



## **Manual**

<http://www.mcps.de>

# Table of Contents

<b>Introduction</b>	<b>6</b>
Welcome.....	6
List of Functions .....	7
New functions .....	9
Basic Concept .....	10
<b>Menus</b>	<b>11</b>
Batch manager .....	11
New session.....	12
Load session .....	25
Config session.....	25
Run session .....	25
Stop session.....	26
Open batch database.....	27
Database operations.....	27
Project.....	29
New.....	29
Project parameters .....	30
Options menu / toolbar .....	40
Group Manager.....	40
Project Pathes.....	41
Project Information.....	42
AutoFile.....	43
Data Reduction .....	46
Automatic data acquisition.....	47
Alarm Settings.....	49
Project defaults .....	51
Automation.....	56
Extras.....	61
Get settings from devices .....	64
Cut, copy and paste channels .....	65
Print project configuration .....	65
Open .....	66
Open project list .....	66
Open from DA Server .....	67
Select new active project .....	67
Select record.....	68
Close .....	68
Configuration.....	68
Data.....	69
Info .....	71
Extras .....	72
Save project as template.....	72
Manage project lists.....	72
Dynamic Alarm File .....	72
Process monitoring editor.....	75
Convert AutoFile to batchfile .....	86
Create successor project.....	86
Printer Setup.....	86
Import.....	86
Import File.....	86
Export.....	88
Export Channel Data .....	88
Export Limit Exceedings .....	89
Export reference measurements .....	89
Window manager .....	90
MServ Startup Snapshot.....	91
Change own login password.....	91
Logout .....	91

Smart Quit .....	91
Quit .....	91
Display .....	92
Numerical Offline Display .....	92
Graphic: Standard Offline Display .....	94
Graphic: Analog + digital trend .....	100
Graphic: Trend of current alarm channels .....	100
Profile display .....	100
Thermo display .....	101
Reference measurement file .....	101
Logfiles .....	102
Event Logfile .....	102
Alarm Logfile .....	102
Global Alarm Logfile .....	102
Global email and sms logfile .....	102
Administrator logfile .....	102
Audittrail .....	103
Comparisons .....	104
Compare AutoFile Records .....	104
Data acquisition .....	105
DA Manager .....	105
Message Window .....	106
Alarm Window .....	107
Log Event .....	108
KEYDRV value .....	108
Update device status .....	108
Manual data saving .....	108
Test .....	109
Reference measurement .....	109
Start .....	110
Pause .....	110
Stop .....	110
Text Window .....	111
Trend Window .....	112
Long Term Trend Window .....	112
Solo Window .....	113
Bargraph .....	114
Profile window .....	115
Image window .....	116
Mathematics .....	117
Statistics .....	117
User Functions .....	118
Polynomials .....	118
New Files .....	118
Time measurements .....	119
Concept .....	119
Run .....	119
Edit .....	120
Configuration .....	121
MCPS settings .....	121
Passwords .....	121
General .....	122
Graphic .....	125
Project .....	126
Admin .....	129
Users .....	130
Directories .....	131
Messages .....	131
Network .....	133
MServ .....	134
SMS/Email .....	135
Batch .....	137





# Introduction

## Welcome

Welcome to MCPS, the modular and powerful software package for data acquisition and evaluation. MCPS was designed for long term data acquisition. It supports a great number of channels and devices. The primary instruments for MCPS are dataloggers, multimeters, powermeters, recorders, controllers, sensor modules, Modbus and OPC devices.

==> To make the first steps, please read the chapter **TUTORIALS**.

==> For instrument specific information, please read the chapter **DEVICES**.

==> Important installation guidelines are available in chapter **IMPORTANT INFORMATION**.

So these types of instruments are used for a certain kind of applications, MCPS is a software package, which meets these requirements. Here some basic features:

During the measurement most of the functions are not blocked! So data evaluation or configuration is possible, while gathering data. You can change the window layout, alarm parameters, math functions and so on.

- MCPS can handle several independent measurements at the same time.
- Alarm surveillance and messaging system (even SMS, relay outputs)
- MCPS can send data and alarms of the measurement computer to other client PCs in the network. So other users can see the online values.
- Powerful mathematical functions, statistics, polynomials and so on.
- Up to 200 different instrument can be used in one measurement (e.g. datalogger + powermeter + view recorder).
- 2D-Display of thermographical devices

# List of Functions

The MCPS software package is a set of various options. Each option provides a subset of new functions. For your application you have to order the base package and all necessary options. Please contact your local dealer (or [info@cad-computer.de](mailto:info@cad-computer.de)) to assemble the correct MCPS system for you.

## **Base package MCPS**

Import of device data (DL, Darwin, LR, OR, VR, MX) from floppy or compact disk  
Numerical and graphic display (no channel restrictions)  
Export to Excel  
Graphic export into clipboard or file  
Graphic and text printout  
Cursor measurements  
Print layouts  
Window layout system

## **Mathematical package I (Option /MATH1)**

Statistics / MKT  
Standard operators +-\*/^  
Mathematical channels  
Bittest function  
Systemvariable (SC,SS,SR)

## **Mathematical package II (Option /MATH2)**

Mathematical package I  
Compare functions  
Summations, integrations, moving average functions, Minimum/Maximum values, ...  
Logical operations  
User functions, polynomials  
Registers  
Calls to external DLL functions  
Stability and difference function  
F-Value computation

## **Device drivers (Option /DARWIN /HR /LR usw.)**

Device driver for an instrument or group of devices

## **Alarm monitoring (Option /ALARM)**

Alarm checking, protocols, messages, actions; Alarm suppression

## **Alarm remote messaging (Option /SMS-E)**

Alarm messages by SMS (mobile phone) or email  
System messages by email

## **Network online visualization (Option /MSERV)**

Online data, alarms and system/device messages on network clients. Acknowledgement of alarms on clients

## **Multi Project Expander (Option /MPE5 /MPE20 /MPE-MX)**

Several measurements at the same time!  
Independent projects  
Access to same instruments is possible

## **Batchmanager (Option /BATCH)**

Batch control window  
Batch database with search and filter functions  
Batch signing (with /ERES)

**BatchImport (Option /BA-IMP)**

Automatic import of device data files (e.g. files uploaded by FTP) into a MCPS batch data base  
Batch database with search and filter functions  
Batch signing (with /ERES)

**FDA 21 Part 11 conformity (Option /ERES)**

Electronic records and signatures  
Audittrail  
User administration with different access levels  
Specific restrictions:  
Delete key is disabled in file requester. Use backspace instead.  
User columns cannot be changed in a project containing data  
A batchfile cannot be deleted  
File masks cannot be changed in a file requester  
Math channels cannot be added or deleted in a project containing data

**Process monitoring (Option /PMON)**

Monitoring with process icons (LED, Analogmeter, Bitmaps, ...)  
Integrated editor to setup process window

**Webserver (Option /WEB)**

Display of customer HTML-pages  
Data values, bar graphs, analog meters, LEDs and trends  
Active server pages for dynamic and flexible page creation

**Thermographical display (Option /IR)**

2D display of infrared cameras  
Definitions of zones or regions of interest to determine maximum, minimum or average value  
Bitmap overlay for better orientation  
Flipping and rotating of images

**Network versions with network dongle and logins**

Only one dongle for all users  
Login from each computer in the network  
Simultaneous access to data files

## New functions

- Index saving: MCPS data is stored in two files (time stamps and channel data). This new mechanism allows data reduction per channel, adding new channels to a project with existing data and even adding new devices to those projects.
- New appearance (True color)
- Several new drivers (e.g. Siemens S7)
- Support for Vista, Server2008 and Windows7
- Integrated web server with dynamic pages (Active Server Pages)
- Start and stop of a project or batch triggered by any project channel
- Support of JPEG bitmaps
- Embedding of bitmaps in PMON or print layouts
- FileGapFilling: Automatic gap filling with device files uploaded by ftp.
- PMON: Full screen mode with fixed windows. Trends, buttons and sliders for user actions even from clients.
- Menu item to select active project
- After a restart of MCPS and a restart of the data acquisition the projects are filled up with missing time stamps to enable gap filling (manually or automatically).
- Download of device tags into the project configuration
- 15 User columns in project configuration instead of 10
- Faster offline trend display
- Mserv snapshot: The connected Mserv server, loaded Mserv projects and the window layout can be saved for automatic reload the next time the client is started.
- Export with data reduction without averaging
- 8 software alarm levels
- Batchmanager: 20 input fields instead of 8
- Print layout manager: Change of page format and orientation now possible in existing layouts

## Basic Concept

In MCPS a data acquisition, its data and its configuration is organized in a so called **project**. A project contains the configuration for channels, samplerate, comments, tags, colors, marker, mathematical functions and so on. To have access to the measured data the project should be loaded into MCPS. Once loaded all project related functions can be used (data display, printouts, export, statistics, data acquisition...). If a project is loaded into the data acquisition manager, it will collect data from the devices and channels specified in the project configuration.

