started.
- New Objects
  - Scientific y(index) graph: This enables data that needs to be displayed using one index only, such as bar graphs (including those with 3D effect), to be produced faster and with less load on the PC, since only the y-axis is defined. Among other things the object provides various options for formatting the index label, including a free-text facility.
  - OpenGL 3D graph: The object uses OpenGL, which can best be described as hardware-supported graph functions that are available with a large number of graphics cards. This allows significantly faster graph production, and some of the options are not available with the other graphs. For instance, you can show several plots in a single 3D graph, you can make your displays appear to cast shadows in a light source, and you can use interactive zoom, panning and rotation. This object can even be used with an array as a data source which is interactively defined.
- New alternative for Online Pages

  QuickView Windows are now available to provide fast data display. These already contain a graph, and all you have to do is define the data sources. Since you can call multiple windows, it is just as quick to configure a number of graphic displays. Various graphic analysis options are available to every window.

- Miscellaneous enhancements
  - The option “*Align objects to grid*” can be switched on or off via the toolbar.
  - Drawing tools can also be positioned in the background within a graph, that is, the grid and curves then lie in front of the drawing object.
  - In the case of the Scientific y(t) Real-time graph it is also possible to specify a different source for the time marker generation than the Δt of the curve. This option has been made specially for script programmers, who can then also set up a script-defined Δt.
  - The option to prevent an object being configured during the run-time of a script or the Measurement Wizard is now defined via the configuration dialog window of the object concerned. In the case of graphs this can only be done by using script commands, as before.
  - It is now possible to define a drop-down menu for a button in the Toolbar object.

Changes in the Database

- catman now supports more channels.
- New export formats support the programs FlexPro® and Remus®.

4.5 What was new in version 3.0?

The significant changes were:

- The Database configuration has moved into a separate dialog window (“Options → Configure Database”).
- The length of each Database channel can be defined individually. Consequently the hard disk volume is used much more efficiently, because the channel length is adapted closely to the expected data volume when acquiring measurement data.

- The size of the Database is no more bound to the 2GB limit (unless Windows 95, Release A is used.)
- The Database can be restored after a system failure.
- The names and units for I/O and Database channels being assigned to each other can be named or given independently. When assigning an I/O channel to a Database channel for the first time, the Database channel's name and unit is identical to the I/O channel. Afterwards these properties can be modified independently.
- version 3.0 also enables text to be entered in the Database channels.
- Single value operations (read or write) within the Database are sped up by read/write caches.
- The parameters being set in Spider8 and MGCplus devices will be stored in the Database together with the measured values and measuring interval. (Traceability of measurements).
- The editor for Online Documents offers now a document-wide overview of all graphical objects. Therefore it is possible to display or delete several objects on a single button click.
- Two new graphical objects (2D Scientific Graph and Scientific Polar / Rose diagram) offer improved display capabilities and export functions. Graphics created by means of these functions can be exported into the Metafile-, BMP- and JPEG-format.
- The newly introduced HTML Browser object enables any HTML file to be viewed at runtime (only then!) and offers any functionality on the basis of the Internet Explorer installed on your system.
- From now on it is possible in the Database channel manager, to adapt channels of different length by reduction or expansion.
- The ML801 amplifier type (used in the MGCplus) with its newly introduced sub-channels can be configured via catman.
- The MGCplus option for acquiring data with several sample rates is supported by catman 3.0.
- The Script editor, which automatically opens after a compiler test cycle, offers an automatic compiler function. After opening the editor the cursor jumps back to the recently edited line. External files being referenced by INCLUDE commands can be directly edited.

4.6 What was new in version 2.2?

The significant changes were:

- Project management has been completely revised and changed into a functionality. It provides improved support for daily work, since all relevant files are displayed in the same fashion as the Explorer window. In addition, a larger number of files can be associated with each area of a project (device setup, data etc.).
- It is now possible to operate *different* devices using *different* sample rates simultaneously. This applies to both catModules and the Measurement Wizard.
- The menu in the main catman window (the main menu) remains visible even when worksheets are open and it no longer changes, since each worksheet now has its own menu bar and toolbar.
- The toolbar below the menu lines has therefore been taken out. A number of new icons have also been added, partly so that frequently used functions can be launched more quickly, and partly so that buttons can be spared from the worksheets to improve their appearance and readability.
- The worksheet formerly known as I/O channels or I/O channel definition is now called *I/O Definition*.
- The *Access Authorization* function has been removed.
- In the Online Document editor, the functions of the table object and the list object have been taken over by the new Spreadsheet object. This allows a simpler yet more flexible configuration and resembles a table in Excel.
- Exchanging data via Excel has been improved: a single mouse click is all it takes to call up Excel and copy all data from the Database to an Excel worksheet.
- Logging of device settings has been improved: just select an area of the Spreadsheet and all device settings will be written into consecutive fields. Special script commands also make it easier to log all settings with the aid of script.
- catModules have been revised in response to many requests and now offer more options.
- You can now use script to create your own device drivers, and these can then be called into the I/O Definition just as easily as HBM device drivers.
- The introduction of the Script development system considerably simplifies the management and supervision of large projects. Individual subroutines can be saved as files, several code modules can be contained in one project, the subroutines in a module can be called up from within another etc.

- The Script editor allows code to be color-contrasted.
- Readings can be taken from hardware so that all settings can be logged.
- It has been made easier to modify device parameters outside device setups (filter settings, taring, etc.).
- Due to the new option to operate a number of devices using different data transfer rates, certain commands have been modified or enhanced in the case of data acquisition.
- FOR...NEXT has been implemented.
- The Profibus DP and CAN bus interfaces have been integrated.
- Conditional compilation is now possible.
- The correct read-block size will be determined by catman automatically.
- The Groupable switch function from version 2.1 has been taken over by the On / off switch and LED object.

4.7 What was new in version 2.1?

The significant changes were:

1. Ever since this version, catman has had three operating levels: catModules, a Measurement Wizard (which works with Online Documents you have created) and script. The Measurement Wizard offers you a basic measurement setup without the need to create your own script. When you have created an Online Document to display your measured values and then want to execute periodic or continuous measurement, open the Measurement Wizard to configure and start the measurement procedure.
2. You may use the Auto Command List (“Measure” menu) to create and store actions which will be automatically executed in a measurement procedure without having to have detailed knowledge of the script language.
3. Three further catModules are available:

- You can use the Long-term measurement catModule to carry out cyclic measurements over long periods.
- The catModule Frequency analysis performs an online frequency analysis of up to two signals (autospectrum, amplitude-phase spectrum, cross spectrum). Here you can choose between the modes “*Single shot*”, “*Repeated single shots*” and “*Continuous sliding window*”.

4. The Database editor has been revised and now has an integrated print function.
5. The print manager has been redesigned and its functionality has been enhanced. In Print Preview it lets you review and print individual pages. Print documents can also be stored in a file for administration purposes.
6. The scope of the basic and post-process mathematical functions has been extended, for example with new functions for frequency analysis (amplitude-phase spectrum, cross spectrum, complex FFT), for rainflow analysis, and digital filters.
7. Innovations with the script language:
   - One of the most significant changes in version 2.1 is the introduction of arrays. catman V2.1 can manage arrays with up to 3 dimensions.
   - The debugging tool has been revised. You can now set any breakpoints you like in any subroutine and use a popup window to examine the contents of variables.
   - script functions for supporting MGC*press* and MGCplus.
   - script offers a new interface for communication to MS Excel. It is based on the facility known as OLE Automation. The script language provides you with an EXCEL object for this purpose.
   - Commands, which can be applied to channel ranges of the I/O Definition (`ACQTAR, ACQTAS, ACQCAL, ACQCPV, ACQLIV`).
8. Support for the Ethernet interface with TCP/IP.

5 Technical support

Should your installation not run perfectly or if you have a problem with catman® Professional, please contact our Hotline in Darmstadt.

☞ Please always have your license number ready so that we know which version you are using. You will find this on your catman CD or in the catman info window (“About catman” in menu “Help”).

E-mail support

Software@hbm.com

Telephone support

The telephone support is available on workdays from 14:00 to 15:00 hours Central European Time.

+49 6151 803-373 (International)

06151 803-373 (from within Germany)

☞ Extended support can be obtained through a service contract.

Fax support

+49 6151 803-9373 (International)

06151 803-9373 (from within Germany)

HBM in the Internet

http://www.hbm.com

Please do us a favor:

If you come across an error in catman® Professional, please let us know. Sub-directory BUGRPT is created in the catman directory during installation. This contains a form called SWQUEST. You may use an Internet browser to call it up and fill it out. Then please either print it out or send it to us by e-mail. Thank you for your assistance. This will support our efforts towards the continuous development and enhancement of catman® Professional.

B Overview

1 Working with catman® Professional

In the current version the catman® Professional program is a very versatile measurement technique tool which offers extensive facilities for acquisition, processing and visualization. This chapter is designed to introduce you to the central theme connecting the individual components of catman® Professional. As you will see, handling a complex program like catman® Professional need not be more difficult than driving an automobile with automatic gears: start up, pick your route, foot on the throttle.

However, before you venture onto the road with a motor vehicle you do need to know the *highway code*. With catman, too, there are some *basic facts* that you ought to know about if you are to achieve the results you want in a reasonable time. We shall therefore start with some background information on the structure and working methods of catman. You will need this in order to understand such things as the difference in the way real-time data and post-process data are handled. This difference has an effect in all areas, from acquisition and display right through to data storage.

The sections below will show you the following:

- The different measurement procedures that catman® Professional provides (four operating levels), what options are available for carrying out measurement, analysis and evaluation etc. in each case, and how long it is likely to take you to learn and start using each operating level.
- How much help catman® Professional can provide to make your work easier.
- How to perform a data acquisition using catman® Professional and Spider8 or MGCplus (Quick Start).
- How to use the catman® Professional online Help system.

☞ This last point is particularly important for your everyday work, since this manual concentrates on the concepts and does not go into *all* details like a reference work.

In catman the right mouse button is used for calling up a *context menu*, in which you then use the left mouse button to select the function you want. You may also directly call the configuration dialog most frequently used for the current element by double-clicking with the left mouse button.

2 catman's core structure and method of working

catman is a highly complex program with a modular structure. You can use it to control a wide variety of measuring instruments (including third-party devices) for the purpose of acquiring measurement data. Data can be processed either during acquisition (real time) or after measurement has taken place (post-process). Data can be exported or imported at will, redisplayed, used in computations, and so on.

Many different kinds of pages can be displayed: One and the same data series can be represented in various ways, perhaps in order to show an overview and an excerpt simultaneously, or in order to fit a visual display and a logfile printout onto their respective screen and paper media. So that this overview does not get submerged beneath the wealth of available options, we shall first of all start with the basics and show you a diagram to illustrate the flow of data and commands.

The four most important components of catman® Professional are:

- The Device drivers
- The central Measurement and Control unit
- The Online Document with its Online Pages
- The Database

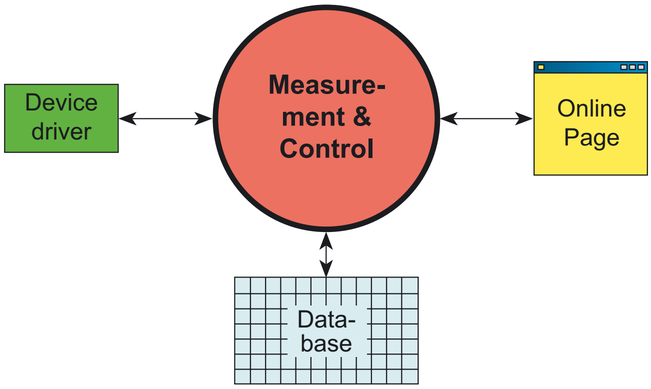


Figure 3: catman® Professional's core components

2.1 catman’s core structure

The device drivers

Device drivers are specific to each device and are installed according to requirements. They are used for the communication or transfer of measured values. Of course, several drivers can be active, i.e. you can combine not only devices of the same type, but also different ones. You are free to choose the interfaces over which the particular device drivers and amplifiers work together, and you can mix them as you wish (e.g. IEC, Ethernet and RS-232). With bus-compliant interfaces (IEC, RS-485, Ethernet) you can also operate several devices of the same type via the same interface. You can even use the Script development system to create your own drivers for other, third-party devices. Drivers of this type are named *Script Drivers*.

The central measurement and control unit

catman’s *central control unit* monitors the interplay between individual components and loads the necessary device drivers from disk. This central control unit therefore has to receive information about the devices, interfaces and device drivers in use. So that the *central unit* also knows which channels on each device will be used for measurement and which will be used for output, everything is defined in something called a *worksheet*. Over and above this, it also implements real-time mathematical functions at this point. The whole Setup procedure starts therefore with the *I/O Definition* worksheet.

When the devices, the channels they are going to use, and possibly a number of real-time computations have been defined, you can start working with catman® Professional at one of the *four operating levels*.

The Online Document

In the first place the Online Document is responsible for displaying data. You may build it up from as many pages as you like (known as Online Pages) and each Online Page may contain both real-time displays and post-process (off-line) objects. In the second place, all user entries take place via buttons, text boxes, lists, etc. using these pages. For changes or for creating new pages, the *Online Document editor* worksheet is used.

The Database

All measured values required for further processing must be saved in the Database. If you just want to monitor a measurement channel but do not need the measured values themselves, you do not need to save them, although you can display the values concerned in a *real-time graph.* The Database is usually created on your hard disk when you start catman.

You can assign an appropriate length and data format (numeric, text) to each measuring channel in the Database for the efficient use of hard disk memory. So that it does not permanently take up space on your hard disk, the Database is deleted (as a default) again when you exit from catman. You can get access to the stored values via the *Database* worksheet.

The illustration Fig. 3 on page B-4 is merely the basic operating diagram, but further components also exist, as you can see from Fig. 4 below.

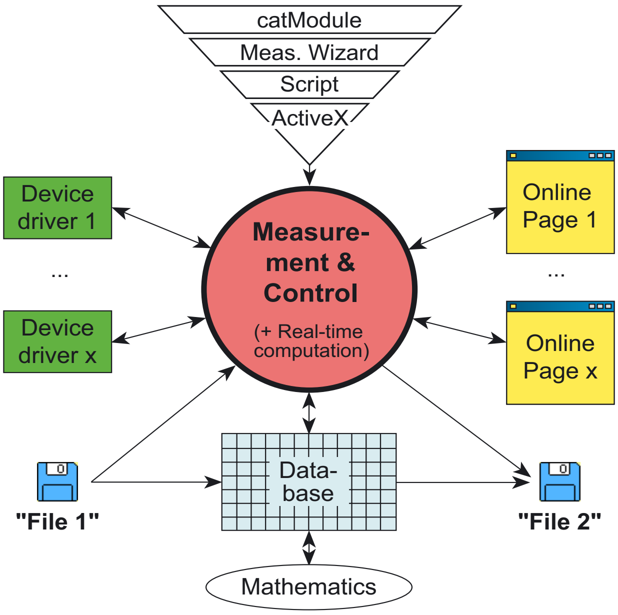


Figure 4: Full functionality of catman® Professional

2.2 catman's method of working

All catman® Professional components are shown in Fig. 4 on page B-6. You will see, for example, that another component exists for post-process mathematics, and this can carry out further processing on data from the Database. Post-process mathematics is tied to the presence of data in the Database and can be accessed through the Database worksheet. Data can of course be imported and exported to and from the Database, and this can even be done in real time whilst measurement is in progress. The triangle above the central unit contains the component of most significance for working practically with catman® Professional. Depending on the operating level, this provides another operating option:

- Measurement with the aid of one of the catModules
- Measurement with the aid of the Measurement Wizard
- Control using a script
- Control using ActiveX

The possible data flows during data processing are shown by arrows in the diagram. There is an important difference in the types of data that are available to be processed and displayed: data that has just been measured (1) or data taken from the Database (2).

1. Data is passed on direct from a device driver (or from a file) to the Online Page graph(s).

   These values are *real-time data* and are only available until the next values are read. The term IO (IO stands for Input/Output) is used for labelling as a means of identification in dialogs. Since the data is usually read in the form of small *blocks*, over 100 values may be present simultaneously per channel, depending on the data transfer rate. However, as a rule these change at least 10 times a second, and are therefore very quickly replaced by new ones.

2. Data is stored in the Database.

   The values here are then available for further computations, display etc., even at a later time, until they are overwritten by another measurement or computation, or are explicitly deleted, or until you exit catman. When this data is displayed in dialog boxes, it is often labelled DB (DB stands for **D**ata**B**ase). A data series may contain measured values, calculated values, values computed using an algorithm without reference to measurements, or values read in from a file.

Not all graphic objects can display real-time data. For y(t) and y(x) real-time display, there are special graphic objects that have their own supplementary buffer and can therefore display real-time data. A normal graph would only be able to display the current read block (the acquisition data usually is not read-in value by value but in the form of blocks) and could therefore only show the measured data for a fraction of a second. In the case of objects that can display only one value at a time (such as an Analog meter or a Digital indicator) only the first value in each (real-time) read block is used.

The components within catman® Professional are accessed with the aid of *worksheets* (see menu items in the main menu):

- Device drivers, complete with the channels and (real-time) computations that will be used in the data acquisition, are defined in the I/O Definition worksheet.
- Online Pages are designed in the Online Document editor worksheet.
- The Database and post-process mathematics are accessed via the Database worksheet.
- If you want to define your own measurement or testing sequences, you must start the Script development system with the aid of the Script editor worksheet.
- Information about VBScript and the ActiveX interface from catman can be found in the ActiveX reference (online Help only).

So that you do not always have to go via the menus to carry out frequently needed actions, a row of icons called a *toolbar* is located below the menu for a given window. If the window is active, *tool tips* are shown if you leave the mouse pointer on an icon for a couple of seconds. On the extreme right of this group of icons you will always find three icons representing the other worksheets. You can use them to switch quickly to the corresponding worksheet without having to use the main menu.

In catman the right mouse button is used for calling up a *context menu*, in which you then use the left mouse button to select the function you want. You may also directly call the configuration most frequently used dialog box for the current element by double-clicking with the left mouse button.

3 The four operating levels of catman® Professional

For the most part, carrying out a measurement task can be divided into three sub-tasks:

1. Set up the devices (*interactive mode*)
2. Carry out the measurement (*run-mode*)
3. Analyze and present the results (interactive mode or controlled by a script in run-mode)

In order to terminate the so-called run-mode (when using a catModule or the Measurement Wizard) you have to use either a button (catModule: Exit), a menu item or close the Online Document window with .

In practice, however, the actual implementation may be quite different, depending on the overall conditions. Quite commonly an onsite measurement is needed in the short term, but further work has to be deferred until it can be analyzed in the laboratory at a later time. Other measurements may require special preparation since they can only be done once, or real-time computation has to take place during the session for control purposes. If you intend to operate a test-rig, you need specially adapted software that will not only carry out the measurement but also control the test procedure.

catman® Professional has the built-in flexibility to meet all the requirements described above. So that catman can provide an optimum solution for all such widely differing applications, it has four different operating levels, giving you four possible ways of using catman® Professional to carry out measurements:

1. Level 1

   Using catModules that contain prebuilt outputs or displays (Online Pages).

   Total time needed to learn all basic procedures: 2 hours.

2. Level 2

   Using the Measurement Wizard in conjunction with Online Pages that you have created yourself.

   Total time needed to learn all basic procedures: 4 hours. If you intend to use existing Online Pages only 2.5 hours are necessary. In this case you only need to learn the functions of the Measurement Wizard.

3. Level 3

   Not only configuring your Online Pages yourself, but also creating your own process run, that is, your own script.

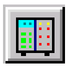
Total time needed to learn the basic procedures: around 2 to 3 days.

4. Level 4

   You work in a programming language of your choice and control catman via the catman object. This level is only suitable for experienced programmers and requires profound knowledge of the programming language used. You will find further information and a description of the properties and methods of the `catman` object in the ActiveX reference. As Level 4 is intended for programmers only, this manual does not explicitly refer to this topic. However, the settings required for catman are the same as in Levels two or three.

Preparations: All four options require you to go through a particular step which is described in Chapter D, *Defining Devices and Channels*, namely setting up the devices and configuring the I/O Definition. Calling the I/O Definition worksheet is therefore nearly always the first thing you do in catman. Set aside half an hour to get to know the most important settings. If you will be wanting to carry out real-time computation, digital I/Os or the like, you should take time to go through the chapter thoroughly and carefully (about 2-3 hours).

If you often work with the same configuration, you can go to the main menu and use “Options → Startup configuration” to specify that the I/O Definition you last used will be loaded every time you start catman. Alternatively you can specify a number of I/O Definitions to appear in the project window, and you may then activate one of them by double-clicking on it. You can also stipulate that a particular project will open automatically when you start catman® Professional. This also allows you to specify that a number of other settings are loaded, such as device setups, see Section 4, *Project view*, page B-13.



Having defined which devices will be used, you can configure via the Sensor Database or with the individual channels, if this has not already been done. You then need to decide on one of the three Levels. A common feature of all three levels is setting up the required sample rate, which may differ from one device to another when the configuration includes several devices. The optional triggering function may also need to be set up. For this reason these aspects are dealt with in a chapter of their own called Chapter E, *Measuring with catman® Professional*. How long it takes you to complete this chapter depends on whether you intend to use just one device (around 0.5 hours) or several different devices (around 1 hour).

Level 1: Data acquisition with catModules

This is certainly the quickest way of carrying out a measurement, since hardly any preparation is needed. You just have to specify which channels will be displayed in which graphs and define the scaling for real-time graphs. One disadvantage is that catModules have only a few different pages for output, and you cannot amend how they are displayed or the type and size of their output objects. If you have already read Chapter D, *Defining Devices and Channels*, of the manual, it will take you around another half hour to learn about the options available to you from one of the modules. You will find the information you need in Chapter F, *catModules and Add-Ins*. A quick introduction to this level provides Section 5, *Getting started (Quick Start)*, page B-22 in this Chapter.

Level 2: Data acquisition with the Measurement Wizard

At this level you are no longer tied by the restriction on catModules regarding output pages, though you do have to create your own output objects on one or more pages. You can also use Online Pages which have been prepared by another user or choose from those available in the Measurement Wizard.

Since not all objects are equally suitable, you must first of all get to know the boundary conditions affecting their use. For this you will need to work through Chapter G, *Online Document*, which will take around 1.5 hours. You will then be able to create your own pages in a matter of a few minutes. In order to take measurements you have to learn how to use the Measurement Wizard (Chapter H). The Wizard offers a whole range of options (approx. 0.5 hours). (Do not forget Chapter E, *Measuring with catman® Professional*, and Chapter D, *Defining Devices and Channels*.)

Level 3: Data acquisition with a script

As well as finding out how to design Online Pages (or documents) you must also learn about the Script development system. This may sound daunting, but thanks to catman's Code Builder and the comprehensive online Help it is not difficult to write a script for acquiring and analyzing measurements after just two or three days familiarization. This level is usually the only effective method to use for controlling a test-rig, but if the test-rig already runs in stand-alone mode, Level 2 is often a practicable and easier alternative. There are so many facilities available in Level 3 that where large-scale projects are concerned you should be prepared for a correspondingly longer development time. A knowledge of how catman works is however helpful in Level 3.

☞ If you are short of time or capacity, just get in touch with HBM. We have our own department dedicated to producing tailor-made script programs.

Level 4: **Data acquisition with ActiveX**

In Level 4 you need only know the commands and objects which are required for the data acquisition. The visualization is not absolutely necessary. This depends though on the project you have planned and must be decided on a case-to-case basis. A knowledge of how catman works is however helpful in Level 4.

Post-processing: Post-processing after a measurement is carried out identically at Levels 1 and 2. At Level 3 it is usually carried out straight from a script, so that the user needs to take as few decisions as possible. Once the data has been stored in the Database, you can use post-process mathematics to perform computations.

Since computations are only executed when you click on the “*Compute*” button in the computation window, if there was a particular measurement that always needed the same computations to be carried out, you would have to call up the dialog box for this every time. Repetitive actions of this kind can be automated thanks to the *Auto Command List* (Chapter J, *The Auto Command List*). This is an intermediate step that is less complex than Level 3 and greatly enhances the facilities available at Levels 1 or 2. In particular it is very easy to define repetitive computations. Also the ability to use a single keystroke to carry out actions that are not provided in the catModule or the Measurement Wizard is a very attractive feature. The more complex the actions the greater the degree of difficulty, and if such actions are far-reaching, Level 3 achieves your aims faster.

4 Project view

You can use catman’s Project window to edit, organize and manage all the components of a measuring project. You can combine any I/O Definitions, Online Pages, device setups, scripts, and so on up to Measurement Wizard and QuickView window settings in a project. Additionally the catModules and licensed Add-Ins are listed in this window. If the size of a worksheet is not optimally adjusted to the existing space, this can be done from the main menu using “Worksheet → Fit window”. If the Project window is not required you can remove it from the screen with “Project view” in the menu “Worksheet”.

Double clicking on an entry in the Project window loads the corresponding file or activates the respective worksheet.

Only one project can be open at a time.

You may create two types of project files:

1. A standard project file (*.CPJ) contains only *cross-references* to the individual component files, not the files itself.

   If you rename or move a file it cannot be called up again; there is, however, the option to search for the file.

2. A packed project file (*.CPR) contains *all* related files in compressed form and is suitable for transferring projects to other PCs.

   Activate “*Pack project files for deployment*” to create this type of project file. If the project has been saved once already, use menu “File → Save project as”.

Information on the file, such as the file path or last editing date, can be obtained via the context menu “File properties”.

Go to the “Options → Projects” main menu to set up the worksheets or settings that should be transferred into a project as a file reference.

4.1 Save/Open Projects

Set up the worksheets or settings, which should be transferred as a file reference into the project from the “Options → Projects” main menu. Create a packed project if you need the files on a different PC.

If you have not saved any worksheets or measurement settings, activate the “Save project” command from the “File” menu to save the information from the existing worksheets and the Measurement Wizard settings. File names will be assigned in accordance with the project name, only the file extensions will be adjusted.

If you activate option “*Add measurement data from Database*” the actual data is saved every time you save the project. In this case old data already stored in the project is overwritten.

If you have not yet created device setup files, you can create and add these automatically for the MGCplus and Spider8: Context menu “Add → Current device settings”.

You can specify particular start-up options for your projects on saving them and thus automate certain tasks when loading the project. As well as loading and activating the first I/O Definition (if available) and the (first) Auto Command List (if available), you can specify in the frame “Execute upon loading” of the “Create project” file dialog window the following options additionally:

- Execute device setups: The device setups of the project will be loaded into the devices. The device setup files are transferred to the devices in the same sequence as the devices appear in the device list, one setup for each device. If the connection to a particular device is interrupted, the setup for that device will be ignored.

  A file is not checked to see whether it contains a suitable setup for a particular device. An error message will be issued if the file cannot be loaded into the device.
- Open Online Document editor
- Execute first Favorites script
- Import measurement data: The first data file is loaded into the Database
- Start Measurement Wizard: The Measurement Wizard is started with the first settings file
- Run the first VBScript: the VBScript is loaded into the Windows Script Host (WSH) and executed.

- Load the first VBScript: the VBScript is loaded into the Windows Script Host.

☞ When opening an I/O Definition, the current measurement settings, the current settings of the Event monitor and the trigger definitions are overwritten.

When manually loading a project, catman will display the loading options for this project which can then be modified. If the project is opened automatically at the start of catman, the options saved in the project are executed. If you open a packed project, all files will be extracted into the same directory and then opened.

☞ Opening a project at the start of catman is set with “Options → Startup options” in catman’s main window, see Chapter C, Section 4.1, *Automatically load project at startup*, page C-14. If you wish to activate the Measurement Wizard immediately after loading a project, enter the Online Document you wish to start in the Measurement Wizard setup. This document will then be opened when the measurement starts. A document entered in the project will only be opened, if you specified “*Open Online Document editor*” for “Execute upon loading” on saving.

Since apart from the device setups only those files are loaded which are at the beginning of a project group you can change the position of the files in a group:

1. Right-click on the corresponding component file.
2. From the context menu select “Move to first position”.

4.2 How to add a component file to a project

To add a component file, proceed as follows:

1. Right-click on the group or a member of the group to which you want to add a file. From the context menu select “Add”.

   or

2. From the “File” menu of the catman main window select “Add to project group”. From the selection displayed select “Current document”.

   or

3. From the selection displayed select “... from file” and choose the file you want.

☞ New components will be added onto the end of a group. A message will be issued if the name selected has already been assigned to another component.

You can create and add a device setup automatically for the MGCplus and Spider8: Context menu “Add → Current device settings”.

4.3 Files and file types of project components

Double-clicking a project group with worksheets opens an empty or—if available—the first worksheet of that group.

When a worksheet is loaded, any previous settings that have not been saved will be lost.

Project (*.CPJ, *.CPR)
In a standard project file only the directory information of the individual component files of the project are stored. Create a *packed project*, if you want to transfer a project to another computer: “Save project as” with option “*Pack project files for deployment*” (*.CPR). Otherwise you must also transfer all the component files with the corresponding directory information in addition to the project file.

You can also edit the *.CPJ file with a text editor.

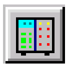
I/O Definitions (*.IOD)

In this project group I/O Definition files are managed.

Device setups

The device setups are managed in this project group. Double-clicking the project group executes all the device setups one after the other. If a device or a device channel is not present, the corresponding setup is skipped. If you just wish to set up a certain device, mark the device in the list at the top of the worksheet I/O Definition and then double click on the device setup file.

☞ With the device setups, the first file is used for the device first defined, the second for the second device and so on. You must therefore modify the order of the files where necessary.



If no setup files exist, this can be done for the Spider8 and MGCplus from the context menu “Add → Current device settings”. For all other devices, select a device in the I/O Definition and click or select “Setup device” in menu “Device”. The device setup module will be called up and the device setup window will open. Save the setup to file before leaving the setup module, and then add this file to the Device setups group.

It may take up more than 60 seconds to create a device setup file for a filly equipped MGCplus.

*.MGC	MGCplus device setup
*.MLX	MGCplus channel setup
*.MCS	MGC device setup
*.S8	Spider8 device *or* channel setup
*.SP8	outdated format for Spider8 device setup
*.DPS	DMC*plus* device setup

Table 1: File extensions for HBM device setups

*.D9S	DMC9012A device setup
*.CFG	UPM60/UPM100 device setup
*.D4S	DMP40 device setup
*.MVD	MVD2555 device setup
*.SCO	Scout55 device setup

Table 1: File extensions for HBM device setups

catModules

In this project group, you find the various catModules of catman® Professional. Double-clicking a catModule will execute it. You can also use “Execute catModule” in the “Measure” menu.

The list of catModules always forms a dedicated group in the project window. If only certain modules are to be displayed there, it can be changed via the catModule window or with a text editor: MODULES.CAT file in the CATMOD sub-directory. However, you should first save the original list under a different name.

Online Documents (*.OPG)

In this project group Online Documents are managed (Chapter G, *Online Document*).

If you would also like to activate the Measurement Wizard with a certain Online Document immediately on loading a project, you should state in the settings for the Measurement Wizard which Online Document the Measurement Wizard is to start. This document is then also opened on starting with a project. A document given in the project is only opened if you specify in “Execute upon loading”: “*Open Online Document editor*” on saving.

Measurement Wizard (*.WIZ)

In this project group Measurement Wizard settings are managed (Chapter H, *The Measurement Wizard*).

Depending on the Measurement Wizard settings, files which should be saved during the measurement (Online data export) might be overwritten with a new measurement. Therefore, specify either a counter for the file name, activate the request for a new file name or append the data to a file. When saving measurement data periodically (after each period), the file name will be checked before saving.

Database Manager

In this project group data files are managed. Double-clicking opens the Database channel manager (Chapter I, Section 2, *The Database*, page I-4). Double-clicking a particular set of data opens the Database channel manager and loads the data file into the Database. If no Marker has been set in the Database channel manager, the data will be saved starting from the first channel. If data will be overwritten, catman® Professional will request confirmation.

*.BIN	data file containing binary data
*.DAT	Database file
*.ASC	data file in ASCII format

Table 2: Common file extensions

☞ When saving data, file extensions will be suggested. However, you do not have to accept these, other file extensions may be used.

Data Explorer

Double-clicking opens the Data Explorer (Chapter I, Section 2.5.1, *The Data Explorer*, page I-15).

Test Manager (*.TST)

Double click on the entry to open the Test Manager dialog window (Chapter E, Section 5, *Test Manager*, page E-21). The Test Manager information is saved in files that have the extension TST. However, these files cannot be inserted into a project.

QuickView windows (.QVT)

Files with settings for a QuickView window can be included into this group (Chapter I, Section 4, *QuickView*, page I-39).

Auto Command Lists (*.ACL)

In this project group Auto Command Lists are managed. After loading a project the first Auto Command List is always active (Chapter J, *The Auto Command List*).

Script project (*.SCP)

In this project group Script projects are managed (Chapter K, *The Script Development System*). In the Script editor you can refer to existing modules, subprograms or compiled object code files.

*.SCT	Script modules (text form)
*.OBJ	Object modules
*.SUB	Subroutines (SUB programs)

Table 3: File extensions for script related files

Favorites (*.SCB)

In this project group executable scripts, i.e. previously compiled scripts (*.SCB), are managed. Double-clicking on a script will execute it.

Add-Ins

Programs which access catman via ActiveX are combined in this project group. The programs must be entered in the file ADDINS.CAT, see ActiveX reference. In catman® Professional you can find the data visualization with the module DataView (Chapter F, Section 8, *DataView*, page F-24). Further programs can be obtained from HBM (MD Explorer) and can then be accessed from this group after licensing. Double click on an entry to run the program.

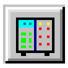
VBScript (*.VBS)

Microsoft Visual Basic scripts, abbreviation VBScripts, can be included in this project group. Double click on a script to run it.

5 Getting started (Quick Start)

☞ You must have power user access rights for working with catman. Additionally write access must be permitted for the catman directory and sub-directories.

The section below shows you how to carry out measurements using one device and a cat-Module. Even if some of the necessary settings have already been discussed in other Chapters, we shall run through all the steps again here for the sake of completeness.



We are going to be using the Data logger catModule and a MGCplus, but you may also use a Spider8. For other devices, you must specify the device type manually. Then, instead of using the Sensor Database, you have to go to the I/O Definition and use  to carry out a device setup, such as setting up a half or full bridge, selecting the measuring range and so on. You should do this as soon as the device and interface used is specified in the I/O Definition worksheet. If the device is not capable of scaling the acquired data, use a user scaling in catman® Professional, see Chapter D, Section 4.9, *Scaling*, page D-42. Apart from this device setup, which depends on the settings available in the device you are using, all HBM devices are operated via catman in the same way.

If you have a MGCplus or Spider8 complete with a transducer connected, you can carry out the actions discussed here on your PC as you read.

5.1 What do you have to do?

1. Define device (activate device driver).
2. Assign active I/O channels.
3. State the required transducer scaling
4. Call the catModule.
5. Specify parameters for measurement sequence.
6. Define which channels shall be displayed in the real-time graph.
7. Run acquisition.

5.2 How to do it

5.2.1 Define device

First call up the I/O Definition worksheet, e.g. with menu “Worksheet”. If you work with MGCplus or Spider8, define the interfaces to be searched through *once* from menu “Options”, tab “General”. In the upper part of the worksheet, frame “Devices”, click on “*Device scan*”.

The device is entered into the device list (see Fig. 5 on page B-24).

5.2.2 Assign active I/O channels

From the automatic connection all channels are already connected. If you don’t need all channels, go to the “Connection” column, point to the corresponding row and select “*Not connected*” from the context menu. You can also clear channels with .

For having a unique identifier for each channel, you should name each channel in the “Name” column, e.g. “*Force1*” (see Fig. 5 on page B-24).

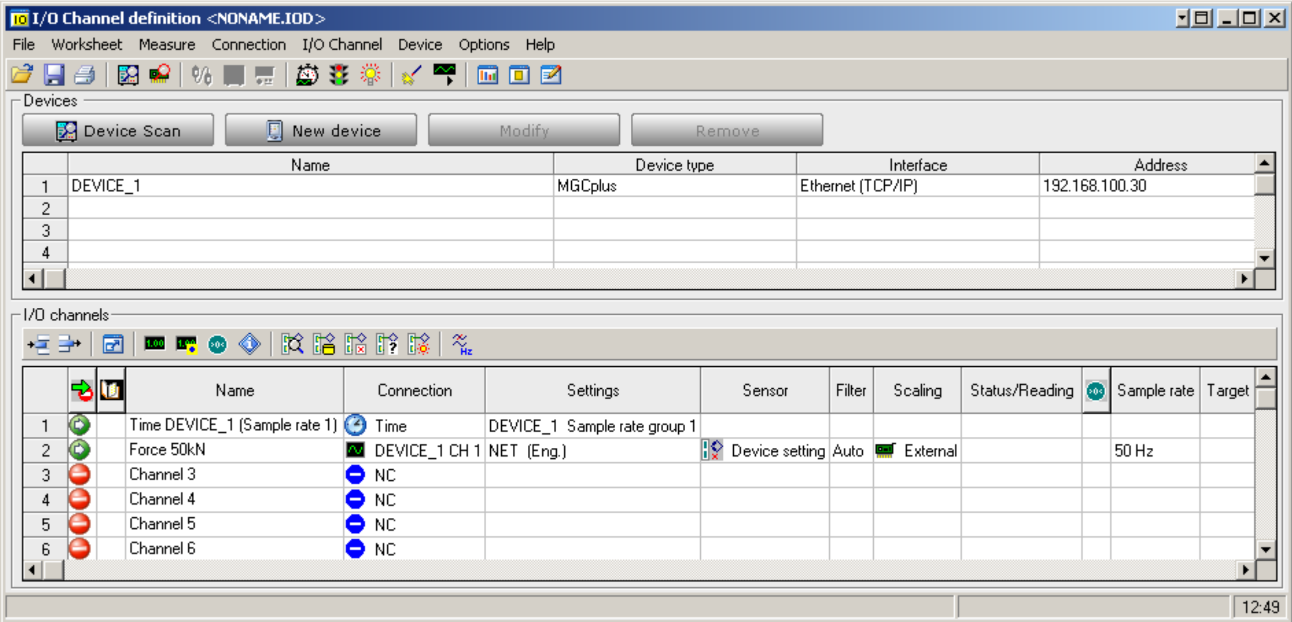


Figure 5: The I/O Definition

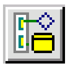
5.2.3 Define transducer scaling

☞ If your measuring chain supports the transducer identification (TEDS/T-ID) feature (MGCplus with connection plates with the identification i and appropriate transducers connected), please proceed as described in Chapter D, Section 6, *TEDS and T-ID*, page D-63.

The Sensor Database is used for the easy setup of the measuring chain according to the sensor used. You only need to specify the sensor once and you can then enter it in the I/O Definition. Instead of transducers with parameters from the calibration protocol, you can also use

Sensor templates. Sensor templates contain the typical settings for a transducer, i. e. the standard settings. For the sensitivity, for example, 2 mV/V is used even if this does not exactly conform to the transducers later used.With some transducers the tolerance to this nominal sensitivity is so slight that the measurement deviation can be neglected. In these cases you can use sensor templates. See also Chapter D, Section 5, *The Sensor Database*, page D-54.

Procedure (both cases)



1. Mark the channel to be set in the column “Sensor”.
2. Open the Sensor Database with or via the context menu “Connect sensor from Database”.
3. Select the transducer connected or a sensor template and click on “*Apply*”.

catman does the rest: Setting the excitation voltage, characteristic, unit, etc. Depending on the device, the scaling to physical units is carried out in the device or in catman (user scaling).

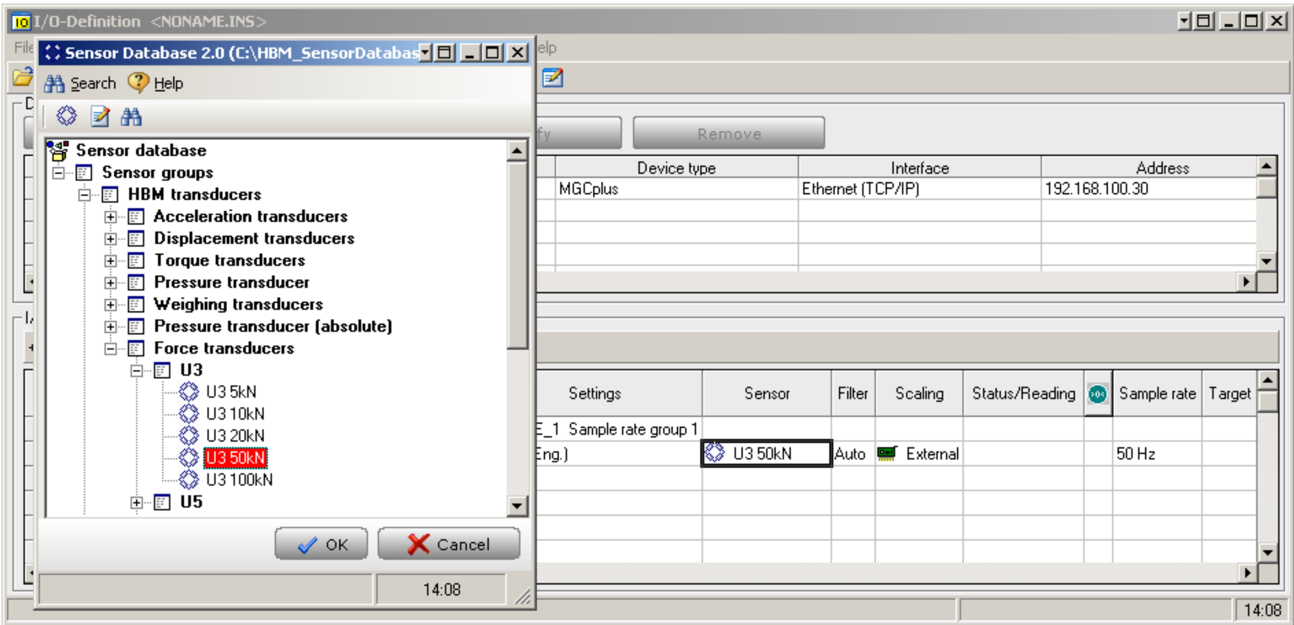


Figure 6: Choosing a sensor from the Sensor Database

We recommend that you mark all active channels in column “Filter setting” and use the setting “*Automatically from sample rate*” (Fig. 6). This prevents alias effects, see Chapter E, Section 2, *Which sample rate is the right one?*, page E-4.

Finally, save this I/O Definition: use menu “File → Save”.

5.2.4 Calling the catModule

Call the *Data logger* catModule. You may use either menu “Measure → Run catModule” and select “Measurement modules → Data logger” in this dialog or the project window by double-clicking on “Data logger” under “catModules → Measuring”. The catModule is initialized and the first page is displayed.

5.2.5 Specify parameters for data acquisition

Click on “*Measurement settings*” and input a “Sample rate”, such as “*300Hz*”. Keep the catman default “*Device internal timing*” (see Fig. 7 on page B-27).

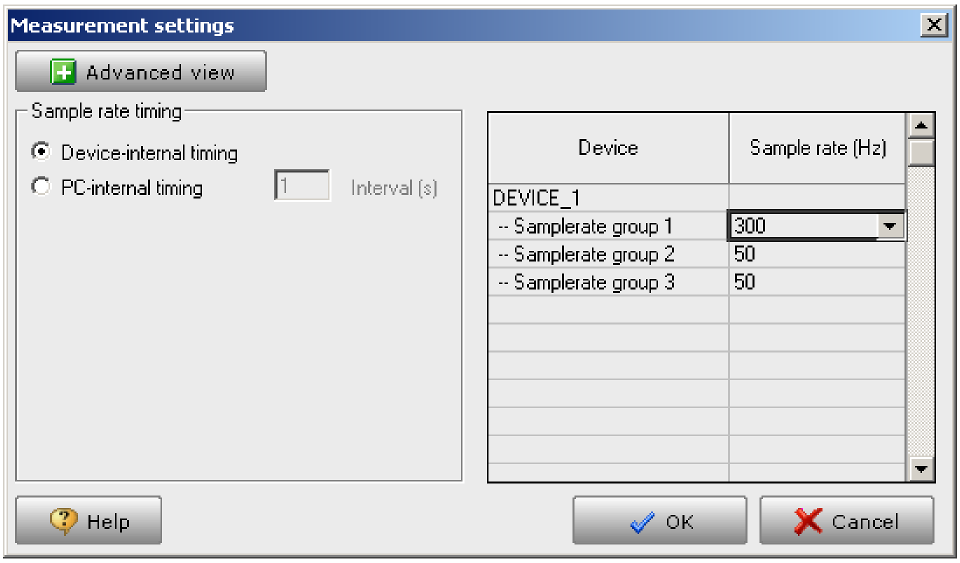


Figure 7: The measurement settings

Close the window again and make sure that the “Visualization window” you are using is “*y(t) Real-time graph*” and that “*Initialize I/O channels before start*” is checked (see Fig. 8 on page B-28). Then click on “*Run*”.

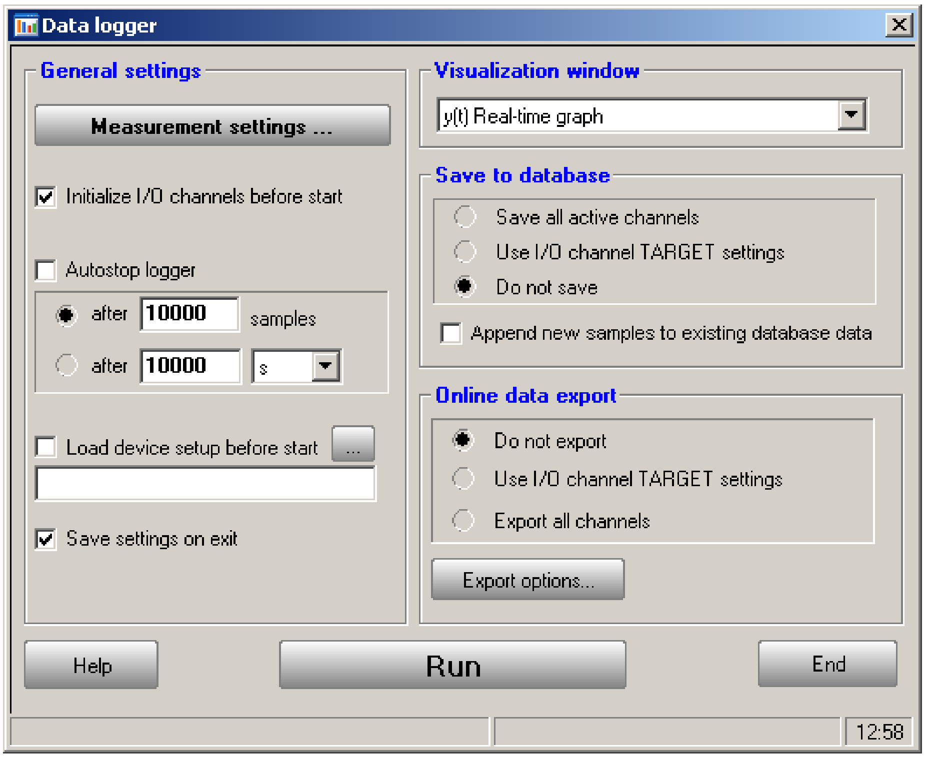


Figure 8: The data logger settings

5.2.6 Configure real-time graph

The second page of the Online Document contains a toolbar for running the acquisition, along with two output objects: Indicator panel and y(t) Scientific graph.

Now let’s define the channel to be shown in the graph: point to the graph and click the right mouse button. The context menu for this object is displayed: click on “Data sources”. The configuration dialog is displayed with tab “Plots”. If no channels are listed in the dialog box yet, drag your channel from the list “y data source” to “*Scaling layer 1*” (Fig. 9). If other, unused channels are entered, clear those channels.

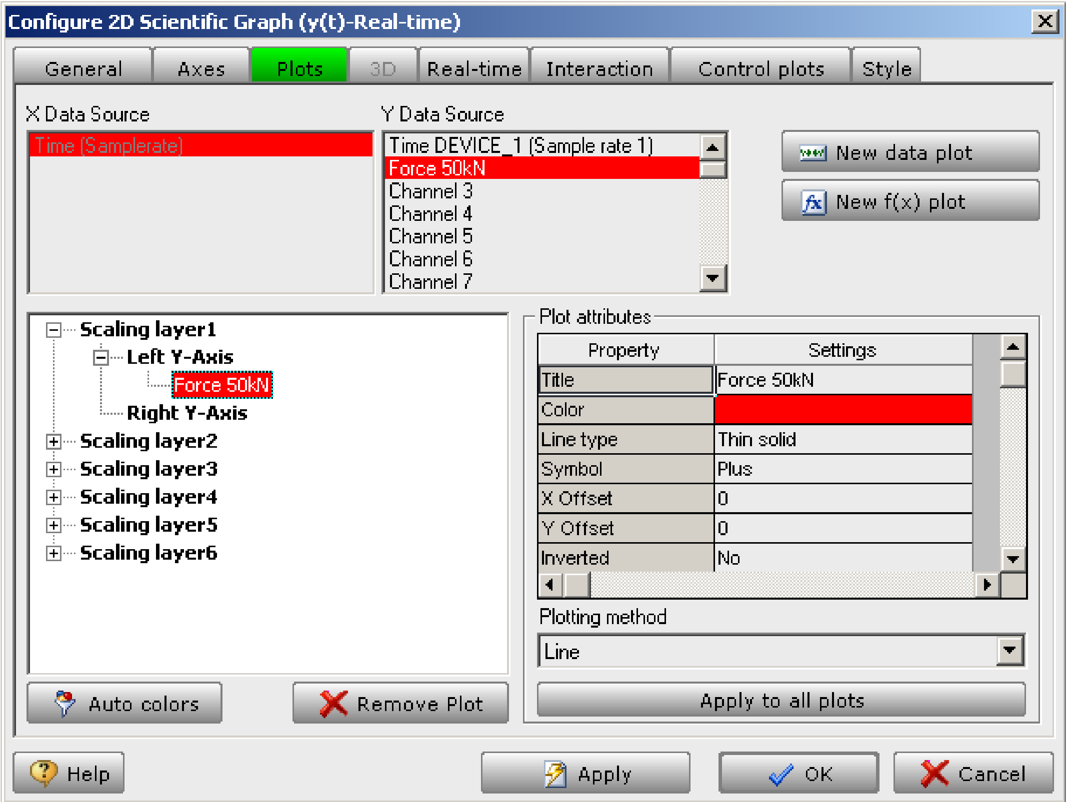


Figure 9: The settings under the “Plots” tab

This real-time graph does carry out automatic scaling. If you want to deactivate this: double - click on the left y axis. Then deactivate “*Auto*” and state in rows “*Minimum*” resp. “*Maximum*” values suitable for your transducer, e.g. “*0*” and “*50*” for a 50 kN force transducer (Fig. 10).

If you do not know which values you will get, enter the measurement range of the transducer or use “*Auto*”.

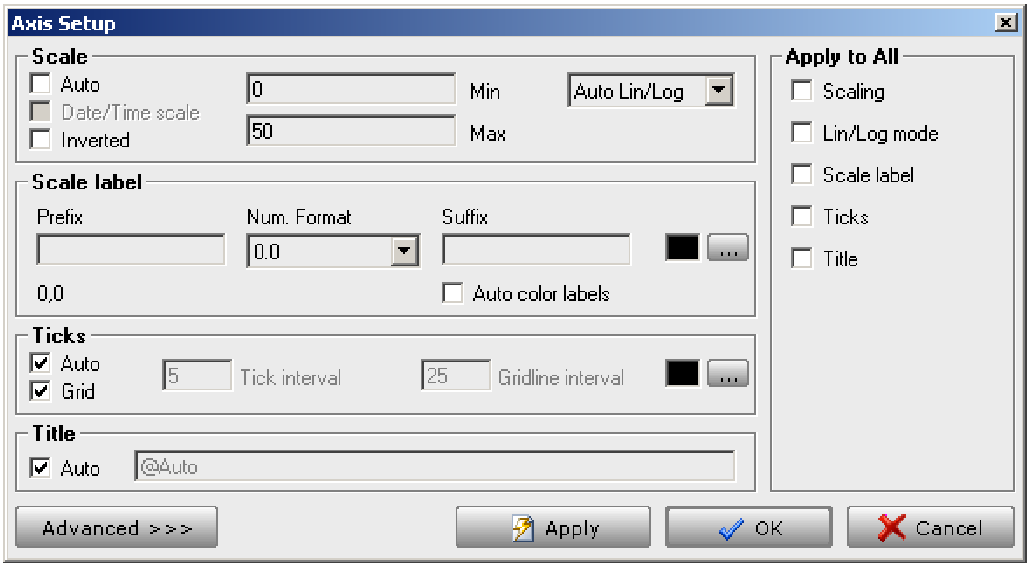


Figure 10: The settings under the “Axes setup” dialog

The catModule is now fully configured.

5.2.7 Run acquisition

Click on or press F5, and your measurement begins.
After the measurement you may use the icons at the top of the window to evaluate your data graphically, by using the Zoom and/or Cursor functions.

6 The catman® Professional online Help system

The catman® Professional online Help contains considerably more information than this manual, particularly the reference section for the script language. Strictly speaking the complete catman Help system consists of several mutually independent Help files:

- Help for the Setup programs for all HBM devices
- Introduction Help file
- Help for worksheet I/O Definition
- Help for worksheet Online Document with an explanation of how to configure the visualization objects
- Help for worksheet Database
- Help for the catModules
- Help on the Script development system

You may access the worksheet-specific help through the “Help” menu, “Help on active worksheet”. When you call this “Help on active worksheet” you usually receive the contents page of the corresponding Help file. You can use the buttons in this window to switch to the “Index” or the “Search” function.

First, use the index, not the search function, because here the individual topics are listed under key words; otherwise a simple full text search is made (all occurrences of a word or phrase).

The individual configuration dialog boxes contain a “*Help*” button that takes you straight to the corresponding information. For example, if you use the “*Help*” button in the Configure object dialog for an Online Page object, you go straight to an explanation about this object. The “*Help*” button in the “Define user scaling” dialog for I/O channels takes you straight to the topic *Scaling I/O channels*. The “*Help*” on device setup gives you a description of device functions.

Function key F1 takes you either to the Help start page (Contents) for the active worksheet or to the context-sensitive Help. In the Script development system, if you select a command and press F1 the list of Help topics about the command concerned is displayed.

You will find further information and additional links in section “The catman Help system” in the Introduction.

7 The catman log file (System log)

☞ Access to the System log file may have been withdrawn via the user administration.

Certain catman events, e.g. program start, will be automatically recorded in the *System log file*. From the “Options → Show log options” main menu, you can specify the entries that should be made. You can also specify that the Event monitor (Chapter E, Section 6, *The Event monitor*, page E-22) makes entries or make manual entries in the Log Viewer. It is also possible to change entries, extend entries by adding a comment, or assign a description file.

☞ The original entries in the underlying MS Access database cannot be changed via catman. Only additional entries can be made.

The first column in the Log Viewer (“Options → Show log options” main menu) shows the event category: stands for information, means warning and stands for error. The second column shows the origin of the entry: created by catman® Professional or script command, manual creation (“Edit → New entry”).

The events can be sorted in ascending or descending order according to the column “Date/Time”.

Description files can be used in order to save additional information (pictures, drawings texts) for the individual events. Any file format can be selected, catman only saves the cross reference to the file entered.

☞ Do not confuse the description files of the log file with the I/O Definition description files (Channel description file). Both are independent of each other and use separate lists.

Procedure for adding description files or modifying event entries

1. Open the dialog window to edit the entry from the context menu or from the menu “Edit → Modify entry”.
2. Insert a description file with .
3. Change the message displayed or the comment in the top section of the window.

   ☞ The fields in the lower section of the window (original entries) cannot be changed.

To display description files, double click on the event or use the context menu (“Edit entry”).

☞ catman can display files of types text (*.TXT), bitmap (*.BMP), JPEG (*.JPG), metafile (*.WMF), gif (*.GIF) and HTML (*.HTM, *.HTML). In other cases, the files are displayed by calling the program which is linked to it under Windows.

C Customizing catman® Professional

1 Introduction

This Chapter explains how you can do the following:

- Modify the size of the Database, that is, alter the number of channels and values that catman® Professional can save to it.
- Activate the user administration, e.g. for suppressing the start dialog or setting different access rights for individual users.
- Arrange for certain settings (I/O Definition, device setup) to be configured automatically as soon as you start catman® Professional.
- Execute a catman® Professional script as soon as you start your PC

You may make a number of other custom adaptations for the Online Document editor (Chapter G, *Online Document*) and Script editor (Chapter K, *The Script Development System*). These topics are therefore dealt with in these chapters.

In general you may modify the background color and/or font used for all worksheets in the corresponding “Options” menu or via the context menu.

2 Configuring the Database

Additional information is available in Chapter I, Section 2, *The Database*, page I-4.

When you reconfigure, all the data in the Database will be overwritten. If you have already captured some measurement data, export it in format “*catman*” first, then modify the size of the Database and reimport the data.

With the “Options” menu you can configure or change the Database.

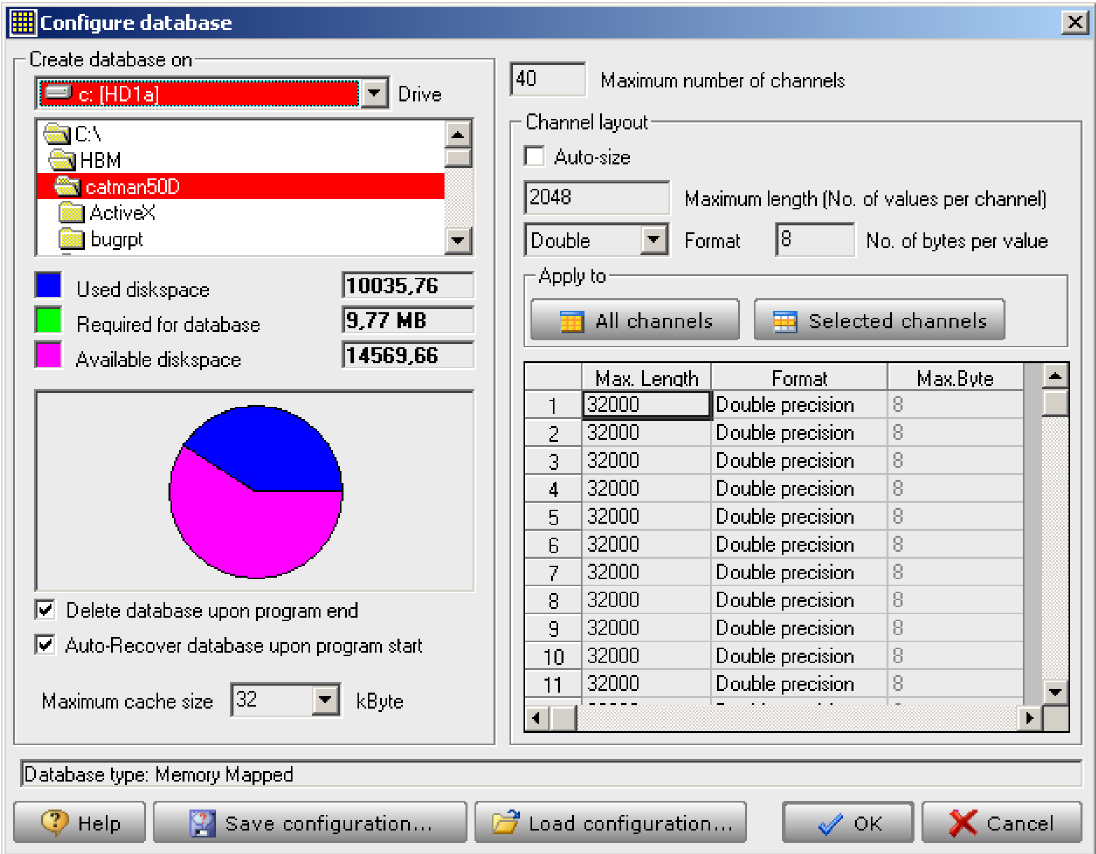


Figure 11: Dialog for configuring the Database

2.1 Define the storage location of the Database

When starting catman® Professional a (default: temporary) Database file will be created. This file contains the values of all channels. catman uses the name DB_000.$_$ and the directory of catman. If several hard disks are available, you can alternatively assign another drive and path.

Do not use a network drive here, as access to the data will be very slow.

2.2 Define the Database size

After the installation only 1,000 channels may be created. As memory is required for the administration of all *possible* channels, this setting does not take up unnecessary space. You can however increase this limit to a maximum of 10,000 channels from main menu “Options → Startup options“. This presetting for the maximum number of channels is stored in the registry and is user-independent. Administrative rights are required to change the setting. Restart catman to activate the changed setting.

With “*Maximum number of channels*” you define the required number of input, output and computation channels. This setting can be changed in the run-time mode of catman (without the need for re-starting), but all data will be overwritten. The maximum number of channels which you can enter here depends on the setting in main menu “Options → Startup options”.

Close the entry with RETURN in order to activate the setting immediately.

The “*Maximum length*” option serves to define the maximum number of measured values for each channel (data depth). The data depth is only restricted by the space which is available on the hard disk for the Database file. If you do not want to mark and set up the channels individually, click on “*Apply to all channels*”.

With less than 512 channels you can use a dynamic Database as in catman® Easy, which increases in size as long as there is space available on the hard disk: “*Auto-size*”. However, a little more time is required when saving data in such a Database, therefore this type of Database is not recommended if you are using high sample rates.

The “*Format*” option defines, which data format will be stored in the corresponding channel. At present only the numerical and text data formats will be supported.

The “*No of bytes per value*” option defines, how many bytes for storing the data format (being set before!) will be reserved at maximum. For the numerical data format 8 bytes (“*Double*”, double precision, default) or 4 bytes (“*Float*”, single precision) are selectable. In the text data format for every entry, i.e. every row of a channel, up to 32 kilobytes can be reserved. This format is usually only interesting in Level 3 if you are working with catman's script language and wish to save not only values but texts in the Database.

For configuring several, *consecutive* channels with the same settings in one step, please follow this procedure:

1. Mark the channels to be configured (hold SHIFT key and click in the left column on the channel number or click and drag with the left mouse key).
2. Set the desired values for “*Maximum length*”, “*Format*” and “*No. of bytes per value*”.
3. Click on “*Selected channels*”.