MCPS allows the user to load several projects at the same time, which will be handled completely independently. So it's possible to compare data between different measurements.

A special feature of MCPS is the capability to run more than one measurement at the same time. Several independent projects can collect data from the same or different instruments (Option /MPE).

MCPS is running all data acquisitions in the background. This feature allows the user to have access to the already measured data even if the measurement is still in progress. This is very useful for long term data acquisition, because you can use the offline functions (display, printout, export, ...) on the project to make some evaluation before the measurement stops.

Several instruments can save measured data to floppy disk or memory card. For specific devices MCPS can import these files and generate compatible projects, which can be handled like all other projects for evaluation.

For online measurements MCPS can combine different interface types (RS232/ RS422A/ GPIB/ ETHERNET) and devices (recorder, loggers, powermeters, controllers, ...) in one project.

Trend data can be combined with 2D thermographical data display.

# Menus

## Batch manager

General concept

### Important:

The batch manager menu is only visible, if activated in [CONFIGURATION/ MCPS SETTINGS/ PASSWORDS](#).

The batch manager of MCPS can handle the monitoring and administration of several batches in different production or test units at the same time. A general control window contains screens for each unit to show the current batches and their status.

Unit 1	Unit 2	Unit 3
Mixer_17 (Running) Product: Rheumatism tablets Serial number: 72345/356 Customer: Pharmaceutical manufacturer Number: 20	Mixer_18 (Waiting) Product: Rheumatism tablets Serial number: 72345/356 Customer: Pharmaceutical manufacturer Number: 20	Mixer_19 (Stopped) Product: Rheumatism tablets Serial number: 72345/356 Customer: Pharmaceutical manufacturer Number: 20
Batch Stop	Batch Start	Batch Start

For each unit a project is setup, which gathers the measured data. So if five test units are available to run batches, you have to define five projects. A batch application in MCPS is called a session. All settings are saved into a session file. It's possible to create several session files to run different applications. A session file can be loaded and run automatically at startup (See [CONFIGURATION/ MCPS SETTINGS/ BATCH](#)).

Before a batch is started in MCPS you have to enter some parameters like batch name and lot number. Additional input parameters are possible, which are defined by the customer in a **batch input mask**. For example you can ask for a serial number, operator, product group and so on. These information can be displayed in the control window. Some parameters of the input mask can also be sent to the device, if supported by the instrument. The Yokogawa DX/MV series for example can handle batch name, lot number and comments.

Everytime a batch is started and data are measured, MCPS creates a new AutoFile for the related project with the batch name entered in the mask.

In the configuration of the batch input mask you can specify to create batchfiles and to use the session related database. In this case, the AutoFile, the project, the batch information and all logfiles are saved in *one* independent file. The path to this batchfile is stored in the database with the batch information. Every session has its own database. Even if there are several input mask with different target directories for the batches, there is only one database. Really it's not a database, but an index file. So there's no need to setup any complex and often slow database. Our index database file contains the batch information, which has been input before the start of a batch. With strong filter functions you can find and show batchfiles.

## New session



This creates a new session file for a batch application. Two different types of operation modes are provided:

**Online batch measurement.** Some projects for data acquisition are defined and controlled by the batch manager. The projects are started and stopped in the batch control window. After the measurement all project related data is written into the batchfile. The batch information are stored in the batch database.

**Offline or import session.** Device data files are imported automatically by MCPS and converted to batch files. Batch information from the files (if available) are stored in the batch database.

In both cases scripts and automatic printouts can be started.

### Session-Common

The screenshot shows the 'Session administration' dialog box with the 'General' tab selected. The 'Activate online batch session' checkbox is checked. The 'Mainframe' section has 'Docking position' set to 'Bottom', 'Height' set to 250, and 'Width' set to 400. The 'Unit settings' section shows 'Number of units' set to 4. Below this is a table with 4 rows, each representing a unit. Each row has a 'Set...' button, a 'Project mode' dropdown set to 'Constant project', and a 'Related project' text field containing a path like '(\$BatchDir)\BATCH\_BASIC\_PROJECT\_1.PRO'. The 'Display mode' section has 'Standard mode' selected, 'Adjust unit windows automatically' checked, 'Units per row' set to 10, and 'Use size of unit 1 for all units' checked. At the bottom are 'OK', 'Abbrechen', and 'Hilfe' buttons.

Unit	Settings	Project mode	Related project
1	Set...	Constant project	(\$BatchDir)\BATCH_BASIC_PROJECT_1.PRO
2	Set...	Constant project	(\$BatchDir)\BATCH_BASIC_PROJECT_2.PRO
3	Set...	Constant project	(\$BatchDir)\BATCH_BASIC_PROJECT_3.PRO
4	Set...	Constant project	(\$BatchDir)\BATCH_BASIC_PROJECT_4.PRO

#### Mainframe:

The batchmanager control window is docked to one side of the main window. If the option **Variable size** is active, the control window can be dragged smaller.

#### Number of units:

Defines the number of your production or test units. Each unit will have a status window in the batch manager control frame. The unit windows can be set independently in size and position or automatically.

**Hide project base windows:**

For every loaded project a small base window is shown in the workspace. If there is a session with a lot of projects, it is helpful to hide these windows. We recommend to activate this button after all settings are made.

**Settings:**

Unit specific settings.

Defines the dimension and the title of the unit window.

**Project mode:**

Normally a project is setup and used permanently in the session (*Constant project*). In some cases a new project should be created, if a new batch is configured. Then the mode must be changed to *Variable Project*. By pressing the BATCH-button a new project is created and saved.

**Related project:**

Every unit needs a project, which handles the data values. Each new batch is saved into a new AutoFile record of the related project.

**Display mode:**

Standard mode:

For each unit MCPS will show a related control window to see the status of the batch. The units can be organized in several lines, which can be scrolled, if the batchmanager window is too small. If adjusted automatically, MCPS will use the sizes in the unit option window as minimum values.

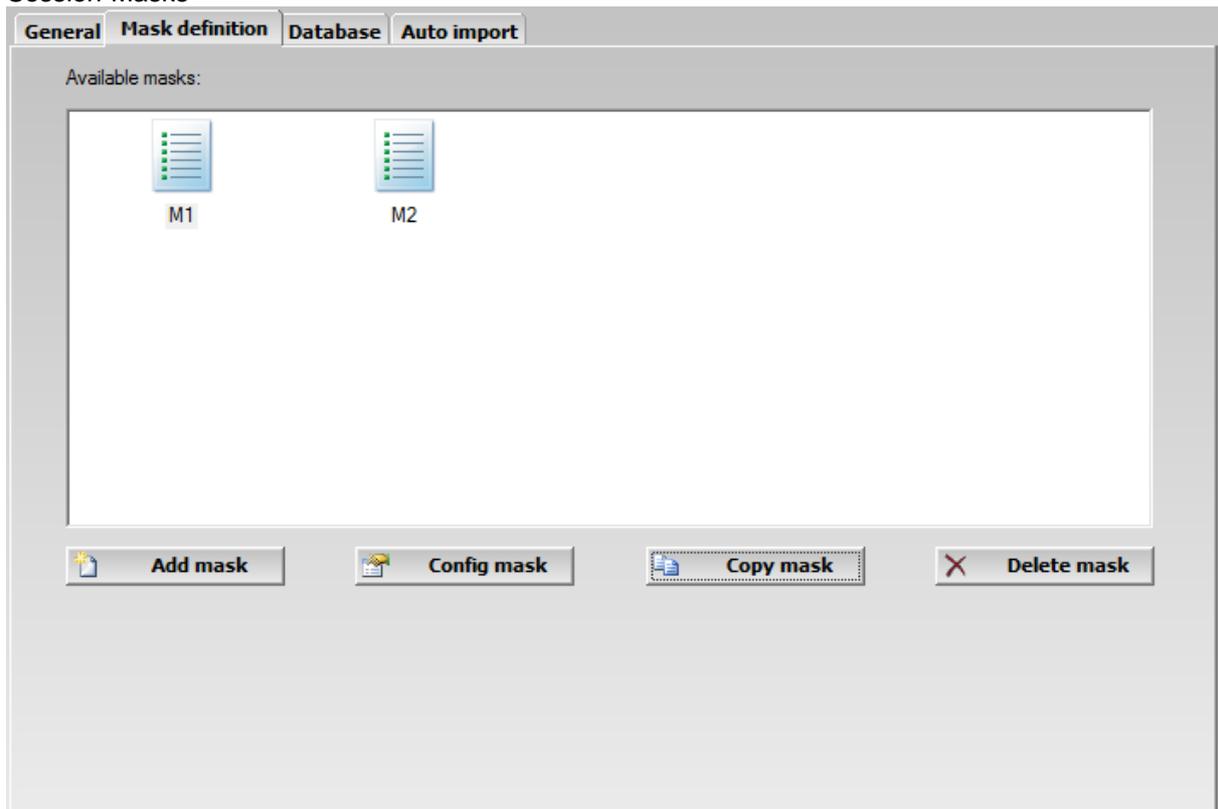
Tree mode:

Only one batch status window is visible and all units and batches are organized in an explorer like tree view. A click on a unit will show the related batch in the control window. This mode is used for a large number of batches.

Hide batch manager:

The batch manager's control window is hidden. This is only useful, if batches are controlled by script or PMON(Process monitor).

## Session-Masks



MCPS can handle several batch input mask to run different applications at the same time. Each mask is project independent and can be used for all or only specific units. The following actions on this page are possible:

- Add Maske: Creates a new and empty mask
- Config mask: Configurs an existing batch mask
- Copy mask: Creates a copy of the selected mask
- Delete mask: Removes the selected mask

For configuration see also [Batchmask-Common](#).

## Session-Database

The screenshot shows the 'Batch files' tab of the 'Session-Database' configuration window. The window has a title bar with 'Session-Database' and a tab bar with 'General', 'DX/CX', 'Automation', 'Batch files', and 'ERES'. The 'Batch files' tab is active. Inside the window, there is a dashed box containing the following settings:

- Create batch files and store batch information in database:
  - Save files in directory: C:\ProgramData\MCPST7\Batches
  - Delete AutoFile record automatically after batch file has been created
  - Link batch to internal database directory: TOLO
  - Always create batch file
  - User can reject creation of batch file
  - Creation of batch file through project channel: 1

If a batch mask is configured to create batchfiles, the batch information and file links are saved in the batch database file. Normally this file is saved in the same directory as the session. In case of network sharing (Client/Server applications) it might be useful to save the database to a different location.

All batchfile information can be stored to one database file or into monthly or yearly created files. If a database file is renamed outside MCPS, the software will create a new file and the renamed file can be loaded manually.

## Session-Auto import

### Activate automatic file import:

This checkbox must be activated to run the automatic file import, when the session is started.

To import versatile files in different directories and to start various actions (script, printing) you have to define one or more so called rules. A rule describes a set of import settings. Press <Config rule> or double click on the rule icon to setup the rule.

### Auto import – General

The screenshot shows the 'Import rule definition' dialog box with the 'General' tab selected. The dialog is titled 'Import rule definition' and has a close button (X) in the top right corner. It contains several sections for configuring import rules:

- Common settings:** A checkbox labeled 'Active' is checked. The 'Name' field contains 'R1'.
- Search paths:** A table with columns '#', 'Path', 'SubDir', and 'Mask'. The first row has '# 1', 'Path s:\ftp\dx', 'SubDir' (checkbox), and 'Mask \*\*'. Below the table are 'Add path' and 'Remove path(es)' buttons.
- File type:** Two radio buttons: 'Binary file format with device id' (selected) and 'ASCII / CSV file'. An 'Import settings...' button is to the right.
- Batch file settings:** 'Destination directory' is 'd:\mcpbatches'. 'Directory in database tree' is empty. 'Batch name' has two radio buttons: 'Original file name' and 'Specific filename' (selected), with the value 'SB-\$YSMSD'.
- Settings for device file:** Two radio buttons: 'Remove file after import' and 'Move file to:' (selected), with the value 'd:\dxabackup'.
- Default project for import:** A checkbox 'Use default project:' is unchecked.
- Joining of batch files:** A checkbox 'Join batch files' is checked. 'Joining criteria' is a dropdown menu set to 'Daily files'.

At the bottom right, there are three buttons: 'OK', 'Abbrechen', and 'Hilfe'.

- Active:

Activate this rule

- Name

Name of the rule

- Search paths

One or more directories to look in for device files to import. If the checkbox <SubDir> is active, MCPS will also search in subdirectories.

- Destination directory

After import MCPS creates a batch file, which is stored in this directory.

- Directory in database tree

The MCPS database itself (not on disk) can have subdirectories to arrange the batch files. With different rules it is possible to order the imported files related to their meaning or application.

- Batch name

The new batch file must have a name. This can be the original filename from the import file or specific filename defined in the text field. This can be a new text with additional time information, which is set with the following control codes:

\$B Batch name (is imported from file)  
\$L Lot number (is imported from file)  
\$Y Year  
\$M Month  
\$D Day  
\$H Hour  
\$I Minute  
\$S Second

- Setting for device file

After the import the original file must leave the search directory. It can be deleted or move to a backup directory.

- Default project

A project with default settings (Tags, comments, colors, ...) can be defined to be used for import. This project configuration is taken and put into the batchfile. The best way is to import a device file manually, change all necessary settings and save the project. Then define here the path to that project.

- Joining of batch files

All batch files of the same device (serial number) are concatenated to one MCPS batch file until a trigger condition is met and a new MCPS batch file is created. The trigger can be time based or message based. Time based trigger will split the data according to the setting. For example a daily trigger will put all data of one day into one batch file. If necessary MCPS will split the device data file in two or more parts. The message trigger finishes joining of device data files, if the trigger message is found in the messages of the device file.

An automatic printout is done after a MCPS batch files is completed (e.g. at midnight).

## **Auto import - Printouts**

### **Graphic printouts:**

#### **- Automatic printout after import**

An automatic trend printout after batch import is started. The option button defines, which channels, window and print layouts should be used. If batches are joined, then the printout is made only at the end of the total batch.

#### **- Numeric printout**

An automatic text printout after batch import is started. If batches are joined, then the printout is made only at the end of the total batch.

#### **- Automatic printout**

The file is automatically printed after import to the specified printer.

#### **- Use every n scan**

To reduce the amount of paper only a part of the data can be printed.

#### **- Use print layout**

The selected print layout is used for printing.

## **Auto import - Automation**

The specified script is started after the file is imported and before any printout is made.

You can add several text items, which are shown in the batch information window after automatic import. For each item you can add a related text by selecting <info and signature>.

## **Auto import - ERES**

If the ERES option is available, it is possible to define up to three signing levels. These levels are stored in the batch file and can be used to sign the batch data.