All prementioned settings influence the amount of hard disk memory occupied by the Database. After each change of the settings the required memory will be updated in the display. Please note that you will need additional space on your hard disk, in order to export data from the temporary Database file, provided that this data should be preserved.

2.3 Which settings for what?

The *correct* setting depends on several conditions. Therefore strict instructions are not possible, however, some proposals given here can give you a good *orientation*.

First of all the number of channels used for acquisition is important. Each real-time computation needs a separate channel at least. To be safe you should project some additional channels more than absolutely necessary. The number of values should normally not exceed 1,000,000 values. Please keep in mind that the restricted screen resolution limits the display of a diagram to between 800 and 1,000 pixels. For example, if you like to display 100,000 values only one pixel is used to display 100 values, all other values are only visible if zoomed in by factor 100.

If you need to acquire many values, another method of measuring may be more suitable: First write all measured values into a file, then read in all values in smaller blocks. See also Chapter F, Section 8, *DataView*, page F-24. Alternatively, you can also operate with “*Auto size*” if you have less than 512 channels. If possible, perform a test measurement when using higher sample rates, as this will initialize the necessary space on the hard disk. The data will then be saved quicker in the succeeding measurement.

2.4 Saving several Database settings and reusing them

Once you have defined settings you can save them by using the “*Save configuration*” button and activate them again with the “*Load configuration*” button.

3 User administration

3.1 Introduction to the user administration

The user administration is used in the first place to limit access to catman dialogs or functions for other users. In addition the catman start procedure can be specified from the user administration.

The user administration is initially *not* activated after the installation of catman. This first occurs when you call the user administration via the main menu “File” and mark “*Activate user administration*”. An activated user administration applies to *all* of the PC's users and not just the user currently logged on.

We recommend that the main user is defined as administrator under his Windows log-on name and that you then delete the original “*admin*” entry.

3.1.1 The possibilities for starting catman

If required, you may enter a project that will be loaded automatically when the program is started.

Case 1: catman will only be used by you, should however request password (and starting mode), as other people do have access to your PC. See Section 3.2.1, page C-10.

Case 2: catman will be used by different users. The users should log in when starting catman and can, but must not necessarily, have different access rights. This case can also be used if the PC uses a general user name when starting Windows, the users however log on under their own names. See Section 3.2.2, page C-11.

Case 3: catman will be used by different users. The users should be automatically logged on and can, but must not necessarily, have different access rights. See Section 3.2.3, page C-11.

3.1.2 User name

You can use any user name. If you use the names which you have already defined for the Windows log on, catman checks whether the user logged in to Windows is entered in the catman user list when starting up. If you have activated "*No log-in prompt upon start*" in the section "*General*" additionally, the user will be automatically logged on to catman.

3.1.3 Administrator rights

⚠ Only users with administrator rights can change their own rights and those of other users. All other users cannot call up the dialog box user administration.

☞ You will also find the settings for the log file (System log) and Add-Ins in the section "*General*".

As long as "*admin*" was marked when the new user name was created, access will be allowed to all other categories.

3.1.4 Further explanations to individual settings

Projects: Even if "*Open project*" is deactivated here for a user, you can specify a project via "*Open project upon start*". Only the user himself cannot load a project in this case.

Data acquisition: This section has the most settings possibilities. You can block individual catModules here as well as the DAQ background process or the Remote Data Server. You can also define different restrictions for the Measurement Wizard or block the access to the trigger dialog or measurement settings.

Online Document: If editing is not allowed, the user cannot make any (remaining) changes, your Online Document is protected from being changed.

Database: If "*Edit Database*" is not allowed, the user can call up the editor, but all channels are write-protected.

3.2 How to set up the starting modes of catman® Professional

3.2.1 Starting catman: Case 1

1. Enter a new user using the user name which you use when logging on to Windows.

   ☞ The password may be different from your Windows password.
2. Only activate the "*Administrator rights*" option under "User access rights" in section "*General*".
3. Restart catman and delete the user "*admin*".

3.2.2 Starting catman: Case 2

1. Enter a new user using the user name which you use when logging on to Windows.

   ☞ The password may be different from your Windows password.
2. Only activate the" *Administrator rights*" option under "User access rights" in section "*General*".
3. Restart catman and delete the user "*admin*".
4. Now enter all the users to be created using their user names under Windows.

   ☞ The passwords may be different from the Windows passwords.
5. Deactivate the "*Administrator rights*" option under "User access rights" in section "*General*"
6. Define the other access rights in the respective categories by deactivating those rights not required.

If you need to create several new users having only a few or the same access rights, first create a user and assign the appropriate rights. Deactivate all rights not required. Then mark this user before you create others. The rights will be copied from the marked user into the new users when they are created.

☞ Even if "*Open project*" in section "*Projects*" is deactivated for a user, you can specify a project via "*Open project upon start*". Only the user himself cannot load a project in this case.

3.2.3 Starting catman: Case 3

1. Enter a new user using the user name which you use when logging on to Windows.

   ☞ The password may be different from your Windows password.
2. Only activate the" *Administrator rights*" option under "User access rights" in section "*General*".
3. Restart catman and delete the user "*admin*".

4. Now enter all the users to be created using their user names under Windows.

   ☞ The passwords may be different from the Windows passwords.

5. Deactivate the "*Administrator rights*" option under "User access rights" in section "*General*"

6. Activate the option "*No log-in prompt upon start*" under "User access rights" in section "*General*".

7. Define the other access rights in the respective categories by deactivating those rights not required.

If you need to create several new users having only a few or the same access rights, first create a user and assign the appropriate rights. Deactivate all rights not required. Then mark this user before you create others. The rights will be copied from the marked user into the new users when they are created.

☞ Even if "*Open project*" in section "*Projects*" is deactivated for a user, you can specify a project via "*Open project upon start*". Only the user himself cannot load a project in this case.

4 Startup options

You can use the “Options” menu to modify catman’s “Startup options”.

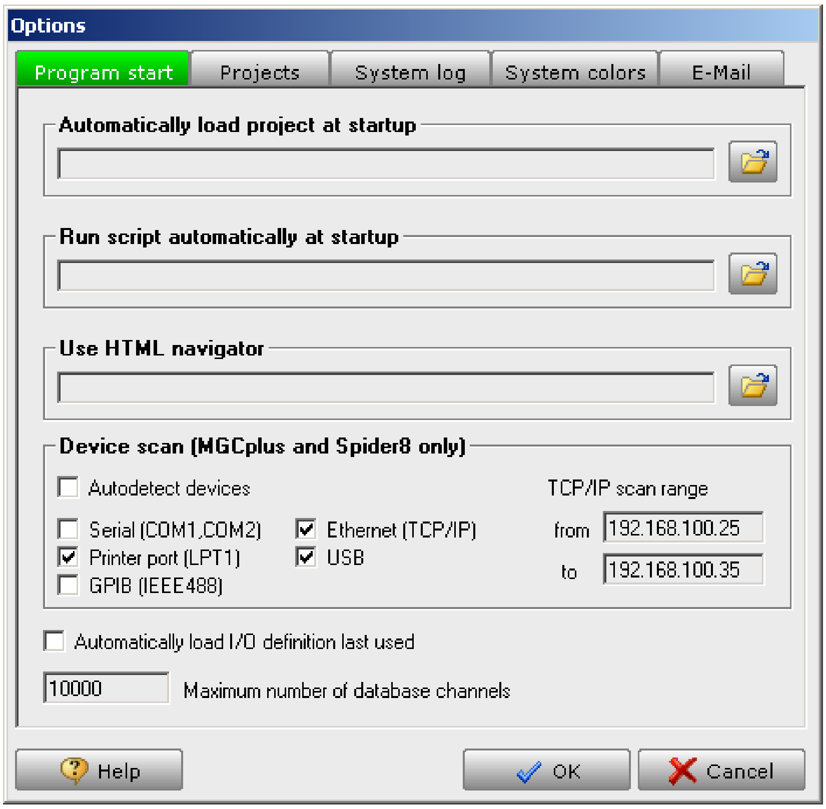


Figure 12: The options on starting catman® Professional

4.1 Automatically load project at startup

If a file has been specified for this (*.CPJ), catman automatically loads the corresponding project file on startup. When a project file is loaded, the first I/O Definition is activated and the Auto Command List is then loaded and activated. Depending what other options were specified when this file was saved, for example, the device setups may also be loaded and activated in the devices.

☞ You may define this functionality to be user dependent, see Section 3, page C-8.

4.2 Run script automatically on startup

This option makes it possible for you to run a binary script immediately on starting up catman. Just enter the file name of the compiled script (binary file, extension *.SCB) or VBScript (*.VBS) here.

☞ If an automatically loaded project contains a binary script file (Favorite file), this will be given priority, i.e. before a file which has been entered under the option "*Run script automatically on startup*".

If catman should execute exclusively a certain script, start catman using the script file as a parameter, see Section 4.7, *Starting with a script*, page C-16.

4.3 Use HTML Navigator

If you use this option, catman starts and displays the HTML page entered. The Project view is first of all hidden, but can be revealed.

4.4 Device scan

You can specify here which interfaces catman should search through for the HBM devices MGCplus and Spider8 during a device scan. This is similar to the tab “Device” with the I/O Definition options. The details for the interfaces are identical with those of the tab “Device” of the I/O Definition. If you change the settings at one of the two positions, the other position will also be changed.

However, if you activate “*Autodetect devices*” in the startup options, this is carried out when catman is started.

4.5 Automatically load last I/O Definition used at startup

If you check this option, the I/O Definition last saved will be loaded and used again by catman the next time it is started up. This option is only retained for compatibility reasons. We recommend to specify the I/O Definition in a project and load this project file on startup. In that case you may also load a device setup.

☞ The first I/O Definition of a project (if available) is *always* activated.

4.6 Maximum number of Database channels

After installation, only 1,000 channels are at first available. As a storage area is required for administration of all possible channels, this setting does not occupy unnecessary storage space.

If you wish to create more channels or, will sometimes need more than 1,000 channels, you can increase this upper limit to a maximum of 10,000 channels from main menu “Options → Startup options“. However, administrative rights are required to change the setting. See also Section 2.2, *Define the Database size*, page C-5. The number of channels actually used can be set from the menu “Options → Database configuration” or via the worksheet Database.

4.7 Starting with a script

If you have written a script, catman can also be started using the script as a parameter. In this version, no actions are possible apart from those which are specified in the script. If a short-cut has been created in the Windows Start-up folder, catman will start with the script as soon as Windows starts.

Proceed as follows:

1. Make a link to the start program of catman (CATRUN.EXE in the catman directory).
2. Under Windows, select “Properties” from the context menu and go to the “Short-cut” tab.
3. Under “*Target*”, enter the path and file name of the binary script after CATRUN.EXE. If the name of the path contains blanks, you should enter the entire line in quotation marks.

Example: `"c:\catman32\catrun.exe d:\measurement\My Script.scb"`

4.8 Suppress traceability information

The determination of the device and channel parameters (traceability data) in the respective channel initialization phases can be suppressed if catman (CATPROF.EXE) is started using the command line parameter “*NOAUTODYNCONF*”. The initialization takes less time (approx. 40 milliseconds per channel) as no time is required to transfer this channel information. However, if you have connected transducers with TEDS or T-ID module and run a scan, this time is negligible as compared to the time required for the TEDS or T-ID scan.

D Defining Devices and Channels

1 Introduction

This chapter describes the settings you must enter in order to use catman® Professional to operate and/or adjust a device, select channels to be used for measurement, execute real-time computation or provide output to devices. These settings *prepare* the data acquisition and need to be entered for all three operating levels. The worksheet used for this purpose is called I/O Definition.

For information on *carrying out* a measurement, please refer to Chapter E, *Measuring with catman® Professional*, through to Chapter H, *The Measurement Wizard*, depending on the operating level you want to use. If you prefer a short introduction, read Chapter B, Section 5, *Getting started (Quick Start)*, page B-22.

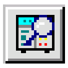
☞ Your access rights to the I/O Definition or certain parts of it may have been withdrawn via the user administration.

Once an I/O Definition has been made, it remains in place, even if you close the worksheet. Only when you exit catman a definition that has not been saved will be lost, therefore catman asks if the active I/O Definition should be saved.

All functions are accessible both via the menu bar and via context menus that you call by clicking the right mouse button in the appropriate columns. For frequently used functions there is also a toolbar below the Window menu.

There are two possibilities depending on the device used:

1. MGCplus and Spider8



- Let catman search for these HBM devices on the interface and connect the available channels: you only need to define the interfaces to be searched through *once* from the context menu “Options → Startup options” (or in the I/O Defininition “Options → Device”). Thereafter when catman is started, you only need to run the device scan, e.g. with , see Section 2.1, page D-6.

  ☞ You can automate this procedure with the Startup options, see Chapter C, Section 4.4, *Device scan*, page C-15.

- Then remove any excess channels.
- That’s all, if MGCplus with TEDS transducers is used. Here, the TEDS information is read out and the measuring chain is fully set up. For other transducers and with Spider8 you

must use the Sensor Database (Section 5, page D-54) or the T-ID (with MGCplus only, Section 6, page D-63) to determine the specific transducer settings for the individual channels

☞ Alternatively, you can enter the transducer data manually. For MGCplus use the device setup, for Spider8 a combination of device setup and user scaling in catman, see *2. All other devices*.

2. All other devices

- First enter the device and the interface used (Section 2.2, page D-6).
- Configure the device (enter the basic data for the transducer, see Section 2.5, page D-16).
- If not done automatically by catman, define the Time channel(s), see Section 3, page D-17.
- Then define the measuring channels and scale the measuring channels (Section 4, page D-21).
- Repeat these steps for all devices which you wish to use for the measurement.

Then in both cases:

- Carry out a zero adjustment (Section 4.1, page D-22).
- Define real-time computations (Section 4.5, page D-25).
- Define an event monitoring (Chapter E, Section 6, *The Event monitor*, page E-22).
- Determine how your measured values will be saved (Section 4.12, page D-48).

☞ To be able to save enough measured values, you should configure the storage space for the Database, see Chapter C, Section 2, *Configuring the Database*, page C-4.

Setting up the measuring devices is device-specific and is *not dealt with here*, since it is explained in the documentation for the devices concerned. Configuration of the time channel is described in a section of its own, since it also depends on the number of devices being used.

shows the channel information (hardware settings like full or half bridge etc.) of a marked channel. Use buttons (Insert I/O channel) and (Remove I/O channel) in frame “I/O channels” to insert new rows or delete those you no longer need. Please note that when having defined real-time computations, one must be careful when deleting channels: the channel numbers in computations are not changed accordingly, so they might refer to different channels afterwards.

Rows or columns may be enlarged or reduced by using the mouse directly on the column header or leftmost column, but these changes are not saved.

After these preparations are finished, use or in the I/O Definition to display measured values in order to check that the measuring chain is functioning properly.

If all devices are defined and all required I/O channels connected and set up, you should save the I/O Definition settings: “File → Save”.

2 Define and activate device

In this section we shall deal with the *upper* part of the I/O Definition worksheet, frame “Devices”. If this is not visible, click on button . This will show the upper part again.

2.1 Use device scan

You can only use the device scan for MGCplus and Spider8.

To be able to use the device scan, you must have defined the interfaces to be searched at least once. This is done from the menu “Options → Devices”. For the MGCplus you can also specify and set up the required number of time channels, for the Spider8 the operation mode of the parallel interface (LPT), see also Chapter A, Section 3.4, *Notes for Spider8 users*, page A-13.

As soon as this specification has been done, just click on “*Device scan*” or , the device is created and all available channels are connected automatically.

You can automate this procedure from the startup options, see Chapter C, Section 4.4, *Device scan*, page C-15.

2.2 Enter the devices that will be used manually

The device definition tells catman which items of *hardware* (i.e. device types) it will be dealing with, and which interface has to be used for making the connection.

Definition is easy:

1. Click on “*New device*” to open the configuration dialog.
2. Here you can either define a name for your device or use the default (DEVICE_1). The device name can be whatever you like, but you cannot use the same name more than once. If you have, say, two MGCplus devices connected, you must give them different names.
3. Next input the device type.
4. Now select the interface over which the device is connected. If the interface is bus-compliant, i.e. if it supports the connection of several devices, you must then specify the address set up on the device. If the interface can be present more than once in the PC, then the entry “*Board no.*” is also accessible. This is also the case with plug-in cards from National Instruments; state the board used when more than one card is installed, otherwise leave the default value as it is.
5. Finally click on “*OK*”.

After the last point, a dialog box is displayed for certain device types, and it asks you whether catman should automatically connect all existing channels on the device to I/O channels. If you will subsequently be using the full range of channels available in the device, we recommend that you click on “*Yes*”. If you need only one or two of the existing channels for your subsequent measurement, you can carry out a manual connection and may click on “*No*”.

☞ With MGCplus the channel names are also transferred (I/O Definition: “Options → Channels”). However, these must have been set in the device. When using several connected Spider8, they are always considered as one device which can then have up to 64 channels (max. 8 housings).

You can use the “Device → Setup device” menu or to start the device setup module, and then set up and adjust the individual channels to the transducers connected. See also Section 2.5, *Configure hardware device*, page D-16.

This catman® Professional manual does not explain how to adjust transducers. For further information on that subject, please refer to the catman online Help and the documentation for the devices concerned.

When you have completed the procedures described in this chapter and no longer need the upper part of the I/O Definition window, you may hide it by using the button  in the lower toolbar (in the “I/O channels” frame). There will then be more space available for displaying the channel configuration.

2.3 Special devices

2.3.1 MD Server, Remote Data Server

Starting with catman® Professional 4.0, a PC running the MD Server software or catman® Professional can be selected as device. The “Remote Data Server” in catman is started from the “Measure” menu in the I/O Definition or from the main menu. As long as the data acquisition runs on this PC, the server, the values acquired can be read by a so-called catman client. The client must have access authorization and be able to access the MD Server via the Ethernet. If necessary, contact your system administrator. Initialize the server in order to determine at the clients which channels are accessible. To transfer data, start the program and go to measuring mode. More information about measuring can also be found in Chapter E, Section 7, *Remote Data Server*, page E-37.

☞ To assign channels for the I/O Definition when using the MD Server you can also use the database file which was created by MC Setup: In this case, it is not necessary to start and initialize the software MD Server.

Select “I/O channel → Configure I/O channel” to display the channels available on the server. Mark the number of channels required in the “Connect” column, mark the required channels in the channel selection window and click on “*Add*”. In order to change a connection, mark the channel in the I/O Definition and click on “*Assign*” in the channel selection window. If the required MD Server directories are accessible (network access), you can also open the

MC Setup database from the corresponding tab. If relevant channels are predefined as *Function groups* on the MD Server, this simplifies selection considerably as only the channels belonging to a certain group can be displayed.

2.3.2 GPS systems

In many fields of application, e.g. road tests in the automotive industry, measurements on agricultural experimental areas, trial runs in nautical engineering, etc. there is the demand, in addition to acquiring the measurement data, of also recording the geographical co-ordinates with the aid of a GPS receiver. In catman, data acquisition from NMEA-0183 compatible GPS receivers is supported, e.g. device GPS III Plus from Garmin Corp. Most of the GPS systems on the market use this NMEA protocol (National Marine Electronics Association), so that also other receivers can be connected.

☞ You can find further information about NMEA under e.g. http://www.vancouver-webpages.com/peter. Information on devices you can find e.g. under www.garmin.com.

Integration into catman

A GPS receiver is a device type which is created just as normal in catman. The interface is usually configured as RS-232 with 4,800 baud, no parity, 8 data bits and 1 stop bit (see Operating Manual of device). Up to 6 I/O channels can be coupled to a device of this type.

1. Geograph. longitude [-180 degrees (westerly direction) to 180 degrees (easterly direction)]
2. Geograph. latitude [-90 degrees (South Pole) to 90 degrees (North Pole)]
3. Height above sea level
4. Speed over the ground
5. UTC time
6. UTC date

GPS data rate

The GPS receiver is optimally measured with a data rate of 1Hz.

Processing GPS data

The GPS data can be saved in the Database. Geograph. co-ordinates are always saved in degrees with up to 6 decimal places being relevant.

2.3.3 Using device type “Unknown” (third-party devices)

With devices which are not directly supported by catman, the type must be selected as “*Unknown*” device. catman offers the possibility of integrating devices not supported directly by catman into the normal data acquisition by using a DLL Driver or a Script Driver. These devices can be then used without further ado in a catModule or the Measurement Wizard and their measurement data can be displayed in real-time graphics or saved in the Database.

☞ Devices without DLL or Script Drivers can only be used in Level 3 with the catman script language.

You will find a template for creating DLL Drivers, for example in C++, in catman’s DRIVERS\DLLDRIVER\DRIVER TEMPLATE sub-directory, see Chapter K, Section 6.13, *DLL Drivers for “unknown devices”*, page K-84. The creation of a Script Driver is done in the script development system of catman and is described in Chapter K, Section 6.14, *Script Drivers for “unknown devices”*, page K-85.

This is how you insert an unknown device into the device list:

1. Select the type of device as “*Unknown*” device.
2. Enter the corresponding driver via “Driver for unknown device” in the “Device” menu.
3. If a manufacturer’s program for the configuration has been supplied for this device, you can enter this as the “*Device setup module*”. The program is then called when you call the

menu item “Setup device” via the “Device” menu. Alternatively, you can also enter a configuration file that will be loaded when the device is initialized.

4. If the driver and the device support the internal generation of a sample rate (time base, i.e. device internal timing) and/or temporary storage of the data, activate the corresponding option under “*Time synchronization*”. Specify the relevant NTP or IRIG-B channel as “*Master channel for time synchronization*”. This channel must be connected to a catman I/O channel (I/O Definition). In case of doubt ask the programmer of the driver.

A time synchronization of the measuring data is only possible if NTP or IRIG-B time data is *additionally* acquired by an MGCplus (channels 20 to 25, see Section 3, *Configure time channels*, page D-17). In the device list, the MGCplus must be *above* the device to be synchronized. Synchronization is not possible with other devices, e.g. Spider8.

2.4 Notes about the interfaces

☞ Access to the dialog “Setup interface” may have been withdrawn via the user administration

The “Device → Setup interface” dialog only changes the PC interface settings, not the settings on the measuring instrument itself. To change the baud rate, for instance, you must then click on the “*Change device baudrate now*” button. Apart from the baud rate, you should not change any settings with HBM devices. If you enter any other specifications for HBM devices, data transmission between catman and the HBM measuring instrument may sometimes no longer work.

You can operate several devices of the *same* type via the same bus-compliant interface (RS-485, NI-GPIB). However, if you want to combine devices of *different types*, you must use *physically different* interfaces, such as RS-485 on COM1 and RS-485 on COM2 or RS-232 on COM1 and NI-GPIB or LPT1 and Ethernet.

There is an exception: If you are using several Spider8 devices, you can connect these over one of the interfaces which are not bus-compliant, LPT1 or LPT2. For further information on the subject, please refer to the Chapter called *Connecting* in the Spider8 Operating Manual and the notes in Chapter A, *Introduction*, Section 3.4, *Notes for Spider8 users*, page A-13, as well as the notes on LPT interfaces in the following subsection.

You can use the “*Offline*” interface to enter all the necessary settings without even having a device attached. If you want to enter a number of devices as “*Offline*”, you must assign different addresses to the individual devices.

2.4.1 Bus-compliant interfaces

If there are problems with the interface, you can use the function “Diagnosis → TCP/IP ping”.

Ethernet (TCP/IP)

This interface is currently only available for the MGCplus and the Remote Data Server. On the PC a network with TCP/IP protocol must be set up.

The important parameter to be entered here is the IP address of the device in the usual notation, e.g. “*172.16.3.48*”. If you do not use the default port (7), e.g. for the Remote Data Server, enter this also here.

If problems occur in establishing a link, make sure the Ethernet cable is correctly terminated (termination resistors). Ask your network administrator if some of the above mentioned terms are unfamiliar to you.

If there are several MGCplus connected and working with high speed data acquisition, it might be useful to increase the Receive buffer size (menu “Device → Setup interface”). On the other hand, a value above 64 kilobytes should be avoided, because this setting is used for all Ethernet buffers of the PC. High buffer size settings consume therefore a lot of RAM memory.

USB

During the installation of catman the drivers needed for this interface are copied into the appropriate Windows system directories. When you connect a device, Windows detects this and starts the Windows Hardware Wizard which searches for these drivers and activates them for Windows, see Chapter A, Section 3.6, *Notes on the USB interface*, page A-15.

You can leave the “*Address*” line free, if only one device is connected. catman® Professional will then use the first device found on the bus. When using several devices, carry out a device scan.

☞ Uppercase/lowercase writing is considered for the USB address.

IEEE 488.2

To use an NI-GPIB you must have a pre-installed and fully configured adapter from National Instruments. For further information please refer to the notes in Chapter A, *Introduction*, Section *Notes on the interfaces IEEE 488.2, CAN bus and Profibus DP*, page A-16. The interface is also designated IEC 625, DIN-IEC 625, HPIB or GPIB.

☞ The factory setting for the IEC address of HBM devices is 4. All other parameters are already suitable for the stated HBM device.

COMx with RS-485

To use the RS-485 interface, your computer must have the appropriate hardware (board) or an adapter on the RS-232 interface. These adapters can be obtained in various versions from computer supply retailers. However, you can only fit the PC with one of these adapters, and not a measuring instrument, since the additional bus commands needed for the interface are not implemented in devices with RS-232. The rest of the settings are identical to those for the RS-232 interface.

CAN bus, Profibus

To use these interfaces, your computer must have the appropriate hardware (board) or adapter. For further information please refer to the notes in Chapter A, *Introduction*, Section 3.7, *Notes on the interfaces IEEE 488.2, CAN bus and Profibus DP*, page A-16.

2.4.2 Non-bus-compliant interfaces

COMx (RS-232)

In the case of the UPM60/UGR60, the baud rate on the device can only be adjusted externally by using DIP switches. After that the “Settings → Setup interface” dialog must be called and the baud rate must be changed. In all other cases call up the dialog, input the required baud rate and click the “*Change device baudrate now*” button.

⚠ Please note that with certain devices, such as the DMC*plus*, a *cold start* resets the interface settings to 9,600 baud (factory setting) or to the value configured by the DIP switches (for a *warm start* set the DIP switch to “*W*”).

Printer port LPT

One of the two printer ports LPT1 or LPT2 can be used for either the MGCplus or the Spider8.

MGCplus For the parallel interface of the MGCplus with CP32 there is only the “*MGCplus Interlink mode*” which is permanently set. The transfer rate here is approx. 25 kilobytes/second, corresponding to approx. 6,000 measured values/s in the 4-byte format.

☞ With the MGCplus the maximum transfer rate cannot be achieved via this interface. For this use either IEEE 488 or (Fast) Ethernet.

Spider8 "*Nibble mode*": This operating mode functions with any parallel interface. The data bytes from the Spider8 are sent as 4-bit packets (nibbles) over the status lines. Data rate: approx. 6,000 measurements/second

"*Bit8 mode*": Here it is assumed that the four control lines leaving the PC are realized as open-collector drivers and the line level is acquired separately as an input. This is the case with some "old" parallel interfaces; modern interfaces in the operating modes PS/2, ECP or EPP do not permit this. Eight bits are transferred simultaneously. Data rate: approx. 20,000 measurements/second.

"*Byte mode*": The transfer from the Spider8 to the PC occurs via the data lines which are switched from the normal direction (output) to input for this purpose (bi-directional mode). Most modern computers enable the use of this operating mode. Data rate: approx. 40,000 measurements/second.

"*EPP mode*": If the interface operates in the EPP mode, the data is transferred, as with the byte mode, bi-directional. Also the processing of the transfer (handshake) occurs directly and therefore very quickly, whereas with all other operating modes the processor handles the processing of the handshake by program. The mode must though be expressly supported by the PC and activated via the BIOS setup on the PC. Data rate: > 76,000 measurements/second.

Printers, which install special printer drivers in the operating system, may cause problems if they try to regularly pass on status information even when measurement data is currently transferred. In these cases you have the following possibilities:

- You can define the standard printer not to be this printer, but instead install some other, unproblematic printer as standard; this only affects the software installation and the printer does not need to be physically present. You then select the printer before the printout explicitly.
- You can define the connection "*Output to file*" for the printer (Windows printer setup).
- You can use the other LPT interface for the printer.

Alternatively, the USB interfaces can also be used either for the printer (if present) or e.g. for the Spider8.

For Spider8 please read also the notes in Chapter A, *Introduction*, Section 3.4, *Notes for Spider8 users*, page A-13.

2.5 Configure hardware device

Your access rights to call up these programs may have been withdrawn via the user administration.

In the I/O Definition, the device setup can be called with icon , the context menu in the device list or the menu “Device → Configure device“. catman starts the appropriate setup module for the device marked. Save the device setup with this program in a separate file. For some devices, this is saved as a text file additionally so that the settings can be added, for example, to a measuring protocol.

Device setup or channel setup

A channel setup is only available for Spider8 and MGCplus. All other devices can only be set up and adjusted with “Setup device”. If you only want to change individual settings for a few channels, this method is quicker. The changes cannot however be saved in the EEPROM of the MGCplus or as a file.

With MGCplus and Spider8, you can also create the current settings automatically as device setup file(s) and insert them into the Project window: in the Project window, call the Device setups context menu and select “Add → Current device settings“.

3 Configure time channels

In this section we shall deal with the *lower* part of the I/O Definition worksheet, frame “I/O channels”.

Time channels are normally created automatically by catman when you add a device. You can however change this default setting in the “Options → Device” menu. For the MGCplus you can also define whether only one or three time channels are automatically created. You can however allocate the data source “Time” to other I/O channels at any time (context menu in the column “Connection”).

A time channel supplies the *time markers* (time increments) of the individual measurement values recorded during acquisition. In general the time base of a measurement can be determined in various ways:

1. Each I/O channel contains the time increment Δt used (since version 3.0).
2. Time channels can be configured in the I/O Definition. catman writes the cumulated time increments Δt into these channels.
3. The MGCplus has so-called hardware time channels which can be connected as *normal analog* I/O channels with the MGCplus channels 17, 18 and 19 („*Time SRGx*“). If you use corresponding hardware in the MGCplus, you can also evaluate the NTP or IRIG-B time information acquired in the MGCplus: channels 20 to 22 (NTP) or 23 to 25 (IRIG-B).

A time channel is essential for:

1. In order to be able to reconstruct the time sequence of the measurement if during the measurement the sample rate of a channel is changed using the sample-rate trigger of the MGCplus. In this case it is mandatory to use the corresponding hardware time channel of the MGCplus as the time source. As “Connection” enter the device and under “Settings” one of the required channels. The channels 17 to 19 are identified as “*Time (SRG1 to 3)*”—SRG means sample rate group. You can also enter the channels 20 to 22 (NTP) or 23 to 25 (IRIG-B), if the corresponding hardware is available.
2. All measurements with non-equidistant time markers, e.g. periodic measurements with interruptions, variable acquisition rates, etc.
3. Computations which need time information, e.g. FFT, spectrum, etc.

☞ One single time channel is sufficient for each sample rate used.

If you go to the menu “I/O channel → Configure time channel” and specify “*Simulation*” as the timing source, the time channel will be filled out with values based on the simulation frequency. At 50 Hz, for example, an entry (time mark) is generated every 20 milliseconds. You may then use a real-time computation to define any functions you like, using this time channel as the x value. By this means you may then generate outputs such as sin(time) with “Connection → Online computation → Hard-coded functions” on the “Algebra” tab.

3.1 Source for time marker generation

If you have only *one* device connected, leave the setting in menu option “I/O channel → Configure time channel” for the “*Source for time marker generation*” on the default “*DEVICE_1*” resp. the name chosen for the device. catman will then decide what to do automatically: depending on the settings in the “Measurement settings” dialog and the devices used either the time marks are created in the device (device-internal timing) or catman generates the time marks.

If several devices are connected, the settings depend on the number of sample rates you are using and on the type of devices connected.

Time channels in the case of different sample rates

You must first decide on the basis of the type of devices: can the devices control their data acquisition rate themselves or not? The devices MGCplus, Spider8, DMC*plus*, DMC9012A and NI-ATMIO have their own *time base (time marker) generation*. Here catman only needs to tell the devices the sample rate at which to work. For all other devices, catman itself must generate the timing clock.

If you are using both device types catman will always derive the time base from those devices which are capable of creating a clock of their own, i.e. the others are automatically synchronized with them. However, the command to measure will only be given if needed, so in each case there is a slight time lag compared with the other devices. However, when observed over long measuring periods this behavior is better than with purely internal clock generation by catman via the PC timer, since minor deviations from the clock pulse (in the range of milliseconds) do not accumulate; instead the *right* clock pulse is always given by the devices with internal clock generation.

With the setting “*PC-internal timing*” each measured value must be requested and read in *individually*. For devices with no capability to generate an internal clock you should therefore not perform measurements faster than 10 values per second. If you use several devices of this type you should not request more than 1 measurement per second per device. In the case of scanning systems which work with several channels, low carrier frequency and 20 milliseconds integration time, you should even have only one cycle (one measurement value for each channel) measured every 5-10 seconds.

☞ In the case of slow measurements (less than one measurement value per second) catman uses (except with the MGCplus) “*PC-internal timing*” without this being explicitly specified in the I/O Definition. However, you must then specify in the settings for the *sample rate* that the time marker will be generated by catman (with the PC timer).

With “*PC-internal timing*” the times are derived from the PC timer. Consequently the regularity of the gaps between two measurements is not as good as with a clock pulse created in the measuring device; fluctuations or inaccuracies in the region of 2 to 5 milliseconds may occur. If you are measuring over lengthy periods these deviations will accumulate. For example, this could give rise to a deviation of 10 seconds in a measurement lasting an hour.

For further information please refer to Chapter E, *Measuring with catman® Professional*, for a discussion on setting the sample rate in Section 2, page E-4.

3.2 Time channel unit

You can use the menu “I/O channel → Configure time channel” to define the unit in which time is to be recorded: milliseconds (ms), seconds (s), minutes (min), hours (h) or as *absolute* time with date and time of day. The time always begins at 0 (zero) counting from the moment measurement starts.

When absolute time is specified, the date and time are read from the PC system clock. The precision of this starting value is only as good as the precision of the clock in the PC. However, in contrast to relative time, resolution is in *whole seconds only*, not fractions of a second.

In the event of triggered measurement, the *first* value that the device *sends* is written to time zero. If you also need the starting time, you must either use the hardware time channels of MGCplus or work at Level 3.

If you have specified for the time to be acquired in ms (milliseconds), you must scale the x-axis in the output graphs accordingly. For a measurement of 10 seconds duration, only a scaling of 0 to 10,000 would show all measurements.

Absolute or relative time?

The use of absolute time marks is recommended only for long-term monitoring with less than a measurement per second. So that absolute time can be fully represented in the Database, specify the format as “*Date + time*”, enable “*Date/time scale*” in the y(x) Post-process graph and input the required starting and finishing dates and times.

For longer recordings in which short-term periods are measured with higher sample rates, you can also define *two* time channels with different units (absolute and relative), because with date/time values the resolution is restricted to one second.

4 Defining input/output (measuring) channels

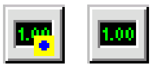
In this section we shall also deal with the *lower* part of the I/O Definition worksheet, frame “I/O channels”. The section “I/O channels” is used for entering not just time channels, but also all channel related specifications, such as which part of the connected devices will be used, how it will be used and what real-time computations are to be performed. Real-time computations are described in Sections 4.5.1 to 4.5.3 starting at page D-27.

For the sake of clarity, this section is divided up in the same way as the headings in the table of the I/O Definition. Just the topic Zeroing is covered separately in Section 4.1.

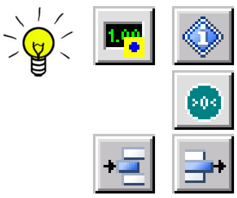
In general proceed as follows:

1. First of all, in the “Connection” column define whether you want a device channel, a file or the Database to return the data for the selected catman channel or channels. If a computation shall be used, you must additionally specify which one.
2. Then enter in the “Settings” column the device channel that you want to use or the name of the file that you want to be read. In the case of Database channels, the one that is read is always the one that belongs to the I/O channel (same number). With device channels in most cases the setting already applied by catman during the first step can be used here.
3. Next specify unique names for the channels (column “Name”): this will make them much easier to assign later on, when you are selecting which data to display in your graphs.
4. If the device you are using does not itself carry out scaling of measured values on the basis of physical units, define these in the “Scaling” column. This allows the unit which the device actually returns to be converted to the right physical size.

   ☞ With MGCplus and Spider8 you should use the Sensor Database, see Section 5, page D-54.



5. If a device is connected, check with [icon] or [icon], whether all active analog channels work as intended and have the correct signal/scaling. Channels connected to a digital input can not be measured in this way.



Click on [icon] to display the channel information (device settings such as full/half bridge etc.) for the channel marked. Press [icon] to start a continuous measurement for *all active* channels which have as source an *analog* signal (“*Analog IN*“). [icon] starts a zero adjustment for *marked* channels. Please also read Section 4.1, *Zeroing channels (zero adjustment)*, page D-22, for zero adjustment. The push-buttons (at “I/O channels“) [icon] and [icon] can be used to insert new lines or delete those lines no longer required.

Note however that channel numbers in computations are not changed accordingly. Therefore the computations might refer to different channels afterwards.

☞ Rows or columns may be enlarged or reduced by using the mouse directly on the column header or leftmost column, but these changes are not saved.

You can also use the arrow keys to move between fields anywhere in the “I/O channels” section of the worksheet.

The individual menus are accessible via the menu bar in the I/O Definition window or by pressing the *right* mouse button in the appropriate column and row. If several rows are selected, settings affect the selected rows only, regardless of the row on which the mouse pointer is positioned.

4.1 Zeroing channels (zero adjustment)

Zeroing (since catman® Professional 4.0) is a function allowing simple and comfortable zero adjustment even when using different devices. The function can however differ slightly depending on the device and the settings in catman. The new procedure is also implemented in the other catman components, e.g. the catModules and the Measurement Wizard. There are basically three possibilities:

1. The device delivers scaled values.

   This is normally the case for MGCplus and for example the scanning systems UPM60/UPM100. In catman, the device *tare function* is used for the zeroing. With MGCplus therefore the *net signal* (I/O Definition) has to be measured, otherwise the zeroing is not visible.
2. The device does not calculate values, the conversion into the unit of the measured physical quantity is carried out via the user scaling in catman.

This is the case for Spider8, DMC*plus* and some other devices, as scaling is not carried out here. Depending on the transducer connected, the device itself delivers, for example, mV/V. In this case, the zeroing is performed by the software, i.e. catman® Professional recognizes the value available at the time of the zeroing and subtracts this from all following measurements. The actual zero value is displayed in the user scaling dialog box and saved with the I/O Definition. The zero and taring memory of the device are not changed in this case by catman but must however *not* be changed later by the user, e.g. via the device setup. We recommend that you do *not* make any zero adjustment in the device setup and to exclude its further use in the device.

3. The device does not calculate values and *no* user scaling is selected.

   The *device zero adjustment* is used for MGCplus. All other devices use the *device tare function*.

If the MGCplus is used as in the third point (“*Electrical base unit*” is set as “*Signal type*“), then either the *gross signal* must be measured (“Configure I/O channel” from the column “Settings“) or *no value unequal to zero* may be set in the *device tare memory*. Otherwise after the zeroing this value will appear as measured value instead of zero.

Lock zeroing

The zero adjustment for a channel can be suppressed (locked) from the column “Status/reading” (see also page D-38). In this case, the channel will not be included in the subsequent zeroing. A zero adjustment or a taring may then only be carried out for the channel concerned from the device setup or via a low-level script command, as in these cases the device is accessed directly. The context menu can also be used to reset the zero value in catman.

4.2 Activate column

Your access rights for this action may have been withdrawn via the user administration.

In this column ( ) you can see whether a channel is active ( ) or currently deactivated ( ). If deactivated all the channel settings are not lost, as is the case with deleting. The channel and its settings are simply ignored. Next to the icons are catman's *internal* numbers for the real-time data channels: 1, 2 etc.

4.3 Description file

You can use description files (column) in order to save additional information (pictures, drawings, texts) about individual channels or the I/O Definition. The file format can be freely selected, catman only saves the cross references to the files and if necessary calls the programs required to display the files. A single description file can be assigned to several I/O channels but only one description file can be assigned to each channel.

Procedure

1. Open the dialog box by double clicking on the required line, select "I/O channel → Assign description file" or use the context menu.
2. Enter one or more description files from the file list.
3. Mark one and then click on "*Assign*".

Display description files

- Double click on in the respective row

  or
- Use "Display description files" (context menu or "Settings" menu).

☞ catman can display files of types text (*.TXT), bitmap (*.BMP), JPEG (*.JPG), metafile (*.WMF), gif (*.GIF) and HTML (*.HTM, *.HTML). In other cases, the files are displayed by calling the program which is linked to it under Windows.

⚠ Do not confuse the I/O Definition description files (Channel description file) with that of a Log file. Both are independent of each other and use separate lists.

4.4 Name

We recommend that you enter a name for the particular channel here.

⚠ Channel names must not start with a number and are restricted to a maximum of 64 characters.

You get to the channel above or below by using the CURSOR UP or CURSOR DOWN keys respectively. You do not have to use the mouse to switch to the next channel.

4.5 Connection

As well as defining a time channel, there are also options to connect a channel to the following:

- an online computation
- a Database channel

- a file
- a device

In most cases, this column is only used to decide how the channel will subsequently be used. The decision about which channel, which file etc. is taken in the “Settings” column. Just for online computations the type of computation *must* be specified here.

Defining online computations

If an “*Online computation*” has been defined in the “Connection” column, you must select one of the computation groups in the sub-menu. The following five menu items are provided:

- Hard-coded functions
- Algebraic computations
- Single strain-gage rosette
- Multiple strain-gage rosettes
- Temperature compensation with strain gages

These computation types are explained in detail in Sections 4.5.1 to 4.5.3 starting at page D-27.

Connection wizard

If you click in column “Connection” with the *right* mouse key, the context menu appears. The first menu item is the “Connection wizard”. This is useful to assign several hardware channels of complex multi-channel devices (MGCplus, Spider8, Centipede) to consecutive channels in an I/O Definition *in one step*.

Database channel (read online)

“Database channel (read online)” is used for making data available from the Database for real-time graphs. As real-time graphs, depending on type, only process one value or one read block, you cannot actually display data from the Database. With this menu item, you simulate renewed data acquisition and can then display data, e.g. as reference values in a real-time graph together with newly measured data (or read in from a file). Set the Database channel to be used by double-clicking into column “Settings”.

If the end of the Database channel is reached during a running measurement, the measurement stops.

File (import online)

You can use “File (import online)” to read data from a file which is then dealt with like data captured in real time, just as in “Database channel (read online)”. In this way, data already captured can be forwarded again *live* into real-time graphs. This function can be useful, for instance, after a run with a large number of measured values, for reading the acquired data again in smaller blocks and displaying it graphically.

With both settings, the data is read against the running index, the time reference will be lost unless you also read in the original time channel. For displaying the data in y(t) real-time graphs you *always must* create an additional time channel (“*Simulation*”).

4.5.1 Hard-coded functions

This setup dialog provides you with functions sorted in groups. You select the required function, e.g. “*a * Channel 1 + b * Channel 2*” on the appropriate tab, in this example “Algebra”. Then you enter the relevant values for the parameters and the I/O channels with which the computation is to be made. The computation is displayed as a check in the field “computation”. You can also state a name and a unit for the computation channel. catman produces no automatic units due to the argument channels which occur in the function.

☞ To keep the formula notation as short as possible, the formula displayed in the I/O Definition does not show the channel names you specified, but just their internal catman abbreviations, C1, C2 etc.

The hard-coded functions operate significantly faster than an algebraic computation of the same type.

Since most of the functions are self-explanatory, the following notes just deal with some special aspects.

Signal analysis

When using FFT or spectrum computations, make sure the read blocks are large enough, say 256 values. In this case you have to turn off automatic read-block size determination for the measurement concerned and specify the block size yourself. For further information please refer to Chapter E, *Measuring with catman® Professional*. You will be unable to display a spectrum without a suitable x-axis. You may create this with the "*Frequency ramp*" function, which is under the "Ramp" tab. Then, for instance, you will be able to display the spectrum in a y(x) Post-process graph.

Ramp

"*Continuous ramp over several read blocks*" counts up in ascending order starting at 1. Each value (i.e. each row) increments the number by 1. "*Ramp over one read block*" begins again at 1 for each read block.

In the case of "*Frequency ramp over one read block*", counting depends on the sample rate of the *Master clock*: the count for the whole of the read block will be between 0 and the value of the sample rate. You can use this data to display the result of an FFT or spectrum computation in real time.

Logic

The result of a comparison is returned as the digit 0 (false) or 1 (true). Use comparisons of identical values with care: no two measured values stored in 8-byte floating point format are hardly ever completely identical.

The function "*Test channel 1 for bit no.*" rounds the value for the specified channel to a 32-bit INTEGER and checks whether the specified bit is set; the result is 1 for set, else 0.

With this function an I/O channel which is, for example, coupled as an analog input to the digital channel of a Spider8 can be split up into the individual signals (see also Section *Analog outputs, digital inputs and outputs*, page D-38).

Filter

The function "*Running average + tolerance band*" enables computation of a plot that lies above or below the running average by a specified "*tolerance width*" (expressed in the unit for "*Channel 1*"). As an example, you may use two such computations to plot a tolerance band and the actual measured values in a stripchart so that you can see whether all measured values are within the tolerance width.

Compression

You may use this function to carry out data compression in real time. The "*Block size*" determines the number of measured values over which the compression computation is performed.

This has *nothing* to do with the *read-block size*. For this purpose the "*Block size*" is equivalent to the compression factor. Please also remember to compress the associated time channel. Unless the compressed values are also accompanied by their times, it will be impossible to produce correct graphics from them.

The "*Center*" setting is provided especially for the time channel compression function. It computes the exact center of the time slice within a block. You may also display compressed values in a y(t) Real-time stripchart or a Cursor graph: select the *computed* time channel as the "*Source for time marks*".

4.5.2 Algebraic computations

This menu item enables you to create any formulas, which you enter as you would on a pocket calculator. You can use the drop-down list boxes in the lower part of the dialog to select available I/O channels or special functions and insert them in the formula.

To keep the formula notation as short as possible, the formula displayed does not show the channel names you specified, but just their catman-internal abbreviations, C1, C2 etc. it is not possible to use channel *names* in the formula.

The hard-coded functions operate significantly faster than an algebraic computation of the same type.

You can access existing formulae via the list “*Formula created last*”. You can also state a “*Name*” and a “*Unit*” for the computation channel. catman produces no automatic units due to the argument channels which occur in the function.

Logical expressions within a formula (e.g. C1 > C2) are always evaluated as 0 (FALSE) or 1 (TRUE).

“*Condition*” (IF), “*Derivative*” and “*Integral*” are also available in the field “Special functions” in addition to “*Bit test*”, “*Random number*” and “*System time*”.

4.5.3 Strain-gage computations and temperature compensation for strain gages

Before you define computations using several results channels, make sure that you have enough free channels available. The number of channels depends on how many channels you specified for the Database in the “Options → Database configuration” menu, see Chapter C, Section 2, *Configuring the Database*, page C-4.

The menu items about strain-gage computations contain special formulas for frequently-used functions in experimental stress analysis.

The measuring grids a, b and c must measure mathematically positive (counter-clockwise) at the stated angles. With HBM strain gage rosettes the grids are therefore labeled with a, b and c and can be immediately entered. If rosettes from other manufacturers are used, check the measuring grid arrangement and labeling.

If the computation is called from the menu “Multiple strain-gage rosettes”, channel names can be assigned automatically which are derived either from the channel names for the measuring grid or consist of a basic name a number appended. The letters from Table 4 are appended to allow you to distinguish between the strain-gage computations:

Abbreviation	**Computation**
AG	Angle
SS1	Principle nominal stress 1
SS2	Principle nominal stress 2
SH	Shear stress
ES	Equivalent stress according to von Mises
SSX	Stress X: stress in the direction of grid a
SSY	Stress Y: stress under 90° to grid a. The angle is evaluated in mathematical positive direction (counter clockwise).
SN1	Principal strain 1: strain in direction of principle nominal stress 1
SN2	Principal strain 2: strain in direction of principle nominal stress 2

Table 4: Abbreviations for strain-gage computations

Abbreviation	Computation
SNX	Strain X: strain in the direction of grid a
SNY	Strain Y: strain under 90° to grid a. The angle is evaluated in mathematical positive direction (counter clockwise).
SNA	Shear strain: strain under 45° to grid a. The angle is evaluated in mathematical positive direction (counter clockwise).

Table 4: Abbreviations for strain-gage computations

☞ The computations also contain special formulae for experimental stress analysis. Details of the computations can be found in the following. Not all computations can be used for all types of rosettes.

0/45/90° rosettes:

Angle

$$\alpha = \frac{1}{2} atan \left| \frac{2\varepsilon_B - \varepsilon_A - \varepsilon_C}{\varepsilon_A - \varepsilon_C} \right|$$

Principle nominal stress 1

$$\sigma_1 = \frac{E}{2(1-\nu)}(\varepsilon_A + \varepsilon_C) + \frac{E\sqrt{2}}{2(1+\nu)}\sqrt{(\varepsilon_A - \varepsilon_B)^2 + (\varepsilon_C - \varepsilon_B)^2}$$

Principle nominal stress 2

$$\sigma_2 = \frac{E}{2(1-\nu)}(\varepsilon_A + \varepsilon_C) - \frac{E\sqrt{2}}{2(1+\nu)}\sqrt{(\varepsilon_A - \varepsilon_B)^2 + (\varepsilon_C - \varepsilon_B)^2}$$

Stress X

$$\sigma_x = \frac{E}{1-\nu^2}(\varepsilon_A + \nu\varepsilon_C)$$

Stress Y

$$\sigma_y = \frac{E}{1-\nu^2}(\varepsilon_C + \nu\varepsilon_A)$$

Principal strain 1

$$\varepsilon_1 = \frac{\varepsilon_A + \varepsilon_C}{2} + \frac{1}{\sqrt{2}}\sqrt{(\varepsilon_A - \varepsilon_B)^2 + (\varepsilon_C - \varepsilon_B)^2}$$

Principal strain 2

$$\varepsilon_2 = \frac{\varepsilon_A + \varepsilon_C}{2} - \frac{1}{\sqrt{2}}\sqrt{(\varepsilon_A - \varepsilon_B)^2 + (\varepsilon_C - \varepsilon_B)^2}$$

Strain X

$$\varepsilon_x = \varepsilon_A$$

Strain Y

$$\varepsilon_y = \varepsilon_C$$

Shear strain

$$\gamma = 2\varepsilon_B - \varepsilon_A - \varepsilon_C$$

0/60/120° rosettes: **Angle**

$$\alpha = \frac{1}{2} atan \left| \frac{\sqrt{3}(\varepsilon_B - \varepsilon_C)}{2\varepsilon_A - \varepsilon_B - \varepsilon_C} \right|$$

Principle nominal stress 1

$$\sigma_1 = \frac{E}{3(1-\nu)}\left(\frac{\varepsilon_A + \varepsilon_B + \varepsilon_C}{3}\right) + \frac{E}{\sqrt{3}(1+\nu)}\sqrt{\frac{(2\varepsilon_A - \varepsilon_B - \varepsilon_C)^2}{3} + (\varepsilon_B + \varepsilon_C)^2}$$

Principle nominal stress 2

$$\sigma_2 = \frac{E}{3(1-\nu)}\left(\frac{\varepsilon_A + \varepsilon_B + \varepsilon_C}{3}\right) - \frac{E}{\sqrt{3}(1+\nu)}\sqrt{\frac{(2\varepsilon_A - \varepsilon_B - \varepsilon_C)^2}{3} + (\varepsilon_B + \varepsilon_C)^2}$$

Stress X

$$\sigma_x = \frac{E}{1-\nu^2}\left(\varepsilon_A + \nu\frac{2}{3}\left(\varepsilon_B + \varepsilon_C - \frac{\varepsilon_A}{2}\right)\right)$$

Stress Y

$$\sigma_y = \frac{E}{1-\nu^2}\left(\frac{2}{3}\left(\varepsilon_B + \varepsilon_C - \frac{\varepsilon_A}{2}\right) + \nu\varepsilon_A\right)$$

Principal strain 1

$$\varepsilon_1 = \frac{\varepsilon_A + \varepsilon_B + \varepsilon_C}{3} + \frac{1}{\sqrt{3}}\sqrt{\frac{(2\varepsilon_A - \varepsilon_B - \varepsilon_C)^2}{3} + (\varepsilon_B - \varepsilon_C)^2}$$

Principal strain 2

$$\varepsilon_2 = \frac{\varepsilon_A + \varepsilon_B + \varepsilon_C}{3} - \frac{1}{\sqrt{3}}\sqrt{\frac{(2\varepsilon_A - \varepsilon_B - \varepsilon_C)^2}{3} + (\varepsilon_B - \varepsilon_C)^2}$$

Strain X

$$\varepsilon_x = \varepsilon_A$$

Strain Y

$$\varepsilon_y = \frac{2}{3}\left(\varepsilon_B + \varepsilon_C - \frac{\varepsilon_A}{2}\right)$$

Shear strain

$$\gamma = \frac{2}{\sqrt{3}}(\varepsilon_B - \varepsilon_C)$$

Diverse: **Shear stress**

$$\tau_{xy} = \frac{\sigma_1 - \sigma_2}{2} sin 2\alpha$$

Equivalent stress according to von Mises

$$\sigma_v = \sqrt{\sigma_1^2 + \sigma_2^2 - \sigma_1\sigma_2}$$

If required, compensation of the transverse sensitivity can also be carried out. Using the “Strain gage analysis” tab, a measurement with one of the hole-drilling or ring-core methods can also be made. For the simple (integral) hole drilling method, bore a hole of the same depth as the drill diameter. As a rule, measurement and evaluation will only be carried out, that is, residual stresses will only be determined, once this has been done. Using this method, no data that is dependent on the depth of the hole can be determined. Using an improved method developed by the MPA Stuttgart, the hole depth is also measured and strain values are computed with the aid of a calibration curve and the hole depth. This provides data for the residual stresses in different surface layers. The ring core method requires special drills and machines. With this method too, you can drill in stages to obtain data about the residual stresses in different surface layers.

On the tab “Thermal compensation” you will find special computation methods for the temperature compensation of single strain gages. However, we recommend to use the “Strain gage scaling” methods from the “User scaling” dialog in column “Scaling” in the first place, see Section 4.9, *Scaling*, page D-42.

☞ For each incoming value, strain-gage computations also produce a new value in the results channel(s).

The formula applied is as follows:

$$\varepsilon(T) = \varepsilon_{measured}(T) - (a_0 + a_1 T + a_2 T^2 + a_3 T^3) - (\alpha_{material} - \alpha_{SG})(T - T_{ref})$$

4.6 Settings

This column is used to define the details of the setting made in column “Connection”. Depending on the type of connection, you can via the context menu (*right* mouse button):

- define the type of time marker generation with “Configure time channel”
- use “Configure I/O channel” to select analog or digital inputs or outputs, device channel or sub-channel number, signal type and sample rate group, depending on the hardware
- choose with “Sensor” a sensor from the Sensor Database or perform a sensor scan
- define with “Scaling” the transducer characteristic manually
- select the sample rate group for a channel when using a MGCplus. As this can be done from dialog “Configure I/O channel” as well, the settings are described in Section 3, *Configure time channels*.

Configuring time channels is dealt with in this Chapter in Section 3, page D-17, and therefore not discussed in the next part.

Double click on a channel in this column to open the dialog for setting the details for the connection defined.

4.6.1 Configure I/O channel

Automatic connection of the channels present in the device

With the devices MGCplus, Spider8 or DMC*plus* the automatic channel detection enables fast assignment.

With some MGCplus channel configurations, e.g. the ML7x-plug-in modules, there may be a link resource conflict under unfavorable conditions. In this case, either change the I/O Definition or go to the device setup and change the configuration of the relevant channels, see Section *Test link resources*, page D-48.

Even though you are free to make the definition as you wish, we recommend putting physically consecutive channels one after another into the I/O Definition unless not special settings are required due to a certain scaling mode or computation.

If the automatically entered channel numbers are not the correct ones, you can reallocate device channels to I/O channels with the option “*Assign subsequent hardware channel numbers for the … following I/O channels*”. Here you then enter the number of the channels to be (re-)connected.

The device connection for the I/O channels must already exist—it is not created.

Connecting a device channel with the *same* signal to two I/O channels is considered a double link and is not allowed. Note that this point is only checked when the channels are initialized before measurement starts. However, connecting, for example, the gross and the net signal of the same MGCplus channel is allowed.

Subchannels

Subchannels are used at present only by MGCplus: both multiple channel slots as well as the CAN bus and Profibus plug-in units use subchannels to access the individual channels as they only occupy one *slot*. At present, you can only enter a signal type in MGCplus, as this device can transfer both scaled values (“*Scaled to engineering units*”) and original measured “raw values” (“*Electrical unit*”) to catman.

Lock (disable) zeroing

If necessary, you can block zeroing by catman for a certain channel. Then a zero adjustment may only be carried out from the device setup or with a low-level script command, as in these cases the device is accessed directly. See also Section 4.1, *Zeroing channels (zero adjustment)*, page D-22.

Analog outputs, digital inputs and outputs

With suitable measuring devices you can also specify analog outputs as well as digital inputs or outputs.

☞ Analog outputs can only be used in Level 3 under the catman script language. Digital inputs and outputs can also be used with the Measurement Wizard or the event monitoring in Level 2.

For digital connections you must define the respective input or output by specifying a bit; the channel therefore contains only binary information about a particular input or output. You are recommended to check the documentation for the device concerned, or catman’s online Help on device setups, to see how many digital input and output options are available, which contacts are used, and how the levels for “*TRUE*” (digit 1 in catman) and “*FALSE*” (digit 0) are defined on the device.

MGCplus: On the MGCplus the eight *Remote-control* contacts of the individual channels can be used separately as digital inputs, the four limit switches can be used as outputs, and the two analog outputs Vo1 and Vo2 can be used with identical values as analog outputs of the channel.

☞ In conjunction with the amplifier versions B the MGCplus offers two additional signals:

- Combined signal S5: defining this signal is only possible with the MGCplus Assistant software.
- "*Bitmask*" of the remote contacts.

The state of all eight remote contacts is output as a "*Remote-control bitmask*" (range of values 0…255 resp. 65,535 with 16 bits). A bit being set corresponds to an active remote contact. With the help of this signal and subsequent online computations ("Hard-coded functions → Logic", "*Check channel 1 for bit no.*") up to eight digital inputs per amplifier can be measured fast and *synchronously* to the analog signals.

Spider8: The Remote-control bitmask function is also available for Spider8. In this case channel 8 has to be defined as an *analog* channel.

Sample rate group

With MGCplus one of the three sample rate groups can be selected for a certain channel. The sample rate actually used is normally defined in the measuring setup just before the data acquisition. See also Section 3, *Configure time channels*, page D-17.

Max. read block

During high speed data acquisition, measured values are not transmitted singly from the measuring instrument to the PC, but in what are known as *blocks*, since the procedure would otherwise take too long. The value suggested by catman is based on the transfer rate at which the device can operate: For fast devices 1,024 and for slow ones usually 10.

☞ If you want to use FFT computations in real time, each computation uses one read block. In such cases it may be necessary to define correspondingly high values for the maximum read-block size.

⚠ The total storage capacity required for the read blocks may not exceed 128,000 values, i.e. with 128 channels not more than 1,024 values per channel are allowed, for 256 channels only 512.

Ring buffer

All ring buffers should be deactivated during normal operation of catman.

If catman should work as Remote Data Server or be controlled via ActiveX and it is necessary to exchange data, we recommend that a ring buffer is created here for all channels whose values will be transferred. This reduces the probability of losing data as this allows more time for the data to be collected for the clients or the control program. Otherwise only the read blocks are used as buffers and are always overwritten by the next block read.

The size of the ring buffer depends on the sample rate, the number of clients and the speed of the network. We recommend that you make the ring buffer as large as possible so that there is a span of at least three to five seconds until the measurements have to be read from the buffers by the clients. The ring buffers are set up in the RAM of the PC. A larger RAM storage space must be available when using many channels. We recommend at least 512 megabytes when operating as Remote Data Server and using more than 500 channels.

4.6.2 File (online import)

The file to be imported is specified by the usual Windows procedure, either following a double-click in the column, via the context menu or through menu item “Settings → Online import”. The file must be available in the format “*Binary for online import*” which can either be produced via the Database or created in a file with continuous recording of the acquired values during the measurement.

☞ You can only specify *one* file name for the data import, since all marked channels are read from the same file. Since the last file name to be specified applies to each case, you can define the file name in any row.

⚠ *All* channels existing in the file must be read in. You cannot select just certain channels. If you do not know how many channels are saved in the file, go to the Database import dialog and click on the file. The number of channels in the file will be displayed in the dialog window.

4.7 Sensor

This column shows the settings from the Sensor Database or TEDS information. You will find further information in Section 5, *The Sensor Database*, page D-54. Please read also the information in Section 4.9, *Scaling*, page D-42, concerning modification of sensor data derived from the Sensor Database ( ) for the current I/O Definition.

4.8 Filter

These functions are only available for the MGCplus and Spider8. The ML801 HD-Filters will not be considered.

☞ You can select a certain filter setting as default e.g. “*20% of sample rate*” from the menu “Options”, “Sensors” tab. The default setting is 15%.

Use to select one of the digital low pass filters available in the device for the marked channels. The list contains most of the filters which can be set for the Spider8 or MGCplus. However, not every filter can be used for each channel. If a setting is not possible, catman will use the next possible one. As a rule, this is the next lower cut-off frequency. The setting is however only determined on initialization.

☞ Particularly with the Spider8, only a few filters can actually be used. This depends on the sample rate, see Spider8-Assistant.

If you use the setting “*Auto*”, the actual filter cut-off frequency will depend first of all on the setting made from the menu “Options”, “Sensors” tab. If possible, a Bessel filter of the frequency will be selected. If a Bessel filter is not available for the sample rate chosen, a Butterworth filter will be applied. If the filter frequency selected is too high, the next possible lower filter frequency will be applied.

A filter with Bessel characteristic is always selected as default filter as this guarantees maximum signal exactness, i.e. the signal is not distorted in the time range. Unfortunately, the interference suppression is not very effective for this filter, as there is a large transition range. High frequency interferences with higher amplitude are only insufficiently suppressed. If you require effective suppression of high frequency interference, you must select a manual filter with Butterworth characteristic. However, note that with this filter the signal is distorted in the time range, e.g. overshooting lies in the area of 10%.

☞ The Spider8 can only use approx. 1/8 of the sample rate as highest frequency. A setting of more than 15% of the sample rate will therefore never be effective.

If you use the setting “*Auto*”, you must initialize the I/O channels each time you change the sample rate otherwise the filter settings will not be activated: Setting “*Initialize I/O channels prior to measurement*” in the Measurement Wizard and catModule.

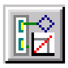
4.9 Scaling

With MGCplus and Spider8 we recommend you use the Sensor Database instead of setting a user scaling manually, see Section 5, page D-54.

In this column you can choose between “External” scaling (that is scaling which is—if possible—carried out in the actual measuring device), user-specific scaling or linearization with standard curves for thermocouples or Pt100.

The last item “Configure scale” in the context menu is only available if you have already selected a user scaling, which you may then modify. Otherwise the setup window for scaling opens when you first select “User”.

☞



As default, a user scaling is no longer allowed for a channel, if a sensor is selected from the Sensor Database or a TEDS sensor is used. However, sensor data derived from the Sensor Database may be changed with “Sensor → Modify Sensor adaptation” or . The modification applies for the selected channel(s) only, the data in the Sensor Database is not affected. This enables you, for example, to set gage and bridge factor for a strain gage channel as well as the temperature compensation method.

Linearization table

To specify the scaling you can use a “Linearization table” containing up to 32 value pairs, a “Polynomial”, a “Function f(x)” or “Strain gage scaling”. In Fig. 13 on page D-44 you can see the scaling for a “Linearization table” and a transducer with an 2 mV/V output signal at a nominal load of 50 kg.

You can also use the “*Measure*” key to apply the actual measured values from the device at the selected row (load level). However, you must then carry out loading with the corresponding data; in the above example, once with 0 kg (transducer unloaded) and once with 50 kg (nominal load). For certain devices, such as the MGCplus, you must also specify the required signal (gross, net etc.).

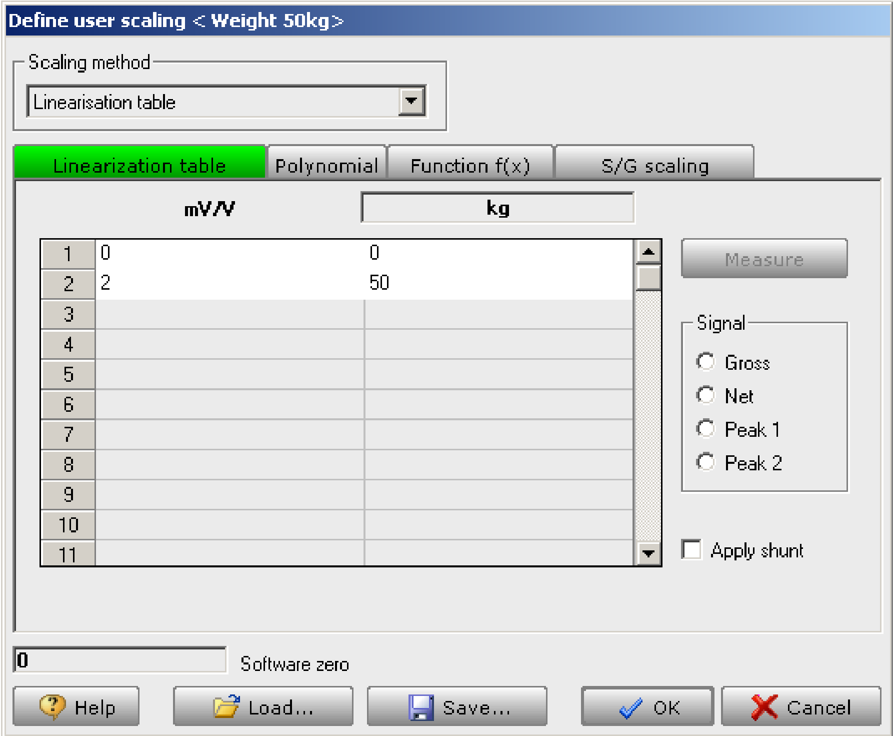


Figure 13: The “Linearization table” with a user scaling

Strain gage scaling

Scaling of the “Strain gage scaling” type enables mV/V values from the bridge circuit to be converted taking account of the gage factor and the bridge factor (number and direction of the strain gages). The resulting unit is always µm/m.

The measuring device must supply mV/V, otherwise the conversion will not be correct. If only the temperature compensation should be carried out as described below, activate “*Temperature compensation only*”.

The three methods used to compensate temperature influences for strain gage (SG) measurements are:

1. “Classic” by using one active SG and another SG which acquires *only the thermal expansion* of the component. Both SGs are linked in a half-bridge and connected as one measuring channel.
2. By using a separate measurement channel with a SG, who *only registers the thermal expansion* of the component. This measurement can now be subtracted from several active channels, all of which contain only *one* active SG. In this way, *several* active channels can be compensated using one compensation SG if the temperatures of these SGs are similar enough. In this case, select “*Temperature compensation with compensation SG*“. The compensating SG is wired to a separate measurement channel and specified in frame “Temperature compensation values from”.

   ☞ The section “Temperature compensation values from” is only superimposed when “*Temperature compensation with compensation SG*” has been activated.
3. By specifying the polynomials on the SG packing and using *one* channel to measure the temperature on the measurement spot. *One* temperature channel is used to compensate for *several* active SGs. In this case, select “*Temperature compensation with polynomial*“. Specify the channel measuring the temperature on the measurement spot in frame “Temperature data from”. In addition to the polynomial, the co-efficients of the thermal expansion must be specified and the temperature at which these co-efficients were determined (“T_{Ref}”).

   ☞ The section “Temperature data from” is only superimposed when “*Temperature compensation with polynomial*” has been activated.

With the devices Spider8-30 and MGCplus with the AP814 or AP14, it is possible to activate a shunt resistance provided in the device and to measure the signal it produces. Due to the shunt, the devices expect an unbalance of 1 mV/V compared to the zero-balance signal and any deviation from this is interpreted as cable loss. From this loss a correction factor is formed with which the result must be multiplied so that the correct value can be specified. The determined factor is given after the measurement and used for further measurements. The measurements are carried out by catman automatically when “*Measure shunt corr. factor*” is clicked.

☞ Only the losses as currently present are acquired. Changes to the losses due to temperature variations on the cable and due to any ensuing resistance variations are not acquired.

Thermocouples and Ptxxx

You only need to specify this linearization if this is not already made in the device itself. The measured mV or Ohm values are then converted into thermal values with the aid of the standard linearization tables for the thermocouples or Pt100, Pt500 or Pt1000 concerned. The Celsius unit °C is always used. If you want a display in Fahrenheit (°F), you must define a real-time computation (“Hard-coded functions → Algebra”) in another channel: (degrees Fahrenheit) = 1.8 * (degrees Celsius) + 32.

⚠ When using thermocouples, catman® Professional can handle the linearization only. The temperature compensation of the interchange point from thermocouple leads to copper wire (cold-junction compensation) must take place in the device. If you are using a particular reference temperature, this must be taken into account by a real-time computation.

☞ In the case of the Spider8 you *must* use a linearization, since in this device only the cold-junction compensation is activated if you specify a thermocouple as the transducer for the channel.

Scaling “None”

The scaling specification “None” enables certain devices to read out AD converter values, i.e. not computed *digits*. This setting should only be used for test purposes.

4.10 Status/Reading

In this column you have the option to “Test device connection” or display the current signal: “Show measurement signal”. If everything is in order with the status check, “*OK*” appears. If an error message appears, click again with the right mouse button and select “Error codes”. A help window is displayed with explanations of the possible error codes and suggestions for correcting the situation.

Press to retrieve a measurement from all active channels. Use to start continuous measuring for *all active* channels which have as source an *analog* signal (“*Analog IN*“).

With a device connection test, the only check made is whether the device can be addressed via an interface and a response is given. There is no check on whether all the measurement channels are connected and set up correctly. However, if you retrieve the current signal, any channel overflow will be displayed as “*OVFL*”.

The zeroing settings are discussed in Section 4.1, *Zeroing channels (zero adjustment)*, page D-22.

4.11 Sample rate

In this column, you can assign individual channels to one of the possible sample rate groups when using a MGCplus. The same possibilities are available as discussed in Section 4.6, *Settings*, page D-36. See also Section 3, *Configure time channels*, page D-17 and Chapter E, Section 2, *Which sample rate is the right one?*, page E-4.

4.12 Target

You may use this column to select individual channels specifically for export to a file or the Database *during* measurement.

Both the catModules and the Measurement Wizard offer the possibility of retrospectively ignoring these settings and of, for example, saving all or no channels. From the settings in the Measurement Wizard, you can specify whether all channels should be saved in *one* file or in *different* files.

The entry of the file name and whether the data from several measurements should be appended to a file should also be done via the catModules or the Measurement Wizard. In the latter you can additionally specify whether the counter in the file name should be increased and a new file should be created.

4.13 Diagnosis functions

☞ Your access rights to “Device → Diagnosis” may have been withdrawn via the user administration.

Besides the menu items already discussed in Section 4.10, *Status/Reading*, these menu items are only needed if you experience communication problems with a device.

4.13.1 Test link resources

☞ This function can only be used with MGCplus.

There could be link resource conflicts if a MGCplus is equipped with many channels. This can occur because all measured values in the MGCplus are transferred in a certain pattern via the internal bus (link). The same amount of space is available for the transfer of data in the multi-channel plug-in modules as in the single channels. However this must be distributed between all channels. Therefore not all available values can be transferred at the same time. As soon as another card, e. g. an ML70 or ML71 (CAN bus), requests measured values, the necessary space is reserved. This happens, e. g. when switching on the MGCplus. If catman is then started and another signal requested, additional space may not be available for the transmission. This will lead to link resource conflicts. catman cannot prevent these conflicts, you can however take measures to remove the conflict.

How could a link resource conflict occur?

- An ML70, ML71 or ML78 read the gross values of a channel from a multi-channel plug-in module in order to process them. At the same time, the settings in the I/O Definition stipulate that catman should read the net values (default setting).
- An ML70, ML71 or ML78 should read both gross values as well as net values from a multi-channel plug-in module.
- The MGCplus is equipped with many channels (> 128). For example, several ML71 may be available which work with many channels (max. 128 per ML71) and in addition other ML801 multi-channel plug-in modules. Because of the number of channels, it is possible that a channel will not find any space in order to transfer its measured values when working with high sample rates. If this is not done before a new value has been received, a link resource conflict occurs.

What can you do to avoid this?

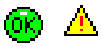
Find out the reason for the conflict. In most cases, you must call the device setup to see the MGCplus internal settings. With an ML70 you must under some circumstances also check the programming for which signals from which channels are requested. Then try either to use less signals or at least to use the same signals everywhere. This also applies to the I/O Definition in catman: Use there the same signals (gross/net), which are also used inside the MGCplus. Via the menu “Options → Channel” you can also define that the gross signal is always connected in catman instead of the net signal.

4.13.2 TCP/IP ping

The TCP/IP ping window is used to check for problems in the Ethernet interface and to check whether a connection can be made to the required address. If this is not successful, there is either a network fault or a firewall is installed which is blocking the access.

Procedure

1. Enter the address to be checked under “Hosts to ping”, e.g. “*192.168.200.5*”. You can also enter names, e.g. “*SERVERPC1*” if a name server (DNS server) or a corresponding host file is available.
2. Usually, the default values for the interval in which the request was sent (“*Ping interval*”) as well as for “*Timeout*” and “*Packets*” are already correct. If there is a lot of traffic on the network, you can increase the setting for the “*Timeout*”.
3. Then click on “*Start auto-ping*”.



The names entered will be resolved and catman tries to reach the addresses. After an answer is received, and “*Echo received*” appears in the “Status” column. If no answer is received, and “*Fail*” is displayed.

In this case, check:

- whether the address is correct
- whether the participant is located in the same network segment
- if the participant is not in the same network segment, whether the subnet mask or a firewall allows the access to other segments
- whether the participant is switched on
- whether other participants can be reached

If none of the checks leads to a solution, you must contact the network administrator.

4.13.3 Shutdown interpreter

This terminates the interpreter of the device, releasing the device for manual operation. This is normally inhibited once the PC has addressed the device for the first time which means that operation via the front panel is no longer possible. With devices which have no front panel, this command has no effect.

4.13.4 Enforce interpreter activation, enforce start initialization (MGCplus)

If this option is activated, catman attempts, during the next channel initialization (e. g. “Show measurement signal”), to reactivate the command interpreter of the device and to re-initialize the device. This can, for example, remedy errors which have arisen because the device has in the meantime been switched off and on again without terminating catman. The Interpreter activation is done only once before the next command to the device.

☞ The interpreter need not be terminated to use this command.

4.13.5 Log communication

☞ The interpretation of the log requires knowledge of the command set of the connected device. The function should normally only be used on instructions from the catman® Professional hotline.

To facilitate error diagnosis, a communication log can be activated and later viewed in a text editor. As default the file bears the name COMLOG.LOG and is located in the directory for your catman installation. This type of log contains the exact sequence of the command transfer to the device as well as the incoming responses, including time markers with 1 millisecond resolution. The number of entries is limited, however, you may change this number as well as the file name via “Options → Communication log”.

4.13.6 Show log file

This displays the communication log.

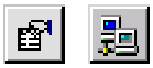
4.13.7 Terminal

☞ Access to this dialog may have been withdrawn via the user administration.

This function should only be used when you are very familiar with the relevant interface and the device used, in particular the device command set.

catman offers an integral terminal module for diagnostic and parameterization purposes. The terminal opens a communication channel to the device selected in the device list. Then you may send commands to this device and incoming responses will be displayed. The commands can be combined in a list and saved as a file. You can load these files later and execute them. This allows you to send an extensive sequence of commands to a device. You can also use special commands to program repeats, decisions, jumps, etc. in the command list ( ). The commands may be written in small/capital letters, the case is not taken into consideration.

☞ The terminal module can be also started as a stand-alone program without catman: catTerminal.exe in the catman directory. In this case, use  to configure the connection and connect it with .



5 The Sensor Database

This functional feature is only available for the MGCplus and Spider8.

5.1 Introduction

The Sensor Database allows the measuring chain to be easily adjusted for the sensor used. You only need enter the characteristic of the sensor once. The sensor can then be assigned to the channel simply by clicking it with the mouse. As long as the device supports it, it will then be adjusted to the characteristic value of the transducer:

- For MGCplus, the characteristic value will be transferred to the device for linear characteristics. MGCplus carries out the scaling.
- For Spider8 only the basic settings, such as the transducer circuit and measuring range, are carried out.

If you assign the transducer data using a program, e.g. catman, the program will perform the functions that the device cannot perform. For example, catman will perform simple scaling for Spider8 and multipoint scaling for Spider8 and MGCplus.

You can also use the sensor templates provided by HBM (HBM transducers group), if the characteristic values from the calibration protocol of a transducer are not available. With sensor templates the standard values given on the data sheet will be used for the characteristic value of the transducer, e.g. 2mV/V as sensitivity for nominal load.

You may also use transducer templates, if you want to input the sensitivity used or—with strain gages—gage and bridge factors at a later point in time. Choose then “Sensor → Modify Sensor adaptation”, as all other settings are already made by the template. A red marker in the upper right-hand corner of the sensor information cell (“Sensor” column) indicates such a modification.

Transducers with T-ID

If you are using transducers with T-ID module, first transfer each sensor into the Sensor Database, see Section 6, *TEDS and T-ID*, page D-63. A new data record with the unique number from the T-ID module will be created for each sensor and added on to the Sensor groups, and the current amplifier settings will be written to the database. Check the settings transferred and, if necessary, change the values using the data from the calibration protocol.

Transducers with TEDS

You can also transfer transducers with TEDS module into the database (same procedure as with T-ID). However, it is better to import the data directly from a TEDS file, otherwise the current amplifier settings will only be written to the database. You can also specify that only the data from the Sensor Database should be used: deactivate “Options → Sensor”, “*Activate TEDS always*”.

☞ The Sensor Database only needs to be available during the transducer definition. Once the required data has been saved in the I/O Definition, access to the Sensor Database is no longer needed.

5.2 Structure of the Sensor Database

Before you enter a transducer into the Sensor Database for the first time, you should consider the future structure of your database. After the installation of catman the “*Sensor templates*” from HBM transducers are available under “*Sensor groups*”. If you would like to create further templates, you should create them within a separate group. In this respect, you can also create other subgroups or subdivide the existing groups, e.g. the group “*Force transducer*” into “*Tensile force transducers*”, “*Compressive force transducers*” and “*Tensile and compressive force transducers*”.

To enter your own transducers in the Database, you should create one or more main groups. Depending on the size and organization of your company, you can, for example, group the transducers according to departments and therefore create a main group for each department (see left-hand side in Fig. 14 on page D-56). However, as with the templates, you can also use the individual types of transducer as main groups or define your company name as a main group and the transducer types as subgroups of it (see right-hand side Fig. 14 on page D-56). The structure is not needed to find transducers, but it is recommended to provide a quicker overview. In the case of doubt, you can always search the Sensor Database for transducers, see Section 5.6, page D-62.

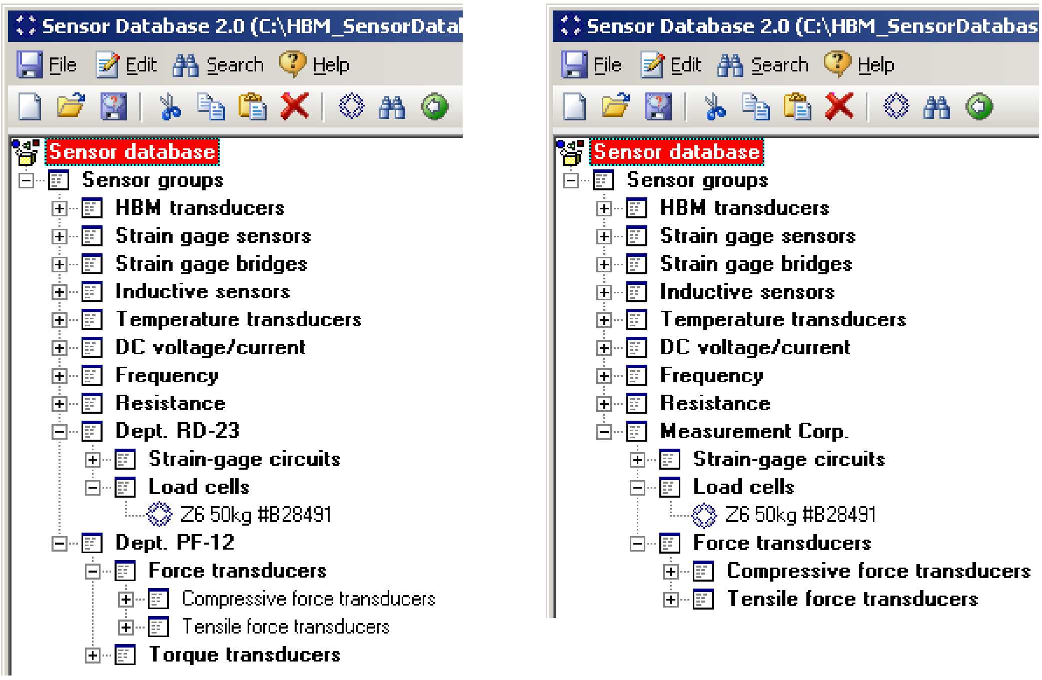


Figure 14: Examples for Sensor Database structures

Creating groups and moving existing transducers/groups

- To create a new group you must mark the higher level group and select “New sensor group” from the context menu.

- To move an existing *transducer*, you must cut it out via the context menu “Cut”, then mark the group to which it is to be appended and select “Paste” from the context menu.

  ☞ You can always only move one transducer with this method—multiple selection is not possible.

- To move an existing *group*, first use the context menu and select “Cut”. You then have two possibilities: if you want to move the group to another position within the same group, mark the group which is immediately *below* where you wish to move it and select “Insert”. If you wish to move the group *into* another group, mark the target group and select “Insert in group”.

5.3 Changing/creating/importing a Sensor Database

⚠ Before installing a new version of catman, copy the existing Sensor Database into another directory, otherwise it could be overwritten.

You can create a new Sensor Database or change to another database at any time via the “File” menu. As default Sensor Database the file HBM_SENSORDATABASE.SDB in the directory of catman is used. When you create a new Sensor Database, it is produced as an empty copy of the HBM database template SENSORS.MDB (without any sensor templates).

☞ It may be necessary to enter a new database file also in catman; otherwise the default setting will be used the next time the Sensor Database is called (I/O Definition, “Options → Sensor”).

Since the Sensor Database is based on an Microsoft Access database (Office 2000 or higher), you can also change the existing one by adding further fields or by creating your own databases. You must define at least the fields and tables used in the HBM database SENSORS.MDB. All entries must have a unique identification (max. 16 characters), the Sensor ID. Furthermore, you can create other fields or tables as required.

If you have imported many transducers or several times, you should then clean up the database: “File → Optimize database storage”.

5.3.1 Importing TEDS files

Use the “File → Import from TEDS” menu to transfer sensors having a TEDS data sheet saved as a file (virtual TEDS) into the Sensor Database. The data in the file will be evaluated and—as far as possible—transferred into the Sensor Database. Afterwards, check that the entries are complete so that no error message will be issued later when sensors are being assigned.

☞ Some TEDS data, e.g. minimum and maximum excitation voltage, will not be transferred as there is no field for this data in the Sensor Database.

You can also connect the sensor to the MGCplus, transfer the data to the amplifier and from there transfer it into the Sensor Database. In this case, however, only the current amplifier settings are transferred. If for example the nominal excitation voltage of the sensor is higher than the current amplifier allows, only the current setting will be transferred, not the nominal excitation voltage entered in the TEDS.

5.3.2 Importing older versions of the Sensor Database

Using the menu “File → Import from Sensor Database” you can import older or other Sensor Databases. You may also import an Access database provided the format corresponds to the SENSORS.MDB template. When importing, new transducers will be created. If entries already exist (same Sensor ID), the current data will be overwritten by the data from the import file.

If you do not wish to import all the transducers from an old Sensor Database, first copy the file (“File → Save Sensor Database as” and “File → Open Sensor Database”), then delete all unwanted groups and transducers from the copy, open the target database and import the remaining groups and transducers from the copy.

5.4 Entering transducers in the Sensor Database

☞ If this is the first transducer which you want to enter, you should first consider the principle structure of your Sensor Database, see Section 5.2, page D-55.

5.4.1 Copying existing transducers

Instead of creating a new transducer, it is often simpler to copy an existing one and correct the relevant data.

Procedure

- Copy the desired transducer via the context menu.
- Then mark the group to which this transducer is to be appended and insert it via the context menu.
- Either leave the suggested “*Sensor ID*” or enter a new one on the “Sensor data” tab.

- Then change the “*Description*” and the relevant data
- After clicking on “*Update entry*”, the data is assigned to the transducer entry.

5.4.2 Entering new transducers

If your transducer has a sensor identification (TEDS/T-ID) and your measurement device supports it, you should then also use it, see Section 6, page D-63.

Procedure

- Mark the group under which this transducer is to appear.
- Create a new entry in this group via the context menu.
- Then fill out the fields on the “Sensor data” tab.

You can also copy a Sensor template (group HBM transducers) and insert it into the required group. You must then correct any data which differs (sensitivity).

5.4.3 Changing existing entries

Entries can be changed at any time: Mark the entry, carry out the changes and click on “*Update entry*”.

5.5 How to use the Sensor Database

The following cases are possible:

1. The transducers to be used are already entered in the database.
2. Only some of the transducers to be used are entered in the database.
3. The transducers are not entered in the database, but you would like to work with standard parameters and not use values from any calibration protocols available: Use the sensor templates.
4. You have not yet entered any of your transducers into the database and would like to use the values from available calibration protocols.

In the second case you can first accept into the I/O Definition, as in Case 1, the transducers already entered. With the other transducers you proceed as required as in Case 3 or 4.

Procedure for entering new transducers

1. Open the Sensor Database via the menu “File → Edit Sensor Database”.
2. Create a new entry in the sensor group under which this transducer is to appear: Mark the required group and select “New sensor” from the context menu.
3. Leave the suggested “*Sensor ID*” (“Sensor data” tab).
4. On the “Sensor data” tab enter an appropriate text as “*Description*”. The “*Description*” may also include the serial number or inventory number of the transducer. You should identify the transducer so clearly that you can identify it at any time with no problem based on this text. The description text is displayed in column “Sensor” of the I/O Definition after connection.
5. Choose the circuit type used for the transducer connection under “Transducer settings” and specify the required excitation voltage.
6. You determine the linearization used via the “Transducer characteristic”. Enter the data from the calibration protocol and the “*Physical unit*”.
7. Assign this data to the transducer entry by clicking on “*Update entry*”.
8. You can now repeat steps 2 to 7 for other transducers. Once all new transducers have been entered into the database, close the window with “*Quit*” and you can now connect transducers from the Sensor Database.

5.6 Searching the Database for transducers

You can search for transducers via the menu “Find” according to the description you have entered, according to the type, the serial number or the Sensor ID. You can also carry out a search in the description fields with .

The results of the search are displayed in the main group “S*earch results*” (subgroup of “*Sensor database*”).

☞ Use “*” as a place-holder for any text and “*?*” as a place-holder for a single character. The place-holders can occur in front, behind and/or in the middle of the search term. Note that a place-holder must also be used at the end if the search term does not match the end of the term. Use of upper/lower case notation is ignored.

Examples of a search

Searching for “**SG*bridge*” finds “*SG quarter bridge*”, “*WB-SG bridge*” and “*SG full bridge*”, but not “*SG bridge with k=2.04*”.

Searching for “**SG*bridge**” finds “*SG quarter bridge*”, “*WB-SG bridge*” and “*SG full bridge*”, and “*SG bridge with k=2.04*”.

Searching for “*SG?bridge**” finds “*SG bridge with k=2.04*”, but not “*SG quarter bridge*”, “*WB-SG bridge*” and “*SG full bridge*”.

6 TEDS and T-ID

These functions are only available with connection plates with the identification i, e.g. AP815i, of the MGCplus. In addition, transducers must be connected which have TEDS or T-ID built in.

Even if there is a connection plate with the identification i, it may be necessary to get a firmware update of the amplifier module in order to be able to use the functionality. This is not necessary if you purchased both items at the same time.

The T-ID (**T**ransducer **Id**entification) allows you to recognize transducers automatically in connection with the Sensor Database and to set the measuring chain accordingly: Plug&Play for the measuring chain. With TEDS (**T**ransducer **E**lectronic **D**ata **S**heet, IEEE P1451.4) even the Sensor Database is not necessary as all information is already contained in the TEDS module. You may however define with “Options → Sensor” that settings from the Sensor Database override the settings from TEDS (deactivate option “*Activate TEDS always*”). See also Section 5, *The Sensor Database*, page D-54.

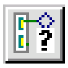
Do not confuse *T-ID* with the *Sensor ID* in the Sensor Database. When entering a transducer in the Sensor Database, the Sensor ID can be freely assigned. For TEDS and T-ID, the transducer or connection plug has a unique worldwide identification, which is stored in a chip (TEDS or T-ID module). The T-ID module contains only this identification; the TEDS module also contains the transducer characteristic. On request, this identification can be transferred as Sensor ID from the module to the amplifier. The sensor is thus *uniquely* identifiable. If only the Sensor Database is used, this could result in an incorrect assignment.

If suitable transducers as well as the relevant amplifier modules and connection plates are available, you can:

1. Perform a sensor scan for all channels

   A connection is established to the amplifier module and the T-ID or TEDS information is requested. If a T-ID module is connected, a search will be made through the Sensor Database and the necessary transducer characteristic is sent to the amplifier for comparison. With TEDS, the settings are directly activated, unless you have deactivated “*Activate TEDS always*” in the “Options → Sensor” menu: in this case, the settings in the Database, not the TEDS settings, will be activated if an entry exists in the Sensor Database. The measuring chain (device and transducer) must be connected.

Call the context menu from the “Sensor” column if you only wish to perform a sensor scan for marked channels.

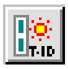


2. Check the Sensor ID

   If the I/O channel is coupled to a sensor from the Sensor Database, you can use this function to check whether this transducer is also connected to the amplifier module. catman reads the Sensor ID of the transducer connected from the device and compares it with that in the Database. The measuring chain (device and transducer) must be connected. This function can also be performed automatically when initializing channels: “Options” menu, “Sensors” tab: “*Sensor ID check on channel initialization*”.

3. Add Sensor ID to Sensor Database

   Using this menu command, you can create a new entry which contains the Sensor ID together with the current settings for the amplifier module. Modifications, e.g. an appropriate transducer description, can also be entered later through the Sensor Database, see Section 5.4, *Entering transducers in the Sensor Database*, page D-59. The measuring chain (device and transducer) must be connected. This menu item may also be used, if the Sensor ID was not found in the Sensor Database.



4. Activate LED on T-ID lmodule (Sensor LED test)

   When using a T-ID module from HBM, you may start the blink mode of the LED on that chip with this symbol.

Notes

- All functions which make a connection with the device and read the TEDS or T-ID information from the transducer need up to half a second per channel. If there are many channels, the operation can therefore take several minutes.
- If a Sensor ID other than that defined for the channel is found, the specified and the actual data will be output with a corresponding message.

E Measuring with catman® Professional

1 Introduction

☞ Chapter B, Section 5, *Getting started (Quick Start)*, page B-22, gives you a step-by-step introduction to data acquisition using a catModule.

Whether you want to work at Level 1 with a catModule or at Level 2 with the Measurement Wizard, in either case you have to consider what measurement speed you want, that is, how many measured values per second you need. In Section 2, *Which sample rate is the right one?*, page E-4, there is a short discussion about how to determine the necessary sample rate.

The settings for sample rate and read-block size using one or several devices are explained in detail in Section 3, *Define measurement settings*, page E-7.

Triggered measurements are explained in Section 4, *Triggered measurements*, page E-16.

Two special types of measurement are explained in the last two sections: Remote Data Server and DAQ background process. The Remote Data Server (starting at page E-37) is an operating mode which acquires data and provides the data for other PCs (clients). The DAQ background process (starting at page E-39) enables a slow data acquisition as background process in Windows.

2 Which sample rate is the right one?

A question often asked in digital measurement technique is what is the right measuring or sampling rate. However, it is not that difficult to find the answer.

Fig. 15 shows what happens if you select a sample rate that is too low for a sampling measurement system.

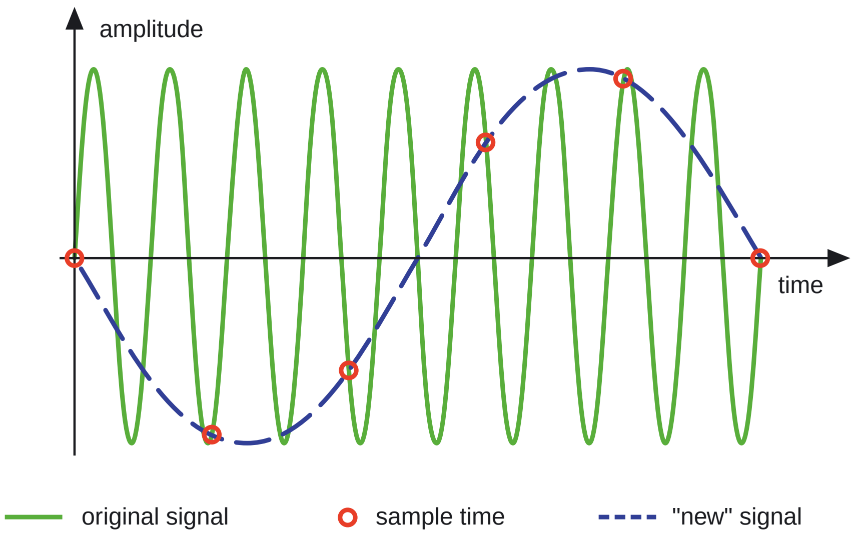


Figure 15: Alias effect with too low sampling rate

In the picture the dashed line shows the curve forming a graph when the sample rate is too low. The solid line shows the original signal which is sampled at certain times. We call the effect alias, i.e. you see a completely different curve. Therefore, many devices which operate by sampling have so-called anti-aliasing filters. With all HBM devices purely integrating AD converters are used with which this type of error cannot occur, because frequencies higher than the conversion rate are strongly suppressed.

However, a measuring error can also occur here if the values measured internally are not read fully from the device. If, for example, the original signal in the figure was measured with 10,000 values, but only 5 of them were transferred to the PC (e.g. MGCplus and sample rate 5 Hz), this will result in an alias effect on the PC.

Spider8 and MGCplus

Activate "*Use auto-filter*" as default via "Options → Channel" and enter a value which is between 5% and 20% of sample rate (see also Chapter D, Section 4.8, *Filter*, page D-41). Then use the setting "*Auto*" in the "Filter" column (I/O Definition).

A filter with Bessel characteristic is always selected as default filter as this guarantees maximum signal exactness, i. e. the signal is not distorted in the time range. Unfortunately, the interference suppression is not very effective for this filter, as there is a large transition range. High frequency interferences with higher amplitude are only insufficiently suppressed. If you require effective suppression of high frequency interference, you must select a manual filter with Butterworth characteristic. However, note that with this filter the signal is distorted in the time range, e.g. overshooting lies in the area of 10%.

☞ In the factory setting the Spider8 always selects a low pass filter which is equivalent to approx. one eight of the required conversion rate. If the settings for the filter you have selected or set up under Options cannot be carried out, either the next possible lower cut-off frequency will be selected or 0.1 Hz will be used.

Other devices, manual setting with MGCplus

The most important question which you must answer here is that regarding the bandwidth of your system. Therefore, check the analog bandwidth of your measuring amplifier. With HBM devices the bandwidth is restricted, depending on the type of amplifier (low or high carrier frequency, etc.), to a few tens of Hz up to a few kilohertz. The AD converter always operates with a conversion rate with is significantly higher (internally more than eight times) than the analog bandwidth. Therefore no error due to aliasing can occur here during AD conversion. However, you need not transfer all values processed by the AD converter to the PC, i. e. display them. If you therefore want to be certain that no values are lost and that you also receive all values measured on the curve, then you must use a sample rate that is as large as possible in catman. In all HBM devices apart from the scanning systems, there are digital low pass filters available after the AD conversion to eliminate unwanted interference. First check to which bandwidth this filter is set. Then set the sample rate to five times this value if you use Butterworth filters to be on the safe side. If you use Bessel filters take ten to 20 times the sample rate.

☞ However, if useful signals instead of interference are present in the region suppressed by the low pass filter, these are not measured and also cannot therefore be detected.

Recommendation

If you do not know the type of signal to be measured, you should first select the largest possible analog bandwidth (maximum frequency of the low pass filter) and the highest sample rate available. In this operating mode you can, for example, determine with the catModule 2-channel frequency analysis which signal frequencies are occurring. Then set the bandwidth with the low pass filter to the value required (to be certain you can specify double the value) and use—depending on the filter—the appropriate factor to calculate the sample rate.

☞ The DMC*plus* uses—if not otherwise specified—a sliding mean computation over all measured values which are not transferred to the computer. With MGCplus a mean value calculation in the CP42 can be activated for sample rates under 60 Hz.

3 Define measurement settings

The dialog Measurement settings is used to input this setting. You can access it via the “Measure → Measurement settings” menu item in the I/O Definition worksheet, or via the “*Measurement settings*” button in the catModules, or via the “Measurement settings” tab in the Measurement Wizard.

The Trigger settings which also appear in the “Measurement settings” dialog box are dealt with in Section 4, *Triggered measurements*, page E-16.

☞ The Measurement settings are saved along with the settings for the I/O Definition worksheet and with the Measurement Wizard settings. The settings loaded last are valid.

3.1 Sample rate setting for one device

If you are measuring with one device only, set up the “Sample rate” as recommended in Section 2, *Which sample rate is the right one?*, page E-4 and keep “*Device-internal timing*”. If you want to measure with a very low sample rate, use “*PC-internal timing*” instead and specify the time interval between two measurements.

3.2 Sample rate settings for more than one device

When you connect more than one device, you must define how the devices are to operate together: All with the same speed or some with different sample rates, all synchronous or just starting as simultaneously as possible, etc. Unfortunately, catman cannot fulfill every wish and the available possibilities also depend on the devices used.

Firstly, the devices can be differentiated into three types:

1. Devices which you can inform about the number of desired measurement values as well as the sample rate to be maintained. These are the devices MGC, DMC*plus*, DMC9012A, Spider8 and the NI cards.
2. Devices having channels to which different sample rates can be assigned and which also enable triggered switchover between different sample rates. Currently, only the MGCplus belongs to this group.
3. Devices which can only supply one measured value (per channel) on request. These are all the devices not listed above.

The internal procedure of catman during a measurement must be quite different depending on the device:

- In the first and second case it is sufficient to start the measurement once and thereafter the values only need to be read in.
- In the third case each measurement value must be requested separately.

Even if the reading in of the measured values does not occur quickly enough, in the first two cases this has no effect on the measurement provided the internal buffer memory of the measurement device does not overflow. With the third type of device the speed of measurement is limited by the time which is required to start the measurement, carry it out and fetch the value (or values, e.g. with scanning systems). With these devices the timing can only occur via catman.

Provided all connected devices belong to the same group, no problem arises, because then the corresponding timing can be selected generally in the I/O Definition and in the measurement settings.

Special conditions arise:

1. For devices which are synchronized via hardware: Case 1.
2. With mixed operation of devices from different groups: Case 2.
3. When the desired sample rates are not available or none of the devices can produce the desired sample rate: Case 3.

3.2.1 Case 1: devices which are synchronized via hardware

With the devices MGCplus and DMC*plus* the synchronization of a number of devices of the same type is possible: Connection via the sockets Synchr. in/out and the switch Master/Slave with MGCplus devices or connection via the socket SYN for DMC*plus* devices. In this case fully synchronous acquisition of all channels of all synchronized devices is possible.

When devices are synchronized via cable, they can no longer be operated in "*Stand alone*" mode (exception: newer MGCplus devices, see below). Here, an error message would occur when starting the measurement, because the devices receive incorrect commands during their initialization and return corresponding messages to catman. All synchronized devices except MGCplus must also be set to "*Master*" or "*Slave*" in catman in the dialog "Measurement settings" depending on the corresponding settings on the device itself.

☞ Devices of the same type can also be operated in the "*Stand alone*" mode without hardware synchronization, even via the same interface. The disadvantage is that during the measurements a time offset between the devices arises, because the start command must be individually sent to each device. The time offset is however very slight and lies in the region of milliseconds, because the start commands are output directly consecutively. Please keep in mind that without synchronization the carrier frequency amplifiers of different devices may interfere with each other.

⚠ Synchronized devices must always be started by the master.

Devices which are synchronized by hardware must be operated with the same sample rate or at least with the same highest sample rate; the device which is defined as master also produces the master time channel (Master clock). The synchronized devices operate as a single device with the sum of the individual channels; the fact that the channels are located in a number of housings does not matter for the measurement. If synchronized devices are not operated with the same sample rate, the measurement of one of the devices operated as a slave is aborted when the master terminates its measurement.

MGCplus: MGCplus devices with a CP32 with at least firmware version 2.06 or with CP42 can be connected via the synchronization cable and still be operated as "*Stand alone*" (switch at the back of the devices set to Master). In this case only the CF amplifiers are synchronized.

3.2.2 Case 2: devices from different groups

If for the clock generation one of the devices having its own clock generation can be used, i.e. this device has the highest sample rate, the time channel of this device becomes the *Master clock*. catman automatically detects this. You can therefore set the “*Source for time marker generation*” of the relevant time channel in the I/O Definition to “*Master clock*” or to device (device name) and all other devices can also be set to the device time marker.

Therefore the setting device internal timer marker generation (choose the device name from the list) is correct in most cases.

All slower devices and all which do not generate a time clock are then synchronized via the master clock. The possible sample rates depend in this case on the sample rate of the device having the master clock and on the possible settings in the drop-down lists of the dialog Measurement settings. Based on the master clock, catman calculates when each measurement value from the other devices is needed and sends the appropriate command to the device.

With devices which take a relatively long time to respond to such requests, e.g. with scanning systems, this can lead to problems. Since catman waits until the requested response is made, the measurement values from other devices participating in the measurement are not yet fetched. It must therefore be ensured that these devices can temporarily save their values acquired in this period. With MGCplus and DMC*plus* this is usually no problem, but may be with Spider8; in any case this depends on the sample rates and the number of channels as well as the time requirement of the slowest device.

Since the start of the measurement cannot occur simultaneously with all devices, a time offset is produced between the individual devices. The extent of this time offset depends in turn on the type of devices. All devices with internal clock generation only need for starting information about how the measurement is to be made and they can then start immediately. The output of this command only lasts a few milliseconds. This is different with devices which do not generate any time clock. Here, catman must request the measurement value and wait until it is available. The ensuing time offset normally lies in the region of a few hundred milliseconds per device, but with scanning systems it may be up to 10 seconds (measuring time of the system).

☞ Except for the MGCplus with hardware time channels, you cannot determine the offset of the measurements, because the time of the read-in and not the time of the measurement is used for the entry in the time channel.

3.2.3 Case 3: none of the devices can produce the desired sample rate

Just to point out at the outset if there is a requirement for a high sample rate: It is hardly possible to obtain more than 10 measurement values per second. The reason for this is that with the devices which supply a value only in response to a command, they must first measure and only then can output the data. With a measurement and response time for these devices of at least 50 to 100 milliseconds a sample rate of less then 10 Hz is produced, even with a fast PC. Another consequence is that you cannot define different sample rates for the individual devices (except when working in Level 3). The time marker generation must always be provided by the PC: “*PC-internal timing*”. However, these restrictions should be less critical for measurements that normally only occur every few minutes or even hours.

☞ Setting up a sample rate for devices that have internal time marker generation has a role to play even here: this speed is used for an individual measurement, that is, the duration of the measurement is the reciprocal of the sample rate. For a sample rate of 9,600 Hz this results in a measurement duration of around 104 µs; at 1 Hz 1 second is measured. Since in most HBM devices integration takes place over this period, you can also achieve suppression of interference in this way.

Consequences for the entry of devices in the I/O Definition:

During a measurement the values are fetched from the devices in the order of their entry in the device list. Therefore, devices which can only execute individual measurements should either:

- be placed at the start of the list if all measurements are to occur with the least time offset

or

- at the end of the list if the starting time of the measurement or the entry in the time channel is to be as accurate as possible.

3.3 Advanced view

3.3.1 Setting up the read-block size

Normally, you should always have this option, “Read block”: “*Determine automatically*”, activated. catman then handles the computation of the read-block size for you. A read block consists of a number of measured values which are read “in one go”. The manual setting of the read-block size is only necessary in special cases.

Reasons for the manual setting of the read-block size

1. A certain size of read block is needed for FFT computations.
2. With fast measurements or large displays the PC performance is not sufficient to enable a delay-free display.

Case 1: Here, a size must be entered which produces 2^n so that the FFT can be executed with all values of the read block.

Case 2: In this case the block size calculated by catman can be increased because it is only in this way that the PC is loaded less frequently with command transfers, screen output, etc. As the block size determined in catman is not visible, first use as an approximation value the highest sample rate that occurs divided by 10. Depending on the result, further changes may be necessary:

1. The result is more than 1,024 resp. the value entered for this channel in “*Max. read block*”.

   Enter the maximum block size (1,024, if not specified otherwise in dialog “Configure I/O channels”) and measure again. If the time delay still occurs, increase the maximum permissible size (worksheet I/O Definition).
2. The result is less than 1.

   Try a read-block size of 1.

If these suggestions do not help, then you only have three possibilities left: Make do without some graphical outputs, measure fewer channels or buy a faster PC.

3.3.2 Timeouts

These settings determine when

- to stop waiting for measured values and issue an error message
- to stop waiting for trigger events and issue an error message.

The read-block timeout setting must be increased only for very slow devices, e.g. for a fully equipped UPM100 with 100 milliseconds integration time per channel. For all other devices, the pre-set time of 10 seconds waiting time after requesting measured values is sufficient.

The trigger time-out must however be adjusted according to the signals monitored:
If you would like to register events which only seldom occur and over larger periods of time, this waiting time must be increased as otherwise catman will end the measuring or monitoring. For example, to define a waiting time of 3 hours, you must enter 10,800.

3.3.3 Suppress cyclical autocalibration

This setting is only relevant for the MGCplus: in this connection, the cyclic autocalibration *for all channels* of the device will be deactivated during the measurement (default).

If when setting up a channel, you have activated the periodical autocalibration (approx. every 5 minutes), in case of a measurement with this channel real-time measurements would be missing: only the transducer signal or a calibration signal can be measured. This option allows you to activate this internal autocalibration via the device setup so that, for example, autocalibration can be carried out between the measuring intervals. Nevertheless, the measurement will not be interrupted by the autocalibration, as this is deactivated for the duration of the measurement itself.

You should deactivate the option only for very slow measurements (less than 1 measured value per second). “Measured values” will also be output from the channel during the autocalibration. This is however *only the last value actually acquired*, which is output at the sample rate specified until the autocalibration is ended.

3.3.4 Communication protocol

This setting refers to the protocol of the data transferred on the interface. It is only interesting if there are problems with the interface. Please read also Chapter D, Section 4.13, *Diagnosis functions*, page D-48.

3.3.5 Ring buffer for all channels

If catman should operate as Remote Data Server (see Section 7, page E-37) or is controlled via ActiveX and it is necessary to exchange data, we recommend that you create a ring buffer for all channels. As soon as the option is activated, you can enter the required storage depth.

The size of the ring buffer depends on the sample rate, the number of clients and the speed of the network. We recommend that you use a ring buffer size corresponding to a measuring time of at least three to five seconds to enable the clients to read out the buffer. The ring buffer is set up in the RAM of the PC; the RAM must be sufficiently large if many channels are used. We recommend at least 512 megabytes RAM when operating as Remote Data Server with more than 500 channels.

4 Triggered measurements

In many cases, instead of starting data acquisition by clicking on a button, it is preferable to start when a particular event occurs, for instance if a certain measured value is exceeded. For this purpose catman offers you the option to define trigger events and enable them for measurement.

☞ The setting “Trigger: *Timeout*” is explained in Section 3.3.2, page E-13. See also Section 6, *The Event monitor*, page E-22.

4.1 Define trigger events

You may call the *Trigger list* via the “Measure” menu. The dialog box allows you to define up to 10 different trigger events. Their default names are “*TRIG_1*”, “*TRIG_2*” etc., but you can specify your own names if you wish. The names can also be used in a script to activate trigger settings. For each trigger event you need to define the device in which trigger monitoring resides; this device then functions as the *Trigger master*. You must also specify whether a “*Pre-trigger*”, in other words a time-related history, must also be captured, as well as what type of trigger should take place and at what level.

Please note the following points in connection with triggered measurement:

1. Trigger events can only be defined for the following devices: MGCplus, DMC*plus*, DMC9012A, Spider8 and NI-MIO.
2. As a rule trigger events can only be defined for *one* device. Monitoring different triggers in different devices, which then also return data at different times, is only possible subject to restrictions (see Section 4.2, *Special cases for triggered measurements using several devices*, page E-18).
3. In any event only *one* Trigger master is allowed per device type, which means it is not possible to set up two groups of synchronized MGCplus devices.
4. When there are several synchronized devices, only trigger events for the device defined as “*Master*” are active; trigger events for devices in “*Slave*” operating mode are ignored.

5. Normally at any *one* time you can only ever have *one* trigger event that is active, i.e. that applies to the start of the next measurement. The MGCplus system is an exception. In this case up to 4 events in the trigger list can be linked by using logical operator AND or OR.
6. With MGCplus a trigger can be used to change the sample rate.

Synchronized devices (MGCplus, DMC*plus* and DMC9012A) are *always* started with a trigger event, upon which the *Master* activates the *Slaves*. catman carries out the necessary settings itself (in former versions of catman the setting “*Immediately*” was used for this. The setting is retained for compatability reasons only).

☞ Trigger events required for the measurement must be flagged in the “*Activate for measurement*” list.

It is possible with some devices to initiate the trigger event through a digital level at a control input on the device. This type of triggering is activated by means of the “*External*” setting.

☞ In the case of the MGCplus, the Remote 7 of a channel (pin 8) is used exclusively for this purpose at the standard single channel amplifiers. Please note that the channel must be activated in the I/O Definition. In the case of the Spider8 and DMC*plus*, please note that the pin used for this function, connector pin 14 of the Digital I/O socket, uses negative logic, that is, the start command is given on transition from high to low (short circuit to ground).

⚠ When using a pre-trigger, the device must save the resulting measured values during this time. As the Spider8 only has enough memory for about 1,000 values, an error message will be issued if the value specified is too large for the selected sample rate and duration. In this case, reduce the pre-trigger.

The setting “*Wait for pre-trigger*” means that in any case a trigger event will not be monitored until the pre-trigger time has elapsed first. The number of measured values is always constant in this case. If this option is not activated, it can happen that the trigger event arises before the complete pre-trigger time has elapsed, i.e. not as many measured values are recorded as was originally requested. Since this can lead to problems in the case of the devices DMC9012A and DMC*plus*, “*Wait for pre-trigger*” is already activated by default with these devices.

☞ The settings for the Trigger list are saved along with the settings for the I/O Definition worksheet.

4.2 Special cases for triggered measurements using several devices

Although in the case of triggered measurements we recommend that you use either one device only or synchronized devices only, you may also work using other configurations. On the other hand there are certain restrictions involved, so you should check each individual case to see whether the measurement you want is feasible. This section introduces a range of different cases and the problems that can arise in each one. The following characteristics are common to all cases:

- The setting "*Values per period*" always refers to the device with the highest sample rate. All other devices measure fewer values, according to their sample rates. In the course of measurement it may happen that due to rounding errors and the read-block sizes being used, fewer measured values are actually acquired from a device than intended, for example only 2,990 values instead of the 3,000 expected.
- With the aid of the "*Period*" and the sample rate, catman converts the relevant specifications into the number of measured values that the device with the highest sample rate is going to return. All HBM devices expect the number of measurements to be stated, it is not possible to specify a time interval.
- Values for the "*Pre-Trigger*" are specified as a percentage; catman converts also this to the number of values needed for the *pre-trigger*.
- It is not permissible to split several devices of the same type into two or more groups that are synchronized only with one another within each group. Only *one* trigger master may ever be defined for each device type.

- Also for triggered measurement, all synchronized devices should operate at the same (maximum) sample rate.
- A time-synchronized real-time display is not feasible; as a rule the display is delayed. This is because the pre-trigger data is only output when triggering takes place.

Case 1: Several devices with internal time-base generation are intended to monitor their own trigger events in each case.

This only works if the devices can buffer all requested data internally, otherwise some measured values get lost. Data is not output until the very last device has had a trigger event.

Case 2: Just one of several devices with internal time-base generation is intended to monitor trigger events and the others are to measure at the same time.

On starting the acquisition, all devices normally start to measure *immediately* without a trigger event, until they have measured their computed number of values. It must be possible for the data to be internally buffered until the triggering device has received a trigger event, otherwise some data gets lost. Data is not output until the trigger event has occurred. However, as a rule this is not the effect that was desired, since for one thing the trigger event may only occur when all the other devices have already completed their measurement.

A remedy is possible if the start of measurement in the triggering device can be output as a digital signal which can be used to start the other devices: for instance in the MGCplus triggering can take place on one of the limit switches (also on a limit switch tie). This signal is present as a digital signal on the rear connector plate and can be used on a DMC*plus* or Spider8 to start the measurement *with external trigger* (observe level). Similarly the Spider8 and DMC*plus* have digital outputs that indicate the status of the measurement and can be used as output in the manner described above. However, since it is only after the devices have been activated (start of measurement, MSR at zero) that the output RDY (Waiting for Trigger) is set to zero, either an external circuit must be used to create a digital tie between outputs RDY and MSR, or edge triggering must be used. As soon as RDY returns to HIGH level, the trigger occurs. The MGCplus has no output to indicate the start of measurement. With this method it must be noted that no device other than the triggering device itself can operate with pre-trigger.

Case 3: One device with internal time-base generation is intended to monitor trigger events and one or more devices without internal time-base generation are to measure at the same time.

As soon as the triggering device returns data, the other devices are also requested to return their data.

5 Test Manager

Data files in the catman formats already offer the possibility of saving a comment for each file. It is now possible to save further information such as tester, department, test object, etc. and to log all files created during a test: use the *Test Manager*. In addition to the (changeable) parameters (“*Date of test*”, “*Operator*” and “*Department*”) already entered, you can define up to 127 parameters and fill them with values.

☞ The “*Test name*” parameter cannot be changed; as the value entered here is used as the file name for the management file.

As soon as you activate “*Use Test Manager*” in the dialog window, the names of the files generated (maximum 128) will be entered automatically in the management file (*.TST), and if you are using the “*catman*” or “*ASCII + channel information*” formats, the cross reference to the management file will be entered with the measurement data.

Procedure

1. Open the Test Manager.
2. Specify the directory where the management file should be saved. Select a directory via or enter a new one.
3. Enter the file name of the management file as parameter “*Test name*”.

   ☞ The parameter name itself cannot be changed.
4. Specify the parameters (name and value) for the measurement. If necessary, change the parameters suggested and/or insert others.

   ☞ If you wish to re-use certain parameters, you can save them in a *template*.

- Activate “*Use Test Manager*” and close the dialog window.
- Perform the measurement. As usual enter which files should be saved and in which formats as well as where the files should be saved.
- After the measurement, reopen the Test Manager and *deactivate* “*Use Test Manager*”.

All information, parameters and the saved files will be displayed in the Data Explorer (Chapter I, Section 2.5.1, *The Data Explorer*, page I-15).

6 The Event monitor

Access to the Event monitor may have been withdrawn via the user administration.

With the Event monitor ( ) you can monitor events in all measuring modes and execute different actions, independent of the Online Page displayed. It does not matter whether you are using a catModule for the measurement, the Measurement Wizard, a script or the DAQ background process.

You must call the window before starting the catModule if you wish to use the event monitoring in other catModules as the Data logger or Long-term monitoring.

You can monitor the following individual events:

- Upper limit detection
- Lower limit detection
- Digital input on HIGH
- Keyboard event
- Database capacity check

In addition you have the possibility to combine several events with AND.

The following actions can be carried out:

- Set digital output
- Set analog output
- Write entry into log file
- Play a sound file (audio output)
- Send e-mail
- Switch on storage to Database
- Switch off storage to Database
- Save single measurement (all active channels) to Database
- Export data to file (from the Database)
- Clear the Database
- Stop and start the Measurement Wizard data acquisition

- Execute catman script command
- Execute VBScript procedure

You also have the possibility of executing several actions at the same time, i.e. when one event occurs.

Notes

- All settings are saved with the I/O Definition. Save the I/O Definition after defining the event monitoring required.
- The settings can also be saved in a separate file and reloaded from it: Menu “File → Export” and “File → Import”.
- The window can be displayed during a measurement and restricted to display the active events only.
- You can deactivate the event monitoring for individual events in the column “A” or for all events on the tab “Events”: “*Deactivate event monitoring*”.
- Explanations about the reaction times can be found in Section 6.2.1, page E-26.

6.1 Procedure

Set up event monitoring as follows:

1. Call the “Event monitor” dialog from the main menu or the I/O Definition with “Measure → Event monitor”.
2. First define the actions required
3. Then define the events to be monitored and enter the actions required.

☞ Events and the connected actions can be modified at any time.

6.1.1 Defining actions

Call the “Event monitor” dialog from the main menu or the I/O Definition using “Measure → Event monitor”. Click on the “Action” tab to display it.

1. Select the “*Action type*” from the list field and click on “*New action*”.
2. The action appears as an entry in the field below. Enter a meaningful name for the action.
3. If parameters are required for the action, e.g. a sound file, enter these.
4. *Click* on “*Apply settings*” to *activate*.

☞ You can only define combined actions after the basic individual actions have been defined. You can also combine actions which are already combined. However, you can only combine actions which appear in the list *above* the new definition. The list is sorted according to the creation time. Use DEL to delete a single action from the combination.

6.1.2 Defining events

Call the “Event monitor” dialog from the main menu or the I/O Definition using “Measure → Event monitor”. Click the “Events” tab to display it.

1. Enter an “*Event name*”, e.g. “*Maximum exceeded*”.
2. Define the type of the event to be monitored: “*Event type*”.

   ☞ You can only combine events if the basic events have first been defined.
3. Define the (catman) channel to be monitored for events based on measured values: “*Monitor channel*”. With a keyboard event, define the “*Key*” to be monitored, with combined events, mark all events you wish to combine via AND with ☒.

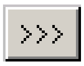
   For digital inputs, if the catman channel contains a “*Remote-control bitmask*” (MGCplus) or the channel 8 (Spider8), you must also select the bit to be checked. In all other cases, select the lowest bit. Keyboard events which are at the same time defined as keyboard events for the active Online Page will be executed by the event monitoring as well as the Online Page.

4. Enter the “*Action*” to be carried out.
5. If the action should be performed several times as long as the event is present, you can use “*Notify again after (ms)*”, to define the number of milliseconds which should elapse before the action is repeated. This is helpful, for example, if a user must react to a certain status and this request should therefore be repeated. Note however that the action can be performed within the specified time period. Do not enter a repeat time of 1,000 milliseconds for a sound file which is 3 seconds long.
6. Create the event definition using “*Add event*”. It will then be displayed in the list of the defined events on the right-hand side.

6.1.3 Changing events

There are two possibilities for changing events:

1. Click on the event in the list which is to be changed, then change the definition on the left-hand side and finish the modification with “*Modify event*”.



2. If you only wish to change one setting, which is possibly used for several events, mark the event you wish to change (by clicking together with SHIFT or CTRL), change the setting and click on .

6.2 Explanations of the actions

Some actions are simple to define: Entry in the log file, Save to Database, etc. In the following section, you will find more information about the individual actions as well as the reaction and execution times.

6.2.1 Reaction times

A check for events is made immediately after the measured values have been read (for script programmers: after `ACQRead`). Depending on the frequency with which the measured values are read in, the actions will also be started: for automatic computation of the read-block size through catman at least every 100 milliseconds. A reaction time of approx. 20 milliseconds can be achieved if the read-block size is reduced. However, this also depends on the PC being used and other conditions, such as the number and size of the graphs to be displayed and the channels.

☞ With the DAQ background process, on a fast PC with 50 to 100 Hz sample rate, read-block sizes of 1 are reached (10 to 20 milliseconds process interrupt time).

All events which are not combined are checked *before* combined events. The actions are carried out only *after all checks have been made*.

6.2.2 Execution times

⚠ Defined actions are only executed after all events have been checked.

However the reaction times are not identical to the execution times: these depend on the actions to be carried out. The reaction times are in the µs-range for actions which catman can execute itself. This is different for actions which are executed using another device or program:

- Set digital output
- Set analog output
- Play a sound file (audio output)
- Send e-mail

In these cases the reaction time of the device or the necessary programs must taken into consideration. It takes e.g. 20 to 50 milliseconds to set a digital output in the MGCplus. This is much faster (< 5 milliseconds) when using Spider8 or the National Instruments AD converter boards with digital outputs. The event monitoring is therefore not suitable for fast real time requirements. Further information can be found in corresponding sections about the actions.

6.2.3 Set digital output (Digital OUT)

Sets a digital output to HIGH or LOW. You can only use I/O channels of type "*Digital OUT*" (I/O Definition) therefore only channels of this type are displayed in "*I/O channel*".

You can choose between two modes:

1. "*Static*": The output remains set until it is reset by another action. You can also reset the output manually with a script command.
2. "*Puls*": The output is reset automatically by catman after n milliseconds, the "*Pulse duration*".

☞ The status is set via commands. With MGCplus, this can take 20 to 50 milliseconds per action. This is much faster with Spider8 or the National Instruments AD converter boards with digital outputs (< 5 milliseconds).

6.2.4 Set analog output (Analog OUT)

Sets an analog output to a certain level value (in Volt). You can only use I/O channels of type “*Analog OUT*”. Only channels of this type are therefore displayed in “*I/O channel*”.

When setting the output you may:

1. Output the limit value of the channel to be monitored: “*Limit value monitor channel*”
2. Output the current measured value of the channel being monitored: “*Current value monitor channel*”
3. Output a manually set value: “*Manual setting*”

☞ The signal is set via commands. With MGCplus, this can take 20 to 50 milliseconds per action. This is much faster with Spider8 or the National Instruments AD converter boards with digital outputs (< 5 milliseconds).

6.2.5 Entry in log file

☞ See also Chapter B, Section 7, *The catman log file (System log)*, page B-32.

Creates an entry in the log file. The text can also contain so-called system text, e.g. the names or the current measured value of the channels being monitored (monitor channel). The group to be assigned to the entry is defined using the log type.

The following system texts are available:

- “*%NM*”: Name of the monitor channel that the action uses
- “*%UM*”: Monitor channel unit
- “*%VM*”: Last (most recent) measured value available from the monitor channel. The formatting is done automatically.
- “*%NCxx*”: Name of any I/O channel. This is defined by the number, e.g. “*%NC25*”.
- “*%UCxx*” or “*%UC{<Name>}*”: Unit of any I/O channel. Either the number or the name of the channel can be entered, e.g. “*%UC25*” or “*%UC{SG_A_Wing_Top}*”.

- “*%VCxx*” or “*%VC{<Name>}*”: Last (most recent) measured value available from any I/O channel. Either the number or the name can be entered, e.g. “*%VC25*” or “*%VC{SG_A_Wing_Top}*”.
- “*%LV*”: Limit value (level) of the event.
- “*%D*”: Date.
- “*%T*”: Time.

6.2.6 Sound output

Plays a sound file (*.WAV). This may take several seconds depending on the size of the file. If you specify a repeat, make sure this time is longer than the duration of the sound.

6.2.7 Send e-mail

From catman 5.0 on the sending of e-mails takes place via CDO (Microsoft Collaboration Data Objects). Therefore, neither MS Exchange nor MS Outlook needs to be installed, but the SMTP service must be available. Use your Windows CD to install this component which is part of the Internet Information Service (IIS), if it is not already installed. Additionally “*SMTP server name*” and “*Sender address*” must have been set (“Options → Startup options”, “E-Mail” tab) and forwarding to an SMTP e-mail server without authorization (anonymous) must be possible. Check with your system administrator, if a Microsoft Exchange Server is used in the intranet. (With default settings this is possible. However, the system administrator can change this so that only specific PCs are allowed to do this.)