## Batchmask-Common

**General** | DX/CX | Automation | Batch files | ERES

**Basic definitions:**

Mask name:   Available for all units  Available for specific units

Batch preset text:  Statisch  Disable batch input field

Lot number:  Use  Auto increment  Reset at midnight  Disable input field

Batch filename definition:

\$B = Batch name, \$L = Lot number, \$U = Unit, \$Y = Year, \$M = Month, \$D = Day, \$H = Hour, \$I = Minute, \$S = Second

**Optional inputs:**

#	Active	Display	Text	List	Items
1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Customer	<input type="checkbox"/>	...
2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Serial number	<input type="checkbox"/>	...
3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Order number	<input type="checkbox"/>	...
4	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	...
5	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	...
6	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	...

Do not use input values of last batch  All input fields must be filled

**Optional settings**

Skip mask input window  Disable >Add next batch< button while batch is running

**Mask name:** Defines the name of the mask (not the batch) to distinguish between several masks.

Available for all/specific units:

The mask can be available for all or only specific units.

**Batch preset text:** Normally the basic batch name is requested, when a new batch is added into a unit. The name can already be preset here.

**Disable batch input field:** The batch input field is locked. Useful for automatic batch start.

**Lotnumber:** Additional number to identify batches.

**Auto increment:** MCPS is counting up the lotnumber automatically.

**Reset at midnight:** Starts again with 1

**Disable input field:** The lotnumber input field is locked. Useful for automatic batch start.

**Batch filename definition:** Defines the name of the AutoFile-record of the project. If the database is activated, this is the name of the batchfile.

**Optional inputs:**

If set to <Active>, these inputs will appear in the batch input mask, when a new batch should be added. If the <Display> button is on, this information is also shown in the unit status window. In the column <Text> you can define your own parameter description. Normally a text input field is shown in the input mask to enter the related item text. If the button <List> is active, a dropdown list with predefined items is displayed instead. Click on the ellipsis button <...> to create or modify selection table.

**Do not use input values of last batch:**

If a new mask window opens, the old settings are set as default. If this button is active all input fields are empty.

**All input fields must be field in:**

The batch input window will not proceed until all input fields are filled in.

**Skip mask input window:**

This feature suppresses the batch mask input window. The mask must be set to create unique batchfile names (increasing lotnumber or date/time information).

Disable <Add next batch> button while batch is running:

Normally you can setup the next waiting batch in a unit, while a batch is running. To suppress this feature, activate the check button. Useful, when scripts are started to configure the instrument after the batch mask has been called.

## Batchmask-DX

Batch and lotnumber can only be sent to the device, if the recorder provides the batch option. Start / stop command will start and stop the data acquisition of the recorder. If an user event is entered for a batch and saved within the project, the text can also be sent to the recorder. MCPS will use the recorder's message text 8 to send a start or stop message.

Three additional inputs can be entered in the batch mask, which can be sent to the recorder to set comment 1 to 3.

## Batchmask-Automation

The screenshot shows the 'Automation' tab of the 'Batchmask-Automation' dialog box. It features two main sections: 'Printouts' and 'Scripts'.

**Printouts:** A checkbox labeled 'Automatic printout after batch is stopped' is currently unchecked. To its right is an 'Options...' button.

**Scripts:** A table with two columns: 'Script type' and 'Script path'. The first row is checked and has a path. The other rows are unchecked and have empty paths. Each row has a three-dot menu icon on the right.

Script type	Script path
<input checked="" type="checkbox"/> Before batch mask:	C:\ProgramData\MCP57\Scripts\dialog.mbs ...
<input type="checkbox"/> After batch mask:	...
<input type="checkbox"/> DAQ start script:	...
<input type="checkbox"/> DAQ stop script:	...
<input type="checkbox"/> Remove waiting batch	...

MCPS can save predefined window and print layouts in the batchfile. When a batchfile is printed from the database (right mouse button on batch), these settings are used automatically to have equal printouts. The printing can be done also at the end of a batch.

To improve batch processing MCPS can run several customer scripts:

- Before batch mask: The script is started before the mask becomes visible. The input fields can be preset e.g. by evaluating external data.
- After batch mask: When the user presses OK this script is called to verify inputs or configure some project parameters (e.g. a triplines related to one input field)
- DAQ start script: This script is called after the start button has been pressed.
- DAQ stop script: This script is called after the stop button has been pressed.
- Remove waiting batch: Called after a waiting batch has been removed by the user to reset project or device parameters

Requires option (/SCRIPT)

## Batchmask-Batchfiles

The screenshot shows the 'Batch files' tab of a configuration window. It contains several options for batch file creation and storage:

- Create batch files and store batch information in database:
  - Save files in directory: C:\ProgramData\MCP57\Batches
  - Delete AutoFile record automatically after batch file has been created
  - Link batch to internal database directory: TOLO
  - Always create batch file
  - User can reject creation of batch file
  - Creation of batch file through project channel: 1

### - Create batch files and store batch information in database

This window enables the database functionality. MCPS will create special batchfiles, which contain the project, the batch information, the data, logfile and alarm infos in one file. The batch information and the path to the batchfile are stored in the session dependent database file (extension .bdb). If a session is loaded and a database file exists, it will be loaded into the database view on the right side. It's possible to load a database from another session or on a client computer without the option /BATCH. In the last case you should also have a look at [CONFIGURATION/ MCPS SETTINGS/ NETWORK](#).

### - Save files in directory

The path were to store the batchfiles

Delete AutoFile record automatically after batch file has been created

During the measurement the data is saved to the normal AutoFile of MCPS. After the stop, the AutoFile data is copied to the batch file and in most cases not necessary anymore. The AutoFiles can be deleted manually or with this button automatically.

Link batch to internal database directory

MCPS can handle one sublevel (directory) in the database to organize batchfiles. By settings this text field as a directory name, the batch filenames created by this mask are put into the related folder. This does not effect the saving procedure of the batch files themselves.

#### Batchfile creation

- Always create batchfile: When stopping the batch the batchfile is created and the database is updated. There is no way to prohibit this procedure.
- User can reject creation of batch file: When stopping the batch a check box can be activated to skip the creation of the batchfile
- Creation of batchfile through project channel: If the value of the project channel is not equal to 0, when the batch is stopped, the batchfile is created. Otherwise the procedure is skipped.

#### Batchmask-Eres

Level:	Description:
1:	User
2:	QA
3:	Supervisor

With the option /ERES you can sign batchfiles with 3 different levels. The permission, who can sign which level, is defined in the user administration ([CONFIGURATION/MCPS SETTINGS/ USER](#)). A batchfile is signed in the database window (See [DATABASE OPERATIONS](#)).

## Load session



Loads a batch session into MCPS. If a related database file is available, it is loaded too.

## Config session



Configures a session, which already has been loaded or created. See [NEW SESSION](#). Not possible, if the session is running.

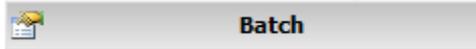
## Run session



(Only with option /BATCH)

Starts a session, which has been loaded or created. All projects of the session file are loaded and started. The device information are received from the instruments and basic communication is tested. The batch control window with the units' windows is opened.

To run a session automatically see [CONFIGURATION/MCPS SETTINGS/BATCH](#).



Batch configuration and action button. If the button is pressed the first time, the batch input mask is opened to define the first batch. The next time a batch configuration window is displayed to perform several actions. These actions depend on the current state of the batch.

### Change settings

Changes the batch information even during and after the measurement. Once the batch is started, the batch and the lotnumber cannot be changed, because they are used for the name of the batchfile. If the option /ERES is installed only empty fields can be modified after the start.

### Log event

An event comment can be entered, which is saved with the project and thus later on with the batchfile. It's the same feature as [DATA ACQUISITION/ LOG EVENT](#), which is disabled for batch projects. So the user do not need to know, which project he has to activate to enter a batch related comment.

### Display trend

This will display the default window layout for data acquisition or ,if not defined, the online trend window. This can also be done by a double click in the unit window.

**Sign batch**

With the option /ERES the batch can be signed after the measurement as defined in the batch type definition.

**Add next batch**

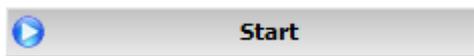
After one batch is running the next waiting batch can be configured.

**Remove (next) batch**

Removes the waiting batch from the queue.

**Test measurement**

A waiting batch can be tested before the data acquisition is started.



Starts a waiting batch. If the project is started by external inputs, the START button is disabled.