Sends an e-mail.

6.2.8 Store to Database ON/OFF

Turns saving of measured values to the Database on or off. No other parameters are required for this action.

If you save measured values via the event monitoring, this will not collide with other storage settings. However, these entries are not changed either. If, for example, you are using the Measurement Wizard and you activate Database saving manually by clicking on , this setting will not be affected by an event which turns the data saving off. In this case, data saving will only be stopped when it is manually turned off in the Measurement Wizard. However, the measured values are not saved twice in catman.

If you use this action, we recommend that you deactivate all other data storage options: markers in the I/O Definition, "*Store to Database*" on the tab for the type of measuring when using the Measurement Wizard, clicking on during the measurement. If data saving was activated with one of these methods, it *cannot be deactivated by an event*.

6.2.9 Save current measured value

Saves the last (most recent) measured value in the current read block of all channels in the Database. No other parameters are required for this action.

6.2.10 Export data from Database

Select the directory where the data files should be saved and enter the file names. catman appends a counter to the names that is increased each time the event occurs.

☞ If possible do not use a network when saving data.

If you are saving large amounts of data, ensure that the internal buffers in connected devices do not overrun. Otherwise the measuring operation will be cancelled. If you are using higher sample rates and slow devices, you should end the measuring operation while saving data and then restart it.

6.2.11 Clear Database

Deletes the entire database. The action is useful if you have already exported data and wish to save new values in the Database.

6.2.12 Stop/start acquisition with Measurement Wizard

The Start acquisition action cannot start an acquisition, if the check already requires a measurement. Therefore you cannot check if, for example, a certain value is exceeded and then start the measurement. However, you may check for key presses (keyboard event) or digital inputs.

It is useful to interrupt the measuring operation if, for example, you wish to save a large number of measured values. As this can take several seconds and could lead to an overrun of the internal device buffers if high sample rates are being used. This would then cancel the measuring operation.

Combine the actions: Stop data acquisition, export values, delete Database and restart acquisition.

6.2.13 Execute (catman) script command

Executes one or more script commands. Several commands can be separated through CRLF, also the use of @ is supported.

Only commands in an Auto Command List can be used, see Chapter J, Section 4.1, *Restrictions for Auto Command Lists*, page J-9.

If you require variables, you can use the catman internal variables `GP16` to `GP32` or declare your own using `VARIABLE`.

6.2.14 Execute VBScript procedure

To be able to use this action, you must load a VBScript into the Windows Script Host, for instance, from the VBScript editor (“Script → Load script in WS Host”), from a script or using the context menu in the Project window.

The action performs a procedure from a loaded VBScript. If a VBScript is loaded, all procedures in this script will be offered for selection.

6.3 Explanations of the events

6.3.1 Limit value

Additionally to level and “*Monitor channel*” you may specify a “*Hysteresis*” value. The hysteresis prevents from executing the action several times in case of superimposed noise: The trigger level is reached several times due to the additional noise level. If condition “*Above limit*” is chosen, the hysteresis is below the limit value, with “*Below limit*” it is above (Fig. 16).

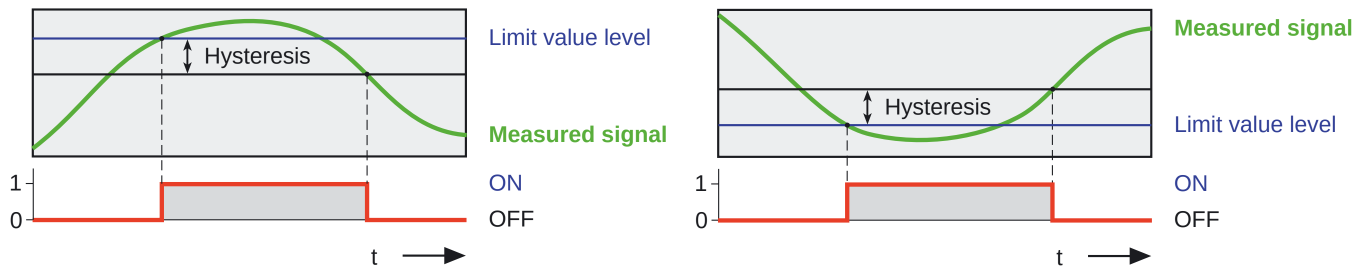


Figure 16: Limit values and hysteresis: left-hand side above limit, right-hand side below limit

6.3.2 Digital input

With most devices you define the digital input including the bit to be used for a catman I/O channel. In this case, select the lowest bit.

With the MGCplus, you may set the channel to "*Analog IN*" and use the "*Remote-control bit-mask*". With the DMC*plus*, you can also use "*Analog IN*" with "*Channel 0 (SY01)*", with Spider8 select "*Analog IN*" and "*Channel 8*". In these cases, all digital inputs will be read parallel to the measured values. You must therefore select the bit to be checked. Remote-control bitmasks are covered in Chapter D, *Defining Devices and Channels*, in the section starting at page D-38.

☞ If a single bit should be requested, catman must explicitly request this information. The procedure is carried out asynchronous after a read block has been read. However, the MGCplus requires up to 50 milliseconds to do this! It is therefore better to use the "*Remote-control bit-mask*" with MGCplus.

6.3.3 Keyboard event (key press)

Select the required key.

☞ Not all keys can be used. Only the key last pressed will be evaluated. With slower acquisitions, e.g. only one read block is read per second, events may get lost.

☞ Keyboard events which are defined at the same time as Keyboard events for the Online Page will be carried out both by the Event monitor as well as from the (active) Online Page.

6.3.4 Database capacity check

Checks the amount of space taken up on a certain channel (number of values stored). The action is performed if the percentage of the space occupied is larger than the limit entered under “*Filled up to %*”.

☞ The channel *must* be entered as catman channels may have different channel lengths (storage capacity).

6.3.5 Combining events

The following restrictions apply when combining events:

- Events can only be combined with AND. If you wish to realize a simple OR, define several events which should start the same action. Complicated conditions with Boolean algebra formulae are not supported at present.
- You can only combine events which have already been defined.
- Combined events are only checked after all other events have been evaluated.
- If one of the events to be combined is a combined event already, this must appear in the list *above* the new event being defined.

6.4 Displaying Events

☞ To be able to use the event monitoring in other catModules as the Data logger or Long-term monitoring, you must call the window before starting the module.

You can use the Event monitor in two different modes:

1. in operating mode
2. in display mode

In display mode, events and their statuses are only displayed in a list. In operating mode, you can define new events and actions or change existing ones.

During a measurement, display mode is always used first. You can however switch to operating mode from the menu View. Note that the reading in of measured values will be interrupted when menus or list fields are selected. This could lead to a buffer overrun in the measuring device, see also Chapter H, Section 2, *The Measurement Wizard status window*, page H-5.

Events where the condition has already been fulfilled, are marked in the column “Status” with , events which were active at least once since the start of catman are marked with .

Position the mouse pointer over the event to display the details: the tool tip contains the date and time of the last time the event occurred and how often.

The column “T” shows the symbol indicating the type of event. You can deactivate individual events by clicking on ☒ in column “A”.

Notes:

- The displays can be deleted with “View → Remove yellow LEDs”.
- If there are many active events, you can find the next active one with F3, once you have clicked on an event.
- You can reset the event counter shown in the tool tip for the event marked or for all events: Menu “Edit”. Resetting the counter also removes the display of .

7 Remote Data Server

☞ Access to this dialog may have been withdrawn via the user administration.

catman has the Remote Data Server function which gives other PCs (“clients”) also running catman access to measurement data. The data is transferred with TCP/IP via an Ethernet network to the client for display or further processing. In this case, the Remote Data Server must be selected as device in the I/O Definition on the client PCs. The total process will be handled automatically by catman.

☞ The speed at which the data is acquired is restricted through the number and speed of the PCs as well as the transfer rate of the network.

For the Remote Data Server, it is recommended that you set up ring buffers for the channels which can be used by the clients: Measurement settings, see Section 3.3.5, *Ring buffer for all channels*, page E-15. The values of the individual channels are then (temporarily) saved in the ring buffers.

The clients must be able read the resulting values fast enough otherwise values will get lost. If ring buffers exist, it is enough if the buffer is read before being overwritten with new values. Without ring buffers, each *read block* must be transferred *immediately*, otherwise data will be lost as the read-block buffer is overwritten by the next read block.

There are two important points to consider in order to determine whether data will be lost:

1. Are the clients quick enough to be able to obtain values in time from the ring buffer or read block buffer?

   Here, in addition to the sample rate of the Remote Data Server, the number of real-time computations and the type and size of the visualization on the client play a role.

2. Can the Remote Data Server answer the client quick enough?

   In this case, there are two important points to check: the network could be the reason why the clients are not being served quickly enough, e.g. because other network participants are active and the network is occupied by them. For an Ethernet, there is no “safe” data transfer rate, the network can be occupied by other participants for more than one second. Another critical area is the server itself: how quickly can the data be acquired and which real-time computations or visualizations should be done on the server. If the performance of the processor is not adequate, the clients will also not be provided regularly with data.

For maximum performance, we recommend, for example, that several connected MGCplus have their own network and data is transferred via a second network. You should keep the number of real-time computations and visualizations on the server to a minimum.

The clients must have the necessary authorization (Windows system administration) in order to be able to access the PC on which the Remote Data Server has been started.

The rate at which measurement values are acquired must be defined on the Remote Data Server. The clients can only *collect* the data acquired, the sample rate cannot normally be set here (there is one exception though!). If necessary you can reduce the data on the client, for example, by using a real-time computation. Exception: On the client, you can specify that only 1 value is retrieved for each "measurement": Set the size of the read block to "*1*" in the measurement settings. In this case, the first value in the current read block will be retrieved for each read operation if this value has not yet been transferred. All other values acquired will be lost. However, there is no guarantee that the same number of values will not be transferred. This depends on the reaction speed: different numbers of read blocks may not be requested if the clients do not query the server fast enough to obtain one value from each read block. For this reason, this setting is recommended only when it is possible to collect the measured values very slowly, e.g. every 5 seconds.

The Server port set-up is reserved for future extensions. At present, it may *not be changed*. If you have problems with this port, please contact the HBM Hotline.

As soon as you start the server ("*Server START*"), it will create the ring buffers specified and monitor the server port for queries from the clients. The measuring is not yet started, this only enables queries from the clients. Next, start a measurement, for example, from the Measurement Wizard in order to allow access to the data. During server operation, you can perform measurements as usual with catman. However, if the measurement is paused while the clients wait for data, then time-out errors will be issued on the clients if no data is available.

The diagnostics functions in the window are only of interest if you are in contact with the Hotline as faults can be easier localized when the interaction of server and clients can be seen.

8 DAQ background process

☞ Access to this dialog may have been withdrawn via the user administration.

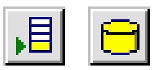
The DAQ background process allows measured values to be read, saved and displayed independent of any user action. The measurement is not interrupted, even if a menu or a dialog box is opened. This is opposite to normal measurement operation as the measuring sequence stops when, for example, a dialog box opens. Using the DAQ background process, an Online Page can be designed or a script written while 100 UPM100/Centipede channels are acquired and saved in the background. New visualization objects can be inserted interactively and will be updated immediately.

⚠ The DAQ background process carries out a continuous measurement until “*Stop*” is clicked.

At the present time, the DAQ background process can only be called from the main menu or from worksheet I/O Definition with “Measure → DAQ background process“. Once the settings have been made, the process can be started with “*Run*”.

8.1 Settings

☞ The sample rate must be defined via the measurement settings, e.g. with in the I/O Definition.



As with normal measurements, you can decide between “*Device-internal timing*” and the PC-internal timing (“*Single measurements*“). In the section “Database storage mode” you can define whether and how the channels are to be saved. The icons and can also be used. saves a single value in the Database and switches continuous saving on and off.

8.1.1 Measurements with device-internal timing

With this type of measurement the process interrupt time must first be specified. This is the time interval in which values are acquired from the device. The time must be more than 50 milliseconds (milliseconds) but should however only be large enough so that no values are lost, i. e. so that the internal measurement buffer of the device does not overrun (Spider8: approx. 1,000 values). Depending on this time and the sample rate set, the "*Read block*" must be calculated such that all values are transferred. You can also activate "*Dynamic read block*" for MGCplus and Spider8. In this case all values available in the output buffer will be read in.

Example:

For a sample rate of 600 values per second and a process interrupt time of 200 milliseconds a read-block size of 120 values is used. This means that 120 values will are fetched five times each second (every 200 milliseconds), i. e. a total of 600 values per second.

8.1.2 Single measurements

When you select "*Single measurement*", the field "*Process interrupt time*" changes into "*Sample interval*". The shortest permissible interval is 50 ms (milliseconds). The read-block size must be set to "*1*" for this type of measuring.

8.1.3 Online export

As with the Measurement Wizard, you can activate data saving for the DAQ background process during the measurement. However not all options can be accessed. With the DAQ background measurement, you may only export all channels or none at all. The file name can be incremented automatically each time the process starts. The process can also request whether the file should be overwritten.

8.1.4 Read-block timeout

catman will wait for the period of time set here for the requested measurement data before issuing an error message. For very slow devices, e.g. the scanning systems, it may be necessary to increase this value. See also Section 3.3.2, *Timeouts*, page E-13.

8.1.5 Ring buffer

You can also activate ring buffering for the Remote Data Server in this operating mode.

8.2 Restrictions

This type of data acquisition is primarily not intended for fast acquisition of large amounts of data. Despite this and depending on the PC hardware higher sample rates can also be used. The tasks to be carried out simultaneously by the PC must however always leave sufficient time for the acquisition and processing. During measurement with the DAQ background process another script can be executed, but this may not be used to collect data. However, computations can be made using the data already acquired.

☞ The following situation leads to an interruption of the process: The mouse key is pressed in the title bar of a window and not moved.

F catMODULES and Add-Ins

1 Introduction

Your access rights to individual catModules or to the measurement setup may have been withdrawn via the user administration.

The catModules cover all *standard* measurement tasks. They remain variable and are therefore suitable for a wide variety of applications. catModules provide a framework for the acquisition, transfer and display of measurement data. Use the I/O Definition to define which channels will be used for measurement and whether real-time computations will be performed. Then in the catModules you specify whether data will be saved to hard disk and/or written to the Database. Everything else up to and including display is handled by the catModules, each of which is optimized for a particular measurement task and contains various Online Pages for displaying output.

If you ever wish to return to the original configuration of the Online Pages of your catModules, copy all files from the CATMOD\BACKUP sub-directory to CATMOD. The sub-directories reside in the catman directory.

Each catModule has a Help function which gives you information on the functioning and operation of the module concerned.

We start with a section giving general hints on using catModules. This is followed by other sections describing the modules.

Basically, there are two different groups of catModules

1. Modules for measurement: Data logger, Periodic measurement, Long-term monitoring and Single value data acquisition
2. Modules for acquisition and post-processing: Single value DAQ to Excel and 2 Channel frequency analyzer

Using catModules is easy:

1. Make all the necessary settings for your device and measurement task, i.e. in the I/O Definition worksheet define your device(s), active channels and any computations. Also carry out the device setup (Chapter D, *Defining Devices and Channels*).
2. Use “Run catModule” from menu “Measure” to open the dialog with the catModules, or open the list of catModules in the Project window.

3. When using the catModule dialog, click on or double-click in the Project window.
4. In the first window (first page of the Online Document of the catModule concerned) define the conditions for your measurement: how long, how fast, save data etc.
5. Select one of the available “Visualization windows”, such as “*y(t) Real-time graph*”, and click on “*Run*”.
6. Specify for the graphs on the Online Page which channels you want to display and in which color etc. Use the context menu of the graph, menu item “Data sources”. In some modules you may also use , , or at the top of the window.
7. Click on or press F5; there is a short initialization phase and then measurement starts. The initialization phase can last longer than usual, if catman has to collect acquisition parameters (channel information) from many MGCplus device channels or UPM100/Centipede.

General instructions for using catModules

If you have called a worksheet and some of the symbols in the toolbar cannot be accessed, one of the catModules is still active. Quit the catModule to allow access to all settings.

Most catModules work with all active I/O channels. Enable the channels to be measured by clicking in the I/O Definition worksheet column marked with or with the “Measure” menu in the same worksheet. A green arrow marks an active input channel.

If the message is displayed during measuring that values cannot be saved any more, this indicates that the depth of a channel (or several channels) in your Database is too small. If necessary, increase the size from the catman® Professional main menu “Options → Database configuration”. See also Chapter C, Section 2, *Configuring the Database*, page C-4.

catModules work in conjunction with a visualization page, i.e. an Online Document. The page can be sized freely in order to view other programs which are running at the same time. When resizing catman tries to match the size of the objects displayed. However not all objects may be reduced, for example, the toolbar.

Dialog boxes and windows opened via the toolbar must normally be closed before you can continue reading in measured values. As the measurement continues during this time, the values already in the buffer of the measuring device are read as fast as possible after the dialog window has been closed: The measured values seem to run must faster in the graph. If the buffer in the device has overrun, the error message "*Buffer overflow in device*" will be issued and the measurement is stopped after the dialog box has been closed.

2 Data logger

☞ Access to this catModule may have been withdrawn via the user administration.

This catModule plots a continuous data stream of all active I/O channels and is particularly suited to recording longer measurement sequences with low sample rates. Several prebuilt Online Pages are available for display purposes. The module is well suited to continuous data recording with visual monitoring.

On the first page of the catModule you have to enter the settings for the data acquisition.

2.1 Settings

Click on “*Measurement settings*” to adjust the sample rate. Enter the number of measurements per second either in the column “Sample rate” or, if no internal time-base generation is available, use the “*PC-internal timing*”. Further information concerning sample rates can be found in Chapter E, *Measuring with catman® Professional*.

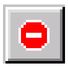
Initialize I/O channels before measurement

This option includes the following actions:

1. A check will be made which device, which interface and address and which hardware channel is used as input channel.
2. The hardware connection is made and checked.
3. The channel information (acquisition parameters), such as measurement range, unit, etc. will be read from the device.

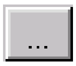
The option “*Initialize I/O channels before measurement*” *must* always be active when *first* running a catModule. Provided you make no changes to the channel setup (I/O Definition or device setup) or, if you are working with the filter setting “*Automatic from measuring rate*”, provided you do not change the measuring rate, you can again deactivate it for following measurements. Time can be saved in this way when starting the measurement, particularly when using MGCplus and UPM60/100. The initialization phase can take longer than usual, if catman has to collect acquisition parameters from many MGCplus device channels or UPM100/Centipede.

Autostop logger



If this option is selected, the acquisition automatically stops after the defined number of measurements have been carried out or after the pre-set time has ended. Otherwise the data acquisition will continue until  is clicked.

Load device setup



Before switching to the visualization window, a device setup file can be loaded and transferred to the connected device. Click on  to open a file selection dialog.

Save settings on exit

If you wish to retrieve your settings the next time you start the catModule, you must activate this field.

Visualization window

Measurements can be displayed in different windows, i.e. Online Pages. All measurement windows (except the measurement value monitor) contain on the left-hand side a series of Indicator panels and on the right-hand side a graph.

If you wish to display measurement values over time, you *must* select one of the y(t) graphs. If you wish to display one channel vs. another one, for example force vs. displacement, select an y(x) graph.

The measurement monitor displays the values of up to 250 channels in Indicator panels.

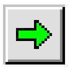
Do not forget to enter the name of the file in which the data should be saved, if you want to store data to file.

Choose the required graph in the “Visualization window” and click on “*Run*”; the actual measurement starts only in the measurement window.

2.2 Perform a measurement

On the second page of the catModule, you define which I/O channels should appear in the graph(s). Position the mouse pointer in the graph and right-click to call the “Data sources” menu item. Insert the required channels to one or more scaling layers (drag&drop or click on channel and “*Add data plot*”). In the default setting all scaling layers use “*Autoscale*”. Indicator panels are already connected with the first I/O channels.

Start the measurement



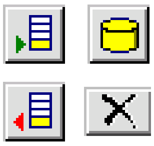
Click on or press F5; after the initialization the measurement starts.

If clicking on F5 does not produce the desired effect, you may need to click once on the Online Page after starting the catModule.

Post-processing

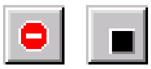
With the symbols in the toolbar you can assign data sources, zoom into curves, use cursor analysis, view or export data from the Database or print out the graph. All symbols have *tool tips*.

Storage during measurement



Use the icon or to save data in the Database during the measuring operation. will turn the continuous registration of *all* channels on or off, saves the current value for all channels. Use to delete the last value saved in the Database. The values are always appended to the end of the Database. deletes the entire Database.

Exit



Stop the measurement with , terminates the catModule.

3 Periodic measurement

☞ Access to this catModule may have been withdrawn via the user administration. To use the Event monitor in this catModule, you must call up the window before the start of the module.

This catModule records a specific number of samples for each active I/O channel and saves the measurement data to the Database. The start of acquisition can be *triggered*, provided the connected measuring device supports this facility. After each period data may be printed or exported to file. The module is especially well-suited to recording measurements with a high sample rate, e.g. 10 kHz. However, it can also be used in long-term monitoring for particular events.

On the first page of the catModule you have to enter the settings for the data acquisition.

3.1 Settings

Click on “*Sample rate*” to set the acquisition speed. Enter the number of measurements per second either in the column “Sample rate” or, if no internal time-base generation is available, use the “*PC-internal timing*”. Further information concerning sample rates can be found in Chapter E, *Measuring with catman® Professional*.

Measurement/Periods

Enter the number of the measurements to be collected per channel. The upper limit for this value is determined by the maximum length of the Database channel, see Chapter C, Section 2, *Configuring the Database*, page C-4.

Number of periods

Enter the number of required periods. This number determines how often a measurement is started.

Trigger measurement

Periodic measuring can be started by a trigger event. Click on “*Trigger*” and specify the event (or for MGCplus up to four events) in the dialog window. Activate the definition(s) in the list on the right-hand side of the dialog window. Instead of a trigger event, you can also use an I/O channel which is connected to a digital input of a device. In this case, mark the command “*Wait for Digital IN on channel...*” in frame “Before each period”. The channel number must be entered as catman I/O channel number, *not* the device hardware channel! For trigger settings, see also Chapter E, Section 4, *Triggered measurements*, page E-16.

Save to Database

In this case *all* active channels are saved to the Database in the course of a measurement run; definition via the “Target” column in the I/O Definition is not possible. However, there are three different ways of saving:

1. All active I/O channels are written 1:1 in the related Database channel. A new measurement will overwrite the data already in the Database from the previous measuring operations. In this mode, periods cannot be compared. Only the last measurement is in the Database. To prevent loss of data, you must export the data after each measurement.
2. All active I/O channels are written 1:1 in the related Database channel and new measurements are added on to the end of the Database channels. The depth of the channel must be sufficient enough, see Chapter C, Section 2, *Configuring the Database*, page C-4
3. All active I/O channels are saved as continuous channel block, starting from any channel in the Database. When this start position is increased before each measurement, several measurements can be saved simultaneously in the Database. The number is restricted by the available Database channels, see Chapter C, Section 2, *Configuring the Database*, page C-4. If you wish to keep more than one period, you must activate “*increase automatically*” for the start position.

You can export the data after each period additionally to saving into the Database. When saving the measurements in a file, we recommend to use format “*catman*” as this is the quickest method of saving. Files in this format can be reloaded at any time and exported in other formats. All *Traceability data* (channel information) is also included.

Delete Database

The Database is normally deleted when the catModule starts. If you deactivate this option, the new measurements can be compared with old ones which have previously been imported into the Database.

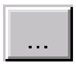
Initialize I/O channels before measurement

This option includes the following actions:

1. A check will be made which device, which interface and address and which hardware channel is used as input channel
2. The hardware connection is made and checked
3. The channel information (acquisition parameters), such as measurement range, unit, etc. will be read from the device
4. will be read from the device.

The option “*Initialize I/O channels before measurement*” *must* always be active when *first* running a catModule. Provided you make no changes to the channel setup (I/O Definition or device setup) or, if you are working with the filter setting “*Automatic from measuring rate*”, provided you do not change the measuring rate, you can again deactivate it for following measurements. Time can be saved in this way when starting the measurement, particularly when using MGCplus and UPM60/100. The initialization phase can take longer than usual, if catman has to collect acquisition parameters from many MGCplus device channels or UPM100/Centipede.

Load device setup



Before switching to the measurement window, a device setup file can be loaded and transferred into the connected device. Click on  to open a file selection dialog.

Save settings on exit

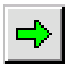
If you wish to retrieve your settings the next time you start the catModule, you must activate this field.

☞ Generally, this module is used with post-process graphics. You can however also use the visualization window “*y(x) Cursor graph + y(t) Real-time graph*” if you need a display *during* the measurement.

3.2 Perform a measurement

Configure the graphs

As autoscaling is carried out, you must just enter the channels to be displayed on the second page of the catModule: Call the context menu in the graph and select “Data sources”. Enter the channels for the x- and y-axes in the dialog window (first click on the x channel, then drag&drop or click on y channel and “*Add data plot*”).



Start the measurement

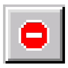
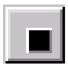
Click on  or press F5; after the initialization the measurement starts.

If clicking on F5 does not produce the desired effect, you may need to click once on the On-line Page after starting the catModule.

Post-processing

With the symbols in the toolbar you can assign data sources, zoom into curves, use cursor analysis, view or export data from the Database or print out the graph. All symbols have *tool tips*.

Exit

Stop the measurement with ,  terminates the catModule.

4 Long-term monitoring

☞ Access to this catModule may have been withdrawn via the user administration.

This catModule is used for long-term monitoring of processes. Unlike the Periodic measurement catModule, with this module only *one* value per measurement is ever measured from all active channels. With this catModule the first I/O channel must *always* be the time channel. Times are captured in absolute terms (Date and Time), the necessary settings are made by the module automatically.

For the long-term monitoring catModule to restart automatically when power is restored, some settings are made in the registry. Please note, that you must have the corresponding access privileges (Windows Registry). Note also that your system must start without the need of a password entry. Otherwise the computer would stall at the password entry prompt. It is recommended to set the option “*Auto-recover Database on program start*”, see Chapter C, Section 2, *Configuring the Database*, page C-4.

Function principle

The measured values and start time for each measurement are determined and stored in the Database. The start time of each measurement is captured in absolute terms (Date and Time) and stored suitably encoded in the first Database channel. Measured values stored in the Database are saved to a *backup file* at the intervals specified. After a power failure catman automatically restarts when the power is restored, and the data is loaded from this backup file first into the Database before the measurement series is resumed. You can also run an extra backup by key-press.

In the case of the Spider8 you must save the device setup to a file and specify this file, since the Spider8 loses its amplifier configuration in the event of a power failure. When the Long-term monitoring catModule restarts after a power failure, this configuration file is automatically uploaded to the device.

You may review the measured values captured in the Database at any time in the Database editor. You have a Spreadsheet and/or 2D Scientific graph available for visualization purposes.

4.1 Settings

I/O Definition and device setup have to be loaded resp. performed prior to starting the cat-Module. The active I/O Definition is saved by the module under a default name. This allows the original status to be restored in the event of a power failure.

The interval between two measurements is specified in the field “*Interval time*” (1 sec. to 24h). Instead of using a fixed interval for the measurement you may define an “*Interval table*” where you can specify different measuring intervals depending on the time already measured. This enables you to use e.g. short intervals at the beginning of a measurement, if the biggest changes occur at the beginning. You can begin or end Long-term monitoring manually, or input any start or stop measurement times you wish. For automatic data acquisition enable the “*Start*” resp. “*Stop measurement at specified time*” checkbox.

The start time of a measurement series must be later than the current time and earlier than the stop time.

Maximum time

The maximum possible measuring time depends on the size of the Database and the acquisition period. If there is not enough time, increase the size of the Database as described in Chapter I, Section 2.2, *Database configuration*, page I-6.

Timeout (s)

Time in seconds in which all measurements must be read into the computer. Otherwise, the measuring will be interrupted and an error message displayed.

Calibrate with every interval

Before each measurement, all active I/O channels will be calibrated (autocalibration of the device). The option will be ignored by systems which do not support autocalibration (for example Spider8).

Automatic backup after n intervals

The measurements from each period will be entered line by line in the Database. The value n determines how often the Database is to be saved in a backup file. The size of the factor n depends on your measuring configuration:

- For long measurement intervals, this value should be set to 1 so that the entire Database can be saved after each measurement.
- For short measurement intervals and many channels, you should select 2 or 10 so that time is not used for the backup after each measurement and the required measurement interval can be kept.

As the backup file is loaded after a power interruption, only the measurement data will be restored which was saved in the backup file. Data which was not yet saved in the Database will be lost.

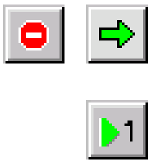
Maximum number of rows in table view

Gives the number of lines of the table which can be seen on the Online Page. As soon as the table is full, the contents will be automatically moved up after each measurement and the new measurements will be added to the end.

4.2 Perform a measurement

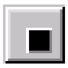
Configure the graphs

You must define which I/O channels should appear in the graphs on the catModule pages with graphical display. Position the mouse pointer in the graph and click the right mouse button to call the “Data sources” menu. Then add the required channels to one or more scaling layers (drag&drop or click on channel and “*Add data plot*”). As all graphs in this catModule use autoscaling, there is no need to set the scaling manually.



Start the measurement

You can start the Long-term monitoring with and stop it with . If you perform a time-controlled measurement, just the monitoring for the start time begins after clicking on . Starting and stopping is also possible with “Start measurement” or “Stop measurement” from the “Function” menu. performs a single value acquisition for all active channels (manual acquisition).



Exit

Clicking on terminates the catModule.

5 Single value data acquisition

☞ Access to this catModule may have been withdrawn via the user administration. To use the Event monitor in this catModule, you must call up the window before the start of the module.

At key-press this catModule records a single measured value from all active I/O channels and saves the values to the Database. After clicking on the “*Measure*” button ( ), one value per channel is entered in the table, saved to the Database and displayed in a y(x) Graph. The last recorded value can be canceled.

Settings for saving and exporting data in the “Target” column of the I/O Definition are not allowed as this catModule stores a value from *all active* channels to the Database on button or key click. If any markers are set, you must remove them.

You can configure the Indicator panel according to your wishes so that if an alarm level is exceeded, a color change and text message appear. This catModule is ideal for processes where the operator has to decide when a measurement should be acquired, e.g. measuring the calibration curve of a transducer characteristic.

The entire Database will be deleted when you start the module.

Application example

You want to carry out a measurement on several channels when certain values are reached in a monitor channel. For this, on the Indicator panel, you monitor arrival at the default values and then measure all active channels at key-press when the value is reached.

Settings

With this catModule the first I/O channel must *always* be a time channel. Times are captured in relativ terms (hours, minutes and seconds since measurement start), the necessary settings are made by the catModule automatically. I/O Definition and device setup have to be loaded resp. performed prior to starting the module.

Mode of operation

In the main window you will see the measured values in the Indicator panel. You can configure the display according to your requirements using the context menu. Once the required value is reached, click on or press F5. The current measured value of all monitor channels will be added to the Database channels and transferred to both the table and the graph. Only active channels appear in the table. Column 1 of the table contains the time when the measured values were transferred. If you want to delete the last recorded value, click on .

Use to review, process and export the data from your Database. This opens the Database editor, in which you can edit your data as usual. To return to the main page of the catModule, close the Database editor. To display or print out measurement data with a full page graphic, click on . Configure the graph on this page with the required Database channels using the context menu item “Data sources” or one of the symbols , or in the toolbar.

☞ The list displayed on the main page holds a maximum of 128 entries (or lines). As soon as the end is reached, the list is cleared and starts again at the first line. However, all data is stored in the Database.

6 Single value DAQ to Excel

Access to this catModule may have been withdrawn via the user administration.

This catModule only works with MGCplus and Spider8.

At key-press this catModule records the values of all active I/O channels of a MGCplus or a Spider8. The data is stored to the Database and exported to Microsoft Excel at the same time. Times are captured in relative terms (hours, minutes and second since measurement start) and transferred to the MS Excel workbook. The actual data is displayed in an Indicator panel. The values recorded last can be canceled from the Database and from MS Excel.

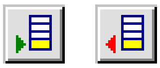
The entire Database will be deleted when you start the module.

The catModule is particularly suitable for controlling measurement by pressing an external key and writing the measurements to an MS Excel workbook.

6.1 Settings

First, an I/O Definition and a setup file may be loaded. Next enter the “*File name*” for the MS Excel file as well as the “*Start row*” and “*Start column*” in the MS Excel workbook. If necessary, the file is created. If no path name is entered, Excel will create the file in the default folder. If required, the channel name from the I/O Definition is exported to Excel. In this case, the channel name is written to the start line. The measurements then follow in the next lines.

The Excel workbook into which the measurements are exported may not be opened manually in Excel and, during the measuring operation, Excel may not be closed manually as catman opens and closes both the file and Excel automatically.



To write the present measurement of all channels in the Database and at the same time transfer to Excel click on . If you want to delete the last recorded value, click on .

6.2 Start with external contact

Data acquisition can also be triggered by an external contact.

MGCplus

Connect a push-button switch to pins 1 (Digital ground) and 8 (Ctrl 7) of female connector Bu2 (Output) of a single channel plug-in module with AP01 in the MGCplus. Then enter the *channel number (slot) in the MGCplus*, not the catman I/O channel number. The channel will be created automatically in the I/O Definition.

☞ The CP42 digital inputs cannot be used with this module.

Spider8

Connect a push-button switch to female connector Digital I/O, pins 2 (Ground) and 22 (Input/output 7) of the *first* Spider8 (seen from the PC side). Instead of pin 2 you can also use pins 7 or 9. The channel required will be created automatically in the I/O Definition. It is not necessary to enter a channel number; you can leave the field empty or simply enter 0.

7 2 Channel frequency analyzer

This module only works with the following devices: MGCplus, Spider8, DMC*plus* and DMC9012A.

The catModule provides for frequency analysis (spectral analysis) of one or two time signals. It is used to obtain a qualitative overview of frequency distribution in the signal.

Three different functions are available for frequency analysis:

1. Auto power spectrum

   This function computes the power spectrum of the time signal. This spectrum is stored in the Database channel designated MagSx. If a second signal channel is measured at the same time, its spectrum is stored in the Database channel designated MagSy. The unit for the spectrum is equal to the unit for the amplitude of the time signal; the figures are derived from the root of the mean square of the amplitude: hence the term RMS (Root Mean Square). This is the most frequently needed function.

2. Amplitude-phase spectrum

   This function computes the power spectrum and associated phase spectrum of the time signal. The amplitude spectrum is stored in the Database channel designated MagSx and the phase spectrum is stored in PhaseSx. If a second signal channel is also being measured, its amplitude spectrum is stored in the Database channel designated MagSy and its phase spectrum in PhaseSy. The unit for the amplitude spectrum is equal to that for the amplitude of the time signal; the figures are derived from the root of the mean square of the amplitude: hence the term RMS (Root Mean Square). The unit for the phase spectrum is the radian (rad).

3. Cross power spectrum

   The cross spectrum calculates the frequency consistency of two time signals, this analysis therefore always requires two measuring channels. The cross power spectrum consists of the product of the two spectra A and B of the time signals. The amplitude spectrum is stored in the Database channel designated MagSx and the phase spectrum is stored in PhaseSx. The unit for the amplitude spectrum is equal to that of the time signal; the figures are derived from the root of the mean square of the amplitude. The unit for the phase spectrum is the radian (rad).

If required, a digital filter can be applied to the time signal before spectra are computed. It is also possible to specify different window functions for the FFT and use a start trigger.

7.1 Settings

In the I/O Definition worksheet, only the channels one to three are taken, only the channels two and, if specified, three are evaluated. Delete all other channels, e.g. computations, from the I/O Definition, since the module needs Channels 4 to 11 for its own computations. These channels must therefore *not be assigned* and the Database must contain these channels. The channel depth should lie above the number selected in the "*Block size*" field so that all measurements can be stored.

☞ If you wish to use a digital filter, enter the respective limit frequency always in the field "*Lower cut-off frequency*" for "*High pass*" and "*Low pass*" filter. Do not forget to activate "*Filter time signal*".

Select the "*Block size*" in relation to the "*Sample rate*", so that for slow measurements one read block is received at least every five seconds, i.e. the "*Block size*" should not be larger than five times the "*Sample rate*". For faster measurements (high sample rate) the "*Block size*" should not be too small, otherwise the data, especially in "*Continuous sliding window*" operating mode cannot be read and calculated quickly enough. The "*Block size*" influences the *resolution of the frequency axis*, and the "*Sample rate*" the *maximum frequency*.

Please note:

- Settings will only be active for the next measurement. They are not effective for the current analysis.
- In the "*Continuous sliding window*" operating mode, the number of "*Points per window*" may not be larger than the "*Block size*". The setup determines how often (with how many values) a new computation is performed. This function is not activated in other operating modes.
- In the "*Continuous sliding window*" operating mode, the PC must be quick enough to be able to calculate and display during the measuring process. If necessary, change to another operating mode.

Avoid static signals (constant load on transducer) as a high amplitude develops for the frequency zero and the other frequencies contained in the signal may not be visible. As far as possible use either a digital high pass filter to suppress the constant share or tare the measurement channels.

7.2 Operating modes

Single shot

Once the “*Start*” button has been clicked the catModule records a specified number of measured values (“*Block size*”) and computes the chosen spectrum.

Repeated single shots

As single shot operating mode, but repeats measurements until “*Stop*” is clicked.

Continuous sliding window

In this operating mode a continuous monitoring process is started. After every n new measured values, where n is specified by the parameter “*Points per window*”, a new spectrum covering k values, where k is specified by the parameter “*Block size*”, is computed and displayed. In this operating mode online frequency analysis is thus possible; real-time performance depends heavily on the capability of your PC, however. With a 200 MHz Pentium Pro, at a “*Sample rate*” of “*1200 Hz*” approximately 10 windows per second (i. e. “*Points per window = 120*”) with a “*Block size*” of “*512*” can be analyzed and displayed.

8 DataView

☞ Access to Add-Ins may have been withdrawn via the user administration.

This Add-In, DataView, enables you to view measurement data in an overview, even of large files with over 100 megabytes, and to display any sections of the original data as a segment. The required original data is loaded from the program into the Database.

You can use the following files:

1. Data files recorded on the internal hard disk by the MGCplus.
2. Binary files created by catman from Version 4.5 onwards (formats “*catman*” and “*Binary for online import*”).

The measurement data must be present in two files for this:

1. A file with compressed values (this is specified by you).
2. A file of the same name with the original values (segments of this are imported as required by the program).

☒ In the left part in the *channel list* the window displays all channels of the recorded files with the compressed values. In the upper right-hand part the *Overview graph* is displayed and in the lower right-hand part the *Segment graph* for the channels marked (☒) on the left.

8.1 Producing compressed data

If the measurements are not present in compressed form (*.ST*) as with a measurement using the MGCplus, create a compressed file from an (uncompressed) catman file via “*File → Create compressed file*”. The compressed file receives the extension .REB for identification. Uncompressed MGCplus files (*.ME*) cannot be compressed.

☞ You may import uncompressed MGCplus files into catman and save in format “*catman*” from the catman Database.

8.2 Procedure

First of all you must select the data for display, then you define the desired segment. This can be changed or reset to view other segments of original data.

8.2.1 Selecting data for display

You can only use files which have either been produced by the MGCplus or binary files (formats "*catman*" and "*Binary for online import*") that have been produced by catman from Version 4.5 onwards.

1. Load a file with compressed data (*.REB or *.ST*) via "File → Add file to channel list" or with . Refer also to Section 8.1, *Producing compressed data*, page F-24.

   ☞ When measuring with a number of sample rates, the MGCplus creates a dedicated *.ST* file for each sample rate. In this case DataView loads all the files for the various sample rates. The *.ME* files contain the original values.

2. If the segment data is to remain saved later in certain channels, enter the first channel for the import: "File → Database offset". Otherwise DataView uses the first free channel in the Database.
3. In the toolbar mark the scaling layer in which the data is to be drawn: Y1, Y2 or Y3 and LY for left y-axis or RY for the right y-axis.

   ☞ When using more than one scaling layer, only horizontal zoom ( ) may be applied.

4. In the channel list click on the channels, the data of which you would like to see. Channels with data are identified by and marked channels by ☒.

All data is displayed in compressed form, both in the Overview graph as well as in the Segment graph, (initially). Repeat Steps 3 and 4 for channels, the data of which is to be displayed.

If you load files containing different amounts of measured values, the entries in the fields "*Number of measured values in current segment*" and "*Number of values by which the segment is moved*" refer to the channel which contains the most original values (reference channel). Only the number of the values is decisive, not the length of the data acquisition. When using files containing data acquired using different sample rates, check which channel is the reference channel. When using high measuring rates, many values are obtained but the curve may still be "shorter" than a curve measured with a low sampling rate. Even if in the Overview graph the two cursors are initially on the left and right *of all values displayed*, the figure shown in the "*Number of measured values in current segment*" may refer to the graphically shorter channel if this contains more measured values.

If required, you can change the default setting that data is drawn against time: Mark the required x channel and set via the context menu "Use as x dataset".

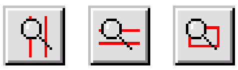
8.2.2 Displaying a segment

The field "*Number of measured values in current segment*" initially shows the number of original values present between the left and right cursors in the Overview graph.

You have a number of ways of displaying a segment:

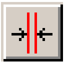
1. Move one of the cursors in the overview.

   Once original data has been loaded, you can then in the Overview graph only move both cursors together, i.e. the segment between the cursors.

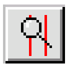


2. Activate one of the cursor functions , or by clicking it and then pull out a section in the Segment graph. You can also activate the functions via the menu "Zoom".

   You can also use this function a number of times. A further click on the selected cursor function in the toolbar deactivates the zoom function.



3. Enter the desired number of values in the segment into the field “*Maximum interval for original measurement data*” and click on or select the menu “Zoom → Zoom into maximum interval”.



☞ When using more than one scaling layer, only horizontal zoom ( ) may be applied.

Compressed or original data is displayed depending on the size of the selected segment. Once the number of values to be displayed drops below that in the field “*Maximum interval for original measurement data*”, original data is displayed. The field “*Number of measured values in current segment*” is highlighted in green. The second line in the heading in the Segment graph also indicates whether original or compressed data is displayed.

If the field “*Number of measured values in current segment*” is highlighted in yellow, then no compressed data is available. This is the case when you add a file with uncompressed original values to the channel list. In this case only the compressed data in the Segment graph is displayed, even for channels, for which compressed and uncompressed data would be available. So only the data from the files in the channel list is displayed and no further associated original data from the linked files is loaded. The status line indicates “Zoom in original values not possible”.

☞ The “*Maximum interval for original measurement data*” is determined by the size of the channels in the Database. The number of channels in the Database limits the maximum number of channels in the Segment graph. You can change the size of the Database via the menu “File → Database configuration”, but data already present is however deleted.

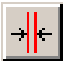
8.2.3 Changing the segment

Move displayed segment:

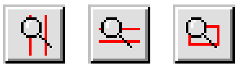
1. Move one of the cursors in the Overview graph. The other cursor is also moved if original data has already been displayed.

2. Select the number of data points by which the segment is to be moved in the field “*Number of values by which the segment is moved*” and click on or .

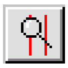
Enlarge or reduce displayed segment:



1. Change the number of values to be displayed in the field “*Maximum interval for original measurement data*” and click on or select the menu “Zoom → Zoom into maximum interval”.

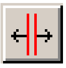


2. Activate one of the zoom functions , or by clicking it and then pull out a section in the Segment graph (only reduction of segment). You can also activate the functions via the menu “Zoom”.



☞ The zoom function must be deactivated by clicking on it again. When using more than one scaling layer, only horizontal zoom ( ) may be applied.

Resetting the segment (clear segment)



Click on or select the menu “Zoom → Reset zoom”. The cursors in the Overview graph are reset to the left and right edges and the Segment graph also displays compressed data.

8.2.4 Removing a file from the channel list

Mark the file symbol in the channel list and select either the menu “File → Remove file from channel list” or “Remove file” from the context menu. Any channels still being displayed are deleted from the graphs.

8.3 Adapting the display

The following options are available to adapt the display to your requirements:

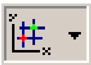
- You can change the proportion of the two sections Overview and Segment by dragging the gray dividing line
- Via the menu “Window” you can mask out the channel list and the Overview graph.
- The titles can be directly changed by clicking them and you can define further subtitles via the menu “Graph → Title → Segment” or “ → Overview”.

  ☞ The first subtitle of the Segment graph is set again to “compressed data” or “original data” after each change to the displayed data by the program, but it can be changed as required at least temporarily.

- You define the font size and type of font along with the foreground and background colors of the graph(s) via the menu “Graph”.

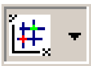
  ☞ The settings always apply to both graphs.

- The axes can be changed with a number of settings: Scaling, scale labeling, title, number and colors of the grid lines and dividing marks, etc. Double click on the relevant axis or use the menu “Axes”.
- You change the legend display via the menu “Graph → Legend”. Apart from the positions at the edge of the graph, you can also display the legend within the relevant scaling layer and select a display which requires less space via the menu point “Legend in single line”.
- The Segment graph can be labeled or you can draw in lines, e.g. for marking, in the graph using annotations. See also Chapter G, Section 3.2.3, *Annotations in Scientific graphs*, page G-32.



- You can display a cursor in the Segment graph with  . You must click on the required point on the curve to move the cursor to that point. The display of the curve values for the cursor position can be located to the right adjacent or within the graph in the relevant scaling layer: “Cursor → Cursor data display”.

You can also move the cursor with the arrow keys: Left/right moves the cursor, up/down changes to the next curve.



☞ The cursor remains active until you deactivate it again by clicking on  .

- If redrawing the graph takes too long when there are a lot of data points, you can speed it up by specifying a higher degree of optimization via the menu “Graph → Drawing speed”. However, you should use values above “*50*” with care.

G Online Document

1 Introduction

Authorization for editing Online Documents may have been withdrawn via the User administration.

The Online Document is the graphic interface of catman consisting of at least one Online Page. You can use the Online Document editor to set up each of these pages according to your requirements. For this you are provided with various graphic objects which you can position on the page and then change their appearance and size, as well as many other properties. The graphic objects you can use are displayed in a little window, the Design Toolbox.

It is important for catman that you have set up at least one printer and enabled it as the Windows standard printer. Otherwise, catman® Professional cannot work correctly because the page size for printing is determined from the printer settings. When starting up catman this will be pointed out to you in the form of an error message and catman will terminate.

The objects on the Online Page can be:

- Directly connected to the I/O channels, and will then automatically display data from the channels during measurement.
- Linked to Database channels, and will then show the contents of the Database.
- Linked to fields of an external database.
- Provided with data and controlled from a script (operating Level 3 in catman® Professional).

Also, there are many interactive objects, such as Buttons, Radio buttons, Checkboxes, etc. These allow to react to user entries. Reaction to user entries is script-driven, therefore you can use it in Level 3 only.

An Online Page is used as:

1. An interface for displaying data, whereby it does not matter whether the data is just measured in real time, has already been measured, mathematically processed or is artificially generated.
2. A script-driven interactive user interface with visualization.
3. An output medium for the creation of presentation graphs and protocols to a printer.

To gain access to the Online Document editor, choose “Online Document editor” from the “Worksheet” menu or “*Online Document*” from the Project window. catman opens a new Online Document and shows you a new, blank Online Page. Also, it shows you the Design Toolbox you can use to create new objects on the current page.

You may also add Online Documents to Projects automatically, see Chapter B, Section 4.1, *Save/Open Projects*, page B-14.

Show or hide the Design Toolbox with menu “Tools → Design Toolbox” or (toolbar).

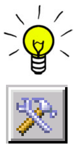
In the default setting you are shown only the *basic objects* you can use for creating a page. Advanced users can configure the Toolbox to their own requirements.

As an alternative to the Online Document you may also use the QuickView windows described in Chapter I, Section 4, *QuickView*, page I-39.

2 Creating Online Documents

2.1 The Design Toolbox

To add objects to a page, use the Design Toolbox. Just click the icon for the object you want and a new object of this type will appear in the upper left corner of the page. You can now move this object and change its size if you wish. You can also configure it using the “Configure object” context menu or by going to the “Edit” menu and choosing “Configure object”.



If the Design Toolbox is the active window (look at the color of the title and if necessary click once in the title bar), you will be shown a note about the object in a *tool tip* if you rest the mouse pointer on an icon. Show or hide the Design Toolbox with  from the toolbar.

You can resize the Design Toolbox simply by dragging with the mouse. The buttons inside the window are then rearranged to suit the new size and shape.

After catman has been installed the Design Toolbox shows the basic objects, i.e. those that are most commonly used. You can then use the context menu in the Design Toolbox, or menu “Tools → Options” and the “Toolbox and Toolbar” tab, to enable the following configurations:

1. “*Basic objects*”.
2. “*Extended*”: further useful output objects.
3. “*Developer*”: also shows input objects that can be used in Level 3 only. See Section 2.3, *Online Documents as user interface*, page G-12.
4. “*All*”: shows all objects, including those that are still supported for the sake of compatibility only. We recommend that you avoid using this setting if at all possible. It is preferable that any objects you require should be added from one of the other three settings only.

☞ The final entries in the “*Visible tools*” list are not displayed in the Design Toolbox but in the toolbar at the top of the Online Document editor window.

The Design Toolbox can also be personalized. Choose one of the default settings and enable additional objects you want to use or disable any objects you are unlikely to need. A note with an explanation about the various entries appears in the right part of the dialog window when you click on an object.

☞ The current settings are saved when you exit from catman. Each user is allowed to save a single personalized configuration, but not multiple configurations.

2.2 Online Documents for displaying measurements

You need to create an Online Page if you want to use the *Measurement Wizard* for your data acquisition, in other words, if you intend working at Level 2. Since the Measurement Wizard assumes control of the data acquisition, your Online Document only needs *output objects*. The Measurement Wizard will ignore any input objects that may be present.

So when creating an Online Document, all you need to do is pick out the objects that are best for your application, position them on your page(s) and adjust their size if necessary. Everything else can be done *after* starting the Measurement Wizard, that is, define the scaling and which channels you want to display. In the section that now follows you can find out which objects you may use for Level 2 and what are the various advantages or disadvantages of each.

If the display of all objects is activated, the range of Design Toolbox is divided into several categories:

- Real-time objects that display *one* value each.
- Graphical output objects for real-time *or* post-process data.
- The Spreadsheet.
- Input objects (Level 3 only).
- Layout objects for headings, borders, pictures etc.

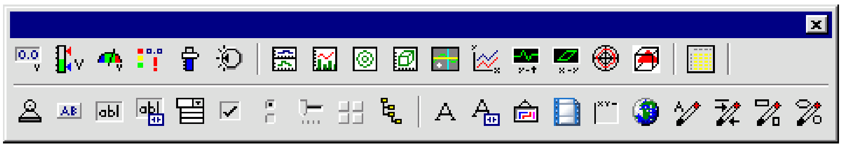
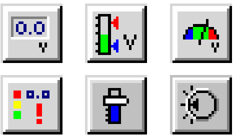


Figure 17: The Design Toolbox

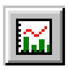
Since the fourth category (lower left part in Fig. 17) can only be used for Level 3, these objects are described in Section 2.3, *Online Documents as user interface*, page G-12.

2.2.1 Real-time objects for displaying one value each

The display objects in this category can display only real-time data, not data from the Database. If measured values are read in in the form of *read blocks* rather than individually, these objects display only the *first* value in each read block concerned. This is usually the case for measurements using the higher rates of data transfer. Since the inertia of the human eye means we can detect movements up to around 20 Hz only, and because a PC screen has a refresh rate (screen frequency) of only about 80 Hz, this is normally not a problem. However, if you manually force a *large block size* at *low sample rates* (dialog “Measurement settings”, frame “Read block”) the display becomes jerky, since it is likely that a new read block and its associated display value are only available a few times a second.



These objects are Digital indicator, Bar indicator, Analog meter, Indicator panel, Bar indicator and Analog meter. The last two objects (Analog meter or Knob control) and (Bar indicator or Slider control) are combi-objects, that is, they can be used for input or output. Of course, at Level 2 you can only use the output capabilities, and this condition is also the default setting for the objects: “*Mode: Indicator*” under the “Pointer” tab in the “Configure object” context menu. Both objects can be configured in many different ways and allow several values from a data series to be displayed: current value, minimum, maximum and average. On the other hand you cannot modify the bar color depending on the measured value, as you can with the Bar indicator object, nor can you vary the color of the background to the scale, as you can with the Analog meter object.



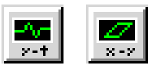
A special case is the Scientific-y(Index)-Graph which can display many I/O channels as *bars*. In the configuration as a “Real-time bargraph indicator” it works similar to the normal Bargraph but is not restricted to just 10 channels.

The Indicator panel can display up to 250 channels.

As only the first value of each read block is displayed, an alias effect may occur under unfavorable conditions, i.e. the Bargraph or the Analog meter oscillates at a frequency which is not found in the original signal. For information about alias effects, see Chapter E, Section 2, *Which sample rate is the right one?*, page E-4.

2.2.2 Graphical output objects for real-time data

The objects (Scientific y(index) graph), (2D Scientific y(x) graph), (Scientific polar graph) and (Cursor graph) can also be used for post-process displays. The preselection “*real-time*” or “*post-process*” must be made during object creation.



The objects (y(t) Real-time stripchart) and (y(x) Real-time graph) are suitable for real-time display *only*.

These objects should not be used for creating any new Online Pages, since they are only retained for the sake of compatibility. Use a y(t) Real-time cursor graph or a Scientific graph instead.

The Scientific-y(Index) Graph also displays many channels comprehensibly as “Real-time bargraph indicator” and is not restricted to 10 channels like the normal Bargraph.

Unlike post-process objects, real-time displays have a special feature in the form of an *internal buffer* for the measured values they have to display. This is necessary because otherwise only the current read block could be displayed in each case. As a result, graphical real-time displays are memory-hungry and this restricts the number of values that can be displayed. With all objects except the y(t) Real-time stripchart the buffer can be defined during configuration: “Real time” tab. The buffer size refers to the number of values that can be displayed per channel. The option “*Clear graph if buffer full*” has the effect of deleting the data *plotted* in the graph so that new values can be viewed. When measurement stops, all values held in the buffer are redrawn, and therefore none of the data in the *buffer* is lost when the *plot* is cleared. A y(t) Real-time graph has no buffer size setting, the buffer is limited to 16,000 values for each channel that is to be displayed.

If you browse to another page of an Online Document, all the temporary buffers for the original page are cleared. The buffers for the real-time graphs are also cleared, which means that if you browse back to the original page, the objects are empty. If however you also save values in the Database, the values can be used to fill the graphs: Tab “Real time”, “*Update on page change: Yes*”. Then you will always see the most recent values in the graph when browsing through pages.

In the case of post-process graphs, clearing the temporary buffer does not matter since the data stays in the Database in any case.

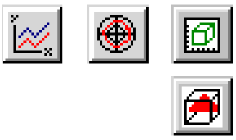
If you want to keep the data, you must:

- Save it to the Database during acquisition.
- Use the Auto Command List to make sure it is saved.
- Work with Level 3 and back up the data as appropriate.

During acquisition, just a *pixel shift* takes place in the simple graphs. Due to rounding errors arising whilst computing how many pixels correspond to the current time scale and the shift therefore required, the displayed time axis may get out of alignment with the actual time. The longer the display continues, the larger the effect of these rounding errors. For this reason, all plots are redrawn (from left to right) when a measurement ends so that the time line is displayed correctly.

2.2.3 Graphical output objects for post-process data

The objects (Scientific y(index) graph), (2D Scientific y(x) graph), (Scientific polar graph) and (Cursor graph) can also be used for post-process displays. The preselection “*real-time*” or “*post-process*” must be made during object creation. With preselection “*post-process*” these objects have *no* internal buffer which means they can *only* display post-process data.



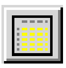
Due to the fact that they do not have internal buffers, the objects (y(x) Post-process graph), ((Post-process polar graph), (3D Scientific graph) and (OpenGL 3D graph) are also only suitable for displaying data from the Database.

All these objects are intended for displaying data after a periodic measurement or computation, or data that has been imported into the Database.

At operating Level 3 it is possible to display data simultaneously with a running acquisition (`Update` command). However, please note that every time the scaling is redrawn on graphs that are set up for automatic scaling, the time overhead is high. Additionally at every update *all* data captured so far is redrawn; therefore this method should only be used for the slower sample rates.

There are only exceptional circumstances in which it makes sense to display real-time data in post-process objects: when large read blocks are used with slow sample rates (manual setting), a read block is in existence for a relatively long time. This is long enough for a post-process object to display the data from the read block. At least 10 read blocks per second are acquired during normal acquisition, so that new data would be shown in post-process objects 10 times every second too. This would have the effect of causing the display to flicker.

2.2.4 The Spreadsheet



The Spreadsheet in the third category is a highly versatile object: it can display real-time data, post-process data, system data (Date/Time), external database data and in particular I/O channel *properties*. It is only worthwhile linking external databases at Level 3.

☞ Selecting “*I/O channel properties*” as the display value makes it possible to *log all amplifier settings*.

Just select a range of cells consisting of 25 fields (MGCplus) in one column, input “*I/O channel properties*” as “*Source*” under the “Data sources” tab and enter the required I/O channel below, then enable “*Create the following keys or fields in range*”.

The Layout toolbar gives you many options for configuration: menu “Configure object”, “General” tab. To help you make your choice, tool tips appear when you rest the mouse pointer over an icon. Define the cell settings either by enabling the “*Layout mode*” or by going to the “Tools → Options” menu, under the “General” tab, and enable “*Lock objects*”.

You may only delete a cell setup by using the Layout toolbar. If you choose “Delete”, from the context menu, it is always the object, in other words the *entire* Spreadsheet, that is deleted.

Not only does the object look like an Excel table, it also works like one. In its default setting its columns are labeled using letters, and its rows using numbers. However, you may use your own labelling system, and you may specify “fixed” columns or rows for additional headings. You may input formulas under the “Formula” tab, and they will be automatically recomputed when new data is written to the original cells (without script). You may of course also fill out a Spreadsheet with data and labels interactively. You can make cells editable as necessary, and then just type in text and figures.

☞ When inputting formulas interactively you can use to force recomputation.

If a whole column is dedicated to a data source, more than one value can be entered at the same time whenever there is an update, for example a whole read block or a number of data records from an external Database. The action taken by a Spreadsheet when the last row is reached can be configured: ignore further updates (“*Stop*”), automatically “*Expand*” Spreadsheet or automatically “*Scroll*” Spreadsheet. With automatic scrolling, the last n values are discarded to make room for the new data.

A Spreadsheet can be printed either as a separate object or as part of the whole page. When printed out as an individual object (context menu “Print object”) all visual effects, such as 3D, overlapping text etc., are retained exactly. Page breaks are automatically inserted so that only complete columns are shown on a page. However, the Spreadsheet is only printed out as far as the last row or column containing data. When printing out a whole Online Page, not all visual effects in the Spreadsheet are retained; in particular you should avoid overlapping text. Also just the part of the Spreadsheet that is *visible* on the page is printed.

A Spreadsheet can hold a maximum of 255 columns and 32,767 rows. Since a Spreadsheet requires a certain amount of memory overhead, it is not recommended to use many Spreadsheets with only a few cells each. Try using a single, larger Spreadsheet instead or use Digital indicators or Indicator panels.

2.3 Online Documents as user interface

In the event of test-rig applications, complex measurement runs or measurements that are intended to be executed by operators with no catman expertise, the Script development system can be used to create a user-friendly, PC-based control facility. To produce the user interface, that is to say the operating elements and output objects, you must also create Online Pages. First of all you need to consider the sequence of events, in other words the process as a whole. When you have completed the design stage, and not before, you can proceed with the implementation. At this point the first matter to deal with is designing the Online Documents that can be used to test the process. Eventually the Online Pages have to be closely interconnected with the script that will be created later.

Excursion: **Structure of a script**

First, therefore, we must touch upon two properties of the catman script language, one being the main structural features of a script and the second being the method of handling inputs and outputs. For further particulars on the script language please refer to Chapter K, *The Script Development System*.

A script consists basically of three parts:

1. The initialization.

2. A wait loop, for the purpose of waiting for events to occur.
3. For each event, a subroutine that is executed when that event occurs.

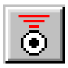
The range of possible events is handled through the menu “Tools → Event constants”, where you may define further events if necessary. Each event consists of a descriptive text, such as `START`, `STOP` or `TERMINATE_SCRIPT`, and a unique number; both together form an *event constant*. When designing an Online Page, one or even several of these event constants can be assigned to virtually any object, for example to detect when the object is clicked, modified etc. The number defined along with the text then receives catman’s `Event` system variable resp. the event queue. In a wait loop the script uses this variable to check whether one of the events has occurred. The text and number simultaneously assigned to an event are therefore used in the catman® Professional script language because although a programmer finds text much easier to read and understand, a program can check a number much faster than text.

Input/output

Another important point is the input and output system for data such as text or digits. Certain objects have general handling methods: real-time graphs or post-process graphs can also be “handled” jointly, for instance in order to display new data. To enable objects to be handled individually for a particular purpose, they must have a *name* which is kept simple: what would you expect to find in a field called *DataTransferRate*? So you need to give a unique *object name* to all objects that you want to handle for a particular purpose. Please do not get confused between the object name and the object title! The latter is just the visible label that you find in the header area of a graph, for example.

☞ Do not give objects on the Online Page exactly the same names as their variables in a script, so that you can tell them both apart more easily. For example you could give each name a prefix letter: o_DataTransferRate for the Online Page object, `g_DataTransferRate` for the global variable data transfer rate etc.

Using these notes you can already make a page ready for a script. It would be best if you first considered what input, output and control objects you need for your measurement task. Make a list, or better still sketch a flowchart with all the objects and necessary event constants, and run through the sequence once. Below you will find an overview of the available objects that allow interactive inputs.



You may use “Tools → Event constants” or the  icon at any time to view or add to the list of defined events.

The Document navigator you call up from menu “File → Document navigator” or with gives you direct access to certain of the context menus and an overview of all the objects in the Online Document.

Virtually all properties of Online Page objects can be modified using the script language: label, scaling, color etc.

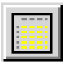
2.3.1 The objects from section 1 of the Design Toolbox: Knob control and Slider control

The two objects (Analog meter or Knob control) and (Bar indicator or Slider control) can be used as both an output (default) and an input. For the latter purpose you need to go to the “Configure object” menu, “Pointer” tab, and under “*Mode*” select “*User input*”. The advantage of these two objects lies in their comprehensive facilities for configuring the “Scale”: not only can you have linear or logarithmic scaling, but the “Scale” dialog also offers discrete subdivision in order to prevent “skewing” of input values due to restricted screen resolution, as well as assignment of value pairs. For example, if you create a slider with a size of 95 pixels on the page, then you can never set up every integer value from 1 to 100. Even with a size of 120 pixels you would always have fractional numbers as the place value. The option “*Discrete values*” prevents this, provided you do not want more values to be generated than there are pixels available. The creation of value pairs enables additional labeling of individual values: the specified label appears in place of the assigned value.

If a numerical value and the text of a value pair overlap, input extra blanks (spaces) in the text to change its position.

If you feel that the control is too plain, put it in a border, define another pointer with another “*Fill color*”, such as dark gray, which you do *not* make “*Visible*”, for which the “*Style*” is specified as “*Invisible*” and which has a “*Fill mode*” of “*Fill to Maximum*”.

2.3.2 The Spreadsheet



The Spreadsheet, already extensively discussed above in Section 2.2.4, *The Spreadsheet*, page G-10, can be used not only for output but also for input. In fact you can even have a *mixed* mode of operation: certain cells for input and others for output! There are many objects for you to choose from under the “Cell type” tab: Text box, Button, Checkbox, Drop-down list box etc. Certain cell formats also provide further options: masked text inputs, numerical inputs, password mode etc.

By using comparatively few script commands, such as

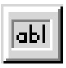
```
Input = Spreadsheet.Row[1].Col[2].Text,
```

a Spreadsheet can be read and modified. If necessary, individual cells can be locked to prevent unauthorized inputs.

☞ Since a Spreadsheet may contain more rows and columns than you can see, you can put several dialog boxes in a single Spreadsheet and if need be, use the properties `.TopRow` and `.LeftCol` to make them visible.

2.3.3 The objects from section 4 of the Design Toolbox

Text box



This object provides the most options relating to the creation of events. “*Got focus*” signifies the typical Windows behavior pattern in which several text boxes can be addressed in succession by using the tab key to move from one to the next. As soon as the text box has “*Got focus*” by this means or by clicking on it, the event constant is written to the `event` variable resp. into the event queue. The sequence used when TAB is pressed can be gathered from the Document navigator (“File” menu), but can be changed by a script with `ObjectName.TabIndex`.

☞ This object can also be linked with a field in an external database. For an explanation of the subject see below in this section as well as Section 5, *Options, tools and tips*, page G-44.

Drop-down list box

This object lets you choose from several specified options. The values concerned can be defined in the design phase or by a script.

Slider control

The object resembles the object Bar indicator or slider control, but is easier to maintain. Since the Slider control has a lower graphical overhead, it reacts faster in a script.

The range of values for the slider must lie between 0 and 32,767.

Checkbox

A standard Windows object used for making a selection.

Radio button

This object is used for making a choice in which all the alternatives are always visible, but only one is enabled at any one time. The active setting can also be highlighted by means of font attributes.

Button, On/off switch and LED

In principle both objects can take on the function of a button. The Button has the advantage that you can specify a function key as a *hotkey*, for instance F5. Disadvantages are that pictures on the Button are not scaled as they are in the case of On / off switch, and that only *one* picture can be specified for *both* status conditions. If object On / off switch and LED is used as LED, an I/O channel may be specified as data source. In this case you can monitor the channel for a certain level.

Toolbar

The toolbar makes it possible to create, say, a toolbar with different functions for the individual icons:

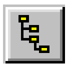
- “*On/off*”: clicking on this changes the status of the switch.

- “*On/off group*”: the switch belongs to a group enclosed by separators—only one button in the group can ever be pressed at any one time.
- “*Command*”: Push button function instead of a switch.
- “*Separator*”.

You may define up to 32 buttons per toolbar. Each button (other than separators) can hold both pictures and text.

A *tool tip* can be assigned to each button. This is a text of not more than 40 characters which appears automatically if you rest the mouse pointer for a moment on the button. Tool tips should always be used with buttons displaying only pictures in order to make the function of each one clear.

TreeView



The TreeView can be used to create a list display in the form of a hierarchical tree-like structure similar to Windows Explorer. The various entries can only be specified by a script, however. During the design stage the display consists only of a frame containing five nodes. Once an image has been assigned to a node, space for an image of the same size is reserved for all the other nodes. You should therefore assign images either to all nodes or to none.

Objects with external database link

Two objects exist in virtual duplicate form; the “duplicate” has additional features: the Label and the Text box have the ability to import or export display contents straight from or to a linked database. No duplicate exists for the Spreadsheet object, since in this case the external database link is separately defined for each cell.

A prerequisite for linking is that you have a database file or similar template that can be linked. The database template need not be the database file that will be used later in script run-time, but it must have the same layout, that is, the same tables and fields.

☞ For further information please refer to Section 5, *Options, tools and tips*, page G-44 and Chapter K, *The Script Development System*.

2.3.4 Miscellaneous

This includes the Keyboard events and the Menu designer; both are used in designing a user interface, but they are not created with the aid of the Design Toolbox.

Keyboard events

This triggers an event if the user presses a certain key. Each page can receive its own list of event constants, so it is possible to generate several events depending on different keys. They are defined with the menu sequence "Tools → Keyboard events" or the icon .

Menu designer

The Menu designer enables you to create a customized menu system that applies to *all* the pages of an Online Document. You may call it by using the menu sequence "Tools → Menu designer" or the icon .

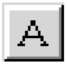
A picture can also be displayed instead of text: after selecting the appropriate cell, go to the "*Picture list*" and double-click on the picture you want.

If you want to display a *shortcut* in the usual Windows way, just insert the character & before the designated letter. If you enter "*&Measure*", the M will be underlined and in run-time you can select the entry by inputting "*M*". To display & input "*&&*".

The menu-designer creates only the graphical representation of the menu. To make it work you must assign an event constant to each entry and test for it in your event loop in the script.

2.4 Other objects

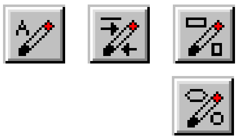
2.4.1 Label



The object Label allows the use of raised or sunken 3D frames. It is suitable for graphical design of titles, explanations, etc. Channel information and system texts such as date and time can also be displayed.

☞ The Label with external database link object can only be used at Level 3 (measurement with a script).

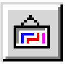
2.4.2 The Drawing tools



These objects are also used for the graphical layout of an Online Page. However, please note that only the Drawing tool: Text can be placed to the front of a standard graph (exception: Cursor graph). All other objects *always* stay to the back, and therefore cannot be used inside a graph nor inside the border object. Among other things, this Drawing tool: Text can be placed inside a standard graph with its lower left corner docked on the scaling of the graph; then on zooming, it moves so that it always occupies the same position within the data display. The following graphic objects support this feature:

- y(x) Post-process graph
- Post-process polar graph
- y(x) Real-time graph
- y(t) Real-time stripchart

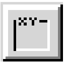
2.4.3 Background picture



This object allows pictures such as logos, drawings of the measurement layout, equipment diagrams, company logos etc. to be displayed on the page. To be able to use the object, you must firstly load the picture into the picture list. The individual pictures in the picture list (menu item “Tools → Picture list”) are also available to other objects, such as the standard y(x) graph, and may be used in an Online Document as many times as you wish.

☞ Save the picture list, in order to make all the separate files into one file containing all the pictures. You may then access this file at any later time and extract individual pictures for other projects.

2.4.4 Grouping frame



The object is used to place a border round a group of other objects and make them into one group (for example Label with Indicator panel). When the “*Show grid*” option is active, it is easiest if you create and position your required objects first, since the background color of the Grouping frame cannot be transparent. Then create the border last and simply move the objects into the area bounded by the border or lay the border over the objects and choose “to back” from the context menu. To remove an object from the group, drag it outside the border.

☞ To hide the Grouping frame, delete the “*Label*” and deactivate “*Show border*”.

2.4.5 Video replay

The Video replay allows video material to be displayed together with measurement data. If DirectShow is installed, all formats supported by DirectShow, including AVI and MPEG videos (standard without DirectShow), can be displayed.

The films are played from the command bar integrated in the object or via script command. Existing graphic objects on the page can also be synchronized with the film: the present position in the film is displayed in the graphs as cursor (vertical line). Synchronization can be used for up to 4 video objects with the following post-process graphic objects:

1. 2D Scientific graph, “Interaction” tab, section “*Cursor*”
2. Cursor graph, “Cursor” tab

The time information for the synchronization of the graph is calculated from the Δt of the channel. The synchronization can therefore only be done for equidistant curves, i.e. when the sample rate did not change during the measurement.

You can also enter a time delay between the start of the video and the start of the data acquisition in order to achieve a real time-synchronized display. The data must however be displayed in a post-process graph. It is not possible to “scroll” through the data.

2.4.6 HTML Browser

The HTML Browser object can display HTML-based files—not only locally, but also via URL connection (e.g. your company’s homepage) in your Online Document at run-time.

The Microsoft Internet Explorer must be installed to be able to use this object. On this technical basis (and its settings) the object displays information at run-time the same way as if you use Internet Explorer directly.

☞ During the design of an Online Page the HTML Browser cannot display HTML code. Instead you will see a gray rectangle, a world icon and the given URL as placeholders.

⚠ There can only be one HTML Browser object on an Online Page at a time. If you like to display two HTML sources, you have to create a second Online Page with a separate HTML Browser.

3 Configuring and editing objects

In the following, Section 3.1, *General*, and Section 3.2, *Objects for displaying measurements*, page G-27, explain the steps that are important for Levels 1 and 2. Section 3.3, *Objects for designing the user interface*, page G-35, gives further information if you want to design a user interface.

Please also read the Section 5, *Options, tools and tips*, page G-44, although the most important items are already briefly dealt with in the subsections that follow.

With menu “Configure object” the option “*Configurable at script run-time*” is available: When this option is active, objects can be configured at script run-time (or while the Measurement Wizard runs), otherwise this function will be locked! Thus you can avoid changes of your settings by users.

3.1 General

All newly created objects are displayed by default in the top left-hand corner of the visible part of the Online Page. To move an item, click in it and drag it to the preferred position. Make sure that the “*Lock objects*” option from the “Tools → Options” menu is not activated (icon , *not* in the toolbar).

You will notice blue squares on all edges of each object which allow you to increase its size in these directions. The squares in the corners allow you to increase the object in both directions.

If you have already defined data sources in a post-process graph, moving or scaling can take a considerable time if the data sources contain much data, since the whole graph is redrawn at every modification. To suppress redrawing of data, you may go to the “Tools → Options” menu, “General” tab, and choose “*Do not update objects*”.

☞ A manual update takes place when you click on (Update page).

To make it easier to position and dimension objects (rectangular, circular) you may use or go to the “Tools → Options” menu, “Grid” tab, enable the item “*Align objects to grid*”, choose “*Grid visible*”, select an appropriate color and set up the “*Grid width*” (distance between grid lines). *Twips* is a Windows unit of measurement: depending on the graphics driver, about 12 to 15 Twips are the equivalent of one pixel on the screen and 576 Twips represent one centimeter when printed out.

There are various options for *editing several objects*. Certain objects can be displayed in multiples: Bar indicator and Indicator panel may consist of not just one, but several elements. Otherwise all objects can be permanently grouped by using the Grouping frame. If there are several objects that you just want to move *temporarily*, simply drag a rectangle round the relevant objects with the mouse. All the objects enclosed in the rectangle can then be copied, moved, cut or deleted in one operation. To move the group, click on any part of the background *inside* the border, *not* on an object or the border itself. To pick up objects that lie behind the Grouping frame, click on the border and choose “to back” from the context menu.

You may use the *layout* function to save individual objects complete with its configuration as a template. The *template* function saves a whole page complete with its configured objects as a template. These templates are always available to be used again and can be pasted into other pages. You will find the command “*Save as layout*” in the dialog to the “Configure object” menu; for whole pages complete with objects you need to use the “Page” menu. If you do not like the standard configuration of the Scientific graph, change the layout and save it “*as default layout*”. This layout will be used when you next create a new Scientific graph. Reset this configuration to the original setting with “*Delete default layout*”.

☞ The Document navigator you call up from menu “File → Document navigator” or with gives you direct access to some of the context menus and an overview of all the objects in the Online Document.

3.1.1 Data source drag&drop

Using the “Data source drag&drop” window (or “Tools → Data source drag&drop” menu) you can very quickly assign data sources to the objects on your Online Page, for instance, link Database channels to graphs or real-time indicators to I/O channels. Open the “Data source drag&drop” window from the context menu.

Drag an I/O channel () onto an empty position on the Online Page to generate a Digital indicator of the channel. A 2D Scientific graph is generated if you drag&drop a Database channel ().

Sources

The window can offer either I/O channels or Database channels as sources. You can also choose between displaying only one channel list (for instance in the case of real-time indicators) or two lists (for example in the case of y(x) graphs). If two lists are displayed, you can only drag&drop channels from the lower list (y). Before doing this you must first select (i.e. highlight) the channel you will be using from the upper list.

You can also use the context menu to change the display.

Channel grouping

The lists show the I/O channels or the Database channels as groups. This provides a more convenient overview, especially if the number of channels is high. Click on a group to open or collapse the list of the channels in that group. You can also use the context menu or to choose the number of channels per group.

You can only drag&drop individual channels, not groups.

Defining data sources (creating curves)

- Digital indicator, Bar indicator, Analog meter, Bar indicator / Slider control, Analog meter / Knob control:

  For these you only need a single y list showing I/O channels. Drag a channel from this list to the object of your choice. If this object is an indicator with several display objects (Bar indicator, Indicator panel), drop the channel onto the sub-object required. The name of the I/O channel will be displayed in the object.

- 2D Scientific graph, Cursor graph, y(x) Graph:

  For these you will usually need two lists (x and y) showing channels. In the x list, mark (i.e. select) the source that will deliver the x values for the curve and then drag your chosen y channel to the graph.
- Spreadsheet for listing Database channels in the form of a table:

  Drag a y channel into a column in the Spreadsheet. The column is then automatically linked to this channel and shows its contents whenever it is updated. If the maximum number of rows in the Spreadsheet is not enough for the number of values in the Database channel, the Spreadsheet is automatically enlarged.
- Spreadsheet for displaying real-time values:

  Drag a y channel into a cell in the Spreadsheet. Then the cell shows the last value of each read block during a measurement.

☞ Even if a y channel contains no data yet, the configuration concerned will still be created. Later, when the channel contains data, this will be displayed when the graph is updated.

3.1.2 Automatic curve colors

Set the list of colors to use for new curves with menu “Tools → Options” and tab “Colors”. As default plot colors are chosen automatically when you define new curves. With every new plot the next color in the list is used. At the end of the color list the list is reused again starting with the first color.

If you want to disable “*Use automatic curve colors*”, you should set “*Color count*” to “*1*” and the default color as “*1. color*”.

⚠ If ever you are unable to see a plot during measurement, check that the plot and background are shown in different colors, in other words they are not black on black, for example. Click on your plot, go to the context menu and look at the “Plot attributes” tab, in the “Line” area, to see which color and line strength are being used for the display. With Scientific graphs check the “Plots” tab to see the “Plot attributes”. Other possible errors: missing or wrong data sources, channel deactivated, trigger activated but no trigger event at device (look at status line).

3.2 Objects for displaying measurements

The most critical settings are without doubt the definition of the data sources and, for real-time objects, the scaling of the axes. Both these steps are also dealt with in Chapter E, *Measuring with catman® Professional*. However, there are a number of other points that are worth knowing, especially since many of them help to make things easier to handle. These are introduced in the next section. At this point we are not going into detail about every menu item. It would not only break the bounds of an overview, it would be simply superfluous to go into great detail: that “*Autoscale*” arranges the minimum and maximum values of an axis, so that all data points are visible, for example, is a fairly obvious point. In the case of other functions you are certainly familiar with the menus if you already use Windows since, for example, the “Font” dialog box is a standard Windows dialog.

We will not cover here two settings that you find with just about every object: the definition of “*Event constants*” and “*Object names*”. These settings are only needed when using an Online Page with script (Level 3) and are discussed in Section 2.3, *Online Documents as user interface*, page G-12.

3.2.1 Alarm

Some objects give you the option to define color coding. Depending on the object, you can do this for the scale (Analog meter), the values display (Bar indicator or Indicator panel) or the color of a separate indicator LED (Digital indicator).

You may only make use of the “*Trigger event*” function at Level 3; at Levels 1 and 2 you receive a *visual* alarm only.

3.2.2 Scientific graphs

Scientific graphs provide very versatile options for displaying two-dimensional and three-dimensional data. 2D graphs provide more versatile design options than y(x) Graphs or Cursor graphs and are equally suitable for real-time or post-process displays. The y(index) graph is most suitable for displaying bar graphs and needs less memory resources, since only the y-values are forwarded. The x-axis contains the count index and can be provided with various labels. A special feature is this graph in the configuration as “Real-time bargraph indicator”: Here, the measured values of several channels can be displayed as bars in real-time. It is similar to the normal Bar indicator where only 10 channels can be displayed.

Among other things a Scientific graph provides you with the following options:

- Vertically arranged y-axes in a single graph
- Additional right axes in each y-axis system
- Autoscaling and automatic division ticks
- Toggling between linear and logarithmic scales according to the range of values
- The ability to display “non-values”, i. e. infinitely large values arising from division by zero, for instance during differentiation of force over displacement plots
- Flexible labeling options: Annotations
- Interactive zoom and panning for 3D and rotation

- Interactive movement of plots for comparison purposes
- Curve segment display
- Powerful output options: printer, file, as metafile, bitmap, JPEG
- Graph can be saved/loaded as a layout, if required complete with data
- Special plot types such as Spline, Surface, Points+Best fit line etc.
- Provides script programmers with the capability to display arrays

From catman version 5.0 onwards, the most important functions have been incorporated into the toolbar above the graph: print, copy/export data, color settings, display graph on screen size, activate/deactivate zoom, cursor mode and specify graph as Sync-Master for the other graphic objects

Scientific graphs are further configured via the context menu with “Configure object”. In addition, some of the other menu items take you straight to other tabs in the same dialog window: “Data sources” goes directly to “Plots”, while “Scaling” and “Configure axes” go straight to the “Axes” tab. The Annotations function, which is available only to these graphic objects, has a dialog window of its own. The “Real time” tab is mainly intended for configuring the necessary memory resources.

The main axis settings can also be carried out in the “Quick axis configuration” dialog window. In fact this dialog can only be accessed if “Interaction”: “*Axis hotspots*” is enabled. If so, just double-click on the axis you want to configure.

Depending on the type of graph, all or only some of the tabs will be available in dialog “Configure object”. Some settings apply to the whole graph, and cannot be used to define each plot or axis separately.

☞ Some settings are only useful if certain settings are also entered on other tabs. For example, specifying “*Data points shaded: 3D bars and surfaces*” on the “General” tab only makes sense if plot method “*Bars*” is selected on the “Plots” tab. Plot methods “*Line*” or “*Point*” would show no effect.

If you are not using one of the graphs in which the x-axis is already defined as the time axis, you have to specify a data source for both the x-axis and the y-axis. When you have chosen a channel or channels, remember to add them to the graphic object concerned: use the "*Add data plot*" button. Only the channels in the lower list with the scaling layers will be displayed. For post-process graphs that are set up to display data over time, specify the appropriate time channel from the Database. In most cases this is "*DB1*". In the case of graphs that display real-time data, you only need make further settings in the "Real time" tab if the time information is not acquired at regular intervals: specify the corresponding time channel under "*Source for time marks*".

☞ In order to be certain that the right time channel setting is used for real-time graphs, you should make sure when selecting from the popup menu that you only choose objects that are preconfigured for displaying over time (y(t)).

You will find all plots defined in the lower left part of the dialog window, in one of the six scaling layers. The six main scaling layers represent a vertically arranged y-axis system, each of which can also contain an axis on the right.

You can also configure plot attributes and delete a plot with the menu that appears when you click on the legend label.

☞ Before you can set up or change a property, you must select the corresponding curve by highlighting it.

If you want to modify the properties of several plots collectively, you may select one or more scaling layers.

If you want your y-axis systems to be displayed not only vertically but also side by side, use the option "*Overlapping axes*" on the "Axes" tab. For the axes concerned you have to set the "*Overlapping axes*" property for the individual axes at *different* numerical values in decreasing order, for instance "*3*", "*2*", "*1*" in the case of three axes. If you assign the same number to several plots, they are in separate plot areas (stacked). Plots with different numbers are overlapped (superimposed).

Speeding up the display

When using high sample rates to acquire data which you wish to display, the real-time graph display will be very slow and there could be a delay between measurement and display. To speed up the display, you can activate the *data compression* on the tab "Real time". Depending on the level of compression, from a certain number of values (as long as available) the smallest and largest value is determined for each read block. Only these min/max values are transferred into the graph. This will considerably speed up the display. However, if you zoom in the graph after the measurement, only compressed data will be displayed. The original values will be lost if they are not also saved in the Database.

A similar feature may be used for post processing: On tab “General”, section “Data points”, you can activate “*Post-process compression*”. This option is suitable for post-process graphs which should display many values (> 100,000). The minimum number of data points, which will be compressed, is entered in the field “*Start compressing at … points*” (default 10,000). The compression factor will be determined from the number of the original values and the minimum number of values for the compression. Then, this factor is used to calculate min/max pair values and the results will be displayed. However as soon as you zoom within an area, a *new* calculation will be made. If the zoom factor is high enough, the original values will be displayed, i.e. the compression operation is *completely transparent*.

☞ No *Segment display* can be used when applying this compression.

If you browse to another page in an Online Document, all the temporary buffers for the original page are cleared. The buffers for the real-time graphs are also cleared, which means that if you browse back to the original page, the objects are empty. If however you also save values in the Database, the values can be used to fill the graphs: Tab “Real time”, “*Update on page change: Yes*”. Then you will always see the most recent values in the graph when browsing through pages.

Displaying a curve segment

There will be no autoscaling of the x axis as long as the segment display is active. If you have activated the “*Post-process compression*” in a post-process graph, you cannot display segments.

Click on the curve legend in the Scientific graph in order to call the dialog where you can enter the curve segment to display:

1. Enter the start and end points for the x-axis
2. Click on “*Display segment*”

The text “Segment display from … to …” will be shown as a subtitle below the plot area in the graph.

☞ The segment display works always on *all* scaling layers. The times of the x points stated are always used as x values (the section is also time-related for y(x) graphs).

Click on “*Reset segment*” in the dialog to restore the original data. You can also close the dialog temporarily or use the zoom function additionally.

The segment display is much faster than zooming within the graph, particularly when having many measurement values.

Please note:

1. A *single* graph can only display channels that are measured with the *same* sample rate.
2. If you want to have a y-axis on the right, you must assign at least *one* plot to that axis. You can do this in the “Plots” tab by dragging a y data source to the “*Right y-axis*” of a layer system.
3. Plots are displayed against a top x-axis for all y-axis layers simultaneously. To create a top x-axis, go to the “Axes” tab and in the upper list under “x-axis settings” specify “*x-axis at top: Yes*”. If you wish the last n curves of all those defined to be plotted against the top x-axis, give the “*Plots against top x-axis*” property (“Axes” tab, upper table) the value “*n*”. It is not possible to allocate plots of your choice individually to the top axis.

3.2.3 Annotations in Scientific graphs

Annotations are drawing objects which may be placed inside a Scientific graph. They are very similar to the objects you may create in a standard graph using the Drawing tools. They are however much more versatile.

You can use annotations to:

- add symbols and text to your graph
- create curves consisting of a few data points only (by having a corresponding number of symbol annotations, via script only)
- create customized axes by means of line annotations

Annotations are created and edited via the “Annotations” command in the context menu. This calls the annotations dialog window in which you can define, edit, move and delete annotations. If “*Annotations hotspots*” are enabled, once you have created an annotation you can click or double-click on it to call its dialog and edit or delete it (depending on the setting under “*Hotspot activation*”, “Interaction” tab). You can make use of a configurable default to specify predetermined text on creating an annotation: you have the option to display the x, y or both coordinates as text. At the same time you can use either the format of the axis concerned (including prefixes and suffixes) or a freely configurable format of your own. These settings are entered in the annotations dialog window, “Options” tab.

3.2.4 Simple graph objects

The Graphics control panel makes it much easier to configure and edit real-time and post-process graphs. You can call it up via the “Tools” menu or the context menu. With your graph in full view you can quickly modify the axis label area, enter new plots, amend colors and so on. You also get support from the Graphics control panel in post-processing situations: the tabs for “Cursor” include options to zoom in on the graph, cut (i.e. extract) a section of a plot between the cursor intersection points, determine differences etc. This window gives you almost every setting that you get from the context menus, plus the analysis functions zoom mode and cursor mode. The analysis functions are explained in detail later in this section.

The Graphics control panel has to be called up again for each object.

If you want to cut out the segment between Cursor 1 and Cursor 2, you should first select one of the channels in the Database as a "*Results channel*", otherwise each of the two data series will be written to the *next* free channel.

☞ If ever you are unable to see a plot during or after measurement, check that the plot and background are shown in different colors, in other words they are not black on black, for example. Call up the Graphics control panel or go to the "Data sources" context menu and look at the "Plot attributes" tab to see which color and line strength are being used for the display.

The layer function gives you the ability to configure more than one scaling, so that plots with different sets of values can be displayed in a single graph. For example, if the values in a data series are in the range 0 to 1 and those in another are in the range 0 to 1,000, if the same scaling were used for both plots the first would be right on the x-axis at zero and would be impossible to see. For this reason, assign one plot to Layer 1 and the other to Layer 2. Most graphs can have more than two layers: Set up the number of layers you want in the "Configure object" window, "Scaling" tab, using the "*Add*" button. Once a graph has more than one layer, when you call the "Set up axes" context menu you are asked which layer you want to configure.

By using the "Set up axes" context menu or the "Set up axes" button in the "Configure object" window, you can modify the layout of the x- and y-axes within wide limits. This procedure lets you modify the position and spacing of the ticks, use the label for the axis for the first or last tick as well etc. The setting "*Position of the axes in the scale layer*" moves the x- or y-axis: the normal position for the axes in a graph is left or below. However, using this option you can position the x-axis complete with its label either in the middle or on some other y value. If the axis is in fact positioned outside the scale, it will not be visible.

The "Function plot" tab in the "Data sources" window lets you input an equation, such as x^2 (x^2), which will then be plotted. You may also enter just one value (a constant) and a horizontal tick then appears on the y-axis at the position for this value. This adjusts the number of points that have to be computed, in line with the function: the more values, the *smoother* the plot appears. For a straight line, two points are enough!

☞ The cursor function "*Snap to point*" is not available for function plots.

The “Plot attributes” tab in the “Data sources” window not only enables plots to be extensively displayed, but also provides a facility to identify multiple plots unambiguously when a graph is printed out in black and white. Set the “Plot type” to “*Connected points*” and specify that a marker will be drawn, say, every 20 measurement points (“*Points/Marker = 20*”). The precise number to specify depends on your graph and the quantity of measured values. For large quantities of measured values, the number should be increased so that the plot is not swamped with markers.

3.3 Objects for designing the user interface

This section first of all contains some notes on the Spreadsheet object and then some information about the objects in the fourth category of the Design Toolbox.

3.3.1 Spreadsheet

The Spreadsheet offers a wide array of configuration options, since each cell can be set up separately and used as an output and/or input field. The Layout toolbar is particularly useful at the design stage. It may even be advantageous to replace the context menu (right mouse button). This then gives you direct access to the column setup dialog box.

Individual cells can be “locked” to stop them being modified. It is even possible to use *hidden* cells; not every cell in a Spreadsheet has to be visible. When the scroll bars at the side are deactivated, you can use the script commands `.TopRow` and `.LeftCol` to scroll the *visible* area. Please note in this event that interactive modification of cell width and height should be disabled.

If the setting “*Allow text overflow*” on the “Edit” tab is enabled, the contents of a cell that cannot be accommodated in the active cell will be extended over the next cell if it is empty. This setting is valid for the whole Spreadsheet.

A Spreadsheet can operate in what is called “*Permanent text input mode*” (“Configure object” menu, “Edit” tab). In this situation you may begin editing an unlocked cell by clicking on it once. Moving on from the active cell to the next position (ENTER key or arrow key) immediately sets the next active cell to edit mode.

☞ A Spreadsheet can hold a maximum of 255 columns and 32,767 rows. Since a Spreadsheet requires a certain amount of memory overhead, it is not recommended to use many Tables with only a few cells each. Try using a single, larger Spreadsheet instead or use Digital indicators.

3.3.2 Text box

“*Numeric input checking*” is carried out during run-time only. If you want to use a default value, you may even enter it directly during the design stage and save the page along with the entry. When you load the page, the field contains the most recent text. You can make use of this behavior if you always want to get the last value entered as default: finally save the Online Document once more.

3.3.3 Drop-down list box

The values for the list can be defined in the design phase or by script.

The Drop-down list box object usually changes position again after you release the mouse button. The move does not happen if you click with the mouse in the top left-hand corner of the object and drag. It is also the only object for which you can *not* call the context menu by using your right mouse button. Use the window menu “Edit → Configure object”.

☞ You receive a new row by pressing CTRL and RETURN (or ENTER) at the same time.

The row height of the object adjusts according to the font size. You may modify the font size via the menu sequence “Edit → Font”.

If you do not want to have the first entry in the object displayed when the page appears, lock all objects first with the aid of the menu “Tools → Options”. Since the object is then unable to move, you can select the entry you require. But do remember to cancel the lock, or you will be unable to move any more objects.

3.3.4 Slider control

The scale on this object is not labeled. Put a border round the control and use a Label to add labels if you need them.

3.3.5 Radio button

If the font for the options is modified it applies to them all: choose “Font” from the context menu. Only the font for the heading can be separately selected in the “Configure object” dialog.

3.3.6 Button

The font color *cannot* be changed for the Button object. This is because the object conforms to the usual Windows behavior and therefore the color of the font must be either black (button active and available for use) or gray (button inactive and not available for use).

You may however put a picture with colored font on the button. If you specify a picture and a text, the text is written below the picture if the available area is large enough. Pictures are not scaled, only the On/off switch and LED object, when configured as “*Switch with Picture*”, scales the picture that will be displayed to fit the size of the object.

3.3.7 On/off switch and LED

On/off switch and LED can also be configured as an output. Various kinds of round and rectangular LEDs can be defined.

3.3.8 TreeView

During the design stage the TreeView shows only of a frame containing five nodes. By this method you can set the font style and color as well as the background color, and configure the object. By default, both text and images are displayed (“*Style*”). Once an image has been assigned to a node, space for an image of the same size is reserved for all the other nodes. You should therefore assign images either to all nodes or to none.

The image is always displayed before the text. In “*Edit mode*: *Automatic*” you can edit node names during run-time (click the text twice). The default is “*Manual*”. You can also show checkboxes in front of the nodes with “*Checkbox: Yes*”. The status can be queried and set through script commands.

☞ If there is insufficient room to display the nodes during script run-time, scroll bars are automatically added.

3.4 Other objects

For information on the Drawing Tools, see Section *The Drawing tools* on page G-19.

3.4.1 Video replay

Enter the video you wish to display in the dialog box “Configure object” in the field “*Video file*“. Click on the field to open the Windows file dialog box. If control via script is required, the control bar and icons can be hidden: “*Hide controls: Yes*“.

☞ Click on the replay button (▶) to display the video. If you wish to recall the configuration, click in the control bar or use the “Edit” menu in the worksheet.

The replay speed can be changed from the list field. After clicking on , a cursor (vertical line) will indicate the current status of the video during replay in all synchronized graphs on the page. The graphs are only synchronized at the moment when you click. Therefore, this function should only be used during a pause. Activate “*Autosync graphs*” for a continuous display, the cursor runs along during the replay time in all synchronizable graphs. The cursor is synchronized approx. 10 times per second, not with each frame of the video.

The time information for the synchronization of the graph is calculated from the Δt of the channel. The synchronization can therefore only be done for equidistant curves, i. e. when the sample rate does not change during the measurement.

☞ The cursor is displayed in a color which is complementary to the background color. However, there may be situations where the cursor is difficult to see. If this happens, go to the configuration menu and change the background color of the plot area.

If the video did not start at the same time as the acquisition of data, a “*Time offset*” can be entered. This is easiest if the starting time of the video is also registered, e. g. via a digital entry: the time at which this channel changes its level is the offset to be entered.

3.4.2 HTML Browser

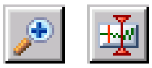
The URL is directly entered via the “Configure object” menu, an HTML file to be displayed can be chosen as well. The HTML information is then displayed within the reserved space on the Online Page. For navigation within the HTML pages the following script functions are available:

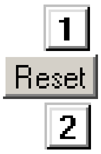
- Display of the previous page stored in the browser’s cache
- Display of the next page stored in the browser’s cache
- Display the Home page set in the Internet Explorer
- Display the Search page set in the Internet Explorer

4 Analyzing graphs

4.1 Zoom and Cursor

☞ Use the Graphics control panel for simple graphs, see Section 4.2.





Using zoom mode and cursor mode you can analyze and evaluate your measurement data in graphical form. Both these icons can be found in the “Online Document editor” worksheet toolbar and the Graphics control panel, “Zoom, Cursor” tab. To enable zoom mode, click on the icon. You can then use the mouse to select the area of a graph that you wish to zoom. After zooming, two icons are displayed at the upper left edge of the graph. If you use 1 you return to the original view, but if you use Reset you make the current view the new original view, i.e. the axis scaling is permanently modified. You can use 2 to look at the last excerpt again. To exit from zoom mode again, click on the icon once more or close the Graphics control panel window.

4.2 Graphics control panel

The Graphics control panel makes it much easier to configure and edit the standard real-time and post-process graphs. You can call it up via the “Tools” menu or the context menu. With your graph in full view you can quickly modify the axis label area, enter new plots, amend colors and so on. A great advantage is that you can immediately see each modification in the graph. You also get support from the Graphics control panel in post-processing situations: the tabs for “Cursor” include options to zoom in on the graph, cut (i.e. extract) a section of a plot between the cursor intersection points, determine differences etc. This window gives you almost every setting that you get from the context menus, plus the analysis functions zoom mode and cursor mode.

The Graphics control panel has to be called up again for each object. The Graphics control panel is *not available* for the Scientific graphs.

4.3 Synchronize graphic objects

There are two types of synchronization for Scientific graphs:

1. Synchronization with a video source (Object Video replay)
2. Synchronization with other graphs, Digital indicators and Spreadsheets

Synchronization with a video source

One of 4 objects of the type Video replay can be selected for the video synchronization of the cursor ("*Video-Sync*", "Interaction" tab). The numbers relate to the *sequence* in which the video objects were *created*. As soon as you start to replay the video, the cursor also moves within the graph in time with the video.

Synchronization with other graphs

You can specify a single Scientific graph using from the toolbar of the object or via the Interaction tab as Sync-Master. As soon as you activate and move the cursor within this graph, the cursor will also move in other Scientific graphs and Cursor graphs belonging to the same page if the cursor of the graph is activated. Digital indicators, which are connected to a Database channel, and Spreadsheet cells, which are connected to a Database channel, (define in both cases no Database channel row) also show the values at the cursor position (same point in time).

5 Options, tools and tips

In this section we show you various useful things that make it easier to create and edit objects. Though some of them have already been mentioned in other sections, we have summarized all the options again at this point and given a brief explanation for each one.

You can modify the background color of a whole Online Page in the “Page → Background color” menu item.

☞ The sizes of graphic objects, if not shown as a percentage or in degrees, are expressed in *Twips*, a Windows unit of measurement for screen displays. Depending on the graphics driver, about 12 to 15 Twips are the equivalent of one pixel on the screen and 576 Twips represent one centimeter when printing.

5.1 Page view, printing out pages

The default page size is determined by the Windows default printer when starting catman® Professional. The default orientation is Landscape (screen format). You may change this with the aid of the window menu “Tools → Options”, and then the “Page” tab.

To print individual objects, the current page or a whole document, just choose the appropriate menu items from the “File” menu. With certain objects you can use the context menu (right mouse button) to print the object out.

5.2 Export options for Scientific graph objects

The Scientific graph objects can be exported in three formats:

- “*Metafile*” format: This format serves for the direct exchange of diagrams with other applications (e.g. word processors). Metafile data can be output to the clipboard, a file or directly to a printer.
- “*Bitmap*” format: This format serves for the external processing (e.g. Image Processing, presentations). As output media you can choose between “*File*” and “*Clipboard*”. Hardcopy is not possible.
- “*JPEG*” format: This format is preferred, when small file size is relevant (e.g. for fast data transfer).

5.3 Templates

A useful function can be found in the “Page → Paste template” menu sequence. If you need certain objects over and over again, start by creating a page containing these objects. Save the page as a *template* and you can then add these objects at a later stage to other ones already on another page. Objects that are already on the page are not deleted by this action.

If you do not like the standard configuration of the Scientific graph, change the layout and save it “*as default layout*”. This layout will be used when you next create a new Scientific graph. Reset this configuration to the original setting with “*Delete default layout*”.

5.4 “Options” in the “Tools” menu

Under the first tab you can define whether:

- The Online Document will be automatically saved when terminating a script or the Measurement Wizard.
- You always have to *confirm* the deletion of an object.
- You want the Design Toolbox, the Scrollbar and the Toolbar below the menu to be *visible*.
- You want all objects to be *locked*. (This affects only the movement of objects and not their scaling.)
- You do not want objects to be updated. This is helpful if you have already defined the data sources in a graph and they amount to 50,000 values. If you then move or scale the graph, all 50,000 values must be redrawn every time. To suppress redrawing these values and save the time this process would take, you can select this option (the axes etc. of the graph will be redrawn in any case). A manual update takes place when you click on .

The second tab, “Grid”, contains settings for the *magnet raster*, on which objects can be aligned.

The third tab, “Page”, defines the *orientation* of a page, that is portrait or landscape, the *insertion point* for Copy/Paste operations, and the option to fit the Online Page to a particular *screen resolution*. This is very practical for designing pages that are intended to be displayed on smaller screens later, since you then get them displayed in the right size.

5.5 List of available pictures

You can call the “Picture list” with the “Tools” menu. This list must include all the pictures that will be used in the document. You can save on disk space by saving picture lists and retrieving individual pictures when needed. The picture list is saved together with the Online Document.

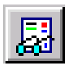
You may load files in the following formats: Bitmap (*.BMP), Windows Meta File (*.WMF), Icon (*.ICO), Graphics Interchange Format (*.GIF) and JPEG compressed images (*.JPG). Please note, however, that there are some objects which cannot accept all these types from the picture list: Bitmap and Icon formats can be placed on Buttons, On / off switches and Toolbars, though the Icon format is not scaled. Note that there are different variants of the GIF and JPEG formats, and it may not always be possible to read every variant. In this event use a conversion program.

5.6 Converting existing Online Documents

If you already have Online Documents that were created using older versions of catman, you can load these straight in. catman recognizes that older versions are involved and automatically carries out a conversion: List and Table are converted into the new Spreadsheet object and the settings remain unchanged. Similarly the former Groupable Switch object is converted into a On / off switch and LED object of the type “*Switch with picture*”.

☞ If you want to convert Online Documents created using catman 1.x, use the window menu sequence “File → Convert V1.x documents”. The dialog box lets you browse the directories and converts Online Documents from versions 1.x to 2.0; the update to the current version takes place when the Online Document is loaded.

5.7 List of all objects



Especially provided as an aid to designing Online Documents for use as a *user interface*, the Document navigator (menu item “File → Document navigator” or) has a list of all existing objects in an Online Document and provides direct access to the context menu.

Selecting an Online Page containing a specific object is done by a simple mouse click on this object in the Document navigator. Deleting several objects on the current page simultaneously is also possible. You can also change the name of an object directly in the window.

5.8 Working with external databases

As a rule this feature is only used at Level 3; however, it can be fully interactively controlled in the design phase, that is with regard to scrolling, reading or writing, with the aid of the “Database control panel”, which you call up via the “Tools” menu. This allows the external database to be used *without* script, data sets can be deleted or created etc. The Database control panel can only ever open *one* table. By changing the table, though, objects or cells can be linked to fields in *different* tables. The script that your Online Document later controls must similarly open several tables.

catman provides the objects Spreadsheet, Text box and Label for linking external databases. Using these objects you can display any field from an external database once the object or, in the case of a Spreadsheet, the cell is linked to a field in the database concerned. However, before you can choose a link you must do the following:

1. Select or create an external database.
2. Select a table from this external database.

The second point is important because most databases do not have just one table to which data is saved, but several interlinked tables. But catman needs to know which table is the one from which the contents of a field have to be read. You can definitely have fields from different tables displayed, that is to say, linked to objects on an Online Page. This being the case, a script must make sure that the necessary tables are open. There can be up to eight *record* *sets* from *one* file (i.e. external database) open at the same time, and up to *four* database files can be open.

If you want to create an external database, a Database designer is integrated into catman; you must already have a detailed knowledge of database techniques in order to make use of it. In particular you need to know how tables can be linked, the meaning of 1:1, 1:n or n:m relationships etc. For information on the subject read the documentation for a database program such as MS Access or other relevant literature.

If you already have a database file, you can link it as a *template*: menu “Tools → Select database template”. The database template need not be the database file that will be used later during script run-time, but it must have the same layout, that is, the same table structure and table names, and the selected tables must have the same field types and field names. These field names will then be selected in the dialog boxes of the objects on the Online Page. Whilst you are simply creating the Online Document, the selected external database *template* need not be filled with actual data; you may even use an empty file. The only important thing is the structure, that is the field type and field name.

The following types of Database are supported:

- MS Access up to and including the Office XP version
- DBASE III and IV
- ODBC databases such as MS SQL Server.
- EXCEL 7.0 (or higher) workbooks

5.9 Creating popup dialog code

This menu item from the “Page” menu is only applicable to Level 3. When the Script editor is open, this menu item enables you to let the objects on the current Online Page be made into script commands for a popup dialog box. This is considerably simpler than for example finding out the correct parameters for the positioning and size of a text box yourself in order to be able to create the command `DLG.Add 2 780 705 1600 315 “1200” <Variable>`.

Another possibility is to use “*Save as dialog file*” (in the dialog box). In this case, the current page is saved as a file which can be called from the script. The page can also contain objects which cannot be created with the command `DLG.Add`. The size of the window must be defined manually or by using the script command `DLG[].Move`. The present size or position is not read. Do not forget to specify the dialog buttons and if necessary the variables for return values.

More information about the object `DLG` can be found in the script Help.

H The Measurement Wizard

1 Introduction

☞ Your access rights to the Measurement Wizard or parts of it may have been withdrawn via the User administration.

The Measurement Wizard enables measurements using your own Online Documents and therefore avoids the restriction of the catModules with regard to the output pages. With the worksheets I/O Definition or Online Document editor open, you can call the Measurement Wizard and configure it via the menu “Measure” or the toolbar.

☞ The Measurement Wizard can also be started directly from a QuickView window. In this case only QuickView windows will be displayed and refreshed. Only if the Measurement Wizard is started from a menu or icon in one of the worksheets both Online Pages and QuickView windows will be displayed.

Operation is simple; there are only four operating modes which you can select via the drop-down list box:

1. Periodic measurement

   Measurement values are acquired for a certain time and then they are processed. Following this, actions such as “*Zero*”, “*Reset time base*”, etc. can be carried out before the next measurement period starts. The start of the measurement can occur in various ways, e.g. also by trigger monitoring.

2. Data logger

   The measurement values are acquired continuously. A maximum time duration or number of values to be acquired can be specified.

3. Manual acquisition

   This operating mode does not need any special settings. After the measurement is started, a continuous measurement runs in the background with all the real-time displays being updated. On a key-press the momentary measurement value is stored into the Database.

4. PC Card recording with MGCplus

   Starts a measurement in the MGCplus, where all data is stored to the internal PC Card memory. You can only use this measuring mode with a MGCplus equipped with CP42 and a PC Card memory (PCMCIA).

The measurement parameters are defined on tab, “Measurement settings”. The tab is identical to the dialog “Measurement settings” called up via menu “Measure”, please refer to Chapter E, *Measuring with catman® Professional*.

A file may be given on tab, “Online Document”, for the Online Document to use for visualization. If no file is available, one of the templates can be used instead.

If you have stated in the Measurement Wizard that the data is to be saved in the Database, then make sure that there is enough memory available. If the data depth is not sufficient, then catman stops the measurement with an error message. The data depth and the maximum number of channels are defined with the main menu “Options → Database configuration”, see Chapter I, Section 2.2, *Database configuration*, page I-6.

The start of the Measurement Wizard does not yet start the measurement. The measurement can only be started with .

The Measurement Wizard can only be ended by closing the window with . If you call one of the other worksheets before the Measurement Wizard has been terminated, not all functions can be accessed (grey icons and/or menus).

2 The Measurement Wizard status window

The status window is displayed when using a Periodic measurement as well as the Data logger and is therefore explained in a separate section.

The status window displays:

1. The number of measured values which are saved in the Database as well as the percentage amount of space occupied in the Database. The background color of the cell will change when the storage capacity is low.
2. Whether an Online data export should take place and if yes, to which file.
3. The measuring and transfer rate for the highest sample rate used (master sample rate). If the two fields do not correspond, the values will be saved in the output buffer of the measuring device. In addition, the display is no longer in real-time but delayed. The time delay is displayed below in the field. The fields have a red background as soon as the transfer rate falls below 75% of the master sample rate or if there is a time delay.

☞ Dialogs and windows which can be called from the toolbar must normally be closed before the measurement can be continued. As the measurement continues during this time, the values available in the measuring device buffer storage area will be read as quickly as possible once a dialog is closed. If the device buffer has overrun, an error message "*Buffer overrun in device*" will be displayed immediately after the dialog is closed and the measurement stopped.

Depending on the measuring device, the buffer storage area for outgoing data may have a different size:

- all single channel devices and the scanning devices do not have any output buffer
- the Spider8 has an output buffer for less than 1,000 values per channel
- The DMC*plus* has an output buffer for more than 100,000 values (all channels) depending on the version
- The MGCplus has a buffer storage area for more than 1,000,000 values (all channels) depending on the version of the CPxx. Details of the storage capacity can be found in the technical data chapter of the device Operating Manual.

⚠ As soon as the buffer storage area is full, catman stops the measurement. Note here that in particular with the Spider8, no delay occurs.

3 Periodic measurement

See also Section 2, *The Measurement Wizard status window*, page H-5.

If you have real-time objects on your Online Page, do not forget to activate the setting "*Update real-time objects during period*", otherwise you will not see any measurement values in these graphics.

Before the first start of a measurement, you must activate "*Initialize I/O channels prior to measurement*". Provided you make no changes to the channel setup (I/O Definition or device setup) or, if you are working with the filter setting "*Auto*"(matically from sample rate), provided you do not change the sample rate, you can again deactivate this for subsequent measurements.

The initialization phase may take a relatively long time with the MGCplus or UPM100/Centipede devices. In such cases carry out a trial measurement without saving and only then start the actual measurement, this time without initialization.

The periodic measurement offers the largest number of configuration options.

3.1 General settings

With "*Values per period*" you specify how many measurement values per I/O channel are to be acquired during a period. Alternatively, you can with "*Period duration*" specify how long is to be measured during each period. Under "*Number of periods*" you define over how many periods the measurement is to be made.

There are four methods (frame "Period start" and the "*Trigger*" button) of starting the acquisition (after clicking on):

1. "*Automatically*": Each period starts immediately once the preceding one has finished.

2. "*Manual using <RETURN>*": You must press RETURN to start the next period.
3. One of these two methods combined with trigger monitoring. To do this, a trigger event must be defined via the button "*Trigger*".

   Triggered measurements are only possible with the devices MGCplus, DMC*plus*, DMC9012A and Spider8. Of all the HBM devices only the MGCplus can currently operate with *more then one* trigger. Further information on the trigger settings can be found in Chapter E, Section 4, *Triggered measurements*, page E-16.
4. Start with the digital signal "*Digital IN on I/O channel x*": This enables you to monitor the signal on a digital input to your device. The measurement starts when the status TRUE (1) occurs; please refer to your device documentation to find out which signal produces this condition. For example with the MGCplus this may be different depending on the connection plate used.

Before each period

Before each period you can allow a certain number of seconds to expire ("*Wait before (s)*"), carry out a (auto) "*Calibration*" (of the device), "*Clear the peak value buffers*" of the devices connected or "*Zero*" all channels. Apart from Zeroing these options are evaluated only, if the hardware is capable of the corresponding action.

Performing a calibration takes time. With a scanning system, e.g. a UPM100/Centipede, this can take several seconds and the acquisition starts later as expected; check also for "*Read-block timeout*".

In the same frame you can specify to reset the time channel to zero ("*Reset time base*") or output a signal ("*DIO OUT on channel*") before each period. The digital output of the hardware channel defined for the specified I/O channel is then set to HIGH for 50 milliseconds before each period.

If you want to append data in a file, you should *not* reset the time channel.

After each period

After each period the currently displayed Online Page can be printed, the data saved in the Database can be exported ("*Export data every n periods*") and a signal can be output. The digital output of the hardware channel defined for the specified I/O channel is then set to HIGH for 50 milliseconds after each period.

3.2 Saving the data

Depending on the settings for saving and exporting data, you can override the settings in the “Target” column of the I/O Definition.

There are different methods of saving data.

1. You can save the measured values into the Database and export these values periodically: “Period storage in Database” and “after each period”, “*Export data*“.
2. You can save in a file during the measurement: “Online data export“. The advantage of this method is that the I/O Definition can be used in order to save only certain channels (column “Target“).

Although you may use both options together there is no benefit, it just takes more time. Files which are created *during* the measurement are not valid as long as Windows is writing the file, i.e. they are lost if you have a system breakdown or power failure. On the other hand the catman-internal Database is safe. This risk is limited to those values still in the RAM buffer (max. 4000).

Periodic storage into the Database has several options:

1. In the frame “Period storage in Database” you can specify that the data is written as normal in the Database: Option “*I/O channel = Database channel*”. Each period then overwrites the values of the last period.
2. You can specify that the data is saved *block by block*: Option “*Block storage starting at channel x*”. Here, all active channels are saved after the measurement in consecutive Database channels, starting with the specified channel. Each period is written to a new block

which in turn uses new, higher channels. You determine how many blocks are used with "*Keep the last n periods*". The oldest periods are then each overwritten.

☞ There must be sufficient Database channels present otherwise catman stops the measurement with an error message. If you would like to save 6 channels plus one time channel and 15 periods, you should have at least 105 channels available in the Database.

3. You can specify that all periods should be appended with "*Append periods in Database channels*". Here, the values are saved consecutively, so pay attention to the maximum data depth. If you are acquiring many values per period, the previous method is recommended.

   ☞ The data depth must be large enough so that all periods are accommodated. If you are acquiring 30 periods each with 25,000 measurement values, the depth of the active channels must be at least 750,000 values.

If you have selected the option "*Block storage starting at channel x*", you should wait with saving the data in a file until the number of periods defined in the field "*Keep the last n periods*" has been measured and only then export the Database: "*Export data every n periods*". Otherwise multiple saving of channels occurs. When using this option, saving in separate files is also recommended.

The button "*Export options*" opens a dialog in which you can enter the file name, file format, etc. for the file to be used. The differences between the formats are explained in Chapter I, Section 2.5, *Importing and exporting data*, page I-15. With some formats you can also specify whether the measured values are appended to existing measurement values in the file each time they are saved or whether each period is to be saved in its own file. The name of the file then consists of the file name and two numbers, the start number and the storage number. With each start of the measurement the start number is incremented by one as is the storage number each time saving takes place.

☞ The periodic storage occurs after a period resp. after the number of periods specified in "*Export data every n periods*".

When using “Online data export” the following applies: if catman is terminated or a settings file for the Measurement Wizard is loaded, the counting is reset and any existing files are then overwritten without warning. When using periodic storage catman checks whether the file with the next counter number exists and asks whether the file shall be replaced.

☞ Not all formats are available for the “Online data export”.

4 Data logger

See also Section 2, *The Measurement Wizard status window*, page H-5.

Before the first start of a measurement, you must activate "*Initialize I/O channels prior to measurement*". Provided you make no changes to the channel setup (I/O Definition or device setup) or, if you are working with the filter setting "*Auto*"(matically from sample rate), provided you do not change the sample rate, you can again deactivate this for subsequent measurements.

Measurement duration

With "*Auto-stop after n values*" you specify how many measurement values per I/O channel are to be acquired. Alternatively, you can also specify with "*Auto-stop after x s (min or h)*" for how long measurements are to be taken.

Saving the data

Depending on the settings for saving and exporting data, you can override the settings in the "Target" column of the I/O Definition.

There are various methods of saving, both for saving in the Database and for exporting, i.e. saving in a file. You can:

1. "*Save*" or "*export all channels*".
2. "*Use I/O channel target setting*": Only those I/O channels which have set the marker are then saved or exported.
3. "*Do not save*" or "*export*".

Only data saved in the Database is available for further processing after the termination of the measurement. The visualization of data in real-time graphics is not affected by this setting.

If the measured values are saved in the Database, then they are overwritten at each renewed start of an acquisition with . If you activate "*Append new data in Database channels*", the new measured values are appended to the values already existing in the Database.

The channel depth must be large enough so that all values are accommodated. If the data depth is not sufficient, then catman stops the measurement with an error message as soon as the Database is “full”.

The button “*Export options*” opens a dialog in which you can enter the file name, file format, etc. for the file to be used. The differences between the formats are explained in Chapter I, Section 2.5, *Importing and exporting data*, page I-15. You can also specify whether the measurement values are to be saved in one file or whether a dedicated file is to be created per channel. The name of the file then consists of either the file name and a number (“Base name” plus counting suffix) or the I/O channel name.

Not all formats are available for the “Online data export”.

Export in the format “*Binary for online import*” is the fastest.

You can also define whether you wish to be asked for a file name each time before a new measurement starts or whether catman should automatically append a counter: “*Increment file name with every new measurement*”. In this way, you can protect files from being overwritten especially if you have loaded the Measurement Wizard settings from a file.

You must enter the *base file name* in the “*Export options*”. The option cannot be used at the same time as the saving mode “*Single file per channel*”.

If you wish to enter a comment for each measurement, activate “*Prompt for file name and comment*” before new measurement. If entered, the dialog will then contain automatically generated file names (if specified) and allows the entry of a comment for each file.

Manual storage during measurement

Use icons or to save measurement data in the Database. switches continuous recording of *all* channels on or off, saves the current value for all channels. Use to delete the last value saved from the Database. The values are always appended to the end of the Database.

5 Manual acquisition

Settings for saving and exporting data in the “Target” columns of the I/O Definition are not allowed as this operating mode stores a value from *all active* channels to the Database on button or key click. If any markers are set, you must remove them.

No further settings are necessary for this operating mode. The measurement starts after clicking . Click or press F5 to enter a single measurement value for all active channels in the Database.

If clicking on F5 does not produce the desired effect, you may need to click once on the Online Page after starting the measurement.

The value recorded last can be cleared from the Database with .

The operating mode is suitable for processes where the operator has to decide when measured values have to be stored, e.g. measuring the calibration curve of a transducer characteristic.

6 PC Card recording with MGCplus

You can only use this measuring mode with a MGCplus equipped with CP42 and a PC Card memory (PCMCIA).

With all active MGCplus in the I/O Definition, the measurements will be recorded on the hard disk and transferred to the PC for monitoring purposes. If Trigger conditions are set, these will also be evaluated. In this case, the start also depends on the trigger conditions defined.

Enter the "*File name*" where the data will be recorded on the MGCplus PC Card memory, the "*Sample count*" (number of measurements) to be recorded as well as—for measurements with start triggers—the number of "*Repetitions*". If you also wish to create a compressed file, enter the compression factor. During the measurement, approx. 10 to 40 measurements per second will be transferred to catman. The actual transfer rate depends on different factors, for instance, how busy the CPU in the MGCplus is. The data can be used for simultaneous monitor display. However *not all* the data collected from the MGCplus is transferred to the PC. *All* data will be saved only on the internal PC Card memory of the MGCplus.

Via "Start mode" you can specify whether the measuring process should be initiated by catman ("*Immediate start*") or whether it should be done via the AB22/32 ("*Prepare for stand-alone operation*"). Here, you must specify which recording parameter set should be used. You can enter this on the AB22/32, see also Operating Manual MGCplus Assistant. If the measurement is initiated by catman, unless trigger conditions have been set, the measuring operation will only start when is clicked.

Please note:

- After starting the measurement, you *may not stop catman*, otherwise the recording in the MGCplus is also stopped.
- As the measurement of the values which are transferred to catman is not carried out equidistant, graphs may not use "*Master clock*" as the "*Source for time marks*". Enter the time channel created by catman (device as source). The irregular time values will be taken into consideration here.
- The recording is carried out with the formats entered on the tab "Measurement settings", "*Advanced view*", i.e. the format "*4-byte float*" is not supported by the Measurement Wizard.

- If more than one sample rate is applied, the MGCplus creates a separate file for each sample rate (*.MEA, *.ME1, *.ME2 as well as *.STO, *.ST1 and *.ST2 for the compressed files).
- If you create compressed files additionally, the original and compressed data can be displayed and evaluated using the Add-In DataView.

You can transfer recorded files to the PC also from the Database worksheet with menu “File → Transfer MGCplus data file”.

7 The methods of data visualization

There are various methods of displaying the data available you can choose from on the “Online Document” tab in the Measurement Wizard.

1. Visualization with the current Online Document

   In this case you should have created a suitable visualization object (e.g. y(t) Real-time graph) on the currently open Online Page.

2. Visualization with an Online Document from a file

   You can state a suitable Online Document file for the display of your measurement data. In this case the specified Online Document is automatically opened when the Measurement Wizard is started.

3. Visualization with an Online Document template

   You can use one of the templates from the list for the display of your measurement data. This is automatically opened at the start of the Measurement Wizard and only needs to be configured.

Use “*Save layout*” from the “Configure” dialogs to store complicated graphic configurations to file. Layouts can be loaded into a graphic object of the same type and will restore all settings previously done.

☞ Other possibilities for visualization provide the QuickView windows described in Chapter I, Section 4, *QuickView*, page I-39.

I Database and Post-processing

1 Introduction

Your access rights to the Database or certain parts of it may have been withdrawn via the user administration. Additionally, write protection may exist for individual channels so that no values can be *changed*.

Usually, measurement is followed by analyzing the data, performing computations and deciding how much of the data must be kept for later. In some cases, evaluation can even be followed by the first documentation of results. All these activities assume that measured values are available in the Database, in other words, that they have either been saved there during measurement or, if the data was initially only saved as a file, that they have been reimported into the Database.

In this Chapter you learn:

- How the Database works, which options are available and how to import and export data.
- Which computations are available for post-process mathematics and how to apply the computations.
- How data can be visualized even without an Online Document.

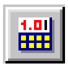
If you would prefer not to generate computations and graphs in catman® Professional, but would rather use Excel, just click on the Excel button to copy all the measured values or all the marked channels directly to Excel.

2 The Database

Your access rights to the Database or certain parts of it may have been withdrawn via the user administration. Additionally, write protection may exist for individual channels so that no values can be *changed*.

The Database worksheet is used for the management of data and for manual processing of individual values through to complete channels. The worksheet provides two different interfaces for the user:

1. the Database channel manager, see Section 2.3, *The Database channel manager*, page I-10
2. the Database editor, see Section 2.4, *The Database editor*, page I-13



When you call up the worksheet, you first open the Database channel manager and from here you access the Database editor via the menu “Worksheet” or with .

Database options

In the “Options” menu you can define the letter font and background color of the Database. Activate “*Confirm critical actions*” if confirmation is required for actions, which incur data loss and whether they are really to be carried out. The deletion of the complete Database must always be confirmed.

Channel information (acquisition parameters of a channel), traceability

This function is only supported for the devices Spider8, MGCplus and UPM100/Centipede.

Traceability of measurement data is an essential requirement for the increasingly important task of quality management. It is defined as the linking of the measurement data to information about the measurement chain setup. Different objects of the Online Page, e.g. Spreadsheet, Label and Drawing tool: Text, have been able to be linked to this information from the I/O channels (system texts) since version 2.2 in order to show the current settings in a log. The present version enables the permanent linking of measurement parameters with measurement values via the Database. The channel information is stored at the start of the channel data into the Database and can even be exported with some export formats.

First, mark the channel, the acquisition parameters of which you would like to view and then call the window “Channel information” via the menu “Edit” in the Database channel manager or the Database editor or use .

Flexible coupling between I/O channels and Database channels

The names and units for two mutually assigned I/O and Database channels can be specified and issued independent of one another. With the assignment of an I/O channel to a Database channel during the channel initialization, the Database channel assumes the name and unit of the I/O channel. However, this can be changed later.

2.1 Data integrity, recovering the Database

One of catman’s important tasks is ensuring data integrity. After a crash of the application or operating system the restoration of the Database therefore has the highest priority. This is indispensable particularly with applications having a measurement process that cannot be repeated, because the test object is damaged. Consequently, catman takes a few precautions to enable the automatic reconstruction of the Database which is managed by catman.

Dynamic Databases (“*Auto-size*”) are managed through the operating system, therefore data is not secure if there is a system crash. This is only the case if catman is managing the Database itself (fixed size).

FAT integrity

Directly after creating the Database file, the file is closed and then immediately opened again. Since the size of the Database file does not then change, the FAT (File Allocation Table) is always up to date, i.e. the operating system always recognizes the file as a valid file.

Saving the channel information (acquisition parameters) in the Database file

In the first part of the Database file all acquisition parameters of a channel (Channel information) is saved and it is only after this that the data section starts. This enables the immediate restoration of the current management structure from the Database file.

Registry entry

Once the Database has been successfully created, the full path name of the Database file is written into the registry.

Therefore, when an application or the operating system crashes, you only lose as a maximum the last values which have either not been written into the file or for which the management information has not been updated.

⚠ If you are acquiring fewer than 4,096 measurements per channel, they are temporarily saved only in RAM to keep the loading on the hard disk low and to achieve higher speeds for reading and writing data. However, this can lead to this number of values per channel being lost as a maximum during a crash.

☞ With the script command `DB[].Flush` you can force the buffer memory to be written to the hard disk from time to time (use a timer).

2.2 Database configuration

☞ After the installation, initially only up to 1,000 channels may be created. As an area of memory is required to manage all possible channels, this setting ensures that unnecessary storage space is not used. You can however increase this limit to 10,000 channels with the main menu “Options → Startup options“. See also Chapter C, Section 4.6, *Maximum number of Database channels*, page C-16 and Section 2.2, *Define the Database size*, page C-5.