**Stop session**

Stops the session. All running batches must be stopped before.

## Open batch database



Loads a batch database, which is different from the current session, or if no session is loaded. On a client computer you can open the database of the measurement computer. To simplify this action, you can set a default directory under **CONFIGURATION/ MCPS SETTINGS/ DIRECTORIES**. One problem occurs, if the path of a batchfile seen from the daq computer is not the same as from the client computer, i.e. the batchfiles are save in C:\Program files\MCP6\batches. The client will normally access this directory for example with \\daqcomputer\c\ Program files\MCP6\batches. So if the batchfile isn't stored on a server, which is seen from the measurement computer and from the client with the same path, we have to exchange the pathes of the batchfiles in the database. This can be done with **CONFIGURATION/ MCPS SETTINGS/NETWORK**.

MCPS looks every 30 seconds on the database file to see, if there are made any changes.

## Database operations

1) If the right mouse button is pressed on the main filter **<ALL>**, the following actions can be performed:

### **Add directory:**

A folder can be added to the database tree. Batch files can be moved to this folder.

2) If the right mouse button is pressed on an **empty directory**, the following actions can be performed:

### **Add directory:**

A sub folder can be added to the database tree.

### **Remove directory:**

The folder is deleted.

3) If the filter button  is pressed, the following actions can be performed to all batches with the check box on:

### **Exclude batches for external storage:**

All marked batch files are moved to common directory. A comment can be entered, which is added to the batch information in the database. Each moved batch is marked with a small CD-icon. The comment can be used to specify where to find the data. The batch files can be put on a server or external media (CD, DVD, ...). If such a file should be loaded again, you have to click on the file as always. MCPS then asks for a directory to find the file and loads the data even from an external drive.

In the filter settings you can define an amount of total disk space the result should be limited to, so you can easily put all files in one step onto a CD or DVD.

### **Remove batches from database and delete batch files**

All marked batch files and related information are completely deleted.

### **Move batch files to different drive or directory**

All marked batch files are moved to a new harddisk or server location. The link in the database to the batch files are also updated.

### **Change directory of batch filenames in database:**

For all marked batches the link to the batch file is updated with a new directory name. E.g. the new path should be \\Server1\data, then a link in the database is changed from c:\Data\batches\M2354.baf to \\Server1\data\M2345.baf. The batch file itself is left untouched.

### **Move batch items to specific database tree folder:**

This action affects only the database and is used to have a better batch overview. The marked batches are moved to and listed in a different folder.



# Project

## New

With PROJECT/NEW the user creates a new project for online data acquisition. To import data see [Project/Import](#).

A project is the basic object to define and configure a measurement.

There are two principle parts in a project:

[Standard parameters](#) as sample rate, groupmanager, AutoFile etc.

[Channel parameters](#) (Settings for each channel)

The channel parameters are organized in a list.

Often a new project should not be completely empty. A template should be used to have most of the parameters already configured. To define a template see [PROJECT/EXTRAS/SAVE PROJECT AS TEMPLATE](#). If templates are available, MCPS shows a window with all templates, which are created locally on the computer. Additionally an explorer item is visible, to load a template from any desired location.

## Project parameters

### Standard Parameters

#### - Sample rate:

The sample rate determines the time interval data is received from the devices, displayed and normally saved. The unit is seconds (S), minutes (M) or hours (H). To influence the kind of data saving see also chapter [data reduction](#) .

#### - Save mode:

This defines, how the measured data is saved to disk:

**Standard:** In this mode all data and time stamps are saved into one data file. This is the classic MCPS saving mode. The disadvantage of the saving type is, that after a project contains data no channels or devices can be added. The project must be copied or the data must be deleted. Only in empty projects channels can be added.

**Index:** The data is saved in to a data file, time stamps and data indices to an index file. Advantage of this mode is, that the data file can contain data blocks with different sizes.

For each channel a data reduction can be defined. The channel is not saved with every scan, but for example with every 10<sup>th</sup> scan or only if the value changes (out of deadband). This decreases the size of data files.

Channels and devices can be added to projects with already existing data. Just add new channels. This is often possible during measurement.

#### - Save:

The parameters of the project will be checked and the project is saved. A file request box allows the user to specify a name. A project, that is already loaded into MCPS cannot be overwritten. MCPS creates several files for a project, a configuration file, data file and some other ones. To copy a project use [PROJECT/DATA/COPY PROJECT](#).

#### - +1:

Adds one channel at the end of the channel configuration list.

#### - +?:

Adds a user defined number of channels at a selectable position.

#### - Delete:

Deletes all marked channels (see [blockoperations](#) ). Math channels can be deleted everytime, raw data channels only, if the project contains no measured data.

## Channel Parameters

#	Device	Channel	Unit	Tag	Comment	Color	YMin	YMax	.*
<input type="checkbox"/> 1	TESTDRV - 1	6	°C	Temp 1	Room 1		-5	45	1
<input type="checkbox"/> 2	TESTDRV - 1	2	%RH	Hum 1	Room 1		0	100	1
<input type="checkbox"/> 3	TESTDRV - 1	3	HP	Pres 1	Room 1		-5	15	2
<input type="checkbox"/> 4	TESTDRV - 1	8	°C	Temp 2	Room 2		-5	45	2
<input type="checkbox"/> 5	TESTDRV - 1	5	%RH	Hum 2	Room 2		0	100	2

All channel parameters are organized in a list. Each line contains a so called **project channel**, which can be a row data channel or a math channel. The project channels are distinguished by a consecutive number(#), because several devices with the same channel name or number are possible in a project. In all MCPS windows this number is the main reference.

### - # (Consecutive number)

Unique number for a project channel. Each line contains a check box at the beginning to be marked for **block operations** or deleting channels.

### - Device

One of the devices configured and set active in the device manager can be selected for data acquisition. The devices are listed with device name and device specifier. Inactive devices are not visible. The device MATH is device independent and declares this line as a computation channel. It should contain a math formula to deliver results. A right mouse button click opens a window with all devices and comments defined in the device manager.

### - Channel

Each device provides a set of channels as defined in the device manager. The channel names are very instrument specific and can change, if the device button is set to another instrument. Even options of the devices like mathematic channels should be set correctly to get all possible channel names.

Only those channels are saved, which are listed in the project. Pure math channels are never saved. They will be always computed during offline display. This feature allows the user to change or add math formulas even after a measurement.

### - Unit

Unit of the channel. If this field is left blank or cleared, MCPS tries to get a unit information before the measurement starts and sets this field.

### - Tag

Field for identification.

### - Comment

Field for identification.

### - Color

Color of the channel curve in the graphic windows

### - YMin/YMax

For each channel in the graphic window the user should define the scaling for the y-axis.

YMin = bottom value    YMax = top value

### - .\*

Number of digits to display after the decimal point. For empty projects this button can be set to **A**. MCPS tries to receive the appropriate information from the instrument before the next data acquisition.

### - Text

This button defines, if a channel is displayed in text windows.

### - Gr

This button defines, if a channel is displayed in graphic windows.

### - Dig

When selecting the [graphic display A+D](#), each channel with the **Dig** button set is shown in a special digital window.

### - Exp

Channel data should/should not be exported.

### - Marker

Each channel curve can be individualized by additional information like markers, number of the project channel (#), name of the device channel or the text in the comment or tag field.

### - Math

This input field contains a math formula (e.g.  $X1 + X2 * X3 - 1.5$ ) to modify the row data values or to compute new values. If the formula is in a line with a regular device and channel, then the row data value is replaced by the result of the math function. This is useful to scale the row data values or to reproduce the original physical values (from voltage input to pressure). A complete description about the math functions can be found in the chapter [Mathematics](#). Maximum number of characters is 512.

### - M-On

Enables or disables the math formula. Only active formulas will be checked for syntax errors. If a formula of a math channel is disabled, this channel won't be deliver any values.