You can define the Database size and the way catman handles the Database file on termination and starting using the worksheet menu “Options → Configure database”. You obtain the same dialog also via the menu “Options” in the catman main window.

All values in the Database are deleted during a new configuration. If you have already acquired measurement data, first export these values in the format “*catman*”, then change the size of the Database and import the values again.

The size of the Database limits the number of channels which are available for collecting measurements and/or for computations as well as the number of values which can be saved for each channel. This affects the graphic displays and mathematical functions: only data which has been saved in the Database can be used in a post-process computation or displayed in a post-process graph.

If, during a measurement operation, there is not enough space available to save all measurements in the Database, the measurement will be stopped and an error message will be displayed.

If you do not know exactly how many measurement values will be acquired or if the number of measurement values is very high (> 1,000,000 values), you can either write the measured values directly to a file or—if you use less than 512 channels—use a database with automatic size adjustment.

Database file location

The Database is created as a file on the hard disk (default: catman directory). In the section to the left of the Database configuration dialog you can choose the drive and a directory in which the Database is to be created. If several hard disks are available, you can alternatively assign a faster or bigger drive. The drives A: or B: are not allowed.

Do not use a network drive, as access to the data will be very slow.

In order to change the directory, double click on the directory name. It is not enough to just mark the directory.

☞ The file name does not need to be stated. The Database is always created with the file name DB_000.$_$.

Database size, channel layout and Database type

The Database layout is defined in the right-hand section of the dialog: Number of channels, type, maximum depth, format and byte width for each channel. Each channel can be edited individually: Mark the channel(s), enter the desired values and click on "*Apply to selected channels*". In the left section you can see, in the numerical display and in the pie chart, the disk space available on the hard disk as well as the free space and the space needed by the currently set configuration.

With "*Maximum number of channels*" you define the required number of input, output and computation channels. The maximum number of channels which you can enter here is 10,000.

☞ After the installation only 1,000 channels may be created. As memory is required for the administration of all possible channels, this setting does not take up unnecessary space. You can however increase this limit to a maximum of 10,000 channels from main menu "Options → Startup options". This presetting for the maximum number of channels is stored in the registry and is user-independent. Administrative rights are required to change the setting. Restart catman to activate the changed setting.

Close the entry with RETURN in order to activate the setting immediately.

"*Auto-size*": With less than 512 channels you can use a dynamic Database as in catman® Easy, which increases in size as long as there is space available on the hard disk. However, a little more time is required when saving data in such a Database, therefore this type of Database is not recommended if you are using high sample rates.

As this type of Database is managed by the operating system, data is not secure if there is a system crash. This is only the case if catman is managing the Database itself.

Individual channel formats

For standard applications it is not necessary to change the data storage “*Format*” or the “*Number of bytes per value*” (record) in the Database. This is interesting if you work with catman's script language and wish to save not just values (“*Double/Float*”) but also “*Text* “in the Database. Texts can enter Database channels in various ways

1. through manual entry in the Database editor, e.g. as extended comment,
2. via Script (Level 3) with the command `DBChan[].Text`,
3. by importing a file which contains text channels.

In format “*Double*” (default) 8 bytes are used for a measured value (double precision), in format “*Float*” 4 bytes (single precision). With the Database editor you can write a “*Maximum length*” number of texts with a length of “*No. of bytes per value*” each into a channel with text format.

Delete Database, Auto-recover Database on start

When starting, catman checks that a Database file is present. If yes, catman asks whether this Database is to be restored. If this is declined, then a further message warns you of the ensuing loss of data. If the Database is not to be deleted or the restoration is to be carried out automatically each time, activate, respectively deactivate the appropriate option. If the Database is not deleted at the end of the program, the data last measured is again available at the next start. On the other hand, the Database permanently occupies space on the hard disk.

We recommend you use a defragmentation tool for the hard disk in question before creating Database files which are larger than 50 Megabytes. In this case it is also recommended *not* to delete the Database on program end, i.e. use a *permanent* Database file.

Optimizing the speed

The presetting of 32 kilobytes for the size of the cache is sufficient for all normal application cases. You can use higher settings if you are working with MGCplus and CP42, many channels and high sample rates.

A corresponding amount of memory (RAM) is required, otherwise the transmission of the measured values will be slowed down.

2.3 The Database channel manager

The Database channel manager is primarily used for the management of the Database. You can change various channel attributes (name, unit, comment) and obtain statistical information about the state of the Database. You must use the Database editor to access the individual values contained in the Database. The Database channel manager changes the complete contents of a Database channel. Irrespective of which action you carry out, whole Database channels are always affected.

For some of the executable actions there are appropriate menu commands as well as the possibility of clicking one of the symbols in the toolbar.

Changed settings, e.g. the names and units of the I/O channels, are normally only updated on opening the worksheet. Therefore, in case of doubt, close the worksheet and open it again.

Basic working method

For the channel or channels which you want to process, set a *marker*. Then choose the required operation. With some functions, e.g. data export or mathematics, the channel or channels to be processed can also be defined in the dialog itself.

If you wish to mark several consecutive channels, click in the left outside column with the channel designations DB1, DB2 etc. You can use the “Channel marker” () to mark non consecutive channels.

Rows or columns may be enlarged or reduced by using the mouse directly on the column header or leftmost column, but these changes are not saved.

2.3.1 The worksheet columns

Type

This column shows, to which data type (numerical, text) the channel has been assigned.

I/O

The symbol is displayed here for every channel connected to a data source in column “Connection” (I/O Definition).

M (Marker)

Here you can set one of the *markers* with the context menu. For some functions it is sufficient just to mark the relevant channel by clicking on it.

Write protection (no label)

If the channels have been locked out via the user administration or the menu “Edit → Set/remove write protection” for editing, the symbol appears in the fourth column. You can also activate the write protection when importing data.

The write protection of a channel only prevents the *modification or overwriting* of values; it is still possible to *delete* all the values of a channel.

Import (no label)

If the channel contains imported data, it is labeled in the fifth column using the symbol .

Name

The channel name assumed from the I/O channel is located here. You can edit the entry by double-clicking on the cell. Once the I/O channel has been given a new name or after initializing channels, the name of the Database channel adapts accordingly, if a connection exists.

Channel names must not start with a number and are restricted to a maximum of 64 characters.

Unit

Shows the unit defined for the channel or indicated by the device. Changing the unit is done in the same way as modifying the “Name”.

Values

Number of the values currently stored in the channel and, in brackets, the maximum possible data depth.

Modified

Date and time of the last change to the channel. With active I/O channels this is the start of the last measurement.

Comment

You can enter a comment (maximum 128 characters) for the channel in this field. If the channel was created using a mathematical function, the formula used is stated as text in this column. When using long channel names or complicated formulas the available space might be insufficient. In this case the text is truncated.

☞ If the file has been imported, the file comment is available separately as *System text*.

2.3.2 Editing channels

The easiest way of copying, moving or appending a channel is to use the drag&drop functions of the Database channel manager:

Using the left mouse button, mark a field in the channel which you want to copy, move or append. Move the cursor to the edge of the marked field until the cursor changes from a cross to an arrow. Holding the left mouse button down, pull the marked channel onto the results channel. When you release the mouse button, a menu opens in which the following actions are offered:

- “Copy to” The source channel is copied to the results channel.
- “Move to” The source channel is moved to the results channel. The data is then no longer available in the source channel.
- “Append here” The data from the source channel is appended to the results channel.

Other functions for editing the channels are located in the “Edit” menu: delete, split, merge, etc. The functions “Split channel” and “Merge channels” are particularly useful for saving values from computations, which, like the FFT functions, for example, lay out their result in several data series with half the length of the initial data series. First mark the results channel. Then mark both of the channels which you want to nest with the “Channel marker” and select “Merge channels” from the “Edit” menu.

The function “Create sub-channel”, in other words, cutting out part of a data series, can also be done graphically: set the results channel in the Database, call the Online Document editor worksheet, display the data series in an y(x) Post-process graph and call the Graphics control panel, choose one of the “Cursor” tabs, display the cursors and move to the desired positions. Then click on the symbol in the Graphics control panel.

The function “Compress channel” is explained under computations in Section 3.12, *Compression*, page I-31.

2.4 The Database editor

Access to the Database editor may have been withdrawn via the user administration. Additionally, write protection may exist for individual channels so that no values can be *changed*.

To call the editor, either use the “Worksheet → Database editor” menu or click on .

If the editor is open, you cannot open any other windows. You have to close this view before you can use other windows or dialogs.

Via the Database editor you have access to the individual values in the Database. You can view the data, edit each individual value and manually enter new values. The individual cells can also be used for computations, that is, you can also enter an expression or a formula, for example 5 * pi. New values can only be entered starting at the top and free fields (lines) are not allowed. Exception: Channels which have been assigned the format “*Text*” may also contain any blank fields.

If you want to insert a new value between existing values, first mark the field above which the value is to be inserted. Then select “Insert value” from the “Edit” menu.

Using or “Find” in the “Edit” menu, you can search through individual channels for certain values.

Updating statistics

The manual entry or modification of values renders the statistical header data (Min., Max.) of a channel invalid. Normally all header data is automatically recalculated. With large amounts of data this may take some time however. If further entry is carried out in this period, the computation is interrupted. With this menu item in the “Edit” menu all values are recalculated.

Printing data

During the printout the currently set formatting of numbers is used.

The printout from the Database editor produces a table which corresponds to the editor display. If you first mark a data range, then only this is printed.

Any markers present in the Database Manager have no significance.

Importing/exporting data

Values can be imported into the Database editor as well as exported from it. When importing, the first import channel is defined by marking at least one field of the required channel. See also Section 2.5, *Importing and exporting data*.

Show channel information (acquisition parameters)

This is equivalent to the “Channel information” in the Database Manager.

Column format

Here you define whether the data is to be shown left, right or centrally justified in a channel in the Database editor. You can carry out this setting also via the toolbar.

2.5 Importing and exporting data

☞ With “Restore Database” (“File” menu), you can reimport a “crash file”, in other words a file generated in the event of a catman crash.

Values can be exported both from the Database channel manager as well as from the Database editor and also imported into the Database. When importing, the first channel into which the import is to be made, must either be provided with a marker or be marked. You can use markers for exporting and you can also use the export dialog for marking channels.

When exporting or saving values in one of the ASCII formats, the currently set number formatting for each channel in the Database editor is used. The default format is with three decimal places. A change can *only* be carried out in the Database editor.

2.5.1 The Data Explorer

Use the Data Explorer (or in the Database worksheet) to view the creation date and comment for all files in a directory that contain measurement data. If you had activated the Test Manager, instead of selecting All importable files, you can also select Test files from the top left. In this case, the test parameters will be displayed together with all the files used for the individual tests. You can fold and unfold the entire file structure from the context menu: “Show all test parameters/Display file comments” ( ) and “Fold tree structure” ( ). You can use the context menu to display the different columns that contain information in the lower section of the Measuring Data Explorer, e.g. the number of measuring values, the measuring rate or the Sensor ID. The columns will remain in the display even if catman is restarted.

First (before calling the Data Explorer) mark the channel from which saving of the imported channels is to start. You can also set the marker . Otherwise the first free channel is used.

The following file formats are recognized as importable files and shown in the overview:

- "*catman*" (*.BIN)
- "*Binary for online import*" (*.BIN, also a catman format)
- "*ASCII + channel information*" (*.ASC)
- files created by the MGCplus in the *.ME* format

  If measurement data from an MGCplus is still present on the internal PC Card memory, you can transfer it via worksheet Database and menu "File → Transfer MGCplus data file" to the PC.
- files which have been created by DIAdem in the *.DAT format (integer, R32, R48 and R64; only channel format, no block format).

Use to open a file dialog, which can also search through a network.

Search function

Use to search through the files in a directory (and its sub-directories) according to different criteria, e.g. according to test parameters (operator etc.), transducers used (Sensor IDs) or channel names. After searching through a directory, use "Folder view" or to display all the files in the directory again.

Use to sort the files according to date and file name. Use to sort in ascending or descending order.

Import

The Data Explorer is closed after each import (default setting). Deactivate "*Close Explorer after import*" ("Options" dialog) if you want to import several files.

Double click on the file you require to import it. Alternatively, you can also mark the file and click on in the toolbar or click on "*Import data*". If you only wish to import individual channels, click on the file and then use the context menu to mark the channels you require ( ).

You can display different information in the lower section of the Data Explorer via the context menu, e.g. the number of measured values, the sample rate or the Sensor ID. You can also mark or unmark all channels with the context menu.

Options for import ()

The settings for the options are retained even with a restart of catman.

In the default setting for catman the file name is appended to the channel name so that you can always distinguish between different measurements, even with several imports of channels with the same name.

The “*write-protected import*” prevents the changing of values. Manual intervention via the Database editor is then no longer possible. However, the values from these channels can be deleted.

Data can also be appended to data already present in the import channels: “*Append data*”.

Additionally you have the possibility of saving the channels at their original position during export: “*Save at original position*”.

The Data Explorer is closed after each import (default setting). Deactivate “*Close Explorer after import*” if you want to import several files.

External ASCII format

When importing in ASCII format there are further options available, e.g. ignore a certain number of rows at the start of the file or evaluate with “*Header*” channel name and/or unit.

catman can also import any ASCII files not produced by catman. However these must be present in the following format:

```
0.0000 <xx> 0.0000 <xx> ... CRLF
0.0000 <xx> 0.0000 <xx> ... CRLF
```

Here, <xx> may be one or more (also different) separation characters. Any non-numerical character is taken as a separating character with the exception of the decimal point, comma, plus and minus arithmetical signs and the exponent characters E or e.

2.5.2 Exporting data to MS Excel

A prerequisite for the export to a Microsoft Excel worksheet or file (*.XLS) is that Microsoft Excel from at least version 7.0 is installed on your computer. Please note, that Excel worksheets may not contain more than 65,536 rows. Any excess data will be ignored.

Direct export to Excel

Mark the channels to be exported with the Marker “MS Excel data exchange” and click on or select “Microsoft Excel → Export selected channels” in the “File” menu. catman calls Excel, copies the data and displays the Excel worksheet. If no channels are marked, catman asks whether all channels which contain data are to be exported.

Via “Microsoft Excel → Options” in the “File” menu you can specify how the data is to be handled for repetitive export processes:

- overwrite existing data
- export in new columns on the same worksheet
- export to a new worksheet

☞ New worksheets will be created automatically, if the active worksheet has not enough columns to accommodate the channels transferred.

If desired, the channel information (acquisition parameters of a channel) can also be transferred. Since a varying number of rows are needed for the header depending on the device, you should define here a certain number of rows after which the measurement values are to be saved.

☞ A number entered in “*Data begins at row*” will also be considered when no channel information is transferred.

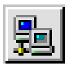
Automatic export to Excel

Via “Microsoft Excel → Options” in the “File” menu you can specify that data from the marked channels is automatically transferred after each measurement period of the Measurement Wizard (Periodic measurement mode) or of the catModule Periodic measurement. In this case Excel is not displayed automatically. Activate Excel after your measurement via “Microsoft Excel → Activate” in the “File” menu or with .

2.5.3 Exporting data

☞ You can only selectively import Database channels, i.e. just single channels, from files saved in the formats “*ASCII + channel information*”, “*catman*” or “*Binary for online import*”.

As an option, you can with some formats select whether the data to be exported is to be appended to an already existing file. However, this assumes that the structure and type of the existing file matches the channels to be exported, i.e. that the number of channels and, with ASCII formats, the selected separating character match.



☞ Use to open a file dialog, which can also search through a network.

The binary catman formats

Files in the format “*catman*” can only be read in again by catman itself. However, the speed of the export is substantially higher than in all other formats. In addition the channel information (acquisition parameters of a channel) is also saved and you can provide a comment for the file. This is then visible later in the Data Explorer without you having to load the file.

ASCII and ASCII + channel information

When exporting or saving values in one of these formats, the currently set number formatting for each channel in the Database editor is used. The default format is with three decimal places. A change can *only* be carried out in the Database editor.

Example: With a sample rate of 9,600 measurements per second, then in the setting with three decimal places, the first five values for the time channel are saved with 0.000 because the value is less than 0.0005. From the sixth value there is then 0.001, etc. Either use more decimal places or one of the scientific formats with an exponent.

Data exported in this form is present as ASCII text file and can be read directly by numerous other programs. The data of a channel is located in a column with the columns being separated from one another by a freely selected “*ASCII separator*”, e.g. TAB or semicolon. Each row is terminated by CRLF (Carriage Return/Linefeed, standard for DOS and Windows). The first rows of the file contain, if required (“+ *channel information*”), acquisition parameters (traceability data).

☞ The traceability data (channel information) can only be restored from a file saved in format "*ASCII + channel information*", if the *parameter IDs* have been saved additionally. Note that export in the ASCII format is slower than the binary export due to the formatting processes.

DIAdem® (*.DAT)

Files exported in this format can be directly read in by the data analysis software DIAdem/DIA-PC from National Instruments Corporation (National Instruments Engineering GmbH & Co KG, former GfS Systemtechnik GmbH & Co. KG). A header file and a data file are produced when exporting data in this format. The header file can be directly interpreted by DIAdem and the data file is in the format REAL64.

You may import files in this format into catman again. See also www.ni.com.

nSoft-E®

catman® Professional supports the export of Database channels for the evaluation and analysis package nSoft from nCode. For this, catman can produce two different nSoft formats:

- Time-series format (*.DAC)

  An equidistant measurement series is written to a file, i. e. one Database channel produces one file. The file name is created with the name provided by you, the channel name (appended with an underscore) and the file extension DAC.
- xy pairs format (*.MDF)

  Two channels are written to one file, one as the x-axis, e. g. displacement and the other as the y-axis, e. g. force. The file extension is MDF.

Files in this format cannot be read in again by catman itself. See also www.ncode.com.

EDASWin® (*.EDT)

Files in this format can be directly processed by the data analysis software EDASWin from MH GmbH. The file extension is EDT.

Files in this format cannot be read in again by catman itself. See also www.mh-gmbh.de.

FlexPro® (*.FPD)

Files in this format can be directly processed by the data analysis software FlexPro.

Files in this format cannot be read in again by catman itself. See also www.weisang-co.com.

CAESAR® Remus (*.RMS)

Files in this format can be directly processed by the data analysis software Remus.

Files in this format cannot be read in again by catman itself. See also www.caesar-datensysteme.de.

RPC® (*.RPC)

Files in this format can be processed directly from the data analysis software RPC III from the MTS® Systems Corporation. As the format does not allow individual channel lengths, so-called groups each with 2,048 values are always saved. If the catman channels to be exported do not contain a multiple of this value, the missing values will be filled with zeroes. You should therefore always acquire or export a multiple of 2,048 values. As integer values with 16 Bit must always be saved in this format, catman uses the minimum and maximum of the respective channels and scales all values accordingly.

Files in this format cannot be read in again by catman itself. See also www.mts.com.

ASAM-ODS (*.ATF)

ASAM e. V. (registered association) is a shortcut for Association for Standardization of Automation and Measuring Systems. ASAM-ODS (Open Data Service) defines a generic data model (for universal interpretation of data), interfaces (for model management, data storage and data retrieval), and a data exchange syntax and format.

Files in this format cannot be read in again by catman itself. See also www.asam.de/new/03_standards_07.php and www.asam.net.

Excel® (*.XLS)

Files in this format can be directly read in by MS Excel. The channel information (acquisition parameters of a channel) may be saved together with the data.

Files in this format cannot be read in again by catman itself. See also www.microsoft.com.

2.6 Printing data

During the printout the currently set formatting of numbers is used. The formatting can be changed in the Database editor *only*.

☞ You can also printout values in the Database via the Database editor, but you cannot set any formatting of the printout with headings or footers there. The printout from the Database editor produces a table which corresponds to the editor display. If you first mark a data range, then only this is printed. Any markers present in the Database channel manager have no significance.

Mark the channels to be printed with the printer symbol. Then select “Print” in the “File” menu. In the following dialog you can specify further options before the printout:

Left margin
Distance of the printout from the left edge of the page (in Twips).

Header line
The text entered here appears in the header line on each page. By selecting the appropriate options below the input field, the page number, date and/or time can be automatically appended to the text. This also applies if no header line text has been entered.

Footer
The text entered here appears in the footer on each page. By selecting the appropriate options below the input field, the page number, date and/or time can be automatically appended to the text. This also applies if no footer text has been entered.

☞ The printout uses the font currently set. Do not choose one which is too large, in case there are several measured values in a channel, as otherwise 10,000 values could easily produce 300 pages of printout.

3 Post-process mathematics

With the “Mathematics” menu you can call up the dialogs for post-process computations, i.e. computations based on the measurement values contained in the Database.

The Script language from catman however contains additional mathematical functions, not just the interactive ones. These can be called via the Auto Command List.

A mathematical computation produces one or more new Database channels. You can set the first results channel either by setting the Marker in the Database channel manager or in the dialog itself. If the computation supplies more than one results channel as with the FFT, then these follow straight after the first one.

Results channels which are following overwrite existing data without warning, only the first results channel is checked.

If you specify a results channel for which the corresponding I/O channel is connected, for example, to a hardware channel or an online computation, then you are asked for confirmation, because the computed values are then overwritten during the next data acquisition. If the results channel already contains values you must also confirm that they are to be overwritten.

You can enter a name for the channel(s) in the “Name” column of the computation dialog; catman displays the type of result, e.g. filtered values, in the first column in this table. If you do not enter anything, catman will issue its own names which characterize the computation. In the “Comment” column, catman creates a detailed description of the formula using the full channel names of the channels involved.

No counter is produced for multiple computations of the same type.

All computations are only executed when the dialog is called and confirmed with “*Compute*”. If a computation is to be carried out over and over again, you can include it in the Auto Command List by clicking “*Include in Auto Command List*”. Here, these computations can be executed, for example, after each period or on pressing a function key, e.g. F9. You will find further information about this in Chapter J, *The Auto Command List*.

catman does not produce any units for the computed channels and you must define these yourself in the “*Unit*” input field.

As some functions just do simple computations, not all computations are explained below.

3.1 f(x) and f(x,y)

This function is called using the “Mathematics → Create f(x) function” resp. “Mathematics → Create f(x, y) function” menu.

This can be used to compute any z = f(x) or z = f(x,y) functions. For this purpose, the function is to be defined in free, algebraic notation, with x or x and y as the argument, e.g. sin(x) + 0.5 * cos(x). The argument values used for computation are determined by predefining the start value “*x0*” or “*y0*”, the step “*dx*” or “*dy*” and the number of data points “*n*” to be generated. The x argument values or y argument values used can also be saved, on request: option “*Create x channel*” or “*Create x, y channels*”.

The function “*f(x)*” produces a channel of length n. When computing the function “*f(x, y)*”, you can choose between two different forms of storage. Normally, all function values are written in a single channel. If you choose the option “*Use matrix to store f(x, y)*”, all the function values for y are written in successive channels. This produces a matrix in the Database, the rows of which represent the x values and the columns of which, the y values. In this case, you must take into consideration the need for free columns!

This function is particularly useful for comparing measured data series with analytical functions. If an f(x) function is merely to be shown as a plot, in the graph object, in the “Data sources” menu, you can also call the “Function plot” tab and enter the formula there.

3.2 Creating an OLE time channel

The function creates a channel for time information. The data is saved in the Windows date and time format (OLE-time). This channel can then be used as an absolute time channel, e.g. in graphs.

The resolution of this time format is 1 second, all decimal places (milliseconds) will be lost.

You can use:

- the sample rate that has been saved in the channel information of the argument channel
- an NTP time channel of the MGCplus
- an IRIG-B time channel of the MGCplus

as output channel.

For NTP and IRIG-B time channels, see Chapter D, Section 3, *Configure time channels*, page D-17.

3.3 Signal analysis

These functions are called using the sub-menus of menu “Mathematics → Signal analysis”.

FFT functions (fast Fourier transformation) usually result in three new channels. If you do not need the frequency channel, you can suppress its creation. You can use an “*Auto power spectrum*” to define the frequency components which occur in the signal. With all FFT computations, you can choose from a number of window functions.

“*Auto power spectrum*” with “*Window function: None*” and “*Scaling: No scaling*” computes the amplitude of the *Power Spectral Density* (PSD). Here, the magnitude and unit of the signal in the time range is retained.

Only 2^n values are used for computation. You should therefore make sure that the number of values recorded is not just below one of these numbers, for example, do not measure 1,000 values, as only 512 of them will be used for computation. It is better in this case, for example, to measure 1,050 values, as 1,024 will be required.

With the correlation function, you can determine both the auto correlation and the cross correlation: for the auto correlation, enter the same channel twice. This gives you evidence of the extent of conformity, i. e. for auto-correlation, whether there are periodic components in the signal and for cross correlation, whether two channels contain the same signals, just with a time lag, or the same signal components.

If the time channel is no longer available, but you still know or checked the sample rate from the channel information, enter it here and activate “*Time channel not available—manual input of meas. frequency*”.

3.4 Peak analysis

This function is called using the “Mathematics → Peak analysis” menu.

The special feature of this computation is that it establishes neither *the* minimum nor *the* maximum of a measurement sequence, but that it can determine, for example, the frequency and time of occurrence of several maxima or minima. This is often also called the search for *local* extreme values. As the computation has to decide as from which level change there is an extreme value available, you must enter a suitable value in the “*Hysteresis*” field (in the unit of the chosen channel). If the hysteresis is too low, even small changes, caused by noise or superimposed interference, count as extreme values, and there are too many hits. The computation produces two new data series, one with the x values, for example the time at which the extreme values occurred, and one with the extreme values themselves.

3.5 Digital filters

This function is called using the “Mathematics → Digital filters” menu.

Various filter types and the functions Low-pass, High-pass, Bandpass and Bandstop are available. For filter types which are not linear throughout their entire range, you can specify the extent of maximum ripple.

3.6 Curve fit

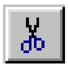
This function is called using the “Mathematics → Curve fit” menu.

An approximation to the measured values can be calculated here for the various Fit models.

The model function must be able to approximate the signal form to obtain meaningful results.

Be sure to comply with the following constraints:

1. A sine-wave signal cannot be approximated with a linear function.
2. With periodic trends, only one period can be approximated. The overall trend is then sufficiently known, as the trend is repeated periodically.
3. The values must be monotone, in other words, there must not be two different y values for one x value, as in the case of force/displacement plots, for example: two different force values emerge for the same displacement value, one for the forward displacement, the other for the backward displacement. Here the two plots must first be *separated* with the “Subchannel” function. Do do this, use the “Edit” menu, or better still, a graph, for example, using the Graphics control panel, one of the “Cursor” tabs and the symbol .



If you have activated an y(x) graph on the Online Page, the “*Create function plot*” button creates the definition of a plot there, with the computed coefficients.

3.7 Curve smoothing

This function is called using the “Mathematics → Curve smoothing” menu.

Both simple averaging and the considerably more effective Savitzky-Golay computations are available. The latter have the advantage that even with small window sizes, interference is effectively suppressed and the original signal is still basically retained. Pure averaging, on the other hand, also reduces the amplitude of the basic frequencies and *smooths* the overall signal.

3.8 Interpolation

This function is called using the “Mathematics → Interpolation” menu.

This function enables you firstly to generate additional values between the measured values and secondly to normalize measured values from different measurement sequences, so that they can be compared by computation.

A typical application for the latter are force/displacement plots from different tests. Both the force and the displacement data are recorded at different times. Usually the number of data points available over the section being examined is also different. This is where interpolation is useful: the y values to be compared are converted using a new x base, resulting in a new y channel.

Enter the measured force values as the “*y channel*”, the measured displacement values as the “*x channel*” and computed (quasi *normalized*) displacement values (function “f(x)” with “*x*” as argument) as the “*x base*”. The interpolation then provides the force values associated with the respective new (equidistant) displacement data, by examining the measured values and computing the relevant points by means of a polynomial. Please be sure to record at least as many measured values as you need for computation later, since in principle there is a reduction in precision due to the approximation.

Interpolation only works for monotonically increasing or monotonically decreasing values of x.

Should the noise component of your measure signal be too high, you should use a *curve smoothing*. If your measure signal contains a hysteresis loop, this must first be separated and interpolation applied separately to both parts.

3.9 Change/convert sample rate

This function is called using the “Mathematics → Change sample rate” menu.

The computation converts data, which has been acquired with a certain sample rate, into the required number of values in order to be able to compare them with another series of measurements with a higher or lower sample rate. Depending on the new sample rate, more or fewer new values are present in the results channel then in the argument channel.

The following interpolation methods are available:

1. Polynomial (3rd order, i.e. four reference points are used)
2. Rational function
3. Cubic spline
4. Linear interpolation (straight line function)

If with one of the Methods 1 to 3, the desired type of interpolation is not possible, e.g. because not enough symmetrical adjacent points exist in the original set of data around the new x reference point, then a linear interpolation is carried out automatically or, if the x reference point lies outside of the original set of data, an extrapolation is carried out.

3.10 Channel length adaptation

This function is called using the “Mathematics → Adapt channel length” menu.

With this function a channel can be adapted in its length through interpolation to the length of another one, the “*Reference channel*”. Here, both shortening and lengthening are possible.

After the computation two channels measured with two different sample rates can also be compared with one another or used in computations.

To carry out the matching, the “*y channel*” to be matched (amplitude), its corresponding “*x channel*”, e.g. a time channel, and the “*Reference channel*” must be specified. The reference channel should be of the same type as the x channel. The computation supplies one results channel of the same length as the reference channel.

Specify a name for the results channel to eliminate later confusion.

3.11 Normalize channel

This function is called using the “Mathematics → Normalize channel” menu.

This computation is especially interesting if you are examining transducers in test equipment. It is possible to establish the deviation from the ideal straight line, for the different definitions. Depending on the definition to be used, choose whether you want to compute the deviation with regard to the “*Zero-endpoint straight lines*”, the “*Best-fit straight lines*”, or similar. The result can be output as a percentage, per thousand, as an absolute value, etc.

3.12 Compression

This function is called using the “Edit → Channel operations → Compress channel” menu or via the “Compression” tab in the “Post-process mathematics” dialog.

If, after a measurement, you establish that too many measured values have been recorded, as the maximum signal frequency was overestimated or because other channels in the same device had to measure at a higher data transfer sample rate, you can use this function to reduce the number of values. This is particularly useful if the values are to be saved. Averaging produces the best results here. In the case of the “*Min/Max*” setting, both the minimum and the maximum are saved in the results channel one immediately after the other. The extreme value which occurs first will also be saved first. This means, for example, that the following arrangement could occur: min_1, max_1, max_2, min_2, max_3, min_3, min_4, max_4 etc. As, in the end, there are two values for each computation, the actual number of values included in the computation is twice as high as the compression factor in order to produce the required reduction.

Do not forget to compress the associated time channel as well, as otherwise the graphs would not have an x channel available, containing the right values.

3.13 Logic functions

This function is called using the “Mathematics → Logic functions” menu.

As a logic function only “*Extract bit*” is currently available. This enables you to extract a certain bit from a channel, in which data is saved from a remote-control bitmask, and write only that to a dedicated channel. You will find information about a remote-control bitmask in Chapter D, *Defining Devices and Channels*, in the section starting at page D-38.

3.14 Strain-gage computations

This function is called using the “Mathematics → SG computations” menu. The computations in the dialog following can implement the compensation of the temperature influence for measurements with strain gages and also determine the material stresses from the measured strains.

The following possible evaluations are available:

1. Stress analysis with strain-gage rosettes with grids at 0/45/90°, 0/60/120° or 0/90°

   The principal normal stresses and the angle with respect to measuring grid a is computed from the three strain channels. For measurements with a plane stress state and known principal directions, 0/90° rosettes can be used with only two measuring channels. If required, the comparative stresses according to von Mises can also be computed and the transverse sensitivity of the strain gage taken into account.
2. Special stress analysis with hole-drilling or ring-core methods

   Various forms of residual stress analysis are possible with these methods.
3. Temperature compensation

   The computations compensate the apparent strains of the objects due to temperature effects or determine the thermal expansion coefficient of a material.

3.14.1 Stress analysis with standard rosettes

You select the appropriate Database channels in the drop-down lists designated “*Rosette strain a, b*” and “*c*”. The designations correspond to the measuring grid labels normally used by HBM. If you are using rosettes from other manufacturers, make sure that the measuring grids have the correct orientation: Starting from a in the mathematically positive sense to c. The angle in the results channel is referenced to measuring grid a in the mathematical positive direction of rotation (counter clockwise). You will find the formulae used for the computations in Chapter D, Section 4.5.3, *Strain-gage computations and temperature compensation for strain gages*, page D-30.

3.14.2 Residual stress analysis

The following methods are available:

- the integral-(simple) hole drilling method
- the improved hole-drilling method (MPA Stuttgart)
- the ring core method

For the simple hole drilling method, bore a hole of the same depth as the drill diameter. This is followed by measurement and evaluation, that is, determining the self-contained stresses. Using this method, no data that is dependent on the depth of the hole can be determined. Using an improved method developed by the MPA Stuttgart, the hole depth is also measured and strain values are computed with the aid of a calibration plot and the hole depth. This provides evidence for the self-contained stresses in different surface layers. The ring core method requires special drills and machines. With this method too, you can drill in stages to obtain evidence about the self-contained stresses in different surface layers.

3.14.3 Temperature compensation

The third tab “Temperature compensation” provides three different functions which are useful if you cannot attach an additional strain gage for compensation of thermal expansion of the object used:

1. Compensation of the thermal expansion of the object using the polynomial on the strain gage packaging, the thermal expansion coefficients of the strain gage and the material, as well as the actual object temperature.
2. Correction of the residual error of the strain gage thermal expansion compensation by linear compensation of this error between two temperature values.
3. Determining the thermal expansion coefficient of the material, to which the strain gage is attached.

☞ In the first case, the following is applicable: as compensation uses the ideal thermal expansion coefficient for the object material, uncertainties or residual errors from fluctuation, different material treatments such as annealing, rolling, etc., remain here as well. Be sure to comply with the temperature limits applicable to the polynomial on the strain gage packaging; outside these limits, deviation may vastly increase.

The computation function used is:

$$\varepsilon(T) = \varepsilon_{measured}(T) - (a_0 + a_1 T + a_2 T^2 + a_3 T^3) - (\alpha_{material} - \alpha_{SG})(T - T_{ref})$$

The second method can only be applied if the behavior of the strain gage, in other words, its residual error, is linear in the required temperature range. Measuring the expansion for two temperatures establishes the strain gage deviation and linear compensation using these values corrects it.

In the third case, the thermal expansion coefficients of the strain gage and the material should be as different as possible, to create high expansion values and to reduce the uncertainty.

☞ The reference temperature is the temperature at which the thermal expansion coefficients of material and strain gage were measured.

The compensation of object strains due to the temperature response for a number of measurement points can also be implemented in that in one channel only the temperature effect is measured on the material used. Then this channel is subtracted from the active channels through a user scaling in the I/O Definition: “User scaling → Strain gage scaling” and “*Temperature compensation with compensation SG*”. To achieve this, the same type of strain gage (foil lot number) is bonded to a separate piece of identical object material which must have the same temperature as the active strain gages and can expand without hindrance.

3.15 Statistics

This function is called using the “Mathematics → Statistics” menu.

”Channel overview” shows statistical descriptive data such as Minimum, Maximum, Average and Standard deviation of a channel.

Averaging over several channels (”Average channels”) makes it possible, for example, if several measurement sequences of a periodic transaction are available in successive channels, to form the average, in other words, a type of *standard plot*.

3.16 Counting

This function is called using the “Mathematics → Statistics → Counting” menu.

There are various techniques available for classifications:

1. Time at level
2. Rainflow matrix
3. Span pairs from rainflow matrix

3.16.1 Time at level counting

This type of classification is similar to a histogram, but it also needs time information and, instead of simple numerical values, it saves the overall time which a signal has spent within a certain amplitude range, i. e. within a class.

As a data source you can use:

- a Database channel. Select the channel under Argument channels.

- a channel from a binary file which has been produced through an online export. Here, you must also state the Channel position in the file, i.e. the position of the channel to be examined amongst those saved in the file.
- a channel from a file which has been produced by the CP32 or CP42 in the MGCplus. The file must be present on the PC hard disk. Here, you must also state the Channel position in the file, i.e. the position of the channel to be examined amongst those saved in the file.

If the measurement data from an MGCplus is still present on the internal PC Card memory, you can transfer it via worksheet Database and menu “File → Transfer MGCplus data file” to the PC. The MGCplus Assistant is not needed in this case.

The following parameters must be specified for the classification:

- *Number of classes*
- *Class minimum*
- *Class maximum*
- *Minimum amplitude*

3.16.2 Rainflow counting (matrix)

This executes a rainflow counting of a data channel according to the four-point algorithm and saves the resulting matrix in the Database. With n classes n channels by n rows are produced.

As a data source you can use:

- a Database channel. Select the channel under Argument channels.
- a channel from a binary file which has been produced through an online export. Here, you must also state the Channel position in the file, i.e. the position of the channel to be examined amongst those saved in the file.

- a channel from a file which has been produced by the CP32 or CP42 in the MGCplus. The file must be present on the PC hard disk. Here, you must also state the Channel position in the file, i.e. the position of the channel to be examined amongst those saved in the file.

If the measurement data from an MGCplus is still present on the internal PC Card memory, you can transfer it via worksheet Database and menu “File → Transfer MGCplus data file” to the PC. The MGCplus Assistant is not needed in this case.

During the classification an n*n matrix is created where n is the Number of classes. Additionally, a channel is created with the residues and another one which contains the multiple of the class width. This channel can be used, for example, for index labelling in a 3D graph. The length of the channel with the residues can be up to twice as large as the chosen number of classes.

The data can be saved in two different formats:

1. In the Database: from the channel marked with the results marker or from the results channel selected in the dialog.
2. In a special rainflow matrix file.

The rainflow matrix file cannot be read in again by catman itself. This form of storage is therefore only practicable in producing files in the same format as they are formed during the MGCplus internal rainflow classification. Files of this type are used, for example, by numerous automotive manufacturers for in-depth analysis.

The following parameters must be specified for the classification:

- *Number of classes*
- *Class minimum*
- *Class maximum*
- *Minimum amplitude*

☞ If required, the classification can be carried out with the Clorman-Seeger correction.

3.16.3 Span pairs from rainflow matrix

The computation first carries out a classification with the rainflow matrix and, based on the rainflow matrix counts for so-called span pairs, it counts how often the signal goes through transitions of a certain amplitude span. The rainflow matrix itself is not saved in the Database.

☞ The parameters of this computation correspond to those of the Rainflow counting.

4 QuickView

The QuickView window is an independent type of data visualization, i. e. it is available additionally to an Online Document. It enables quick and simple visualization of measurement data in independent windows, each with a 2D Scientific graph and an easy selection facility for the channels to be displayed.

QuickView windows can be opened via the main menu or the Database channel manager, menu “Worksheet → New QuickView window”.

The most important settings can be configured directly by a click on the axes, title or a legend entry. Other parameters are accessible via the window menu or toolbar; the configuration however offers fewer possibilities than that of a Scientific graph on an Online Page.

With post-process data sources a QuickView window can manage Database channels (always visible) and files. A special feature with the display of channels from a file is that this data is shown directly and must not first be imported into the Database. If a lot of data is to be displayed, just a part of the data (Min/Max pairs) can be selected for display with “Use data reduction” in the “Graph” menu; the display is then faster.

☞ Settings which have been made once can be saved using “File → Save layout as”. This file can also be included into a project, see Chapter B, Section 4.1, *Save/Open Projects*, page B-14. The setting is reactivated with “File → Open layout”.

General window operations

Any number of windows can be opened simultaneously and they can be cascaded, tiled vertically or horizontally (“Window” menu).

The subdivision of the window between the channel list and graphics can be changed by pulling with the mouse. To do this, the mouse is moved over the separation line between both objects so that the mouse pointer becomes a double arrow. Then the separation line can be moved to the left or right by keeping the left mouse button pressed. The channel list can also be completely masked out or shown again with “Window → Channel list visible”.

In the channel list (left section of the QuickView window) a context menu is available with which, for example, data can be deleted or sets of data can be inserted or removed. In the graphics area various dialogs can be called with a double-click; also, the y coordinates are displayed with a double-click on the curve.

4.1 Displaying Database channels

On the left side the QuickView window contains the list of Database channels. The channels are combined into groups which can be expanded and collapsed individually: To do this, click on the plus or minus symbol of the range symbol. The size of the groups can be set via the menu “Window” with “Channels per node”; the default is 10.

Channels which contain values are labeled with a prefixed . The data is visually displayed by marking a data source in the list; when the box is clicked again, the set of data is removed from the graph.

4.2 Displaying files

A QuickView window can read the binary catman files, MGCplus measurement files (*.MEA) and optimized MGCplus measurement files (*.MEP). The data is displayed directly from the file and it does not need to be first imported into the Database.

Files can be inserted into the channel list in addition to the Database channels using “Insert file” in the “File” menu. In the list the file name is shown as a further group *right at the bottom* with a prefixed . Below the node all channels contained in the file are listed. The file group can also be expanded or collapsed.

The data is visually displayed by marking a data source in the list; when the box is clicked again, the set of data is removed from the graph.

A file (and with it all its sets of data) can be removed again from the channel list by selecting the file node and calling “Remove file” from the “File” menu or the context menu.

☞ Data sets from Database channels and from files can be displayed simultaneously in the graph.

To be able to display the data as quickly as possible, buffering via the RAM is activated in the default settings. If you have less RAM, you can deactivate this option via “Graph → Curve caching”.

4.3 Real-time operation

Apart from the post-process visualization of data, real-time data (I/O channels) can also be displayed: select “Graph → Use as Real-time stripchart”. In this case the left section shows the available I/O channels from the I/O Definition. If necessary, the buffer size must be adapted using “Stripchart buffer size” in the menu “Graph”.

The Measurement Wizard can also be started directly from a QuickView window. In this case only QuickView windows will be displayed and refreshed. Only if the Measurement Wizard is started from an Online Document both Online Pages and QuickView windows will be displayed.

4.4 Graph configuration

The graph configuration is basically carried out as for the 2D Scientific graph, just the access to the configuration dialogs can only occur in two ways:

1. via a click in the axis, title or curve name in the legend
2. via the menus “Graph”, “Axes”, “Zoom” or “Cursor”
3. via the icons in the toolbar

When the mouse pointer is located over a section to be changed in the graph and turns into a hand symbol ( ), you can call the appropriate dialog by a click.

Defining the x-axis

Irrespective of the type of displayed sets of data (Database channel, I/O channel or from a file), the selected x data source applies to all sets of data in the graph. If the displayed sets of data are of different length, then the y and x values in the unused data sections of the shorter channels are set to zero and not displayed. The selection of the x data source can occur either before inserting sets of data or later. In the second case the graph is completely redrawn based on the new x data.

The sets of data can be plotted against the x-axis in three different ways:

1. Point index

   The y data is plotted via a counter index. This type of display can be used when no x information at all is available. The x starting point is always zero with this display.

2. Time (channel information)

   The y data is plotted using the Δt saved in the channel information. Here, it is assumed that the series of measurements has been recorded equidistant with respect to time. If the channel information does not include any Δt, plotting takes place using the point index.

   ☞ If the x-axis unit is not to be displayed in the default setting "*Seconds*", the unit to be applied must be defined via the menu "Axes → Time unit for x-axis". The Δt in the channel information is always present in milliseconds (ms).

3. Set of data

   The y data is plotted against the data of a previously selected set of data, e.g. for a force/displacement display. The selection of a set of data as the x data set occurs by marking the channel in the channel list and stating "Use as x data set" in the context menu. The channel selected in this way is displayed in the x list as default.

   ☞ No data reduction is possible with this type of display.

Curves

You can change the curve attributes after clicking on the relevant curve name in the legend.

Post-process compression (Menu "Graph")

This option is suitable when a lot of values (> 100,000) are to be displayed. Since here much time is required for (re)drawing, you can activate a compression. You enter the minimum number of data points from which compression takes place under "*Compress from … points*" (default setting 10,000). For the compression, the compression factor is first determined from the number of original values and the minimum number of values for the compression. In the next step min/max pairs are computed from the original values using this factor and displayed. However, once you zoom in on a section, a new computation is carried out. In the end, with a sufficient zoom factor only the original values are displayed, i.e. the compression process is completely transparent to you.

Drawing speed

With a large number of measurements a coarser “grid” can be set when drawing curves by increasing this number, i.e. how many values are taken into account for drawing one pixel. Consequently, the screen formation is quicker, but with higher values a loss of resolution occurs. Normally, it is better to activate the post-process compression.

Annotations

Annotations are additional details which can be positioned within the graph. You have the choice of the following types: “*Symbol annotation*” (with or without text), “*horizontal line*”, “*vertical line*”, “*x axis annotation*” (vertical text at the stated place on the lower x axis), “*y axis annotation*” (horizontal text at the stated place on the left y axis), “*line*”, “*rectangle*”, “*rectangle with rounded edges*” and “*ellipse*”.

Procedure: You create annotations via and entry of the coordinates.

Symbol annotations can be most easily produced directly in the graph by double clicking on a point on the curve. If required, you can modify the annotation via the context menu (*left* mouse key) “Edit label”.

☞ Always, only the information of *one* annotation is superimposed in the first line. Once an annotation has been produced, you must call up the edit line via the context menu “Edit label”. During editing, the button “*OK*” is displayed on the left side of the line instead of “*New*”.

With Rectangle, Rectangle with rounded edges and Ellipse you must enter the corner coordinates within which the objects lie.

The entries for x and x1 as well as for y and y1 must be made such that x/y defines the left *upper* corner and x1/y1 defines the right *lower* corner, e.g. x = 115, y = 100 and x1 = 120, y1 = 75.

4.5 Post-process mathematics

Some post-process computations can be accessed directly from the QuickView window via menu “Math”. The functionality is the same as when you perform computations in the Database worksheet. Also here Database channels have to be entered as result channel (see also Section 3, *Post-process mathematics*, page I-23).

J The Auto Command List

1 Introduction

If you use catman at operating Level 1 or 2, that is, with the catModules or the Measurement Wizard, you can record measured values and save them in the Database or in a file. By using the Database, you can then call the post-process mathematics to continue processing the data. This must be repeated, however, for each data record, as catman is not using automatic recomputation.

However, with the Auto Command List (ACL), this is precisely what happens: Call the desired post-process functions, but do not execute them, simply "*Include in Auto Command List*".

Although the data acquisition options available with the catModules and the Measurement Wizard are extremely extensive, a small supplement is sometimes useful. The Auto Command List can also serve you well in this situation: it enables you to create command sequences which can then be processed after each measurement period, for example. The advantage is that you only have to specify the additional functions required in script syntax, not the entire measurement sequence.

As there is a vast difference in the degree of difficulty of the two examples given here, we have split this Chapter into the following sections for an explanation of the working principles of the Auto Command List:

1. Section *General settings and time of execution for the commands*, page J-4
2. Section *Post-processing with the Auto Command List*, page J-7
3. Section *Auto Command Lists for script programmers*, page J-9

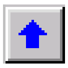
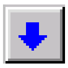
☞ The latter two sections each contain examples for a better understanding. You can find the examples in catman's EXAMPLES\MANUAL sub-directory.

2 General settings and time of execution for the commands

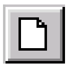
2.1 General settings

To create an Auto Command List it need not be open and it is sufficient with the post-process computations to click on the button “*Include in Auto Command List*”. If the window with the Auto Command List is open, you can mark a line by clicking on it, then copy it, cut it out, insert it, etc.

☞ Lines are deleted either via the “Edit” menu or by cutting out the relevant line. When inserting, you can also mark an already completed line which is then moved downwards and not overwritten.

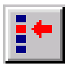
To edit a line you must double-click it so that the text cursor appears. To move lines upwards or downwards, mark them and click on  or . On the right-hand side in the column “Execute” you define the time of execution. In the left section of the window you will find a list of the possible commands, sorted according to groups. Operation takes place as with the Code Builder in the Script editor worksheet.



If you only want to use the Auto Command List later, it can be deactivated before the window is closed or, after saving the list, you can simply click on  (New Auto Command List) and create a new (blank) list. You then load the Auto Command List to be executed before the start of the measurement and it is translated and activated with a click on “*OK*”.

☞ An Auto Command List can also be included into a project. On loading a project the *first* list is always activated.

2.2 Time of execution for the commands



You can define different starting times for execution in the column “Execute”, both for the catModules, the Measurement Wizard and also for your own scripts. By clicking on  execution times can also be defined for the marked lines. If a number of instructions of one type are present, then they are processed in the order in which they occur in the Auto Command List, from the top to the bottom.

The available execution time-points are shown in Table 5; the parameter values given on the left are only of interest to script programmers.

Parameter	Execution time-point
0	At the start of the module, i.e. on calling the catModule, Measurement Wizard or script.
1	At the end of the module, i.e. on terminating the catModule, Measurement Wizard or script.
2	After each measurement period. With a long-term measurement and measurement on key depression the commands are executed after each single measurement. With the frequency analysis the commands are executed after each read block.
3	Before each measurement period. With a long-term measurement and measurement on key depression the commands are executed before each single measurement. With the frequency analysis the commands are executed before the start of the first measurement.
4	Before the start of the continuous measurement (catModule or Measurement Wizard).
5	Before the termination of the continuous measurement (catModule or Measurement Wizard).

Table 5: Execution time-points of the Auto Command List

Parameter	Execution time-point
6 to 14	Manual request with the function keys F9 to F1. These execution time-points are available to you in all catModules and in the Measurement Wizard.
15 to 18	User1 to User4. These execution time points are only available to you when catman® Professional is controlled in Level 3 via a self-written script which supports these functions.

Table 5: Execution time-points of the Auto Command List

3 Post-processing with the Auto Command List

For periodic measurements with post-processing or generally with computations which are executed over and over again, the Auto Command List can handle the automatic execution: Call the dialog for the required post-process functions in the Database worksheet, but do not compute the function and instead just use "*Include in Auto Command List*".

For creating an Auto Command List it is helpful to have a set of data available for testing, since you can then immediately see whether the planned computations lead to the desired result in the intended sequence. Open the Database worksheet and load a set of data similar to that which is to be processed later. Take a sheet of paper and work out once manually all the computations which are to be used in the list later and write down the sequence as well as all the parameters of the functions used. If the sequence is correct and all computations fixed:

1. Call the first computation again and enter all parameters. Give confirmation, but not with "*Compute*", instead use "*Include in Auto Command List*".
2. Proceed in the same way with all other computations. In doing this, check whether all argument and results channels contain the correct channels and—if desired—the names and units of the results channels are given.
3. Now call up the Auto Command List via the "Measure" menu or the project window. In this list there are the defined computations and instructions in the script syntax in the "Command" column; "*After measurement period*" is entered in the "Execute" column. Click on the floppy disk symbol to save the list and quit the dialog.
4. When you now start a periodic measurement, the computations just defined will be executed after each period.

If the time of execution, "*After measurement period*", is not correct for your application, there is a choice of others, see Section 2.2, *Time of execution for the commands*, page J-5.

Example

You want to filter the data after measurement since they are overlaid with heavy interference.

1. Open the Database worksheet.
2. Call the sequence "Mathematics → Smooth curves" from the window menu.
3. In order to affect the original signal as little as possible, we shall use "*Savitzky-Golay*" smoothing as a "*4th order filter*" with a "*window size*" of 64 (values). Input your argument channel, that is, the channel where the acquired data will be located later (2 in our exam-

ple), and the results channel, which must be a free Database channel (10 in this example). If you wish, you can assign the channel a name in the “*Name*” column.

4. Click on “*Include in Auto Command List*” and close the dialog.
5. Use the “Measure” menu to call the “Auto Command List”.
6. Click on the diskette symbol to save the list.

Instructions in the Auto Command List for this Example

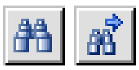
```
DB[10] = SAVGOL DB[2] 64 4          (After meas. period)
DB[10].Name = "Force smoothed"      (After meas. period)
```

This completes the Auto Command List. Create a post-process graph on your Online Page and as the data source, input the Database channel that will contain the computed values. If you then start a Periodic measurement using the Measurement Wizard, for example, you will see the smoothed data after each period.

4 Auto Command Lists for script programmers

With an Auto Command List catman® Professional can almost be controlled as with a script. To create a list you must open the Auto Command List via the “Measure” menu or the project window (double-click). Then insert the instructions either by direct entry in the relevant line or via the Code Builder on the left-hand side. After insertion via the Code Builder, you must replace the parameters quoted in angled brackets by values or variables. If you are not clear how to use a command or its parameters, just mark the desired keyword and press the key F1. You then obtain the list of available Help topics for the script command.

☞ You should already have some initial experience in using script, that is, you should already be familiar with the notation for commands, the *Object.Property* concept etc., before writing your own Auto Command Lists.



Use to search through the entries according to a key word. You can continue the search with . All text entries are searched, but not the command lines themselves. Just enter the beginning of a word to search for (without any wildcard characters) and catman will find all matching words.

4.1 Restrictions for Auto Command Lists

Unfortunately not all the possibilities of a script are provided in the Auto Command List.

The following restrictions apply for Auto Command Lists:

- Variables can only be declared with the instruction `Variable`. Up to 100 variables can be declared.
- The script, in the context of which the list is executed, has no access to these variables.
- An Auto Command List has no access to the variables of a script.
- The use and declaration of fields is not permissible.
- The use and declaration of constants with `#DEFINE` is not permissible.

- The use of flow control instructions such as `SELECT CASE, CALL, RUN, GOTO` etc. is not permissible.
- Blocks like `DO...LOOP` or `IF...ENDIF` must be completely of the same execution type.
- Each command must be placed in a separate row, i.e. using `@` is not allowed here.

The following are permissible:

- `DO...LOOP`
- `FOR...NEXT`
- `BREAK`
- `IF...ENDIF`

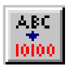
Here, nesting is also allowed as in a normal script.

4.2 Loading, compiling and executing Auto Command Lists

Loading Auto Command Lists via a script command

An Auto Command List can be loaded with `APP.LoadAutoCommandList <Filename>`. The list is translated and activated if the translation was free of errors. You can load and activate a new list at any time.

Compiling (translating) the Auto Command List



If the list has not been created by accepting computations from a mathematics dialog, it must still be translated for catman® Professional (compiled). When you click on "*OK*", the list is compiled and the window closed. If you just want to compile the list, select  from the toolbar. If errors are found during the translation, an error message is displayed and the relevant line marked in red.

☞ The output of warnings can be deactivated via the menu “Tools → Options” in the Script editor on the tab “Compiler”.

Executing the Auto Command List

Sequences in the current Auto Command List are executed with the command `APP.Execute P1`. The parameter P1 gives here the execution time-point at which the command lines are to be processed.

In the scripts of the catModules or of the Measurement Wizard (MEASWIZ.SCT in catman’s sub-directory CATMOD) commands for the execution of the Auto Command List are also present, showing you how they are used.

4.3 Examples of Auto Command Lists

Example 1

After each measurement this Auto Command List computes the maximum of Database channel 2 and displays the computed value in the Digital indicator o_Maxtemp on the Online Page. The instructions can be produced by the Code Builder.

Produced by the Code Builder:

```
VARIABLE <Name>                          At module start
<Variable> = DB[<Chan. no.>].Max         After measurement period
<Objectname>.Value = <Value>             After measurement period
```

After replacing the place holders:

```
VARIABLE MaxTemp                         At module start
MaxTemp = DB[2].Max                      After measurement period
o_Maxtemp.Value = MaxTemp                After measurement period
```

Example 2

This Auto Command List automatically transfers measurement data from Database channel 2 into a new column of an MS Excel worksheet after each measurement. The instructions can be produced by the Code Builder.

Produced by the Code Builder:

```
VARIABLE <Name>                       At module start
EXCEL.Open <Filename *.XLS>           At module start
EXCEL.WORKSHEETS.Activate <Value>     After measurement period
INC <Variable>                        After measurement period
EXCEL.DBWrite <DB chan.> <Column>     After measurement period
EXCEL.Save                            At module end
EXCEL.Close                           At module end
```

After replacing the place holders:

```
VARIABLE Col                          At module start
EXCEL.Open "TestData.xls"             At module start
EXCEL.WORKSHEETS.Activate 1           After measurement period
INC Col                               After measurement period
EXCEL.DBWrite 2 Col                   After measurement period
EXCEL.Save                            At module end
EXCEL.Close                           At module end
```

Example 3

On pressing F9 this Auto Command List saves measurement data from Database channels 10 and 11 in a file. The file is created *anew* each time with the file name consisting of the basic name TESTDATA followed by a three-figure number and the file extension *.DAT.

```
VARIABLE OrgFileName FileCounter                    At module start
VARIABLE FullFileName FileCountString               At module start
OrgFileName = "TestData"                            At module start
EXPORT.Format = 0                                   At module start
DB[10].Export = 1                                   At module start
DB[11].Export = 1                                   At module start
INC FileCounter                                     Manual (F9 key)
CVSTR FileCountString FileCounter "000"             Manual (F9 key)
STRCAT FullFileName OrgFileName FileCountString ".dat"  M. (F9 key)
EXPORT.FileName = FullFileName                      Manual (F9 key)
EXPORT.Execute                                      Manual (F9 key)
```

If the export formatting and marking can be overwritten by other program sections, then the lines with `EXPORT.Format` and `DB[x].Export` must also be set to "*Manual (F9 key)*" and also the export markers for other channels may have to be deleted.

Example 4

This Auto Command List saves the time, displacement and force values after each period for which the maximum has appeared in the force channel. The time is measured in channel 1, the force in channel 2 and the displacement in channel 3. The data will be written in a binary catman file. The number of periods is only restricted by the data depth of the results channels.

```
VARIABLE M1 M2 M3 M4 M5                    At module start
DB[5].Export = 1                           At module start
DB[6].Export = 1                           At module start
DB[7].Export = 1                           At module start
M5 = 1                                     At module start
EXPORT.Filename = "C:\Data.bin"            At module start
EXPORT.Format = 1                          At module start
M2 = DB[2].Max                             After measurement period
M4 = DB[2].MaxIndex                        After measurement period
M1 = DB[1].ROW[M4].Value                   After measurement period
M3 = DB[3].ROW[M4].Value                   After measurement period
DB[5].ROW[M5].Value = M1                   After measurement period
DB[6].ROW[M5].Value = M2                   After measurement period
DB[7].ROW[M5].Value = M3                   After measurement period
INC M5                                     After measurement period
EXPORT.Execute                             After measurement period
```

4.4 Using other editors

Auto Command List files are pure ASCII text files which can also be changed or created from new using any text editor. A valid file must contain the entry CATMAN AUTO COMMAND

LIST in the first line (one space between each of the words). The other lines contain the instructions in the form:

```
Execution time-point @ Command
```

Example

```
CATMAN AUTO COMMAND LIST
2 @ DB[10] = DIGFILT[DB2] 1000 0 8 40
```

This list uses a digital filter on channel 2 after each measurement period (execution time-point = 2). The filter is computed for a 1kHz sample rate and a Butterworth low-pass filter of the 8th order with f_c = 40Hz is used. The result is saved in Database channel 10.

K The Script Development System

1 Introduction

1.1 Overview

One of catman's special properties is that you can customize measurement runs and test runs, so as to set up a predetermined response to events. It is possible to control systems with analog or digital signals, to query operating states, to call a totally different script and run it or even to divert just the outputs to other Online Pages or Documents. But first you have to learn how to deal with the catman script language. It is useful, but not essential to have some programming experience. However, it is a condition that you already know your way around catman, that is, that you can operate catman® Professional in Level 2.

In keeping with the different levels of previous experience, we have formulated some questions, the answers to which will show you how to proceed.

You have *never written a program*? You have two possible options:

1. Work through all the sections, take a look at the examples and maybe some catModule scripts too, and you will see that creating scripts is not that hard. If you take note how a program is formed and structured, then with some practice you will be creating your own (small) scripts in just a few days. However, it is helpful if a colleague with programing experience is available and able to support you, especially if "logic errors" occur in your program code.
2. Speak to us: the HBM system house can create customized scripts to match your requirements. This solution is recommended if you do not have the time to learn all about programming, or the task is so extensive that it would tie up too much of your capacity. Our specialists will gladly help you turn your ideas into reality.

You have *already used Basic* or maybe an even more advanced programming language? Read the following sections carefully and you will see that script syntax is very similar to that of Basic. It is easy to learn how to use the catman script language, and with the aid of Code Builder and the context-sensitive online Help, you will soon be developing scripts of your own.

Although you have no knowledge of the catman script language, you have some *experience of programming*? Take a quick look at Section 2, *Main principles of script creation*, to see the concept of programming with catman, then read Section 3, *The elements of the catman script language*, Section 4, *Editing, compiling and linking script modules*, Section 5, *Help on creat-*

ing scripts and Section 7, *Restrictions and limitations*, and in the case of Section 6, *Special topics*, the subjects of your choice.

You are familiar with an *earlier version of* catman (except version 1.x) and have some knowledge of script programming? Check Section 2, *Main principles of script creation*, Section 4, *Editing, compiling and linking script modules* and Section 5, *Help on creating scripts* in order to get to know the options and the changes in this version. The online Help also contains more information about new functions. If you last worked with version 1.x, you should work your way through all sections, as a great many changes have been made since this version. catman contains a conversion program for 1.x versions, although you should consider whether it would be more effective to rewrite the code, with all the options of the current version.

All catModules are freely accessible and can be opened. Because they contain detailed comments, they can also be used as sample scripts.

HBM also provides seminars on working with catman and as an introduction to script creation. Speak to us or our representatives if you have special requirements, need an onsite seminar, etc.

1.2 What's new in Version 5.0?

See also “Worksheet Script editor → Help → Help on active worksheet → Version history → What’s new in catman 5.0?”.

Changes which make adjustments to old scripts necessary:

- The property `ACQ.MaxDynReadBlock`, with which the maximum read-block size can be set for all sample rates, is omitted. Instead of this use

IODEVICE[].SetMaxDynReadBlock: Here you can specify a dedicated maximum read-block size for each sample rate (and for each device).