### - Alarm

Basic settings							Alarm option						
Level	Type	Value	Color	DR	AF	Priority	Alarm delay	Hysteresis	Logging	Action	Set device	Options	
<input checked="" type="checkbox"/>	1: L	2	[Red]	<input type="checkbox"/>	<input type="checkbox"/>	1	0	...	0	<input type="checkbox"/>	Set	<input checked="" type="checkbox"/>	Set
<input checked="" type="checkbox"/>	2: L	4	[Red]	<input type="checkbox"/>	<input type="checkbox"/>	2	0	...	0	<input type="checkbox"/>	Set	<input checked="" type="checkbox"/>	Set
<input checked="" type="checkbox"/>	3: H	6	[Red]	<input type="checkbox"/>	<input type="checkbox"/>	3	2	...	0.5	<input checked="" type="checkbox"/>	Set	<input checked="" type="checkbox"/>	Set
<input checked="" type="checkbox"/>	4: H	8	[Red]	<input type="checkbox"/>	<input type="checkbox"/>	4	5	...	0.2	<input checked="" type="checkbox"/>	Set	<input checked="" type="checkbox"/>	Set
<input type="checkbox"/>	5: H	0	[Red]	<input type="checkbox"/>	<input type="checkbox"/>	5	0	...	0	<input type="checkbox"/>	Set	<input type="checkbox"/>	Set
<input type="checkbox"/>	6: H	0	[Red]	<input type="checkbox"/>	<input type="checkbox"/>	6	0	...	0	<input type="checkbox"/>	Set	<input type="checkbox"/>	Set
<input type="checkbox"/>	7: H	0	[Red]	<input type="checkbox"/>	<input type="checkbox"/>	7	0	...	0	<input type="checkbox"/>	Set	<input type="checkbox"/>	Set
<input type="checkbox"/>	8: H	0	[Red]	<input type="checkbox"/>	<input type="checkbox"/>	8	0	...	0	<input type="checkbox"/>	Set	<input type="checkbox"/>	Set

External device alarms

Check external alarm [Red]   9 0 ...  Set Set

Dynamic alarm file settings

High limit: [Red]   10 0 ...  Set Set

Low limit: [Red]   11 0 ...  Set Set

OK Help Options... Cancel

The alarm button enables the user to define several alarm levels and resulting actions. In the base package MCPS supports 8 alarm levels to change the color of a channel in text or graphic windows. Additionally an alarm can trigger the [AutoFile](#) or the [Data Reduction](#) function.

The Option /ALARM enables the following features:

Detection of external alarms (Alarms of devices can be examined)

Alarm logging (Alarms are written into an alarm logfile of the project and/or into the global alarm list)

Alarm actions (Alarm messages, relay output, SMS, ...)

Dynamic alarm files

- Level:

MCPS will check alarm conditions starting with level 1 and continues with 2,3, ... For color change this means, that level 2 has a higher priority than level 1.

- Type:

H (High alarm) Alarm, if value is above limit

L (Low alarm) Alarm, if value is below limit

DH (Delta high) Alarm, if difference between value and last value is above limit (High gradient)

DL (Delta low) Alarm, if difference between value and last value is below limit (Low gradient)

DHS (Delta high saved) Alarm, if difference between value and last saved value is above limit (High gradient)

DHS(Delta low saved) Alarm, if difference between value and last saved value is below limit (Low gradient)

ERR Alarm, if any error is detected (Overflow, invalid data, ...)

OFL Overflow alarm

UFL Underflow alarm

H+OFL Alarm, if value is above limit or overflow

L+UFL Alarm, if value is below limit or underflow

ND Alarm, if channel returns <No data>. (E.g. INV in math formulas)

DHS and DHL are only useful, if the data reduction is in use and not every scan is saved to disk. Then this types are important, if the input signal is drifting slowly higher or lower. DH and DL will not generate an alarm, because the difference to the latest scan is not great enough, but the difference to the latest saved scan might be.

External alarms Alarms, which are created by the instruments themselves and reported to MCPS.

Dynamic alarm file For the [Yokogawa-Darwin](#) system see the related driver section. A reference curve is used for alarm checking. [See here](#).

- Value

The alarm limit value

- Color

The channel color for graphic or numerical display in case of alarm

- Priority

If several alarms are active, the color of the alarm with the highest priority (highest value) is used in trend windows or for LEDs in the process window.

- DR

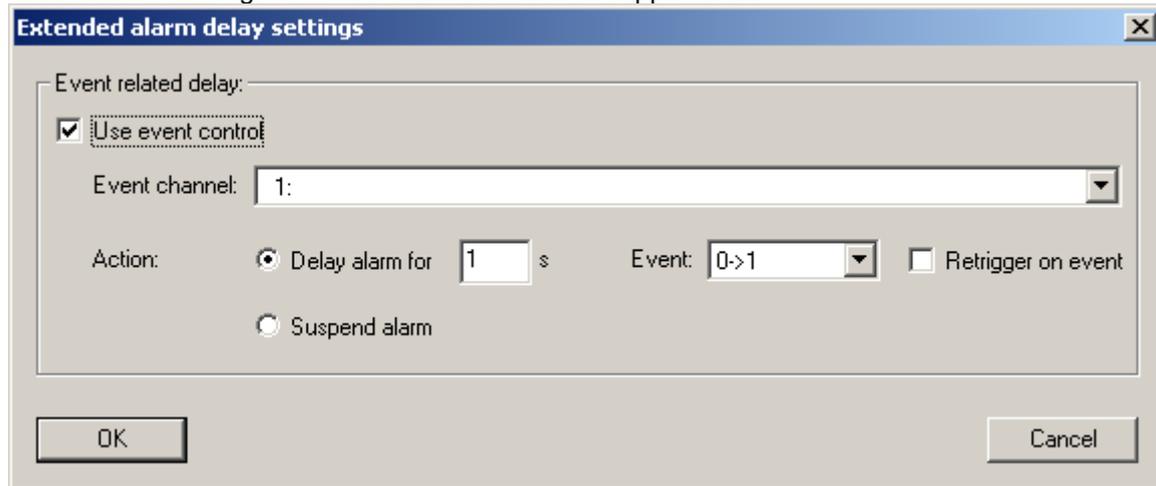
The **data reduction** can be controlled by alarm events. If an alarm occurs and the related button DR is set, MCPS will store the measured data every scan.

- AF

The **AutoFile** function can be triggered by alarm events. If set, an alarm will generate a new data file to store the measured values.

- Alarm delay

This value determines, how many seconds an alarm must be active to generate an alarm event in MCPS. This can avoid alarm events, if the input signal contains often short peaks. The option button allows further settings to control alarm detection or suppression:



a) The alarm can be delayed, if the value of an other channel follows the event condition. If the event condition happens, the timer is started and suspends any alarm for this channel and this level until the timer is inactive again. There are several events to start the timer . With <Retrigger> the timer can be restarted, if the event has been detected again before the timer has been finished.

Trigger events

0->1: Timer is started, when event channel value changes from 0 to 1

1->0: Timer is started, when event channel value changes from 1 to 0

0->1 or 1->0: Timer is started in both cases

1: Timer is started, when event channel is 1

b) The alarm can be suspended, while the event channel is 1.

Both types are often used in conjunction with digital input channels, which reflect the status of a machine, door, and so on.

#### - Hysteresis

An alarm is detected, if the channel value is above/below the limit value (and the delay time is over). By default the alarm becomes inactive, if the value is lower/higher than the limit. If a hysteresis value was entered, the alarm is still active until the value is lower/higher than the limit and the hysteresis:

Example: High alarm at 10, hysteresis 2-> Alarm becomes active above 10, Alarm inactive below 8

#### - Logging

If this button is set, the alarm (channel, level, type, message) information is saved to disk. There are two possible target files. First into the project's alarm file (default) and/or second into a global alarm list.

The global alarm list will contain all alarms of each project. See [MCPS Settings/General](#) for more information.