- The security functions, which have now been integrated into the newer versions of Outlook, prevent data being sent via the MAILSERVER object, as in previous versions of catman. Therefore the MAILSERVER object has been changed: two additional parameters and the SMTP service are required. It is now called up via CDO (Microsoft Collaboration Data Objects).
- From version 5.0, the Scientific graph has its own toolbar. However, the position data for the graph apply only for the graph itself, not for the (additional) toolbar. Use MyGraph.ShowToolBar = 0 to hide the toolbar or when positioning the graph, ensure you take into consideration the additional space required above the graph.
- The property .CellText of the Spreadsheet is omitted or rather has been renamed for reasons of consistency and is now named .CellNote.
- The decimal separator for a Spreadsheet .DecimalChar no longer applies to the whole Spreadsheet, but is rather cell related. Consequently, the syntax has also changed: ObjectName.ROW[x].COL[y].DecimalChar.

General new commands and possibilities

- The differentiation between upper and lower case notation is now generally omitted.
- The Code Builder now includes a function for finding commands.
- The TEST object for managing measurement data.
- Since the logging of the interface communication has been changed (cyclical buffer), there are new parameters for COMLOG.
- The EXCEL object has a number of properties for the formatting and color arrangement of cells.
- File handling has been improved in many objects: IMPORT.GetDataFileInfo, GETFOLDER can be opened with a certain folder. FASTFILECOPY copies asynchronously in the background and is therefore quicker than copying using the operating system.
- The DIALOG object enables access to additional internal catman dialog boxes.
- You can also address the system log via the script language: APP.ErrorLogging.
- The drivers for the IEEE and fieldbus cards are now dynamically loaded.

- The synchronization of devices from other manufacturers has been improved. Generally, we recommend interfacing via DLL Drivers instead of via the catman Script Drivers. Although the latter continue to be supported, the new functions (time synchronization) are only available for DLL Drivers (Section 6.13, *DLL Drivers for "unknown devices"*, page K-84).
- Due to a modified calling procedure, the start of non-synchronized MGCplus systems now occurs almost simultaneously (< 1millisecond).
- Reading data via TCP/IP now takes place asynchronously so that with a number of devices the speed is increased.

I/O Definition

- The newly available traceability data (e.g. CANHEAD) are also accessible via the `IOCHAN` object.
- Channels can now be inserted and deleted script-controlled: `APP.InsertIOChannel` and `APP.RemoveIOChannel`.

Database

- The new options for creating the database can be found with `DBCREATE`.
- The newly available traceability data are also accessible via the script language `DBCHAN.CLCFactor`, `DBCHAN.SerialNumber`.
- The Channel manager worksheet can be separately refreshed to immediately reflect changes made, e.g. also changes made in the I/O Definition, even if the worksheet is still open: `DBREFRESH`.
- New mathematical functions: Elimination of outliers (freak values) with `ELIMINATE`, `DIGFILT` now includes a low-pass filter with Bessel characteristic and several FFT computations for a data record are possible with the STFT power spectrum `AUTOPOWERSPECTRUM_EX`.

Online Document

- The Scientific graphs have some new settings, e.g. a dedicated toolbar.
- The new version of the Spreadsheet object offers new options: Insertion and deletion of rows or columns. In order to be able to control the time required for refreshing with exten-

sive Spreadsheets, `.NoAutoupdate` and `UPDATE -t` (only refresh Spreadsheet objects) have been introduced.

Where necessary, appropriate commands have been added for the changes made in other worksheets which have been discussed already in the other parts of this Operating Manual.

1.3 What was new in catman® Professional 4.5?

Changes which make adjustments to old scripts necessary:

- The plot methods of Scientific graphs have been changed to "*Plotting method*". This also requires a change in the script language. As the property does not now refer to a scaling layer but to a curve, the sub-object has now also changed. The new parameters are different too. If you have used the property `.AXIS[].PlotMethod` or `.AXIS[].PlotMethodII` in your scripts, you must adjust this: `.CURVE[].PlotMethod`.
- If you previously used negative user scaling in catman, the trigger mode was "turned around": this must be set to "*smaller or equal to*" for triggering when values were exceeded and vice versa. If you corrected this by reversing the trigger mode selected by the user in your code, you must remove this. The error has been fixed in catman 4.5.

☞ If you wish to use scripts which have been created with a version higher than 2.0, these must be recompiled for the current version. Check whether commands of the type OnlineObject`.Name` (before 3.0) have been used. These must be changed to OnlineObject`.Caption`. You should also load and re-save the Online Documents and I/O Definitions used in the current version. This will ensure that all necessary conversions are made. If you cannot load an I/O Definition, check the settings for the maximum number of channels. Database configuration files (*.DCF) must generally be created again.

General new commands and possibilities

- A new variable type, the List variable, has been introduced. You can use it to save not only texts but also texts and numbers in one variable. This allows you to overcome the limitations of field variables which do not allow text to be saved. The number of elements of the variables can even be increased during run-time without loss of any of the contents.
- Influencing the CPU load: If catman is in a wait loop (`DO...LOOP`), the windows task manager will show the CPU load as 100%. However this means only that catman has requested all available CPU time. As soon as another process with the same priority requests CPU time, this will be assigned. You can however also lower the CPU load with `APP.Sleep`.
- The location of the directory of the script currently being run can now be queried with `APP.Script`.
- You can now also request the version number and the type of the catman program which is running with `APP.GetVersionInfo`.
- catman will first save up to 4,096 values per channel in a RAM cache. If there are less values from your data acquisition, these will be processed entirely in the RAM storage. The disadvantage here is that if catman crashes, these values are not saved in the Database file. You can save this data on the hard disk using `DBCHAN[].Flush`.
- Channel names can now be requested if the index is known: `APP.GetChannelFromName`.
- With the new log functionality the object `LOGMGR` is available for own scripts. The log functionality depends on catman's log file and you can use it to create and manage your own log files.
- catman can be set up as a Web server which, for example, will use current measurement results to refresh a webpage. Measured data can be displayed as numbers or graphically in a browser.
- Calling VBScript has been improved and the interaction with VBScript has been extended, see Section 6.16, *Using VBScript*, page K-94.
- With the new possibilities of the MGCplus amplifier, catman supports all parameter sets including reading data to and from the EEPROM: `IOCHAN[].ReadXDD` and `IOCHAN[].WriteXDD`.

Commands for the new functions in catman® Professional 4.5

- Some commands have been introduced for event monitoring: `ACQ.AutoEventChecking`, `ACQ.AddEvent`, `ACQ.CheckEvents`, `ACQ.ImportEventMonitor`, `ACQ.SetEventProperty` and `ACQ.ShowEventMonitor`.
- The object `DESCRIPTORS` has been introduced for the I/O Definition description files.
- You can also use derivative (`deriv`), Integration (`integral`) and `IF` queries in formulae for real-time computations in scripts.
- There are numerous new commands to use the Sensor Database and transducer identification (T-ID) with the script language. We recommend you use only the new properties `IOCHAN[].Sensor_xxx` in all scripts. You should also convert older commands to the new commands to call the functions. You can find the new functions with the object `IOCHAN[].Sensor_xxx` and `IOCHAN[].ConnectSensorFromDatabase`, `IOCHAN[].ConnectSensorFromTID`, `IOCHAN[].CheckTID`, `IOCHAN[].CheckTIDOnInit`, `IOCHAN[].DisconnectFromSensor`, `IOCHAN[].GetSensorInfo`. See also `APP.SensorDatabase`.
- There are new possibilities in the I/O-Definition to set up the filter cut-off frequency. These can be found under `IOCHAN[].AutoFilter`, `IOCHAN[].Sensor_FiltChar` and `IOCHAN[].Sensor_FiltFreq`.
- You can set, or remove, write protection for individual Database channels using `DBCHAN[].WriteProtect`.
- `CONVERSAMPLERATE` re-calculates measured values which were collected using a certain sample rate. You can convert to the required number of values for comparing them with data collected using a different sample rate.
- The new possibilities to compress data and speed up the display of Scientific graphs are also accessible from the script language. Use `.RTCompression`, `.AutoCompress` and `.CompressLimit`.
- You can display segments in Scientific graphs using script commands `.XSectionStart` and `.XSectionEnd`.
- `.FillFromDatabase` is used to display existing data in real-time graphs when changing to a different Online Page.
- The object TreeView may now also contain check boxes.

1.4 What was new in catman® Professional 4.0?

The most important change in catman 4.0 concerns the Online Pages: the user can change their size at any time. It is also possible to move them in order to see other applications which are running at the same time. If you do not wish to use this facility, there are some new commands which allow you to suppress these possibilities so that the program will function as in previous versions: `APP.MinimizeBox, APP.MaximizeBox, APP.SysMenu, APP.Sizeable.`

The second big change concerns the zeroing function (see also Chapter D, Section 4.1, *Zeroing channels (zero adjustment)*, page D-22). This standardizes the earlier zero adjustment and tare functions. They only worked as expected if either no user scaling was used or the characteristic curve passed through the zero point. catman can now also prevent any zero adjustment via catman's high-level commands on a per channel basis. With `IOCHAN[].Zero` and the intelligent zeroing (`SoftwareZero` and `AutoZero`) all device-internal possibilities as well as catman's own possibilities can be used.

Other new features

- The Video replay can be synchronized with post-process graphs (Scientific graph and Cursor graph) and can be script-controlled.
- The device setup can be script-controlled saved in a file with `<Devicename>.SaveSetup <DateiName>`.
- Channel information (previously: channel header) can be deleted and copied to other channels.
- New statistics functions for Database channels: `.Sum`, `.Median`, `.RMS`, `.Variance`, `.Moment`.
- When exporting, you can now specify the values (rows) to be exported: tagged export. You use a marker to define in a separate channel whether the row should be exported or not. As an alternative you may enter start and end rows.
- You can also import data using the channel name (up to now only via the number).
- New mathematical functions: `BITEXTRACT`, real inverse FFT
- `STRREPLACE` improves the text processing capabilites.
- Additional commands for the object `CANBUS`.

- The ActiveX interface introduced in the last version has been extended. Concerning ActiveX a special Help is available. Use the “Start” menu of Windows and menu item “ActiveX reference” in the catman program group to call up this Help.

1.5 What was new in catman 3.1?

- common variables for all code modules can be defined with `COMMON`.
- Improved handling of events with the event queue: when checking for an event, the value is not written into the system variable `Event` but instead into a separate variable which has to be specified in the command.
- Major enhancements have been made to the `TIMER` object.
- The Traceability data now contains also the catman user scaling. The scaling can be read out with `IOCHAN[].ScalePoints`.
- Functions for generating and sending an e-mail.
- A script can be canceled at run-time with CTRL-B. The script then enters TRACE mode and the debugging tool is shown.
- New: Popup menus.
- New features for the `DLG` object.
- Master/slave mode in MGCplus can be changed via script, the minimum pre-trigger value can be adjusted.
- New functions: `SPLITPATH, GETFOLDER, ShellExecute, APP.GetDiskInfo`
- New parameters for command `DIALOG`.
- New parameters for command `MESSAGE`.

A number of commands in the catman script language, which so far had been retained for the sake of compatibility with older versions, are no longer allowed with immediate effect.

2 Main principles of script creation

In the first section, we show you the working environment of the Script development system. This is followed by a general description of the procedure for creating a script, which is then put into concrete form in the subsequent section. The example uses all the available features of the Script development system and can be extended for your own requirements.

You can find the scripts discussed here with the associated Online Pages in the catman directory on your hard disk in the EXAMPLES\MANUAL sub-directory.

2.1 The Script editor

☞ Your access rights for creating, editing or running a script may have been withdrawn via the user administration.

The script development system is called by using menu “Worksheet → Script editor” or via the Project window. The worksheet window is split into two: on the left-hand side there is an *Administration window* and on the right-hand side the *Script editor*. An executable script may consist of one or more code modules. The development environment manages these modules in the form of a project in the usual display for directory structures. Only one project at a time can be edited. In addition to the modules, the administration window also displays all the subroutines (SUBs) of each particular module. The “File” menu and the context menu which you access with the right mouse button in the administration window provide various project management functions.

☞ Some functions, for example “Remove module”, are only available if an appropriate object is also selected-in this example, of course, a module.

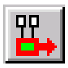
At the beginning, Unnamed.SCP with module MODUL1.SC containing the *MAIN* program part is shown.

We recommend to change the default name MODUL1.SCT as soon as you start writing your own code.

You can (and will) assemble a script not just from one code module (`MAIN`), but you will also define *several* subroutines (`SUB`), as this will make the sequence far clearer. To enable you to reuse these subroutines in other programs, there is the new structure with *start module* `MAIN`, to which other code modules (programs) with subroutines can be added as required. Individual subroutines can also be saved as a separate file, which can then be added to other modules. Of course, only *one* module can ever be the *start module*, but this can then be used by all the subroutines of the other modules. The start module is the only one which can contain *commands* (code), all the other modules can only contain the *declaration part* (variables etc.). The Script project is used to bring together all the modules and subroutines used in a script.

By using the “File” menu or using the context menu in the column with the Administration window, you can add, delete, compile or print code modules, create, save new workspaces, etc.

☞ Unlike older versions, it is no longer enough to compile a script: since different modules can belong together, these not only have to be compiled, they also have to be *joined together.* Therefore, your final transaction for all the individual modules which have already been compiled is “Make executable file” (Script editor worksheet, menu item in “Compiler” menu). This transaction is also called *linking*.



Instead of compiling each module individually, you can also call the linking process directly by using the “Compiler → Make executable file” menu or . All modules which are not yet compiled are then compiled automatically.

2.2 Procedure

Creating a script is not difficult: catman makes the job easier for you with the Code Builder and an extensive context-sensitive Help. However, before we show you how to create a program, we would like to give you some more information about the script language: the basic structure of a script, the method used for input and output to the Online Documents, the command notation and how to proceed when creating a script.

2.2.1 The structure of a script

A script basically comprises three parts:

1. Initialization.
2. An endless loop, during which the occurance of all possible events is checked.
3. A subroutine for each event, to be executed when the event occurs.

Possible events can be user actions or internal conditions such as wait time expired, limit exceeded by measured value etc. All events are normally managed using the “Tools → Event constants” menu and if required, additional events can be defined there, even for catman internal purposes. Each event comprises a descriptive text, such as `START`, `STOP` or `TERMINATE_SCRIPT`, and a (unique) number, both together making up the *event constant*. These event constants are allocated to the individual objects when an Online Page is designed. When the specified action takes place, the numerical value defined at the same time as the text is assigned the *system variable* `Event` and stored into the *event queue*. In the wait loop, the script checks whether one of the possible events has occurred.

2.2.2 Input and output to Online Documents

The method used to input and output data such as text or numerical values, is an additional characteristic of catman. With some objects, common methods of address are used: real-time graphs or post-process graphs can also all be addressed using common commands, to display new measured values, for example. However, so that individual objects can also be specifically addressed, they must be given a unique *object name*.

Do not give the variables in a script exactly the same names as the Online Page objects, so that it is easier to distinguish them. You could, for example, put a letter in front of all the names: *g_DataTransferRate* for the *global* variable DataTransferRate, *o_DataTransferRate* for the *Online Page* object (Text box), etc.

So you can see that the design of a program is closely linked to the design of the associated Online Documents. Creating a script must therefore go hand in hand with designing Online Pages. In this context, the event constants to be used play a particularly important part in determining which reactions have to be triggered by each of the objects on the Online Page.

⚠ To view the defined events in the “Tools → Event constants” menu, load the Online Document concerned. As soon as you load a new page, old definitions will be overwritten.

2.2.3 The notation of commands

☞ The upper/lower case notation is not important for catman (from Version 5.0 onwards) and is only used to facilitate readability and to differentiate between variables and constants.

It was quite common in programming languages similar to basic, to use a separate command for each purpose. With versatile programs, this quickly leads to having to learn hundreds of commands. Which is why a different method is frequently used nowadays: the *Object.Property* notation. To start with, this needs a bit of getting used to, but it does make the work much easier, as many properties can be controlled by using similar commands and you do not need separate command sequences. Commands can also easily be combined to make groups, such as the commands which control program behavior (scripts), the APP object. The notation is also used to create macros for MS Word or MS Excel: `Selection.Find.Execute`.

Related to catman, this means, for example, that the property `Caption` (label) can be found both with `APP` (program output window) and `MENU` (menu bar) objects, as well as with the output objects `Label`, `Button` and `Checkbox`. The advantage for you: less script commands and shorter notations for expressions.

This method also works with several “sublevels”: with the instruction

```
MENU[1].ROW[2].Caption = "Save project"
```

the menu item for the second row in the first menu column is given the name “Save project” (as in the case of catman). For the second level, there are only a few terms which can be used with some objects: `ROW, COL` or `LAYER`. `FIELD` is only of interest to advanced programmers, it allows access to the individual fields of a data record in an external database. `ROW` is a frequently used object which allows access to elements that are arranged in rows, e.g. a row in a Database channel (`DBCHAN[Index].ROW[Index]`) or a read block (`IOCHAN[Index].ROW[Index]`). You use `LAYER[Index]` to address a specific scaling layer in a graph. There is also a third level for graphs or the Spreadsheet: `COL[Index]` and `AXIS[Index]`. For example, you can address a specific axis of y(x) Post-process graph with

```
ObjectName.LAYER[Index].AXIS[Index]
```

As you already know, data input and output from or to the Online Page uses the name of the particular object. Using the *Object.Property* notation, it is a simple matter to read or to set a property or the contents of an object, which, after all, is really only a property:

- `ObjectName.Value = VariableName` sets the object with the specified name to the value of the variable.
- `VariableName = ObjectName.Value` sets the variable to the value of the object with the specified name.

Other properties can also be addressed in the same way as the numerical contents (`.value`).

2.2.4 Writing script commands (code)

Even if it looks simple to create a program, you should never *just start* writing a script without previously drafting a rough outline. A simple program outline could look something like this.

1. First catman and the measuring devices must be initialized.
2. Then the program should wait for input from the user. When this happens, the relevant action, such as measurement, should be performed.
3. Data shall be visualized even during the measurement.

To create a program, call the Script editor by using the “Worksheet → Script editor” menu. To open the Code Builder, use the Script editor window menu “Tools → Code Builder”. Modify the position and window size of the Code Builder, so that you can see as many commands as possible, without restricting the working area of the Script editor window. To generate a command, click the plus sign in Code Builder to open a command group, open this as well, if necessary and double-click on the relevant action.

If you select a command in the Script editor (by double-clicking with the left mouse button) and then press F1, you get the list of Help topics available. The Help topics consist of comprehensive information—often with examples—about that particular command.

2.3 Event-driven scripts (example)

2.3.1 Preparation

We assume that you have already connected and configured a device. The device setup file and I/O Definition file are both used in this section.

In addition, you should have available an Online Document containing output objects in the form of one or more real-time graphs and four Buttons: “*Start*”, “*Stop*”, “*Set sample rates*” and “*Quit*”. In this example the file ExampleScript1.OPG is used.

You can find the example discussed here with the associated Online Page on your hard disk in in the catman’s EXAMPLES\MANUAL sub-directory: ExampleScript1.SCT.

Start by drafting an outline of the task (Fig. 18 on page K-18).

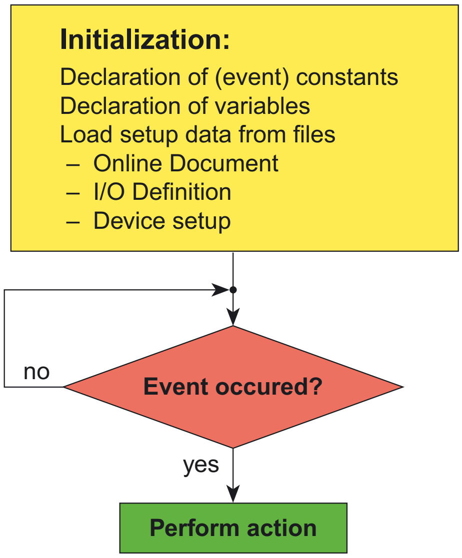


Figure 18: Flowchart of the planned script

2.3.2 Writing the script

To create the script, open the Script editor worksheet and proceed as follows:

Step 1: First catman and the devices have to be initialized. To initialize catman, we first need the definition of the event constants, so that during compilation, we can replace the texts with the numerical values. The required definitions can be created quickly using the “Tools → Event constants” menu. If you are later going to use your own, customized event constants, you should first load the relevant Online Documents, as the defined constants are also saved with them. In our example, we are only using constants which have already been defined and are designated as *system constants*. Create the declaration for `START, STOP, SETTINGSPOPUP` (for the “*Measurement settings*” `Button`) and `QUIT`: Select the constants and click on “*Create constant declaration*”.

Step 2: The following commands are created using the Code Builder, therefore open the Code Builder via the “Tools” menu.

Step 3: In the next step, you specify the variables, if necessary. In catman there are *four sorts* of variables:

- Common variables (`COMMON`). These are known in *all* code modules of a script project.
- Global variables (`GLOBAL`). These are known in the particular code module and in all its subroutines. They must be declared in the code module, as declarations are not allowed in a `SUB` program.
- Local variables (`LOCAL`). These are only known in the particular subroutine and have to be specified there before they can be used. Local variables cannot be used in a main module or a code module. If a local variable has the same name as a global one, the local variable will be used.
- Transfer variables (`PARAM`). These are used to transfer values to subroutines. Instead of using global variables, it is better to use this form, as global variables are only known to the subroutines of the *same* code module. As soon as you call a subroutine of a different code module, only those global variables defined in the `MAIN` section there are known.

In our simple first example we need only one global variable to request the actual event: `EvtVar`. Create this variable with the Code Builder and “Constants, variables and arrays → Variables”.

☞ All the parameters you define with the Code Builder appear in *angled brackets*. When you replace them, you must not forget to *delete* the angled brackets as well!

Step 4: There is not much more to do to initialize catman: load the Online Document to obtain our user interface. In the Code Builder, use the “*Online Document*” category, then item “*Open Online Document*” under “*General*”. One parameter is missing from the command line which appears in the Script editor: the file name.

Unfortunately it is not quite so simple to specify file names, especially with the full path. However, you have several options, you can:

- Not specify a path. catman will then browse through specific directories.
- Use the “File browser” tab in the Code Builder window.

Click on the “File browser” tab and use “*Add file*” to find the relevant files. Then, in the Script editor window, select the parameter to be replaced (including the angled bracket), select the file in the file list of the File browser and click on “*Insert file name*”.

catman browses through directories in the following order:

1. the directory where the script currently being executed is located.
2. the catman’s CATMOD sub-directory.
3. the catman’s SCRIPT sub-directory.
4. the directory where catman.exe is located
5. the currently active directory of the operating system.

☞ Some objects, e.g. object `FILE` for low-level accesses, do not follow this pattern (see also Section 6.8, *Specifying path names*, page K-76) or search just in the current directory, if no path is specified.

Step 5: We can now concentrate on the devices. First catman has to know which device or devices are to be used, then these devices can be set up. In Code Builder, choose the “*I/O Definition: devices (instruments) and I/O channels*” group. In the “*Files and setups*” category, you can find “*Load I/O Definition*”. Create the commands for loading the I/O Definition and the device setup file: “*Load device setup*” and “*Load device setup (use device index)*”. Then enter your files in place of the expressions in angled brackets. With `IODevice[1]` the first defined device is addressed. In this way you save having to give the complete device name.

☞ `IODEVICE[Index].DeviceType` even lets you query the type of a device.

Step 6: This completes the initialization phase and we come to part 2 of the script: the *wait loop*, which responds to user input. Insert the module “*Loop with event queue checking*” from “*Event handling (event constants)*”. The additionally inserted line `GLOBAL EvtVar` must be deleted as this line already exists in our script.

Step 7: Now only the event blocks need to be appropriately completed.

The command for the event *Quit* can be found in “*Program flow control → Quit and restart → Stop all script execution*”. This command has to be placed after the line with CASE EvtVar.

Step 8: Now we have to consider the button “*Measurement settings*”. The easiest way of setting these parameters is to call a catman-internal dialog. In this case catman can handle all connected settings as well, e.g. calculate the block size. The constant SETTINGSPOPUP shall call up this dialog. You will find the command for displaying a dialog in which you can set the sample rate in “*Popup dialogs → Show internal catman dialogs → Show dialog ‘Measurement settings’*”.

Step 9: Since the part after the START event will certainly be larger than just two or three lines, it is practicable to use a subroutine (SUB program) here: “*Program flow control → subroutines (SUBs) → Call subroutine (SUB)*”. This keeps the script clearly laid out.

Step 10: Finally, the subroutine with the actual measurement program, *Measure*, must be written. To do this, use “Insert new SUB program” via the menu “SUB”.

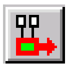
Step 11: Now, insert the macro “*Continuous acquisition with event checking*” via “*Devices and I/O channels → Simple data acquisition macros*”.

This module needs to be modified slightly for our purposes: The command ACQInit ALL should best be copied into the section MAIN after the loading of the setup files. This initialization must take place only once after a new I/O Definition is loaded. If the command is used outside the measurement loop, the data acquisition starts without delay.

Step 12: The interface <Timeout> can, for example, be set to 10,000 (milliseconds).

In the following code, the STOP event is already defined, the remaining CASE sections are not needed in our case and can therefore be deleted.

Each measurement cycle must contain the commands ACQStart—ACQRead—ACQStop in that order. A further ACQStart without an ACQStop leads to an error message, the only one which could be dispensed with is ACQRead.



This completes the script as all necessary commands are stated: save to Database (`ACQStore`), update real-time graphs (`UPDATE -r`) and “freeze” simple graphs (`CHARTREDRAW`). You can now create an executable file. As a check is made, to see whether the individual code modules have already been compiled in the current version, you can click immediately on  or use the “Compiler → Make executable file” menu.

ExampleScript (MAIN)

```
#DEFINE START 100
#DEFINE STOP 104
#DEFINE SETTINGSPOPUP 1060
#DEFINE QUIT 101

GLOBAL EvtVar

OPEN -o "ExampleScript1.opg"
OPEN -i "IO_Definition.iod"
IODEVICE[1].Setup "MGCplus_Setup.mgc"

ACQInit ALL

DO 0

  GETEVENT -1 EvtVar

  SELECT CASE

    CASE EvtVar = QUIT
      APP.Terminate

    CASE EvtVar = SETTINGSPOPUP
      DIALOG 8

    CASE EvtVar = START
      CALL Measure

  END SELECT
LOOP
```

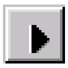
Subroutine (SUB) Measure

```
LOCAL BlockSize

ACQStart 0 0
BlockSize = ACQ.ReadBlock
```

```
DO 0
   ACQRead BlockSize 10000
   ACQStore
   UPDATE -r
   GETEVENT -1 EvtVar
   SELECT CASE
      CASE EvtVar = STOP
         BREAK
   END SELECT
LOOP
ACQStop
CHARTREDRAW
```

If you have created and entered everything properly, the Compiler output window will open and “Compilation terminated—Errors: 0” and “LINK: Executable file created” will be displayed. To close the window, use F4.



If the transducers and devices are connected correctly and you have *specified your files* for I/O Definition and device setup, the script will run immediately (”Compiler → Run script” or ), even if you connect more than one device or work with different data transfer rates.

2.3.3 Disadvantages and limitations of the example

Unfortunately there are some flaws with the code you have just generated:

1. Depending on your device, the initialization process can take a relatively long time. Even after device setup, you have to wait some time for the command `ACQInit ALL` to finish processing. But the user knows nothing about this, he only sees that nothing happens if he clicks on a Button immediately after script start. The initialization phase can last longer

than usual, if the measuring parameters of the MGCplus or UPM100/Centipede devices are saved to the Database channels.

2. All `Buttons` are always active, even if they have absolutely no effect. Our program is set up so that you can only exit the program or call the “Measurement settings” dialog after a `STOP` event. The (unfamiliar) user knows nothing about this, either.
3. There is no evaluation option.
4. The Online Page can be edited after measurement has stopped.
5. This is the worst shortcoming: this script only works for devices with internal time-base generation. Even if you specify “*PC-internal timing*” in the “Measurement settings”, the script does not work with the time set there.

2.3.4 Improvements

Listed below are the numbers from Section 2.3.3, *Disadvantages and limitations of the example*, accompanied by the suggested solutions.

1. To show the user what is happening, catman uses a status line which is located at the bottom of the output window. You can use the command `APP.Status` to display any text. However, do not forget to delete the messages afterwards (`APP.Status = “”`). The APP object gives you another option: changing the mouse pointer to the familiar hourglass. The command to do this is `APP.MousePointer = <parameter>`, the parameter for the hourglass is 11.
2. Deactivate objects which cannot be accessed. Use `ObjectName.Enabled = 0/1` for not active/active. This is not limited to buttons, it can also be used for many other objects, such as menu items, I/O channels, etc.
3. Either create your own toolbar or use the toolbar in catman with the command `APP.Toolbar = <parameter>`. Zoom, Cursor and/or Print are available here, for example; look up the correct parameters in the online Help.

An extended version of the toolbar can be configured using `APP.SetButton`.

4. Of course, you can exit catman straight away at the end of the script. Alternatively, you can load another Online Document or use the command `APP.OLClose = 0/1`. Parameter 0 closes the Online Document, 1 (default) leaves it open. The command should be placed at the beginning of the `MAIN` section before initialization so that you can see immediately how the program is to end.
5. The explanation for this point is somewhat longer and will therefore be addressed in detail below.

Devices with internal time-base generation

These are the devices MGCplus, DMC*plus*, DMC9012A, Spider8 and the NI cards. You can use any of the sample rates which can be set in the device. The script needs only to initiate measurements, to collect and display the recorded data (usually in read blocks). The sample rate and the number of measurements will either be included in the command, or determined individually for each device (the second parameter for `ACQStart` will then be 0). Make sure that the timeout value is not exceeded for sample rates with less than one measurement per second. At present however, only MGCplus can do this, otherwise you must create the time interval for less than one measurement per second in catman via the PC timer.

Devices without internal time-base generation

With these devices, the time interval must *always be generated* by catman. This includes primarily the scanning systems UPM60 and UPM100, as well as single-channel devices such as the MVD2555, Scout55 etc. There is an additional problem with the scanning systems in particular: Depending on the number of channels, the selected integration time and type of amplifiers chosen, these devices may need a relatively long time for data acquisition. This can mean that the cycle time required (all channels measured again) is not achieved. Although the device would be working as quickly as possible, it would only be able to transfer the values *after* the next cycle would actually be due. You should take this into consideration when programming and issue a warning.

☞ The sample rate specified for a device is also significant (except for MGCplus), if the sample rate is generated by catman internally. It determines the period of time required for the measurement (integration time). If you set the sample rate to 10 and the device supports internal time-base generation, the measured signal will be integrated over a time of 100 milliseconds.

Program code for slow measurement and devices without time-base generation

In both cases, timing must be created by catman. Only *one* measurement must ever be requested and collected (*one* measurement means that for *all* active channels of the device each *one value* is acquired).

The script for Example 2

```
#DEFINE START 100
#DEFINE STOP 104
#DEFINE SETTINGSPOPUP 1060
#DEFINE QUIT 101
#DEFINE TIMEREVENT 110
#DEFINE TRUE 1
#DEFINE FALSE 0
#DEFINE TOOLON 1000
#DEFINE HOURGLASS 11

GLOBAL EvtVar

OPEN -o "ExampleScript2.opg"
APP.Toolbar = 0
APP.MinimizeBox = 0
APP.MaximizeBox = 0
APP.Sizeable = 0
APP.SysMenu = 0
APP.Status = "Initializing devices"
OPEN -i "IO_Definition.iod"
IODEVICE[1].Setup "MGCplus_Setup.mgc"
APP.MousePointer = HOURGLASS
APP.Status = "Initializing I/O channels"
ACQInit ALL
CALL EnableButtons
APP.OLClose = 0
APP.Status = ""
APP.MousePointer = FALSE
```

```
DO 0
  GETEVENT -1 EvtVar                REM Request oldest event
  SELECT CASE
    CASE EvtVar = QUIT
      APP.Terminate
    CASE EvtVar = SETTINGSPOPUP
      DIALOG 8
    CASE EvtVar = TOOLON
      APP.Toolbar = 2
    CASE EvtVar = START
      CALL Measure2
  END SELECT
LOOP
```

Subroutine Measure2

```
LOCAL BlockSize TimeMode MInterval
LOCAL StatMsg test

CALL DisableButtons
TimeMode = ACQ.TimingMode

IF TimeMode = 1
  REM PC-internal timing

  MInterval = ACQ.TimerInterval
  CVSTR Meldung MInterval "####0.0#"
  STRCAT StatMsg "Measurement running, interval: " StatMsg " s"
  APP.Status = StatMsg
  CREATEOBJECT "Timer1" "TIMER"
  Timer1.Interval = MInterval*1000   REM Time in milliseconds
  Timer1.Event = TIMEREVENT
  Timer1.Enabled = TRUE
  ADDEVENT TIMEREVENT                REM The first time at once!
```

```
   DO 0
      GETEVENT -1 EvtVar                  REM Request oldest event
      SELECT CASE
         CASE EvtVar = STOP
            Timer1.Delete
            BREAK
         CASE EvtVar = TIMEREVENT
            ACQStart 1 0
            ACQRead 1 10000
            ACQStore
            ACQStop
            UPDATE ALL
            PEEKEVENT -1 EvtVar
            IF EvtVar = TIMEREVENT
               APP.Status = "ATTENTION: Interval too small or device too
slow."
            ELSE
               APP.Status = StatMsg
            ENDIF
      END SELECT
   LOOP
ELSE
   REM Device-internal timing
   test = STOP
   APP.Status = "Measurement running"
   ACQStart 0 0
   BlockSize = ACQ.ReadBlock
   DO 0
      ACQRead BlockSize 10000
      ACQStore
      UPDATE -r
      GETEVENT -3 test                    REM Check for STOP
      IF GP1 = 1
            CLEAREVENTS
            BREAK
      END IF
```

```
  LOOP
  ACQStop
ENDIF
:MeasEnd
CHARTREDRAW
CALL EnableButtons
CLEAREVENTS
APP.Status = ""
```

Subroutine EnableButtons

```
StopButton.Enabled = FALSE
MeasRateButton.Enabled = TRUE
ToolButton.Enabled = TRUE
QuitButton.Enabled = TRUE
StartButton.Enabled = TRUE
```

Subroutine DisableButtons

```
MeasRateButton.Enabled = FALSE
ToolButton.Enabled = FALSE
QuitButton.Enabled = FALSE
StartButton.Enabled = FALSE
StopButton.Enabled = TRUE
```

3 The elements of the catman script language

The following section explains the various elements that make up catman’s script language.

3.1 Modules and SUB programs (subroutines)

The start module

The start module is that part of the script which has a first line that starts the whole thing running. When the last line of the start module has been executed, the script terminates.

A script can never begin running in a subroutine.

Code modules

All code modules except the start module only contain the declaration part, there is *no* code. They are used to group related subroutines.

Apart from the start module, none of the modules can contain program code.

As all declarations only ever apply to one module and its subroutines, you can also transfer all the common declarations to a separate file for storage and use them in all modules with `#INCLUDE`.

SUB programs (subroutines)

Subroutines, called `SUB` programs in catman, are used to retain clarity in the start module or to combine identical command sequences, which are used at various points of a module, in a single code block. Then if the module needs to run this command sequence, just use `CALL <subroutine name> [<parameter>...]`. This method improves clarity and at the same time reduces the size of a module.

Usually, all the subroutines of a script (i.e. all subroutines of all object modules in the executable file) are loaded into RAM at the beginning of the module. Although this ensures that subroutines can be called very quickly (< 1ms), longer scripts do require a correspondingly large amount of RAM. On average, an executable line of script requires approx. 80 bytes, so a script with 1,000 lines, for example, would occupy approx. 80 kilobytes of RAM. You can use `#PRELOAD` and `#LOADBYCALL` to specify when a subroutine is to be loaded.

With `#LOADBYCALL`, the subroutine is loaded at the `CALL` instruction and deleted again when you exit the subroutine. This means that the overall memory requirement is never greater than:

Number of loaded (`#PRELOAD`) subroutines + largest subroutine to be reloaded.

The compiler directives `#PRELOAD` and `#LOADBYCALL` must be located within the subroutine they are to affect. Although the position is irrelevant, we recommend that for the sake of clarity, you insert the instructions at the beginning of the subroutine.

Be aware of the following details in connection with the memory management of subroutines:

- The start module is always loaded.
- It takes more time to reload a subroutine than to change to a subroutine which is already loaded. It is therefore better if subroutines used in loops where speed is critical or in call sequences, are loaded permanently.
- A subroutine with a `#LOADBYCALL` directive remains loaded (and thus quick to access), until another subroutine is called by a `#LOADBYCALL` command or you return to another subroutine of this type.

Large code blocks, executed in response to a user action, are particularly suitable reloadable subroutines.

☞ We strongly recommend the use of subroutines in the interest of the quality and clarity of your code.

If you already have experience in programming with other programming languages, you will already be familiar with the term subroutine (or function). However, catman® Professional subroutines do not always use the same subroutine methods as these rather more complex languages: The following applies:

- Local variables are possible, although they must be declared before they can be used.

- catman® Professional subroutines do not return a value, i.e. they cannot be used in the form `Variable = SubroutineName`.
- Some script commands are not permitted in subroutines.
- Recursive calls are not allowed.
- Arrays cannot be transferred.

To start a new subroutine, click on the code module for which you want to create a subroutine and then choose “Add new SUB” from the context menu. You will be asked for the name of the new subroutine and you will then be given a new page in the editor window. You can then write all the commands that you want to include in the subroutine. When you have finished creating the subroutine, simply click on `MAIN` in the left side of the script window, to bring the main program part into the display.

In this way you can create as many subroutines as you need. The names of all subroutines have an entry in the Administration window. Just clicking on an entry in the list fetches the corresponding subroutine for editing. To delete a subroutine, call it into the editor and then choose “SUB → Remove SUB program”.

☞ Every call of a subroutine uses up a certain amount of space in what is called the *Call Stack memory area*. Since this area is limited, the *CallTree* cannot be any length you please. If the call stack is full and you attempt to call another subroutine, you receive the run-time error message.

3.2 Objects, properties and methods

Starting with version 2.0 of catman *objects* together with their associated *methods* and *properties* were introduced. The main aim of this is to simplify notation and reduce the number of commands needed to change the characteristics of objects (contents of a Spreadsheet cell, line color of a plot etc.).

With a notation that is common to many objects and easy to remember, the properties of an object can be changed

```
Object.Property = ...
```

or queried

```
Variable = Object.Property
```

Methods differ from properties in that they are not only associated with a quantity, but exercise control over an object by defining one (or more) parameters:

```
Object.Method Parameter1 Parameter2 ...
```

3.2.1 Objects

In catman, objects can be of the most widely differing types, e.g. an I/O channel or a Database channel is an object, and so is a defined device or a graphic object on an Online Page. catman makes the distinction between *internal* and *named* objects: a named object is only *one* from a series of objects of the same type. So the name is necessary, to provide unique identification.

catman recognizes the following named objects:

- All objects in an Online Document, e.g. Text boxes, Bar indicators, etc. The object name in this case is the name that you can assign to an object in the configuration dialog.
- All defined devices. In this case the device name is the name you may specify when defining the device in the I/O Definition worksheet.
- All entries in the trigger list. The name of an entry in the trigger list (e.g. `TRIG_1`) is assigned in the “Edit trigger list” dialog (I/O Definition worksheet, menu “Measure”).
- Timer objects. The name has to be specified on creation.
- Popup menus. Also here the name has to be specified on creation.

In the case of internal objects, catman distinguishes between indexed and unindexed objects. An indexed object is a member of a collection of identical objects (e.g. a Database channel) and must therefore be identified unambiguously by a unique count index. The count index is enclosed between square brackets:

```
Object[Index].Property
```

Indexes can be numerical constants, e.g. 5, or expressions e.g. i+3.

Some objects contain other sub-objects. For instance a y(x) Post-process graph has different layers (`LAYER`), which can, in turn, have different axes (`AXIS`). In order to distinguish between them, there are additional *sub-objects* which are separated by a period or point:

```
Object[level 1].Sub-object[level 2].Property
```

Objects for level 2 and higher are always indexed objects.

Example: A row in a Database channel is a sub-object and must be identified by an index.

```
x = DBCHAN[5].ROW[237].VALUE
```

Objects for level 1

- Use the `ACQ` (ACQuisition) object to read out the data acquisition settings.
- The `APP` object (APPlication) controls the behavior of your script as a whole. For instance this object allows you to influence behavior in the event of an error, to define critical program sections without user interaction, etc.
- The `CANBUS` object supports low-level access to this bus.
- The `DAO` object (Data Access Object) makes a number of useful properties and methods available for accessing external databases.
- The `DBCHAN[Index]` object gives you access to individual Database channels and the channel information (acquisition parameters). You can also use `DB[Index]` as an abbreviation and `db(channelNo, rowNo)` for quick access to a specific value.
- The `DESCRIPTORS` object handles the description files of the I/O Definition (sensor description files).

- The DLG (DiaLoG) object enables you to use popup dialog boxes on the Online Page.
- When setting up Script Drivers, you can use the DRIVER object to access various settings such as read block, allocated I/O channel or the input buffer of a device.
- Use the ERROR object to access run-time error messages (number or text).
- The EXCEL object allows you to exchange data with MS Excel.
- The EXPORT object controls the data export from the catman Database.
- The FILE object gives you flexible access to any files.
- The IMPORT object controls the data import to the catman Database.
- The IOCHAN[Index] object gives you access to individual I/O channels, such as the data contained in a read block and the channel information (acquisition parameters) of the device connected. You can also use C[Index] as an abbreviation.
- The IODEVICE[Index] object gives you access to devices, without needing to know their name.
- The LOGMGR object enables you to create and use your own log files.
- The MAILSERVER object enables you to send e-mail from a script.
- Use the MENU object to affect the appearance of a user menu defined in the Online Document.
- The PAGE object gives you access to an Online Document page, for instance in order to run direct, graphic-related methods or to change the page.
- The PRINTER object makes mainly direct graphic-related methods available for the print page.
- The PROFIBUS object supports low-level access to this bus.
- The REGISTRY object enables you to access the Windows Registry.
- The SCRIPT object supports the transfer of data to programs written in VBScript or Java Script.
- The SERVER object enables you to provide data for other PCs.
- The TERMINAL object gives you an interface for direct communication with connected hardware by the user.
- The TEST object enables you to access the Test Manager.

Objects for level 2

- The `ROW[Index]` sub-object allows access to objects that are arranged in rows, e.g. a row in a Database channel (`DBCHAN[Index].ROW[RowIndex]`) or the individual bars in a Bar indicator.
- The `COL[Index]` sub-object allows access to objects that are arranged in columns, e.g. a column of a Spreadsheet on the Online Page.
- The `BUTTON[Index]` sub-object allows access to individual buttons of Toolbar and Radio button.
- The `NODE[Index]` sub-object allows access to individual nodes of the TreeView.
- The `POINTER[Index]` sub-object allows access to individual controls of a Bar indicator or slider control as well as a Analog meter or knob control.
- The `CURSOR[Index]` sub-object addresses a particular cursor in a graph. This sub-object is not available for Scientific graphs.
- The `ANNOTATION[Index]` sub-object addresses a particular annotation in a Scientific graph.
- The `AXIS[Index]` sub-object selects a scaling layer in a Scientific graph.
- The `LAYER[Index]` sub-object addresses a particular scaling layer in a standard graph.
- The `CURVE[Index]` sub-object addresses a particular plot in a graph.
- The `RECORDSET[Index]` sub-object allows you to access a particular data record in the `DAO` object.
- The `WORKSHEETS` sub-object is used to access a particular workbook in MS Excel.

Objects for level 3

- The `COL[Index]` sub-object can also appear as a level 3 sub-object. In this case it addresses a particular column in sub-objects that are arranged in the form of a matrix, e.g. the column of a Spreadsheet on the Online page.
- The `AXIS[Index]` sub-object allows you to position the axes of a scaling layer in a standard graph. Index 0 stands for the x-axis and index 1 stands for the y-axis.
- The `FIELD[Name]` sub-object allows you to access a particular data record field in the `DAO` object.

3.2.2 Properties

With properties you can modify the status of an object, e.g. its background color, the scaling of a y(x) Post-process graph, the text in a Text box, etc., or query the status of an object, such as the status of an On / off switch or Slider control.

Examples: Write “Tester” in the Text box Operator:
`Operator.Text = "Tester"`
Change the x scaling in layer 1 of the graph Chart:
`Chart.LAYER[1].Xmin = 4*x`
Change the x axis label in layer 1 of the graph Diagram:
`Diagram.LAYER[1].AXIS[0].Caption = "Force at the top"`
Set the second sub-menu entry in column 3 to "Evaluate":
`MENU[3].ROW[2].Caption = "Evaluate"`
Set program window label:
`APP.Caption = "TEST-RIG CONTROL V4.5|manual program"`

Even carrying out widely differing actions requires only a few terms, and in the cases to which they are applicable, they are always utilized in the same way.

Please note the following points when using properties:

- `Object.Property` must not be part of a numerical expression. The expression `r = Control.Value + 5` is not allowed.
- Some properties are *write-only properties*, i.e. you can change this type of property, but not query it. (e.g. `.BackColor` for the Digital indicator, but not for the Text box.) If you need the current value, you must store it in a variable.
- An `Object.Property` expression must only appear on one side of an equation. If you want to assign the property of one object to another, you have to use two expressions to do it.

Example: Assign the property of an object to another object
`Input = Text1.Text @ Text2.Text = Input`

3.2.3 Methods

Methods are mostly the same as normal script functions, merely in the same notation as for properties. However, there is no assignment to a variable or from a constant or variable. It is possible to define one or more parameters, depending on the object or method.

Method notation is as follows:

```
Object.Method P1 P2 P3...
```

In the above, P1, P2, etc., are parameters.

3.3 Assignments

An assignment in the catman script language is a mathematical equation containing variables or objects as arguments. Evaluation takes place during program run-time. The result appears on the left-hand side or is written in the variable on the left-hand side, whilst the expression to be computed appears on the right-hand side.

The expressions used in assignments are mathematical formulas containing variables or data channels as arguments. An expression may contain only the mathematical operators +, -, *, / (plus, minus, multiplication, division), brackets, and the basic mathematical functions (see next section). Special functions, such as `FFT`, `INTEGRATE` etc., cannot be used in an expression.

Examples:

```
y = 2*x + cos(2*PI*100*t)
SampleRate.Index = 2*i
DB[3] = DB[1]+DB[2]/2
ACQInit x+1
y = DBChan[c+1].Row[2*k].Value
ACQStart NofValues 2*BaseSampleRate
```

An expression is illegal:

- If part of the expression is a string.

- If it contains a property, e.g. `x = Button.Value + 1`. (First assign `Button.Value` to `x` and then increment `x` with `INC x`.)

Numbers in catman® Professional can only be written with a decimal *point*, a comma (German notation) is *not* allowed.

3.4 Basic mathematical functions and operators

The operators and functions listed below can be used within numerical expressions in script.

Operators

`AND, OR, <, >, =, <>, >=, <=, +, -, *, /, mod, ^, ( )`

If, for division (and for modulo), the divisor = 0, this will result in a run-time error.

Basic mathematical functions

```
abs(), acos(), asin(), atan(), atan2(), bit(), ceil(), cos() exp(),
floor(), hypot(), int(), ln(), log(), max(), min(), mod, rnd(),
sgn(), sin(), sqrt(), tan().
```

`atan2()` returns the tangent, taking the sign into account (i.e. the quadrant).

If the function argument for the logarithm or the root is less than or equal to zero, this will result in a run-time error.

With `floor()` and `ceil()`, the sign is taken into account.

3.5 System variables

The system variables allow access to internal catman data. These variables can be used without prior declaration. They are recognized in all code modules and subroutines.

You *must not* declare system variables. Normal variables declared with `COMMON, GLOBAL, LOCAL` or `PARAM` are not allowed to have the same name as a system variable.

Some system variables are read-only variables, that is, they cannot appear on the left-hand side of the assignment operator (=).

In older versions of catman the variable `Event`, which contains the number of the last event, was quite important. This variable must have write access in order to reset the variable after an event has been processed: `event = 0`. However we recommend to use always the event queue, see Section 6.5, *Event-driven scripts*, page K-66 and Section 2.3, *Event-driven scripts (example)*, page K-17.

A further important variable is `ERR` or the object `ERROR`, which contains the number or the number and text of the last run-time error. Among other things it allows you to set up your own error handling routine. To do this, you must first use `APP.ErrMsg = 1` to deactivate catman's own error handling routine, so that you can work with the error number in `ERROR.Value` or the error text in `ERROR.Text`.

A list of error numbers and their meanings can be found in the online Help.

The `KEYPRESS` variable contains the ASCII key code of the last key pressed by the user. This enables you to respond to a user action, without having to define a Button with a hotkey or a keyboard event in the Online Document.

The variables `GP1` to `GP32` are frequently used by mathematical functions to store results. They are also useful for transferring data between a parent script and a daughter script (`RUN` command).

The variable `DB_MAXLEN` has no importance any more. In versions prior to 3.0 this variable specified the maximum number of values (data depth) for a Database channel. The variable is read only.

☞ Starting with catman® Professional version 3.0 Database channels may have individual lengths. Use `DBCHAN[].MaxLen` for reading out the maximum data depth of a channel.

`DB_MAXCHAN` stores the number of Database channels. The variable is read only.

3.6 Event constants

You can use the “Tools → Event constants” menu to create name constants for any events. These constants must be declared in the relevant code module with the `#DEFINE` command, before they can be used. For the sake of clarity, you are recommended to place the declaration at the beginning of the script. catman’s compiler replaces this short form with actual values before starting to compile the code module. A constant comprises a name (e.g. `HIGH_SAMPLERATE`) and the value to be used (e.g. `9600`) or the code (e.g. `DELAY 5000`). You can even replace a single term with entire command sequences.

Examples:

```
#DEFINE HIGH_SAMPLING_RATE 9600
#DEFINE BREAK? IF Event = 200 @ BREAK @ ENDIF
```

☞ Please note:

- If you try to define a constant that has already been defined, or to assign it a value, a compilation error occurs.
- A constant definition may not include another constant.

If you include a computation in a constant declaration, the compiler will automatically perform it. This means it is the result, and not the original computation, that will be substituted. The expression 2*(17+4) will therefore be computed and replaced by 42. Please note that no spaces are used in this notation of the formula. The compiler would otherwise interpret this as a third parameter and ignore it.

3.7 Variables and constants

3.7.1 Variables

As opposed to other programming languages, the catman script language differentiates between only two types of variable: numeric, double precision floating point numbers and character strings, often just called strings.

Arrays and *Lists* are special types of variables. They make it possible to save several values under one single variable name, see Section 3.7.2, page K-43 and Section 3.7.3, page K-46.

Before you can use a variable, it has to be declared. A variable may be declared as

1. common (command `COMMON`), i.e. valid for all code modules of a script project
2. global (command `GLOBAL`), i.e. valid for the code module concerned
3. local (command `LOCAL`), i.e. valid for each subroutine individually.

☞ When declaring variables, nothing is stated yet about their type, this does not take place until during the run-time of your script using the assignment of a value to a variable.

⚠ In the case of the assignment of a string constant, this must be enclosed in quotes, e.g. `t = “Sample”.` The notation `t = Sample` would make the compiler interpret `Sample` as a variable name and usually lead to a compilation error (invalid expression).

⚠ The inclusion of a variable of the string type in an `IF` or `CASE` query, e.g. `IF name = “John”,` is not allowed since it would produce an incorrect evaluation of the query. To compare character strings, use the command `STRCMP` instead.

⚠ If a variable has taken on the type string, it must not be part of an expression; in this way, for instance, the code sequence `a = “Hello” @ b = “World” @ c = a + b` in c would not store the expected string “`Hello World”,` but an undefined number. To connect strings together, use the command `STRCAT`.

Sometimes a character is to be included in a string which cannot be entered via the keyboard (e.g. a carriage return, ASCII 13), or if the string itself is to include quotation marks. You can insert characters of this kind with the aid of the `STRCAT` command and special string constants containing a $ sign as the first character.

Example: Appending carriage return/line feed (CRLF) to the string *Output*

```
STRCAT Output Output "$13" "$10"
```

Useful special characters:

```
"$10"   Linefeed
"$13"   Carriage return
"$9"    Tabulator (TAB)
"$34"   Quotes
```

3.7.2 Arrays

Arrays are a special sort of variable, which allow you to save several values under the same variable name. Imagine, for example, a series of measured values, which have to be edited in your script. Instead of now defining a separate variable for each measured value (e.g. Value1, Value2, etc.), it would be better to store these values under a single name *MeasuredValue*, and to address the individual values through an index, such as MeasuredValue(1), MeasuredValue(2) etc. This is precisely what arrays allow you to do. In addition to saving space, arrays have the advantage of being able to address the individual elements with variables in loops, for example.

Declaring arrays

To create a variable of the array type, you have to use the `GLOBAL` command. The variable name also contains the number of array dimensions enclosed in brackets, as a parameter.

Examples: `GLOBAL v(1)` for a vector with the name v.

`GLOBAL M(2)` for a matrix with the name `M`.

`GLOBAL T(3)` for a three-dimensional array with the name `T`.

☞ Please note:

- Do not confuse the number of dimensions with the number of elements in a dimension.
- Arrays can only be created by global variables (`GLOBAL`).
- Array elements cannot contain strings.

Creating arrays

The array declaration by itself is not enough to allow you to work with the array. You also have to tell catman the maximum number of elements (i.e. numbers) to be saved per array dimension. This value is called the upper limit of the array dimension. catman needs this information so that it can reserve sufficient storage space for the array. Specify the upper limits by using the command `ARRAY_ALLOC` and specifying the individual dimension limits as parameters.

Example:

```
GLOBAL Matrix(2)
ARRAY_ALLOC Matrix 10 10
```

Storage space is allocated for a 10 x 10 matrix.

Freeing arrays

Arrays, once they have been created, use up storage space, so if an array is no longer required, you can free it: `ARRAY_FREE variable`.

An array that has been freed can be reallocated at any time with `ARRAY_ALLOC`. The upper memory limit can also be varied. With this mechanism, you can dynamically adapt the use you make of the storage space to meet the requirements of your script.

☞ At the end of the script, catman automatically frees all the memory areas which have been allocated.

Array elements

To use an individual array element in an instruction or a computation, the element has to be specified (addressed) explicitly. Use the following syntax to do this:

- `Variable(Index1)` for a one-dimensional array, e.g. `MeasuredValue(i - 1)`.
- `Variable(Index1, Index2)` for a two-dimensional array, e.g. `matrix(i + 1, j + 2)`.
- `Variable(Index1, Index2, Index3)` for a three-dimensional array, e.g. `tensor(i, j, k)`.

The index information here comprises any numerical expressions, e.g. `a(1), a(i + 1), matrix(i, j - 1)` etc.

When allocating the index, make sure that you do not exceed the upper limit of the associated dimension, otherwise a run-time error occurs.

☞ The first index of a dimension is 1.

Using array elements

An array element can be used wherever a numerical expression is allowed. It can also appear in certain methods and functions which write results in variables.

Examples:

```
a = m(i, j) + 5 * sin(x(100))
v(k) = v(k) * 2 * m(i, j)
MyText.Value = a(i)
a(i) = MyText.Value
Registry.Read "MGCplus" "MEASRATE" Measrate(i)
File[1].Write Value(j + 1)
File[1].Read Value(j + 1)
CVNUM a(k) MyString
STRCMP a(k) String1 String2
```

Limitations in the use of arrays

- Arrays can be allocated as global variables only
- Array elements cannot contain strings.

- The index expression of an array dimension must not itself be an array element, e.g. `a = Matrix(i, v(k))`
- The index expression of an array dimension must not contain a function, e. g. `a = Matrix(i,sin(x))`

3.7.3 Lists

Lists are variables possessing a number of elements. Therefore, they are an alternative to arrays, but have the following advantages:

- Lists can also include text passages.
- Lists can even have *mixed* string and numerical elements.
- Lists can be declared locally, i.e. in a subroutine (SUB program).
- Lists can be passed as parameters to subroutines (SUB programs).
- Lists can be reduced or enlarged without the content being lost.

The first element of a list has index 1.

The list assignment `.MakeList` converts each variable into a list. Through the assignment of a value, e.g. `ListVar = 0`, a list again becomes a normal variable.

Lists or list elements have the following disadvantages:

- You can only create unidimensional lists.
- Lists may not be used in expressions (for expressions, see Section 3.3, *Assignments*, page K-38).

Provided you do not need any multi-dimensional arrays, you can use lists as complete substitutes. In particular, lists can be enlarged as required without the content of the elements being lost.

List declaration

Lists can be created both in the form of global variables and also in subroutines as local variables: `GLOBAL ListVar1`. The following instructions are only permissible in a subroutine (SUB program)

```
PARAM ListVar2
LOCAL ListVar3
```

Creating lists

Lists are created with the method `.MakeList`. You state the required number of elements (entries) as parameter:

```
ListVar1.MakeList 20
ListVar2.MakeList 14
ListVar3.MakeList 200
```

The following method is used to change the number of entries: `ListVar1.ResizeList 150`.

Editing list elements

A number of commands are available for editing list elements (list entries):

- Write content of Parameter in element with index idx: `ListVar1.SetListItem idx Parameter`
- Read content of element with index idx in Variable: `ListVar1.GetListItem idx Variable`
- Determine number of elements: `Count = ListVar1.MaxItems`
- Copy content of List 1 to List 2: `ListVar1.CopyList ListVar2`

  ☞ The command is only present so that the commands of the list variables are consistent with those of the other objects. You can also copy one list to another list with a simple assignment: `ListVar2 = ListVar1`.

Releasing/deleting lists

To delete a list you assign any value to the list variable, e.g. 0: `ListVar1 = 0`. This frees memory occupied by the list.

3.7.4 Constants

catman only knows two mathematical constants: `Pi` and `e`. Use of upper and lower case letters for the constants is ignored, accuracy is 14 decimal places.

4 Editing, compiling and linking script modules

4.1 Editing modules

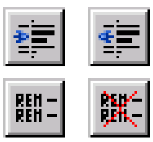
4.1.1 The Script editor

The Script editor corresponds to a conventional word processor, although it cannot use different fonts or formatting to format the script, simply different colors for the individual components of the script language. The type and size of the font is applied to all the code module lines. If you want to use a different editor or want to modify the display, e.g. the color coded syntax, please read the Section 4.1.2, *Customizing the editor*, page K-50.

When you create your script, that is, when you generate and edit it, the following rules apply:

- You can utilize Code Builder to insert a script command, or, if you know the command, you can input it directly from the keyboard.
- A line may contain up to 8 instructions, in which case they must be separated by an @ sign.
- A line may have a maximum of 255 characters.
- Most commands expect one or more parameters—these must be separated by blank characters (spaces), not by commas.
- The script does not distinguish between upper and lower case letters. It makes no difference whether keywords, commands, labels or variables are entered in capitals or lower case letters.

The editor provides you with context-sensitive Help for every script command. Just select the required command (by double-clicking) and press F1. The list of available Help topics concerning the command in question will appear.



With or CTRL-I, you can indent the selected lines, with or CTRL-O, you can outdent them again. It is enough to select a single character in each line. With or CTRL-K you can outcomment selected lines, i.e. the line contains the `REM` command at the beginning. Use or CTRL-J, to delete the `REM` again. It is enough to select a single character in each line.

In the line at the bottom of the Script editor window you can see, in which line you are positioned.

With effect from version 3.0, the editor can use the “Undo” (CTRL-Z) command (for one action).

4.1.2 Customizing the editor

Use the “Tools → Options” menu and the “Editor” tab to modify the background of the editor window, as well as the default font and the color-coding for the text. With “*Keyword always uppercase*” you can specify that catman keywords must always be written in upper case letters. The replacement is made as soon as you exit the line, e.g. press ENTER.

With “*Save code module automatically*”, you can specify whether and at what intervals you want catman to save your files automatically.

4.1.3 Using other editors

You can create a code module with any ASCII text editor you like. Please note the following, simple rules when using other editors:

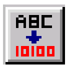
- The code for your main program (`MAIN`) must be located ahead of the first subroutine in the text file
- Each subroutine must be located between two lines: `_PROC_` and `_ENDPROC`
- The line following `_PROC_` must contain the name of the subroutine

- There must be no blank lines between `_PROC_` and the name
- You cannot create an executable file directly from an external editor—you have to load the code into the catman® Professional Script editor for compiling.

4.2 Compiling modules

Before you can run a script, it has to be translated (i.e. compiled). This is done using the “Compiler → Compile script” menu. Only a *raw* translation of the individual code modules is provided, however, and so-called *object modules* are created. If several modules are involved in the same script, all the object modules that have been produced during compilation ultimately have to be joined together to create an *executable file*. This creation of an executable file is also called *linking*.

☞ Since it is not enough simply to *compile* a code module, you can also start the linking process immediately. If there are still any uncompiled modules, these will first be compiled and then linked.



If you click on  in the toolbar, you can compile *just the current* code module. This is useful if changes have been made in *this* code module only (or in an associated `SUB` program). The big advantage of separate compilation is that any error messages which may arise come only from this module.

Before the compilation process begins, catman saves the particular code module as a text file. If the module does not have a name, you will be asked for a file name. During the compilation process, catman keeps you informed about errors in the *Compiler output window*. If there have been no errors, once the compilation procedure is complete, the message “Compilation terminated—Errors: 0” and some statistical information, such as the compilation time, the number of variables used, etc., is displayed. Specific errors, such as a device which has not been defined or an Online Page object which could not be found, simply generate a warning, an object module is still created.

If the compiler finds errors, you must track down the cause: a double-click on the error message takes you to the offending line. Take into account, however, that in some cases the error may even be in a previous line, for example, a forgotten bracket, etc.

Additional information on dealing with errors can be found in Section 5, *Help on creating scripts*, page K-54.

Use the “Tools → Options” menu, “Compiler” tab to suppress the output of warnings.

Once the module has been compiled without errors, the associated object module file is automatically generated and saved. This has the same name as the code module text file (a pure ASCII file), but instead of *.SCT, it has the file extension *.OBJ. If there is a compilation error, an object module file of this type is not generated; however, the text file is always saved regardless of any errors that may be present.

To close or reopen the Compiler output window, use F4 or “Compiler → Compiler output window”.

4.3 Creating executable files

The “Make executable file” command can be found in the “Compiler” menu. The Compiler output window shows that individual object modules are being linked: any already existing object modules are shown with “LINK: Object module x is up to date”, the successful completion of the process is indicated by “LINK: Executable file has been created”.

Instead of using the menu, you can also start generating an executable file with .

If individual code modules have not yet been compiled, this will automatically be done.

The successfully generated executable file has the same name as the start module and the file extension *.SCB. catman can execute a binary file of this type at any time, for example, by using the menu command “Execute script from file” in the “Measure” menu.

In the Script editor worksheet, you can start your script by using the “Compiler → Run script” menu or .

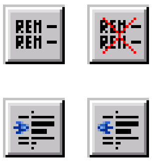
As the linker itself does not actually compile the code, any error messages or warnings which appear come from the compiler. If the compiler has not generated an object module, the linker is not even started, you simply see the message “Object module not created”.

You can even choose the “Run script” command, if your script contains non-compiled modifications in the meantime. With version 3.0 catman compiles modified modules automatically, starts the link process again (if necessary) and then starts the executable file.

5 Help on creating scripts

catman provides you with various forms of help when creating scripts.

1. The Code Builder, which lists numerous actions. Making a selection often downloads not just one command, but an entire command sequence to the editor. If you have already had some experience in dealing with the Code Builder, you may even be able to record some command sequences of your own.
2. catman's context-sensitive online Help. One way of using it is to select a command and press F1; the list of available Help topics is then displayed. The Help topics consist of comprehensive information and will often include an example. Another way is to call Help as a whole and browse for commands or functions (click on the "Find" tab).
3. The debugging tool is particularly useful if a script does not behave as it should: Use it to run the script step by step and look for errors.
4. Fourthly you can directly edit files being referenced by `#INCLUDE` statements in the Script editor: Place the cursor in a line with an `#INCLUDE` statement, open the context menu, and choose "Edit #INCLUDE". The contents of the included file is displayed in a separate window, where it can be edited and also saved.



There are other help facilities to make your work easier: the "Edit" menu and the toolbar both include the commands "Outcomment rows" () and "Undo outcomment" (). You can use these functions to deactivate (or reactivate) any part of your script that you have selected, whenever you want to test a change you have made. Two other useful commands are "Indent text" () and "Outdent text" (). These will let you quickly give your script a pleasing layout without having to use the tabulator key.

☞ The Script editor is opened and the line edited last in the code module opened last is shown, after each compile and test cycle.

5.1 The Code Builder

To open the Code Builder, go to the Script editor worksheet and select the “Tools → Code Builder” menu item. The default is “*Always on top*” activated.

Use to search through the entries according to a key word. You can continue the search with . All text entries are searched, but not the command lines themselves. Just enter the beginning of a word to search for (without any wildcard characters) and catman will find all matching words.

Inserting a code block

Choose a (command) group, open it and any necessary subgroups, by clicking on the plus sign to the left of the line, and double-click on the entry you require. The code block is inserted at the current mouse pointer position in the Script editor, and the mouse pointer is then positioned on the line following the block.

In general, script commands expect parameters (e.g. variables). At these points the Code Builder therefore contains *placeholders* between angled brackets. The placeholders, *including* the angled brackets, have to be replaced by relevant variables, expressions or constants.

File browser

You can use the file browser to insert file names with full path details into your script. For files with long path names, this saves typing and prevents errors.

First, gather all the file names you need from the “*File list*”: open the standard file dialog box by using the “*Add file*” button and select the required file. To insert a file name, position the mouse pointer at the required point in the script, or select the area to be replaced and double-click on the file in the “*File list*” (or select the file in the list and click on “*Insert file name*”).

Use the “*Remove file*” button to remove a selected file from the list.

Insert own code block

If you are missing a function in the Code Builder, it is easy to extend. It is simplest if you first create and check the code to be inserted in the editor window. Then copy it to the clipboard. In the Code Builder on the “New code block” tab, select either the group after which you want to insert a new group or the entry after which the new code block is to appear. You can use “*New group*” to create a group or subgroup yourself. Otherwise, just insert the block from the clipboard into the “*Code*” window, enter a description (”*Description for code block*”) and click on “*Add*”.

Removing a code block

Select the block from the list and click on the “*Delete*” button. You can only delete blocks one at a time—deleting a whole group is not possible on safety grounds.

Editing code blocks

This section is intended for experienced users only. It is essential to create a backup copy of the MACROS.TXT file before making any changes.

All Code Builder code blocks are stored in the MACROS.TXT file in your catman directory. This file can also be edited with a word processor.

The file structure is as follows:

- `FOLDER=<group name>` initializes a group.
- `BEGINMACRO <block name>` initializes a block.
- All lines after `BEGINMACRO` as far as the line `ENDMACRO` form the code block. They are then used in the Script editor exactly as they appear in the file (complete with blank lines, TAB etc.).
- `ENDMACRO`
- `ENDFOLDER`

If subgroups are to be set up in a group, `FOLDER = <subgroup_name>` starts a new subgroup and `ENDFOLDER` ends it. You can also specify more than one sublevel.

Example:

```
FOLDER=My macros

BEGINMACRO Quick measurement
ACQStart 16384 4800
ACQRead 1024 10000
REM Caution: ACQStore needed?
ACQStore
ACQStop
ENDMACRO

BEGINMACRO Slow measurement
ACQStart 1000 5
ACQRead 1 10000
```

```
ACQStore
ACQStop
ENDMACRO

ENDFOLDER
```

5.2 Script language online Help

The “Contents” tab of the Help appears whenever, in the Script editor, you use the “Help” menu to call “Help on active worksheet”. The online Help is divided into several sections.

The “*Reference*” contains the error codes, all the commands sorted alphabetically, the object reference, the system variables and the script restrictions. In the object reference, you can find all the information on:

- internal objects
- Online Document objects
- named objects, e.g. timer, trigger, devices etc.

Use the “Find” tab to browse through the entire Help file for keywords. If the Search functions of the standard version do not meet your needs, get the Windows Help system to create a new database (”*Re-create*” button) and activate all the options, even, for example, including similar topics.

☞ When a command is selected (double-click), F1 will immediately give you the list of Help topics available for this command.

We recommend that you also use the Help facility when you are searching for examples and related commands. With commands which contain parameters, it is advisable to check whether and how parameters have to be specified. Although, as far as possible, catman® Professional tries to handle all devices in the same way, in certain circumstances, a particular device may need different parameters or certain parameters may not work with some devices.

5.3 The debugging tool

Unfortunately, error-free creation of an executable file does not necessarily mean that the whole script will work exactly as you had intended. This phenomenon is well known among programmers—it is simply a matter of *false reasoning* when preparing the script, for example, assigning the wrong value to a variable or unintentionally calling a subroutine. Here, too, catman helps you to locate and solve the problem. This particular help feature is known as a *debugging tool*, and enables you to run the script under controlled conditions so that you can see what it does.

The debugging tool offers you two facilities:

1. You can run your script step-by-step or line-by-line and inspect the variables etc. after each line.
2. You may let the script run as far as a certain point and then continue from there in step-by-step mode or *in stages*.

The method you choose depends on the circumstances: if the only error is in a later part of the script, you should use method 2 so that you can quickly get to the critical part. If something is not happening right at the start, you can use method 1 to monitor the initialization phase.

After compilation, choose “Compiler → Run in debug mode” from the menu or click on in the toolbar. The debugging window is then displayed, and you can review the status of your script. If you are interested in a particular part of your script, double-click into the corresponding line, press F3 or position the cursor in the code line and click on . Then click on or use F5. The script stops running when it reaches the start of the line you have marked and the debugging window is displayed again. You can then use another break point or continue the script step-by-step with and watch the behavior of the script. The next line to be executed is indicated by a green font color.

If during script execution an error occurs, you can start the debugging tool as well: choose “*Debug*” in the error message window. If the source code of the script is available, the code of the active program or subroutine is shown, the next line to be executed is marked.

With the list of subroutines and the CallTree list you can also look at other program segments. The row in which a subroutine was exited is highlighted in bold blue text.

After clicking on , a click on a variable will directly show the variable in the lower part of the debugging tool window. To simulate events, choose one of the defined events in the list on the upper right side and trigger it with . The event is queued into the event queue resp. written into the variable `Event` and you will see the reaction of the script to this event.

If the new contents of the variable are intended to be a character string, the text must be in quotes.

If the script stops running, the following information is displayed in the debugging window:

- Name of the binary script currently running
- Name of section currently running (`MAIN` or subroutine)
- A code listing
- All active `COMMON, GLOBAL, LOCAL` and system variables as well as subroutine parameters (`PARAM`)
- The contents of the global and the parameter stack

The common variables and their values as well as the paramter stack must be viewed on separate tabs.

This procedure lets you find your way to the point where the error has occurred. Always bear in mind that the computer only does as it is told, but it takes everything literally and carries it out *to the letter*!

Additional debugging tool functions

- Click on the symbol to terminate script execution altogether and close the debugging window. When the script has ended, the command `ACQStop` is automatically executed in order to switch the connected devices to STOP status.
- If this option is enabled (button pressed), then in the event of a `CALL` command, step-by-step mode will continue in the subroutine that is called.
- If this option is enabled in step-by-step mode (button pressed), then in the event of a `CALL` command, the called subroutine, along with any others that it calls, will be executed and the script will only be halted again on the line following the `CALL` command.

The font and background color of the code window correspond to that of the Script editor.

6 Special topics

6.1 Modifying catModules

The experienced user can modify or add catModules easily. To do this it is only necessary to load and edit the associated scripts and Online Documents.

The files for the catModules are listed in the online Help, they are installed to catman's CATMOD sub-directory.

If a modified script shall be executed when the original catModule is called up, just put the compiled binary script (*.SCB) into the CATMOD sub-directory. As an alternative you can save the file under a new name and add this script to the catModule list. Use the catman® Professional main menu "Measure → Execute catModule" to call up the dialog "catModules" and add the new script file with "Action → Add script".

☞ The list of catModules shown as well as the module descriptions are stored in the file MODULES.CAT. This file can be edited with any ASCII editor.

6.2 User-defined scripts as catModules

catman allows the user to make his own scripts permanent parts of the catModule system:

1. Open the catModule dialog with "Execute catModule" from the "Measure" menu
2. Mark the entry in the list beneath which the module is to be inserted
3. Select "Add script" from the "Action" menu.
4. In the dialog box which follows specify the "*Name*" under which the module is displayed in the list, an explanatory "*Module description*" and the "*Executable script file*" (*.SCB).

You can insert a new group header entry with “Create new group” from the “Action” menu. New groups are always placed at the end of the list.

To delete an entry from the module system use the “Remove module (script)” menu item in the “Action” menu. This operation deletes only the entries in the file MODULES.CAT; scripts and Online Documents are not affected.

☞ catman saves the entire catModule structure to the MODULES.CAT file, which is located in catman's CATMOD sub-directory. This file can be edited with any ASCII editor.

6.3 Devices, I/O channel connections and scaling

Normally devices are defined interactively on the I/O Definition worksheet. Here you can also set up I/O channel connections and scaling. The script language also offers experienced users the option of using script commands to create and modify devices, I/O channel connections and I/O channel scaling.

6.3.1 Creating devices (instruments)

You create a new device with the help of the `CREATEDEVICE` command. This command only has a single parameter, the device name. catman creates a device here using the name specified—the device type, interface and address are not specified. These are set up afterwards with the appropriate properties (`.DeviceType, .Port, .Address`) of the `IODEVICE[]` object.

Example:

```
#DEFINE UPM100 100
#DEFINE PORT_GPIB 488
...
CREATEDEVICE "Scanner1"
Scanner1.DeviceType = UPM100
Scanner1.Port = PORT_GPIB
Scanner1.Address = 4
```

With `CREATEDEVICE`, a device from the list of existing devices (e.g. interactively defined devices) is added.

To retain full control over the list position of your newly generated device, you should firstly delete all defined devices with the `CLEARDEVICES` command before creating devices from the script.

Example:

```
#DEFINE UPM100 100
#DEFINE PORT_GPIB 488
...
CLEARDEVICES
CREATEDEVICE "Scanner1"
Scanner1.DeviceType = UPM100
Scanner1.Port = PORT_GPIB
Scanner1.Address = 4
```

The device Scanner1 now has the list index 1 (it is the first defined device), the next one created with `CREATEDEVICE` would have index 2 etc. We shall see in the following Section *Setting up I/O channel connections* what this index is needed for.

Notes:

- Removing all devices with `CLEARDEVICES` is not possible as long as at least one I/O channel holds a connection to one of the devices. Firstly, set all connections to “*NC*” (not connected) using `IOCHAN[].Connect` (see next section).
- It is not possible to remove an individual device
- Devices featuring an internal time-base generation (such as for instance MGCplus or Spider8) usually require the setting of a sample rate. Use `IODEVICE[].SetSamplerate` for this purpose.

6.3.2 Setting up I/O channel connections

You use `IOCHAN[].Connect` to connect an I/O channel with a data source (device, time, etc.). We expand on our example from the previous section and connect I/O channel no. 1 with the "*TIME*" data source in the unit minutes and the following 10 channels with hardware channels 20 to 29 of our device (no. 1 in the device list). The type of hardware connection is "*Analog IN*", parameters P4 and P5 can, therefore, be left out.

Example:

```
#DEFINE UPM100 100
#DEFINE PORT_GPIB 488
...
CLEARDEVICES
CREATEDEVICE "Scanner1"
Scanner1.DeviceType = UPM100
Scanner1.Port = PORT_GPIB
Scanner1.Address = 4
...
REM Connection of I/O channel 1 with TIME (all parameters apart from
the first can be left out)
IOCHAN[1].Connect -1
IOCHAN[1].Unit = "min"
IOCHAN[1].Active = 1
...
i = 2
DO 10
  IOCHAN[i].Connect 1 20+i-2 0
  REM activate channel
  IOCHAN[i].Active = 1
  INC i
LOOP
```

Notes:

- Algebraic computations are *not* set up using this method, but using `IOCHAN.Formula`
- *Hard-coded* functions cannot be created or modified using a script
- All I/O channel connections must be made before channel initialization (`ACQInit`), i.e., after each modification to a channel connection, initialization must be carried out.
- After connecting a channel for MGCplus you may use further properties of the `IOCHAN` object to define `IOCHAN[].HWSubChan` (sub channel), `IOCHAN[].Signal` (signal) or `IOCHAN[].SRGroup` to assign the I/O channel to a sample rate group.

6.3.3 Scaling an I/O channel

A scaling type can be assigned to an I/O channel using `IOCHAN[].ScaleMode` of the object. If user scaling is set up, catman automatically creates a 2 point 1:1 scale. The easiest way to modify this scaling is to load a scaling made previously by a user interactively with `IOCHAN[].LoadUserScale`.

Example:

```
IOCHAN[1].ScaleMode = 1
IOCHAN[1].LoadUserScale "C:\CATMAN\SCALINGS\Z6.USC"
```

User scalings of type "*Linearization table*" can also be changed via `IOCHAN[].ScalePoints` (number of linearization points) and `IOCHAN[].Scale` (value pairs).

Example:

```
IOCHAN[1].ScaleMode = 1
IOCHAN[1].ScalePoints = 2
IOCHAN[1].Scale 1 0.5 0
IOCHAN[1].Scale 2 2.5 4000
```

Notes:

- When setting up user scaling for an I/O channel, make sure you use the commands in the *same sequence* as our example above.
- The additional user scaling types "*Polynomial*" and "*Free Function f(x)*" *cannot* be set up by script-controlled means. You may however use `IOCHAN[].LoadUserScale` to load predefined scalings.

6.4 Modifying online computations

An online computation assigned as a data source to an I/O channel is normally defined interactively on the I/O Definition worksheet. However, it is frequently desirable to modify computation parameters (e.g. a multiplication factor) or even the complete computation formula itself, depending on a measurement result or a user input or to recreate them. catman provides for this purpose the `IOCHAN[].Formula` command. You can now assign a computation formula to this property—just like you do interactively on the I/O channels worksheet. But the difference lies in the fact that the formula, apart from the usual Cnn notation (with which you name I/O channels), can also include script variables. During a data acquisition session, the current contents of the script variables are always used—i.e. by simply modifying the variables used in the formula, you can change computation parameters, without a new formula having to be assigned.

☞ With `IOCHAN[].Formula` always an *Algebraic computation* is created, *Hard-coded functions cannot* be created or modified using a script.

Example

```
GLOBAL i f1 f2
REM I/O channel 1 has had the hard-coded function "continuous ramp"
interactively assigned to it is used as a time channel for both of
the following computation channels. In both of the next lines we
define the computation expressions for I/O channels 2 and 3:
IOCHAN[2].Formula = "sin(2 * Pi * f1 * C1)"
IOCHAN[3].Formula = "cos(2 * Pi * f2 * C1)"
REM I/O channel 1 is expressed in the formula as C1
f1 = 1
f2 = 0.5
ACQInit ALL
ACQStart 0 1000
DO 0
   INC i
   IF i > 1000
      i = 0
      REM Modifying frequency variables, online computations are
carried out from now on with the new frequencies.
      f2 = f2 + 0.5
   ENDIF
   ACQRead 5
   UPDATE -r
LOOP
```

Examples of permissible formula inputs:

```
IOCHAN[1].Formula = 2 * c1 + 5 * Delta
REM is translated during compilation time
IOCHAN[1].Formula = "2 * c1 + 5 * Delta"
```

```
REM is translated during run-time)
Expr = "2 * c1 + 5 * Delta"
IOCHAN[1].Formula = Expr
REM is translated during run-time
```

The last form in particular is specially useful—for example, it allows a new formula to be input by the user of your script.

Notes:

- If a computation is assigned to an I/O channel using the `.Formula` property, the data source of this channel will be automatically set to “*Algebraic computation*”. It is not necessary (even though it is permissible) to interactively assign a computation to it in advance on the I/O Definition worksheet.
- The assignment of the computation formula must be applied before channel initialization with `ACQInit`. If the formula is modified, the channel has to be activated again with `IOCHAN[].Active`.
- If a computation has already been interactively assigned to an I/O channel, an assignment to the `IOCHAN[].Formula` property will overwrite this computation.

6.5 Event-driven scripts

The catman script language provides you with facilities to make your script event-driven. These events can be triggered in various ways:

1. By objects in the Online Document: Buttons, Switches, Sliders, Drop-down lists, Check boxes, Text boxes or Labels.
2. By keyboard events (Section 6.5.1, *Keyboard events*, page K-67).
3. By Online Document objects with an alarm monitoring function: Digital indicators, Bar indicators, Analog meters or Indicator panels.
4. By catman's `TIMER` object (Section 6.5.2, *Timer event triggering*, page K-68).

In the following the use of the `TIMER` object and the event handling is explained in detail.

6.5.1 Keyboard events

☞ See also Event monitor in Chapter E, Section 6.3.3, *Keyboard event (key press)*, page E-34.

User presses a key

You can use the system variable KEYPRESS to query the ASCII code of the key last pressed by the user. The value in KEYPRESS is maintained until either a new event occurs or the variable is changed by a script command. It is therefore recommended to set KEYPRESS to 0 after the query.

User presses any key or key combination

In the Online Document editor, define a so-called keyboard event, i.e. assign an event constant to a key like you assign an event constant to an object in your Online Document. This allows a rather more convenient analysis of keyboard events: for instance key combinations such as SHIFT-F5 are possible.

User presses a function key, e.g. F1

You can assign a so-called *Hotkey* to a Button in the Online Document. If the user presses this key, it is the same as a mouse click on this Button and the event assigned to the Button is triggered.

6.5.2 Timer event triggering

Up to version 3.0 Release 2 catman only had available the instruction `ON TIMER` for implementing time control in the script. This had a number of disadvantages:

- There was only one timer.
- Using it in script objects with a number of modules was problematical.
- Only one variable could be set when the timer triggered. This made the initiation of a subroutine difficult: You had to use the system variable `Event` and couple the required subroutine via the `#IRQ` instruction to the event value.
- The use of the system variable `Event` in `ON TIMER` is critical because then, with short intervals, other events can be overwritten.

Therefore, the object `TIMER` was introduced. This can be created by the script in any number and with properties. In particular each object can have its own time interval. Each object can treat the running of its time interval (the “triggering” of the timer) in two different ways:

1. By setting the characteristic `.Event` of the object. The new event is then inserted into the queue. Even when the time interval expires simultaneously for a number of timers, nothing is lost. The script can then treat the events as previously or via `GETEVENT`.
2. By the direct execution of a subroutine allocated to the object `TIMER`. This subroutine can be assigned to the object `TIMER` at the run-time of the script. If on triggering a timer, catman finds that it is coupled to a subroutine, the associated subroutine is immediately executed. Any further `TIMER` coupled with a subroutine is then however only handled during the next event check. It has to be this way, because the principle of the interrupt handler is that the next line to be executed is replaced by a `CALL` instruction.

Create new timer

```
CREATEOBJECT <Name> "TIMER"
```

Example:

```
#DEFINE MEASURE 500
#DEFINE STORE 501

CREATEOBJECT "Timer1" "TIMER"
CREATEOBJECT "Timer2" "TIMER"
Timer1.Interval = 500
Timer1.Event = MEASURE
Timer2.Interval = 10000
Timer2.Event = STORE
```

```
Timer1.Enabled = 1
Timer2.Enabled = 1
```

6.5.3 Checking for events (event queue)

If an event is triggered, catman enters this event in the variable `Event` and the *event queue*. The variable or the queue can then be checked to initiate a suitable reaction.

Method 1

☞ This method has only been retained for reasons of compatibility.

The system variable `Event` contains the last incoming event. Set up an endless loop in your script which contains a `SELECT CASE` structure or a number of `IF...ENDIF` blocks for checking the value of the variable `Event`.

⚠ If a number of events occur quickly one after the other and your script does not react fast enough to all arising events, then events are lost because `Event` always only contains the latest value. Additionally the variable `Event` has to be reset when “evaluating” an event. Otherwise the event is evaluated twice. Use line `Event = 0` immediately after the `CASE Event =` line.

Method 2

☞ This method is recommended from catman version 3.1 onwards.

`GETEVENT` is used to fetch and remove the oldest event from the queue. Another call supplies the next event, etc.

Example:

```
GLOBAL EvtVar
DO 0
  GETEVENT -1 EvtVar
  SELECT CASE
    CASE EvtVar = 100
      CALL Measure
...
  END SELECT
LOOP
```

With this method no event is lost.

Event checking with event queue

Events are saved in the *event queue* when they occur and can be called with `GETEVENT`. The event queue can save up to 512 events and it is only beyond this that the oldest events are lost. The command `GETEVENT P1 P2` returns, depending on P1, a certain value from the queue in P2 and *simultaneously removes* it from the queue. If no event is available, the command returns 0.

In order to ensure upwards compatibility, all events are written both to the variable `Event` as well as to the event queue, but you must only use one of the two methods so that events are not evaluated in duplicate.

For P2 the variable `Event` should not be used, but instead another variable, e.g. `gEvent` or `EvtVar`.

If a check is to be made for an event without removing the event from the queue, `PEEKEVENT` can be used.

`GETEVENT -3 EventValue` can also be used for checking for a certain event. If the event is present in the queue, the system variable `GP1` contains the value 1, otherwise `GP1 = 0`.

IRQ subroutines

A special case occurs in conjunction with `#IRQ` subroutines, i.e. SUB programs, which are linked to a certain event via an `#IRQ` instruction (Section 6.5.4, *Interrupt subroutines*). The aim of the `#IRQ` instruction is to execute such subroutines immediately when an event occurs. The interrupt handler of catman checks the complete event queue for the presence of a certain event. If the specific event is found, the event is removed from the queue and the `SUB` program executed.

6.5.4 Interrupt subroutines

Normally events simply set the event variable resp. the value is queued into the event queue—there is no reaction to the event yet. This only happens if query loops provided by the programmer check this and then introduce appropriate actions. However, this method is not satisfactory under certain circumstances: for example, imagine a button with the function EMERGENCY OFF. If this is pressed, for instance, a digital output should be switched in immediately. But if your script is in a long computation loop at this time, within which the event variable is not checked, the reaction to the EMERGENCY OFF button can take place too late.

To deal with this situation catman has provided Interrupt routines. Any SUB program can, with the aid of the `#IRQ` command, be declared as an interrupt routine and linked with a particular event constant, e.g. `#IRQ EMERGENCY_OFF SystemStop`. catman now checks constantly throughout a script run whether the `Event` variable or event queue has received an event value of this type. If this is the case, the script switches at once to the appropriate IRQ subroutine, the normal script run only being resumed if this interrupt routine has finished. Interrupt routines can be chained, i.e., during the processing of one interrupt another can occur. In this case the interrupt routine being processed is interrupted and program execution continued with the interrupt routine of the second interrupt.

catman does not allow recursive subroutines, interrupt routines are of necessity non-reentrant, i.e., if an interrupt routine is being processed and the same interrupt occurs, it will be ignored.

For this reason, an interrupt subroutine itself, as well as all further subroutines called by it, must not have any calls to other subroutines declared as interrupt routines—a run-time error “Invalid recursive call” would be the result.

Notes:

- catman can only respond to an event when the command currently being executed has been fully processed. This means that during a long-lasting command, e.g. `ACQRead`, mathematical computations or data export, the event will simply be put into a queue—execution only takes place once the current command has finished. If the event in question is a timer event, then the instruction connected to the timer will be executed immediately after the previous operation has finished. If several timer events occur in the course of a lengthy operation, they are lost.
- You can use `APP.Critical` to temporarily disable event monitoring. This may be necessary, for instance, in steps that have critical safety aspects, during which you wish the user to take no action. Turning off event monitoring also makes your script run faster.

☞ Remember to enable event monitoring again after reaching the end of a critical step, otherwise the user will no longer be able to operate the interface (Online Page).

6.6 Popup dialogs

catman's Online Document editor offers you a powerful tool for creating graphical pages and input masks. Sometimes however, you only want to display a smaller dialog box in which the user can make a few settings, e.g. text inputs or via a check box, without having to move from the current Online Page. The advantages of this method are obvious:

1. faster image building (changing pages is time-consuming)
2. lower memory requirement for the Online Document—the dialog box is generated in run-time only
3. Online Page still visible

The `DLG` (dialog) object, therefore provides you with a facility to create and display popup dialogs. They are called popup dialogs because the dialog box is displayed over the visible Online Page. With the `DLG[x]` object up to 16 dialogs can be created and even displayed together. The objects are created with an index, the first index is 1. The dialog box can be called up modal, i.e. you cannot take any action outside the window until it is closed, or non modal. A non modal window remains open until it is closed when a button is pressed or with `DLG[x].Close`. Buttons should therefore only be used for modal windows. In any case the `.CloseEvent` stated will be created.

☞ For compatibility reasons `DLG[1]` and `DLG` are identical. If the index is skipped, always the first object is used.

You can use the `MESSAGE` statement to produce even simpler dialog boxes that output only a message or present the user with a “*YES/NO*” prompt.

Before displaying the dialog, first of all define the various objects—for this purpose use the `.Add` method. When you call this method you specify the type of object, its position and size, a label etc. One alternative is to load the entire dialog from a file. You can also combine the two methods, i.e. first load a file and then add other objects.

Menu option “Create popup dialog code” in the Online Document editor, menu “Page”, automatically generates the `.Add` calls needed to display the current page as a popup dialog. Objects that are not allowed in popup dialogs (e.g. graphs) are ignored. However, if you save the page as a file, these objects will also be saved.

Another possibility is to use “*Save as dialog file*” (in the dialog box). In this case, the current page is saved as a file which can be called from the script. The page can also contain objects which cannot be created with the command `DLG.Add`. The size of the window must be defined manually or by using the script command `DLG[].Move`. The present size or position is not read. Do not forget to specify the dialog buttons and if necessary the variables for return values.

Depending on configuration (`.Show`), the dialog box contains an “*OK*” button and/or a “*Cancel*” button, showing a Help button is also possible. The chosen button is returned in system variable `GP1`.

In addition the popup dialog object supports the following

- Specify window title: `DLG[x].Caption`
- Choose background color: `DLG[x].BackColor`
- Center, maximize dialog box: `DLG[x].WindowState`
- Set up window size and position: `DLG[x].Move`
- Automatically terminate the dialog if no user input is received: `DLG[x].Timeout`

☞ Since all dialog objects (with the exception of the rectangle) contain text, the font to be used can be set up with the `FONT` command before the `.Add` method is executed.

The objects of a dialog can also be accessed. Use the normal object commands preceded by a `DLG[x].`

Examples:

```
DLG[1].ObjectName.ForeColor = COLOR_RED
DLG[1].ObjectName.Caption = "Test"
AnswerText = DLG[1].ObjectName.Text
AnswerValue = DLG[1].ObjectName.Value
```

☞ Not all object properties can be used.

6.7 Smart ACQRead for MGCplus

From catman version 3.0 Release 2 onwards there is the possibility for MGCplus systems of setting the parameter P1 of the `ACQRead` command to -1. From catman Version 4.0 onwards, this method can also be used for the Spider8 if this has a firmware status of at least P22.

⚠ This functionality must not be used when reading data from file in real time (online import). If you read from file only, you may specify any value, as in this case the same read-block size is used as when storing to file. If you want to read in data from devices and file together, you must use the same read-block size for both device and file, i.e. the same read-block size as was used when storing the file.

If `P1` equals `-1`, catman always reads for each `ACQRead` just as many values from the device as are present in its buffer. This procedure offers the best possible performance, because the size of the data blocks that are read in is matched to the current conditions. For example, if the buffer of an MGCplus system fills, then `ACQRead` will always request larger blocks, but never more than the number of values per channel specified with `ACQ.MaxDynReadBlock` (default: 1,024 values). Consequently, an attempt is made to empty the buffer and to keep pace with the measurement.

Another field of application for this method is with the use of the sample rate trigger, i.e. the trigger-controlled selection of the sample rate of one of the three sample rate groups of the MGCplus. If operation were to take place with a fixed read block, then this would either be too small for the usually significantly faster trigger sample rate or too large for the slower base sample rate.

Example:

```
ACQStart 0 0
DO 0
  ACQRead -1
  NoOfValues = NrOfValues + IOCHAN[x].Len
  REM Use an active channel which must have values (highest sample
rate)
  ACQStore
  If NoOfValues >= MaxValues
    Break           REM We have enough values
  Endif
LOOP
```

Notes:

- The actual number of the values read for the fastest sample rate group is returned in the system variable `GP1` when values are acquired via a measuring device. This is however only done for the fastest sample rate group and is not done when files are being read. We therefore recommend that you determine the number of values read in via `IOCHAN[].Len` from one of the I/O channels involved in the measuring.
- Do not use any catman time channel (only MGCplus Hardware time channels) to determine the number of values read in.

- With this type of reading method all participating I/O channels should use the setting 1,024 for the read-block size (I/O Definition, "*Max. read block*", or `IOCHAN[].BufferSize`) or restrict the size via `ACQ.MaxDynReadBlock`.

6.8 Specifying path names

With some commands, you have to specify a file name. You can either specify the full path name or just the file name and let catman find the file. catman will generally search the following directories:

1. the directory where the script currently being executed is located.
2. the catman's CATMOD sub-directory.
3. the catman's SCRIPT sub-directory.
4. the directory where catman.exe is located
5. the currently active directory of the operating system.

There are some exceptions: loading a picture file for some Online Page objects and the `FILE` object for low-level accesses. With the latter, if you do not specify the full path name, the *only* directory searched is the one in which the currently executed script is located.

6.9 Saving and recalling data in structured ASCII files

With complex scripts, it frequently happens that, apart from pure measurement data, which is stored or read using the export/import commands, you also want to manage other information, e.g. operator name, part no. etc., in external files. In this respect, catman gives you the opportunity to write texts (strings) to specially structured ASCII files, so-called *profile files*, from which you can also read information into variables.

Profile files provide the facility to arrange stored text data in groups (known as *sections*), and to provide each individual entry in a group with a keyword (usually just called a *key*). When reading from the file, you can specify the section and key for targeted access to a particular entry.

Example:

```
Section:  [MEASUREMENT]
Key:      sample rate = 1200
Key:      Filter = 3
Key:      DeviceID = DMCplus_001
Section:  [OPERATOR]
Key:      Surname = Tester
Key:      Given_name = Bill
```

☞ In the file created the words `Section:` and `Key:` used here for clarification are not stored.

The most prominent example of this file type is probably the file WIN.INI from Windows. catman offers you two commands, with which you can write data to a profile file (`FWRITEPRIV`) or read data from one (`FREADPRIV`).

☞ If your application makes heavy demands on the structuring of data or has a major requirement for database facilities, it is preferable to use catman's integrated database engine (see Section 6.10, *Accessing external databases*) instead of profile files.

6.10 Accessing external databases

catman offers you the facility to call up information from an external database (such as MS Access) or to save information to a database.

Communication with a database:

- So-called data links create a connection between objects in the Online Document and the fields of a database.
- Methods and properties of the object DAO are used to control the actual data exchange.

Linking database fields to Online Document objects

When they are created, objects in an Online Document are linked to fields in what is known as a database template. Use the “Tools → Select database template” menu to choose an existing database file as a *database template*. However, with the assistance of catman’s integrated database designer from the “Tools” menu, you can design a database yourself. The only condition, in both cases, is that the data layout, i.e. the name, type, number of fields etc., matches the database file used later.

Select a table from this database template. The fields in this table will then be offered to you for selection in the setup dialogs for Online Document objects. Once a link of this kind has been created between Online Document objects and database fields, catman always makes sure that if the current data record changes during script run-time, the objects are automatically updated, that is, the database fields have the same contents as the objects. Additional information can be found in Chapter G, *Online Document*, in Section 5, *Options, tools and tips*, page G-44.

6.10.1 Communicating with a database using a script

Once we have defined all our connections, in the second step the script language makes available some commands, with which we can control communication between catman and the database.

☞ Communication always takes place with what is called a *record set*. This is a virtual image of a table, and contains all or some part of the data records from a table in the database.

A record set can even contain data records with fields derived from a number of tables. For each open database file (`DAO[1...4]`) catman can keep as many as 8 such record sets open at a time: `DAO[1...4].RECORDSET[1...8]`.

Command overview:

- Open a specific database file: `.OpenDatabase`
- Open a record set for the current file using search criteria: `.Open`
- Close database: `.Close`
- Browse data records: `.Browse`
- Find data records: `.Find`
- Add data records to a record set: `.AddRecord`
- Delete data records from a record set: `.DeleteRecord`
- Update all linked Online Document objects with field contents: `.UpdateControls`
- Update all database fields with the contents of linked Online Document objects: `.UpdateFields`

Of course, you can also communicate with the database without having to link Online Document objects to fields. Commands are provided that let you write directly to or read directly from database fields. By this means even entire database channels can be written to or read from *LARGE BINARY fields* in the database.

Command overview:

- Read field contents into a variable: `.Read`
- Begin edit mode: `.BeginEdit`
- End edit mode: `.EndEdit`
- Write to a data record field: `.Write`
- Retrieve a database channel from a LARGE BINARY field: `.ReadChan`
- Pack database channel into a LARGE BINARY field: `.WriteChan`

Please note that writing directly with `.Write` will only work if edit mode has been opened, and then closed again after the last field has been written. Your changes are lost if you switch from the current data record without first closing edit mode. Attempting to write without ending edit mode causes a run-time error.

6.10.2 Transactions

Transactions are usually carried out when a series of write operations needs to be considered as a unit. For instance, an accounting transaction from one bank account to another consists of two transactions: a debit posting to one account and a corresponding credit posting to the other account. In practice both aspects of the transaction must always be completed successfully; a one-sided, single transaction is prohibited. In this example, the debit and credit postings constitute a transaction. If you initiate a transaction with `.BeginTrans` before starting to change the data records, in the event of an error, e.g. failed update attempt with `.EndEdit`, you can cancel all the changes made up to that point in all open record sets with `.Rollback`. If you have carried out all your changes successfully, you can close off the transaction with `.CommitTrans`.

Once a transaction has been closed off with `.CommitTrans`, rollback is no longer possible unless the transaction was part of another transaction. If this is now canceled, then all the other transactions it contains are automatically canceled also even if they have already been closed off with `.CommitTrans`. Please note here that a transaction on a higher level can only be terminated or canceled if all other transactions at this level have been terminated or canceled.

Notes:

- Calling the methods `.CommitTrans` or `.Rollback` without having first opened a transaction with `.BeginTrans` leads to a run-time error.
- Transactions always apply to all currently open record sets, that is, it is not possible to carry out transactions for individual `DAO[x]` or `DAO[y].RECORDSET[z]` objects. To make matters clear, we recommend the notation `DAO.BeginTrans`, `DAO.CommitTrans`, and `DAO.Rollback`.

- When you close the database that is currently open with `.Close`, all transactions not yet closed off will be canceled.
- Besides guaranteeing the integrity of your data, you can use transactions to enhance system performance. In general, operations that run within transaction blocks are buffered, and this drastically reduces the number of disk accesses actually needed.

6.10.3 Examples for database access

Example 1 The example opens the database SENSOR.MDB and there the Sensors table. A new data record is entered and the values of the variables *Type*, *SN* and *Unit*, are written into the database. The write operation is closed after `.EndEdit` with `.CommitTrans`. After `.CommitTrans` another check could be made for write errors.

```
DAO[1].OpenDatabase "Sensor.mdb"
REM Open table
DAO[1].RECORDSET[1].Open "Sensors"

DAO.BeginTrans
DAO[1].RECORDSET[1].AddRecord 1
DAO[1].RECORDSET[1].Write "Type" Type
DAO[1].RECORDSET[1].Write "Serial" SN
DAO[1].RECORDSET[1].Write "Unit" Unit
DAO[1].RECORDSET[1].EndEdit
DAO.CommitTrans

REM Close table
DAO[1].RECORDSET[1].Close
REM Close database
DAO[1].Close
```

Example 2 The example opens the database SENSOR.MDB and there the Sensors table. A search is made there for a certain sensor (Condition). If the search is successful, the data record then becomes the current one and may be read out.

```
DAO[1].OpenDatabase "Sensor.mdb"
REM Open table
DAO[1].RECORDSET[1].Open "Sensors"
REM Go to the last data record. The data record entered in the search
command may not be found if the current data record is that being
searched for.
DAO[1].RECORDSET[1].Browse 3
SerialNO = "12345"
STRCAT Condition "SN = '" SerialNO "'"
REM Start with search from first record
DAO[1].RECORDSET[1].Find 2 Condition
IF GP1 = 0
  DAO[1].RECORDSET[1].Read "Type" Type
  REM Write type of transducers into field on Online Page
  TypeField.Text = Type
  ...
ELSE
  MESSAGE "Transducer not found" 6 gDummy "ATTENTION"
ENDIF
REM Close table
DAO[1].RECORDSET[1].Close
REM Close database
DAO[1].Close
```

6.11 The Controls collection

catman provides the option to address objects in the Online Document without having to use the object name. With the aid of an indexed object group called the *Controls collection*, it is possible to modify or read out object properties interactively, i.e. using loops. Suppose for instance that you want to deactivate all the text boxes on a page. Instead of addressing each object individually, by using the Controls collection you can now write more elegantly:

```
#DEFINE CTRL_TEXTBOX 4
...
```

```
CtlCount = Controls.Count
FOR i = 1 TO CtlCount
   CtlStyle = Controls[i].Style
   IF CtlStyle = CTL_TEXTBOX
      Controls[i].Enabled = 0
   ENDIF
NEXT
```

As you see, you can use each property or method on an element of the Controls collection just as if you had addressed the object by using its name in the customary way. The group also provides some useful properties which you can use to determine, for instance, the total number of objects on a page, or the type of a particular object.

As well as the option to address objects in a general, name-independent form, the Controls collection also offers you the capability to create (`.Add`), remove (`.Remove`) or reposition (`.Move`) objects during run-time.

☞ Please note that when dynamically creating objects, they all have standard settings immediately after they are created. As a rule you then need to set the required properties in the script (e.g. axis scaling). On the other hand, many properties of an object are not accessible from within a script (for example the background picture for a Button). Simple objects such as Buttons, Checkboxes, Labels or Text boxes can therefore be dynamically created without any problem; the majority of their properties are accessible from a script. But as a rule, complex objects such as a Spreadsheet should not be created dynamically.

Properties

- `Controls.Count`: current number of objects on the page (read-only access).
- `Controls[Index].Left/.Top/.Width/.Height`: left/top edge resp. width/height of the object.
- `Controls[Index].Name`: sets or reads the name of the object.
- `Controls[Index].Style`: the object type (read-only access).

Methods

- `Controls.Add`: Creates a new object.
- `Controls[Index].Move`: Moves and resizes the object.
- `Controls[Index].Remove`: Removes an object from the object group.

6.12 The Annotations collection

In contrast to the Controls collection the Annotations collection is not an independent group, it is always a sub-object to a specific Scientific graph. The behavior of all annotations of a graph can be controlled via this Annotations collection: create one or more new annotations, delete annotations or set properties.

☞ Setting annotation properties is controlled by the `.ImmediateUpdates` property—if this setting equals 1 the graph will be redrawn after each property change. When changing many properties at once you should thus set this property to 0 and redraw the graph once after all properties are set with `.Redraw`.

6.13 DLL Drivers for “unknown devices”

To both improve the linking of other measurement devices via standard interfaces and to facilitate the inclusion of cards which can be inserted into the PC, catman from version 4.5 supports the creation of DLLs which are then called by catman to initiate the acquisition of measurements and the return of the data. In catman 5.0 these functions have been significantly improved and extended; therefore we recommend that the (older) Script Drivers are no longer used.

Advantages of the DLL Drivers compared to Script Drivers

- DLL Drivers can be created with a programming environment, e.g. C/C++.
- All functions and commands of the programming language used are permissible, i.e. there are no restrictions as with the Script Drivers.
- The speed of execution is substantially higher.
- DLL Drivers can call and use other DLLs, e.g. the NIDAQ driver.
- Full access to the internal catman data structures is retained.

To simplify the creation of DLL Drivers you will find in catman’s DRIVERS\DLLDRIVER\DRIVERTEMPLATE sub-directory a template and in DRIVERS\DLLDRIVER\AGILENT_54622D and DRIVERS\DLLDRIVER\AGILENT_34401A programming examples. The document “DAQ Drivers.doc” in the DRIVERS\DLLDRIVER sub-directory also contains comprehensive information.

Note the synchronization of measurement data introduced in catman 5.0: If you are working with an MGCplus with a CP42 and with one or more other devices which can also transfer NTP or IRIG-B time information, then have all measurements converted to time-synchronous measurement data already during acquisition.

6.14 Script Drivers for “unknown devices”

☞ We recommend the exclusive use of DLL Drivers from catman 5.0 onwards.

6.14.1 General

catman provides the script programmer with the ability to use third-party devices in high-level data acquisition, that is, those devices not supported directly by catman, with the aid of a *Script Driver*. These devices can then be used without any further special precautions in a catModule or the Measurement Wizard, their measurement data can be displayed in real-time graphs, saved to the Database with `ACQStore` etc.

☞ Devices for which you intend to create a Script Driver must have a standard interface (RS-232, GPIB or TCP/IP) and be controllable with a command set. Devices that are addressed through a driver DLL of their own (as a rule this means all PC plug-in boards) cannot at present be controlled by using a Script Driver.

For the most part, a Script Driver consists of an executable script containing just the subroutines `ACQInitDevice, ACQInitChan, ACQStart, ACQRead` and `ACQStop`. As you can see, these subroutines correspond precisely to the script commands used for data acquisition in each case. If a catModule, the Measurement Wizard or your own script now executes for instance the `ACQStart` command, the data acquisition kernel automatically calls the `ACQStart` subroutine of your driver.

6.14.2 Creating a driver

To create a Script Driver, open the Script editor and choose “New Script Driver” from the “File” menu. catman creates a code module that already contains the required (empty) subroutines. As you would expect, the module also contains a `MAIN` program. However, this may only be used for declaring variables and constants. If it contains executable code, this is ignored. Similarly, at the present time any subroutines other than the 5 standard programs are also ignored.

ACQInitDevice

Add to this subroutine all the one-off actions that have to be carried out when the device is initialized. catman calls this subroutine whenever `ACQInit ALL` or `ACQInit <Device>` resp. `<Device>.Initialize` are used.

ACQInitChan

Add to this subroutine all the actions that have to be carried out whenever a channel is initialized. catman calls this subroutine whenever `ACQInit ALL` or `ACQInit <Channel>` resp. `IOCHAN[x].Initialize` are used.

ACQStart

Add to this subroutine all the actions that have to be carried out in order to start data acquisition. catman calls this subroutine whenever the `ACQStart` command is used.

ACQRead

Add to this subroutine all the actions that have to be carried out in order to read measurement data. catman calls this subroutine whenever the `ACQRead` command is used. In general this subroutine is the most important—in theory all the others can be left blank.

ACQStop

Add to this subroutine all the actions that are needed to exit data acquisition. catman calls this subroutine whenever the `ACQStop` command is used.

After having filled out the required subroutines, build the executable Script Driver by using the menu option “Compile as Script Driver” from the “Compiler” menu. The file is automatically given the file extension *.DRV.

The driver cannot be built by using “Make executable file”.

6.14.3 Script driver commands

A subset of the catman script language is available for programming your driver, together with the `DRIVER` object.

The following language elements are allowed within the driver:

- Global variable declarations with `GLOBAL`
- Declaration of constants with `#DECLARE`
- Variable assignment

- Loops with `DO...LOOP` or `FOR...NEXT`
- Branches using `IF...ELSE...ENDIF`
- Breaking loops or exiting a subroutine with `BREAK` and `EXIT`
- Commands for handling strings `STRCAT`, `STRMID`, `STRSEARCH`, `STRLEFT`, `STRRIGHT`, `STRTRIM`, `CVNUM` and `CVSTR`
- Some properties and methods of the object `IOCHAN`:
  - `.PutBuffer`
  - `.ADURange`, `.ScaleFactor` and `.Offset`
  - `.HWChan` and `.HWSubChan`
  - `.Unit`

All other language elements and instructions are prohibited.

6.14.4 Making a driver

You will find one example for a Script Driver for the DK38 in catman's DRIVERS sub-directory.

Interfaces

It is neither necessary nor possible for you to open or close interfaces in your driver. This is handled by catman before the individual functions of your driver are called.

Scaling measured values

Devices often return your measured values only in the form of binary raw data, e.g. 4,096 digits in the case of 12 bit AD converters. These then have to be converted into meaningful physical values. The easiest way is to leave this task to catman. All values in I/O read block buffers are scaled according to:

$$\text{Physical Quantity} = \frac{\text{Raw value}}{\text{ADURange}} \times \text{ScaleFactor} + \text{Offset}$$

`ADURange` specifies the span of the binary raw data (in digits) which a device can image, e.g. 4,096 for a 12 bit AD converter, `ScaleFactor` specifies the physical measuring range, e.g. in V, to which this span corresponds, and `Offset` specifies an offset that is to be added in physical units.

Just specify these values once in the `ACQInit` subroutine of your driver: `IOCHAN[].ADURange, IOCHAN[].ScaleFactor` and `IOCHAN[].Offset`.

☞ By default these values are preassigned for each I/O channel to give a 1:1 scaling: `.ADURange = 1, .ScaleFactor = 1, .Offset = 0`.

Return values from driver routines

On exiting or quitting a driver routine, you should inform catman's data acquisition kernel whether your routine was successful or not. You may do this by using the `EXIT` command with a parameter P1 which returns the status of the routine. The data acquisition kernel handles return values as follows:

- 0: Successful, no error
- -199: Data not available yet within the read block currently requested. The data acquisition kernel will then call your `ACQRead` subroutine once more, and keep doing so until either the values are received or a reading timeout occurs.
- < 0: Error. You should use error numbers smaller than -16,000 in order not to conflict with the internal error codes of the data acquisition kernel.

You may insert a meaningful error message for each number in the run-time error file RUNTERR.TXT provided by catman.

☞ If errors occur within a method or command in the driver, e.g. low-level timeout for a powered down device, the driver automatically returns an error code to the calling layer. You therefore do not need to concern yourself with errors of this type.

6.14.5 Installing a driver

Once your driver is finished, you can assign it to a device of type “*Unknown*” in the I/O Definition worksheet with menu sequence “Settings → Driver for unknown instrument”. Specify or choose your compiled driver (*.DRV).

⚠ The driver file must be accessible during the run-time of a catModule, the Measurement Wizard or a script.

☞ Script Drivers are only loaded when a script, a catModule or the Measurement Wizard is started, and they are unloaded again on quitting the script.

6.15 catman as web server

To operate catman as a web server which can provide measurement data for any web pages, you must do two things:

1. start the service web server in catman,
2. create an HTML page which can be displayed in a browser and contains appropriate fields for the output of data.

6.15.1 Start web server in catman

To do this you must modify the file ADDINS.CAT in the catman directory. Open the file with a text editor and add the following line:

```
NAME=Start web server,Class=catWebServer.catWeb
```

Then start catman as usual. In the project window and in the main menu the item “Start web server” is available under “Add-Ins”. Double clicking the item in the project window starts the web server.

6.15.2 Create an HTML page for displaying data

The *first* page to be created must be saved in catman’s WEBSERVER sub-directory, so that the catman web server has access to the file. You may however link to other pages in other directories.

Working principle

1. The web browser requests the file page1.htm from catman’s web server, as soon as you type in http://localhost/page1.htm (use the computer’s name or address, if you open the page from another PC).
2. The page contains sections with instructions (code) in Microsoft Visual Basic Script (VB-Script) which is first executed by the server.
3. The results of these VBScript routines are supplied to the browser together with the rest of the page.
4. The web browser displays the page including the information from catman.

Realization

If you would like to use other languages in your HTML document, you must insert them *after* the start tag with the script language for the server. In addition, the place holder must occur *before* the start tag.

The executing VBScript code is placed for identification between the tags `<SCRIPT LANGUAGE="VBScript" RUNAT SERVER>` and `</SCRIPT>`. The required VBScript routines, at least the section Main, are placed within these tags. The commands which you can use in these sections are described in the ActiveX reference. The result of these commands is written to a special variable, the placeholder, together with other HTML code needed for the display. This placeholder is in turn located in the normal HTML code of the page at the place where the result supplied by the catman web server is to appear.

The three possible placeholders are described in the Reference on the `Web server`.

6.15.3 Web server: Example 1

The example reads the last (= most recent) value from Database channel 5 and instructs the catman web server to write this value at the location in the HTML document occupied by the placeholder `%MVAL`. In the case of an error, the error text (`LastErrorDescription`) is output.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
  <head>
    <title></title>
    <link rel="stylesheet" media=screen href="Screen.css"
type="text/css">
    <link rel="stylesheet" media=print href="Print.css"
type="text/css">
  </head>
  <body>
    <h1>Webpage XY</h1>
    ...
    <h2>Last value of catman channel 5:</h2>
    <p> </p>
    %MVAL
    <SCRIPT LANGUAGE="VBScript" RUNAT SERVER>
```

```
        Sub Main()
          Dim Reading
          Dim Err
          Reading = catman.IO.MVal(5,0)
          Err = catman.LastError
          If Err < 0 Then
            server.ReplaceTag "%MVAL", catman.LastErrorDescription
          Else
            server.ReplaceTag "%MVAL", Reading
          End If
        End Sub
      </SCRIPT>
    ...
    <script language="JavaScript" src="Java.js"></script>
    ...
    </body>
</html>
```

☞ The selection of `%MVAL` has been taken at random here, you can also use any other designation. The properties and methods of catman’s ActiveX interface are described in the ActiveX reference.

6.15.4 Web server: Example 2

The example copies a Scientific graph which is located on the current Online Page and instructs the catman web server to write this graph in the WMF format at the location in the HTML document occupied by the placeholder `%IMAGE`. In the case of an error, the error text (`LastErrorDescription`) is output.

```
<SCRIPT LANGUAGE="VBScript" RUNAT SERVER>
  Sub Main()
    Dim Err
    Dim q
```

```
     q = String(1,34)  <!-- double quotes: " -->
     catman.ExportGraph "GRAPH_1",
"D:\catman\Webserver\Images\Graph1.WMF", 0
     Err = catman.LastError
     If Err < 0 Then
       server.InsertText "%IMAGE", catman.LastErrorDescription
     Else
       server.ReplaceTag "%IMAGE", "<IMG SRC=" & q & "Graph1.WMF" & q
& ">"
     End If
   End Sub
</SCRIPT>
```

☞ The selection of `%IMAGE` has been taken at random here, you can also use any other designation. In this example the graph must be stored in catman’s WEBSERVER\IMAGES sub-directory, as there is no path specified. The properties and methods of catman’s ActiveX interface are described in the ActiveX reference.

6.16 Using VBScript

There are various reasons for using VBScript:

1. Higher performance than catman’s script language (up to a factor of 10).
2. Standard programming language with no restrictions as with catman’s script language.
3. Access to a vast number of components and programs from FileSystemObject through to Excel.

Unfortunately, there are also some disadvantages:

- The control of Online Pages is possible only with restrictions.
- Access to the catman mathematics is possible only with restrictions.
- Debugging difficult or impossible.

Basically, you can use VBScript for two different tasks:

1. You can execute it (instead of a catman script).
2. You can load it into the Microsoft® Windows® Script Host (WSH) and execute it from catman.

To use a VBScript you must:

1. Write the script.
2. Load the script into WSH.
3. Execute the script.

You can carry out the last two points in different ways, partially also with one command:

- You can include VBScripts in a project (project window). You can then execute the script via the context menu, but you can also load it or edit it in the WSH.
- When loading a project, you can execute the first VBScript or load it into the WSH.
- You can also execute a VBScript via the catman main menu using “Measure → Execute script from file”.
- You can load and execute a VBScript via the catman script command `SCRIPT.Execute`.
- You run a procedure in VBScript via the catman script command `SCRIPT.Run <ProcedureName>`.

6.16.1 Writing a VBScript

Use the editor built into catman to write a VBScript: Main menu “Worksheet → VBScript editor”. Alternatively, you can also use any text editor or the VBScript development environment. The integrated editor has the advantage that it offers a code builder which lists the most important functions of catman’s ActiveX interface. Then you have access to the functions available to catman for interchanging data and sending commands. You will find a comprehensive reference about these functions in the ActiveX reference (Online Help only).

The integrated development environment offers in addition the possibility of running individual VBScript procedures for certain statuses of a catModule, script or of the Measurement Wizard, e.g. `Execute: After measurement period`. Such procedures are run analogous to the commands in the Auto Command List on attaining this status.

VBScripts are pure text files. They can, but do not have to consist of one general section (`General`) followed by procedures (`SUB ... END SUB`) and functions (`FUNCTION ... END FUNCTION`). The relevant code is located within these blocks and at the end of the block follows the code for the time of execution. The general section usually contains only constants and variable declarations.

☞ If the section `General` also includes executable code, then it is executed already during loading of the script into the WSH.

☞ From catman 4.5 no procedure `MAIN` is required any longer in the section `General`.

6.16.2 Loading a VBScript

You have various possibilities of loading a script into the Windows Script Host (WSH):

1. When loading a project, you can load the first VBScript directly.
2. You can load one of the VBScripts present in the project via the context menu.
3. You can execute a VBScript via the catman main menu using “Measure → Execute script from file”. If the section `General` does not contain any executable code, the script is only loaded.
4. If the section `General` does not contain any executable code, you can load a VBScript via the catman script command `SCRIPT.Execute`.
5. Alternatively to Method 4, you can also load a VBScript via the following catman script commands:

```
SCRIPT.Init
SCRIPT.Language = "VBScript"
SCRIPT.Reset
SCRIPT.AddcatInterface
SCRIPT.AddCodeFromFile <Dateiname> REM Loads the code from file
   or
SCRIPT.AddCode <Code> REM Code is given as parameters
```

☞ When loading into the WSH, first only the syntactic validity of the script is checked and no further action is initiated. Running a script only executes the code located in the section `General`.

6.16.3 Executing a VBScript

☞ From catman 4.5 onwards no procedure `MAIN` is needed any longer in the section `General`.

There are two different ways of executing a VBScript:

1. Manual execution.
2. Automatic or script-controlled execution.

Methods of manual execution

1. You can execute one of the VBScripts present in the project via the context menu.
2. You can execute a VBScript via the catman main menu using “Measure → Execute script from file”.

   ☞ If the section `General` does not contain any executable code or procedure call, the script is only loaded.

Methods of automatic and script-controlled execution

☞ For a VBScript to be able to be executed, at least one line of executable code must be included in the section General. This may also be a procedure call, e.g. `CALL <ProcedureName>`. If no code is included, then the script execution corresponds simply to loading into the WSH.

1. When loading a project, you can execute the first VBScript directly.
2. The Event monitor can execute a procedure on the occurrence of an event.
3. If procedures in the script are assigned to certain execution conditions, e.g. `Execute: After measurement period`, they are executed on the occurrence of the condition automatically by all catModules, the Measurement Wizard and all catman scripts which previously supported the execution modes of the Auto Command List. To enable this, the scripts must include the line `APP.Execute`.
4. You can load a VBScript via the catman script command `SCRIPT.Execute` and execute the code contained in section `General`, e.g. `CALL <ProcedureName>`.
5. You can run a procedure via the catman script command `SCRIPT.Run <ProcedureName>`.

Example 1:

```
SCRIPT.Init
SCRIPT.Language = "VBScript"
SCRIPT.Reset
SCRIPT.AddcatInterface
SCRIPT.AddCodeFromFile "MyScript.vbs"
SCRIPT.Run "MyProcedure" Parameter1 Parameter2
```

Example 2: The example assumes that no code is present in the `General` section.

```
SCRIPT.Execute "MyScript.vbs"
SCRIPT.Run "MyProcedure" Parameter1 Parameter2
```

Example 3: The following line is located in the program code of a script after the end of a periodic measurement and runs the VBScript procedures for which the execution time point is set to Execute: After measurement period. Only VBScripts are executed, no Auto Command List.

```
APP.Execute 2 1
```

6.17 Script optimization

Even though script is a really fast, semi-compiled language, speed of execution can be further increased by taking the appropriate measures.

High-speed data transfer (ACQRead)

Depending on your option, use read blocks that are no smaller than 100. A value between 100 and 512 is suitable; larger blocks will not be read any faster.

Read-block size x number of activated I/O channels x 8 must not exceed 65,000.

Another time-consuming operation during high-speed data acquisition is saving to a Database (`ACQStore`). In this case it is not advisable to use `ACQStore` in the form `ACQStore C[Index]`, i.e. do not try to save specific I/O channels. It is much faster to use `ACQStore` without any parameter. In this way all I/O channels that are active during the current data acquisition session are saved. The storage task runs even faster if you do not use `ACQStore`, but instead set the appropriate marker in the I/O Definition or use `IOCHAN[].Storage`.

Rapid updating of real-time graphs

The larger the read block used for updating, the faster the plotting and scrolling. However, please note that the smoothness of the scrolling decreases as the size of the read blocks increases.

Switch off event checking during lengthy computation loops

Use `APP.Critical` to do this. With event checking disabled, a computation loop operates up to 40% faster.

Use the commands INC or DEC to increment or decrement

These commands are significantly faster than the explicit notation

```
Variable = Variable + 1 (or -1)
```

Avoid numerical expressions and numerical constants in object indexes

Use simple variables instead.

Example:

```
Valuexy = 83
Channel1 = i+5
```

```
Channel 2 = Channel 1 + 3
DO 10000
  INC r
  DBChan[Channel1].Row[r].Value = r
  DBChan[Channel2].Row[r].Value = Valuexy
LOOP
```

Do not use numerical constants during speed-critical computation loops

It is much better to save the constant to a variable and work with this variable in the loop.

Do not use over-complex formula expressions

The more complex an expression is, the longer it takes to determine the result during your script run-time.

Avoid formula expressions in channel numbers, such as `DB[i + 2 * x]`.

When performing computations with entire Database channels, use the special `DB_xxx` mathematical commands instead of explicit formula expressions whenever possible, since they are about 10 times faster.

7 Restrictions and limitations

Command line:

- Maximum 255 characters
- Maximum 9 commands separated by @ signs
- computations with Database channels, e.g. `DB[] = f(DB[], DB[], DB[]`, may not involve more than 50 channels as argument.

Variables:

- Maximum number of common variables (`COMMON`, all code modules of a script project): 100
- Maximum number of global variables (`GLOBAL`, all code modules): 1,023
- Maximum number of local variables (`LOCAL`, per `SUB` program): 1,023

  ☞ You are restricted to a maximum of 11 variables per command line.
- Maximum number of call parameters (`PARAM`, per `SUB` program): 11
- Maximum number of array dimensions: 3
- Do not use any reserved words, e.g. `IF, GOTO, SETVALUE` etc., as names of variables.
- Variable names must not be identical to `#DEFINE` placeholders.
- Variable names must not be identical to `X` followed by a number.
- Text variables may contain a maximum of 255 characters.

Nesting, calls

A nesting is the number of identical commands inside each other, e.g. a loop inside a loop.

- Maximum 5 `#INCLUDE` levels
- Maximum 5 `DO...LOOP` levels per `SUB` program
- Maximum 5 `IF...ENDIF` levels per `SUB` program
- Nesting of `SELECT CASE...END SELECT` commands is prohibited
- Max. call sequence of `SUB` programs (CallTree): 32
- Maximum number of daughter scripts (command `RUN`): 32
- Maximum number of transactions (object `DAO`): 5
- Maximum number of DLLs: 32

- Maximum number of functions in DLLs: 1,000
- Maximum number of files when using online export in separate files: 256

Event queue, Stack

- Maximum number of entries in the event queue: 512.
- Maximum number of variables in global stack (command `PUSH/POP`): 50

Branch labels

- Maximum 50 per script

Code limitations

- Maximum number of code modules: 128
- Maximum number of `SUB` programs per code module: 256
- Maximum 32,767 characters per `SUB` program
- Maximum 32,767 characters in `MAIN` program

Commands not allowed in subroutines

```
#DEFINE
#IRQ
#INCLUDE
COMMON
GLOBAL
DLLDECL
```

Online Document, graphics

- Maximum number of pages per document: 128
- Maximum number of plots per page: 1,023

Auto Command List

- The Auto Command List may have up to 100 with `VARIABLE` declared variables.

The following commands are not permitted in Auto Command Lists:

SELECT CASE...END SELECT
IF...ELSE...ENDIF (IF...ENDIF is permitted)
ARRAY_ALLOC
ARRAY_FREE
ARRAY_STORE_xxx
ARRAY_READ_xxx
TRACE
RUN
COMMON
GLOBAL
LOCAL
PARAM
CALL
DLLDECL
DLLCALL
#IRQ
#DEFINE
CONTINUE
EXIT
:<Branch label>
GOTO
@, i.e. it is not allowed to use more than one command per line
ON ERROR
RESUME
RESUME NEXT

L Index

F

G

H

I

K

L

M

N

O

B0528-71 en

Hottinger Baldwin Messtechnik GmbH

PO box 100151 • D-64201 Darmstadt
Im Tiefen See 45 • D-64293 Darmstadt
Tel.: 06151/803-0; • Fax: 06151/8039100
E-mail: support@hbm.com • www.hbm.com

measurement with confidence