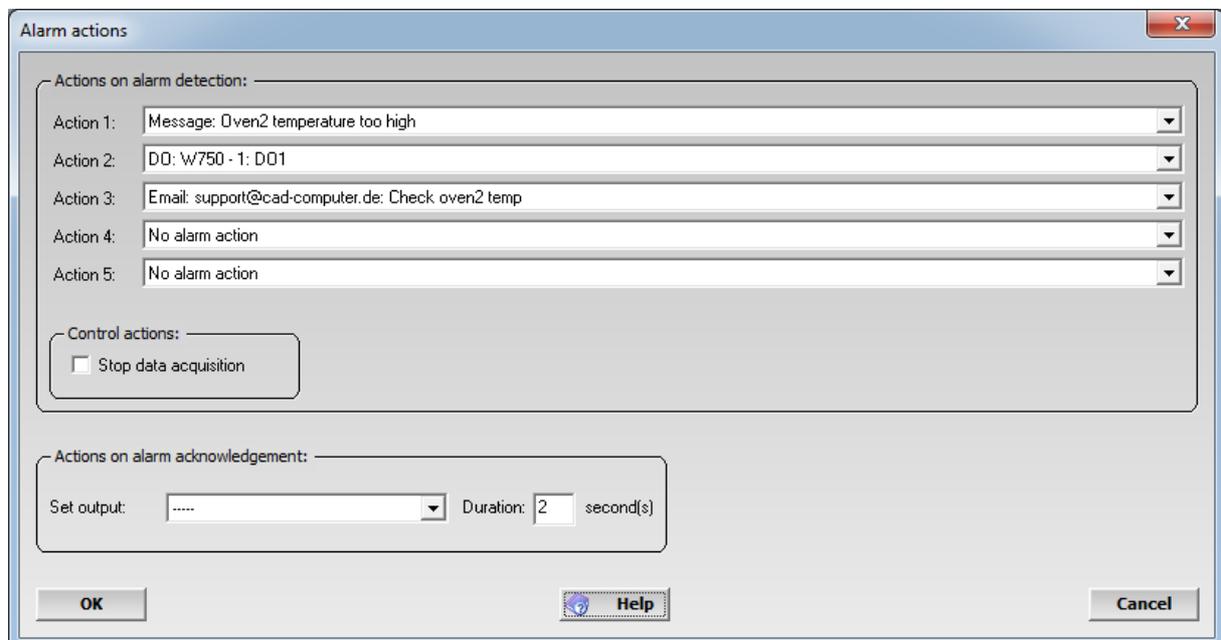
The alarms, which have been logged into the project's alarm file can be listed using menu [Display/Logfiles/Alarm logfile](#). The global alarm list is shown with [Display/Logfiles/ Global alarm logfile](#).

#### - Action

MCPS can respond to an alarm with several actions. The actions themselves must be defined in the [alarm setting window](#) of the project. The SET button allows you to select up to 5 actions from the alarm action list.

Messages, which are defined as alarm action, will be displayed in the [Online-Alarm-Window](#). All alarms of all projects are listed and will stay there until acknowledged by the user.

Only alarms with an alarm message will be put into the online alarm window!



#### - Set device

The alarm state, type and value of the project configuration is sent to the device (related channel), when a measurement is started or the parameters are changed during the data acquisition.

#### - Options

The alarm start time can be taken with the exceeding the limit or after the alarm becomes valid (after the delay time)

The alarm status can be reset immediately, when the acknowledge button is pressed. This is only useful together with alarm delay. A specific delay time can be defined, which is active after the alarm has been acknowledged.

An active alarm is normally set to inactive, when the data from the device is invalid due to communication failure. To hold the alarm status, the check box must be set. This is in particular useful, when the alarm triggers the AutoFiles to avoid unwanted files.

Infolink can point to a document, PDF or Html file to hold instructions for the user in case of alarm. The online alarm window contains the column <Info>. If the info link is set, an icon is displayed in this column. By clicking on this icon, the related file is opened.

### Define alarm groups

MCPS supports 26 global alarm groups, which are project independent. Each alarm level can belong to one or several alarm groups. In the user administration (see DATA ACQUISITION/ALARM WINDOW) each alarm group can be allowed for each individual user. This mechanism is used to hide specific alarms of a project to certain users. So only the alarms of the allowed groups are visible in the alarm online window.

### Channel global alarm options:

- Output of alarms to analog outputs

The bit combination of active alarms is sent to the configured devices.

### - Reference measurement (Calibration)

(Not visible by default, see OPTIONS/EXTRAS/USER COLUMNS)

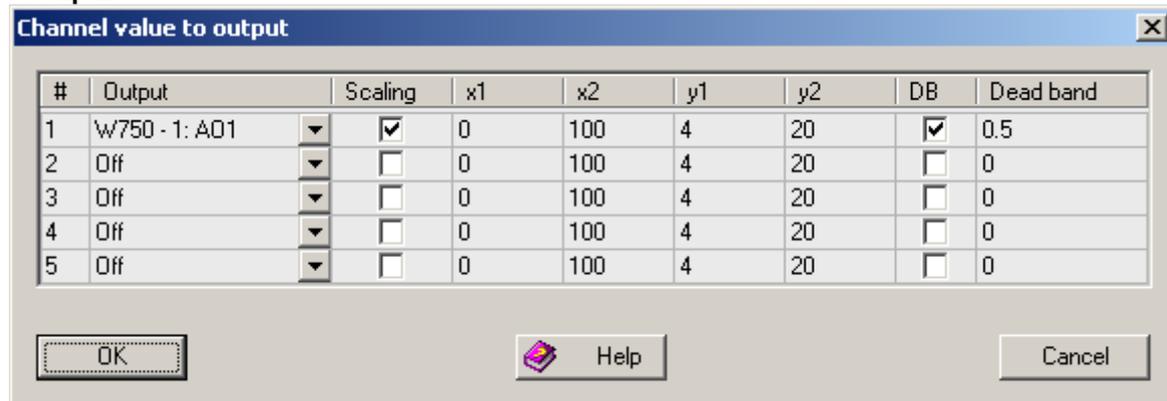
Defines how to correct the raw data value, if a **reference measurement** has been made:

OFF: No correction

Raw data: The raw data offset during the calibration is subtracted.

Math data: The math result offset during the calibration is subtracted.

### - Output



Some devices can receive data to output voltage (analog output modules) or to use the data in their systems (OPC). If MCPS finds a device in the device manager, which supports analog output channels, these outputs are listed here. The value of the related MCPS channel is send to the device. Because even math channels can be output, complex control patterns are possible (See user curves). In most cases the output signal is not equal to the channel value. Therefore a scaling can be defined to convert channel data range (x1/x2) to output value range (y1/y2). The image above can be used to convert a temperature from 0°C to 100°C into 4-20mA.

To avoid permanent output in case of stable channel values a deadband can be defined. If the difference between new value and last value sent to device is equal or higher than the deadband the new value is sent.

### - User columns

It is possible to add 15 user columns in MCPS to be used for additional channel information (See **OPTIONS/EXTRAS/COLUMNS**). The enabled columns will be shown with their header text. According to the type of input field, a check box, a list window or a text field is visible.

### - Data

On: Data is shown and stored normally.

Off: Data is set to <no data>.

Hold: Data of last scan is used again.

These settings are helpful if a device or sensor calibration is done during the measurement. To avoid getting strange data the affected channels can be disabled or set to hold.

### - Log

Logarithmic data display of channel data. The scaling limits YMIN and YMAX represents the exponents!

**- ST**

SaveType. In case of **index saving**, the you can define a saving factor (FCT) or a dead band (DB). See below STV to set the related value. FCT 5 means, that every 5<sup>th</sup> scan the channel data is saved. A DB of 0.2 means, that the new channel value must differ more then 0.2 from the last saved valued to be stored.

**- STV**

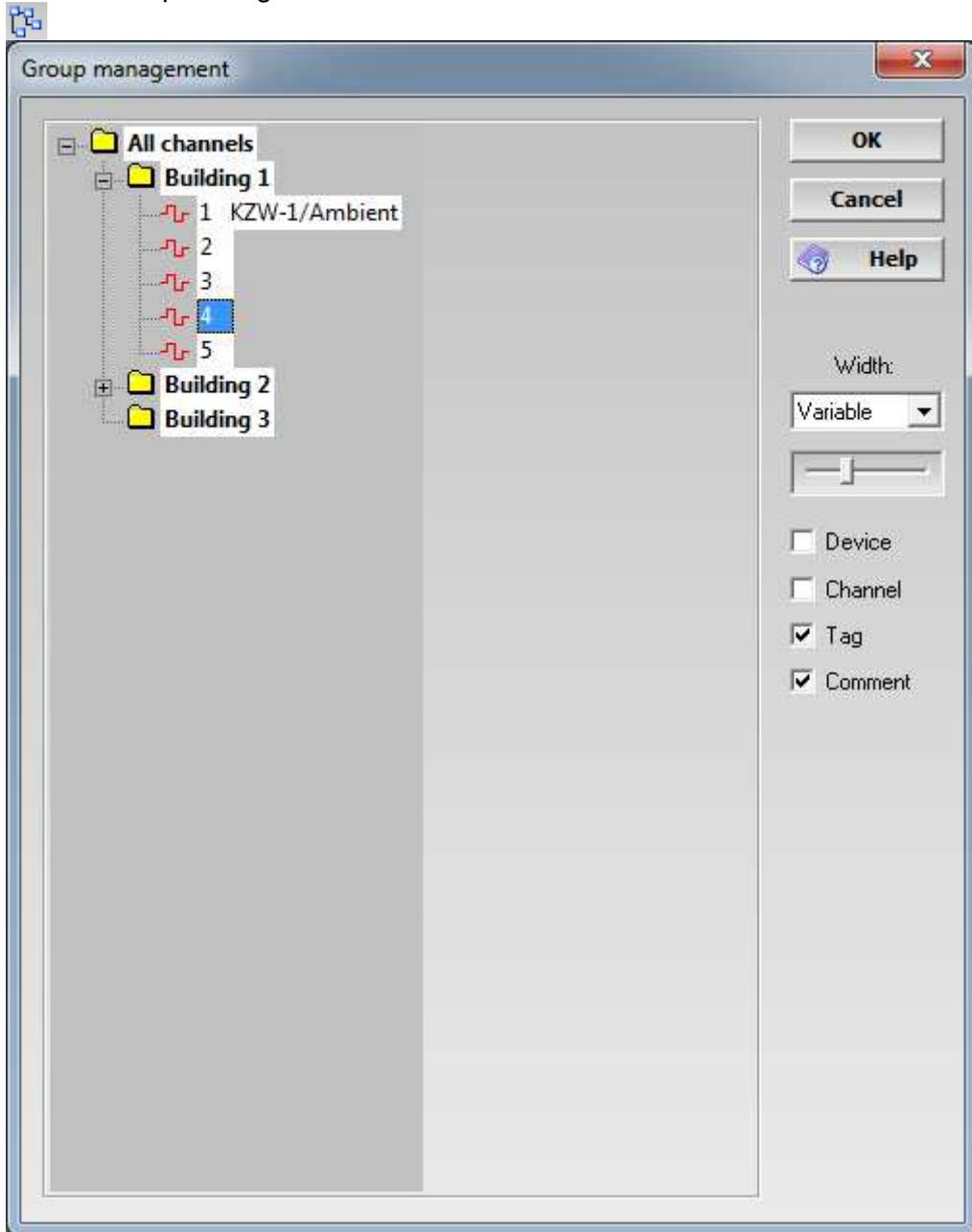
SaveTypeValue: The related value for the selected SaveType. See above.

### Block Operations

Often it's necessary to change a parameter of several channels to the same value. To avoid modifying each line MCPS provides block operations to change the settings in one step. First all lines, that should be modified, must be marked with the check box at the left. This can be done with a mouse click on each button or with the **select button #**. The user can enter a list of project channels (e.g. 1,3,5-7,12-20,33), which should be marked. If channels are defined in the group manager, a group can be used to mark channels. Afterwards the title button of the parameter to modify (MARKER, DEVICE, TAG, ...) should be clicked. A window appears to enter/select the new value. As a result all marked lines will be changed.

## Options menu / toolbar

### Group Manager



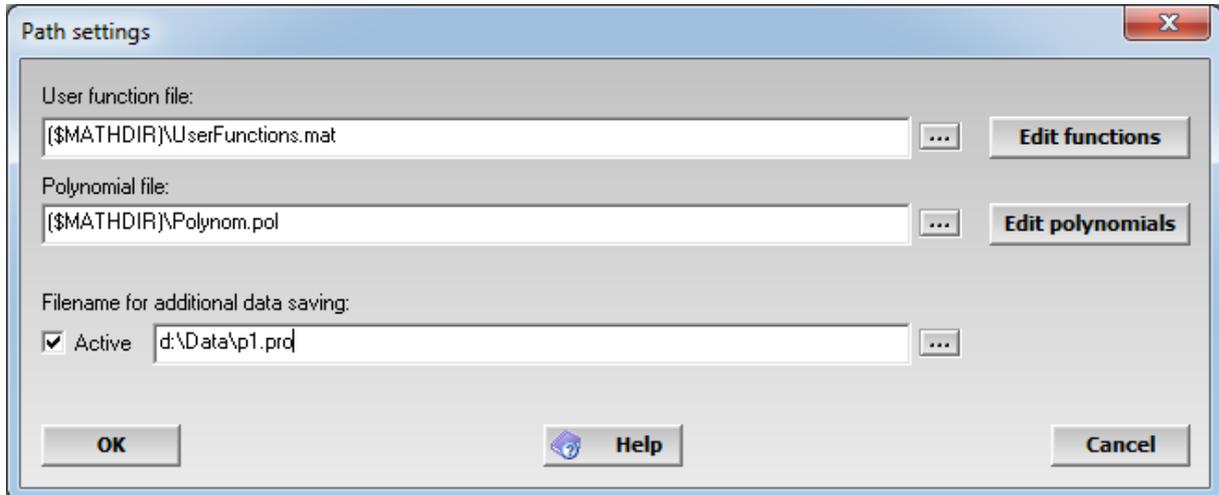
The group manager can be used to get a better channel aspect in text or graphic windows. The user can define and name groups and subgroups to organize his channels. The main item is **ALL CHANNELS**. With the right mouse button clicked on the root item a submenu appears. This menu controls all actions to create or modify the tree of the group manager. Groups or channels are inserted at the marked (highlighted) item. Channels can be added in the following format: 1,3,5-9,11-20.

The width of the group manager in text and graphic windows can be defined in three steps or variable size (use slider).

If groups are defined, the group manager will be automatically active in graphic windows. For all other windows the user have to activate it by the window's OPTION menü.

Channels or groups can be moved or copied to other groups with a hold mouse button. With shift-key or ctrl-key several channels can be selected and moved or copied. Several groups can only be selected with the ctrl-key.

## Project Pathes



A project can use independent files, that are shared with other projects. These files are for example the math files (user functions or polynomials). If they are used in the math formula of a channel, then MCPS needs to know, where to find the related files.

### 1) User functions

Change file: A file request box appears to select the user function file

Edit user functions: A file request box appears to select a user function file, which can be edited.

### 2) Polynomials

Change file: A file request box appears to select the polynomial file

Edit polynomials: A file request box appears to select a polynomial file, which can be edited.

For more info see MATHEMATICS/USER FUNCTIONS and MATHEMATICS/POLYNOMIALS .

A copy of the project and its data can be activated. The data file of the copy is updated simultaneously to the original file.

## Project Information



Project info

Company: CAD Computer GmbH & Co. KG

Group: Development

Operator: Gunther Rossol

#	Comment
1	System test: Alpha34-2
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	

OK Help Cancel

This window can be used to enter project specific information. When using layouts for printing, these information can be printed by using specific control codes. See [layout manager](#) for more information.

## AutoFile



**Autofile setup** **Batchfiles**

Autofile mode

Off  Standard  Extended  Batch manager

Standard autofile

Trigger type:

Extended autofile

Data filename definition:

Start number:  End number:   Stop at end  Restart with:

Interval trigger Day(s):  Hour(s):  Minute(s):

Fixpoint trigger

Alarm trigger

File size trigger  MB

Scan count trigger

Automatic file deletion:

Number of files to keep:

MCPS can distribute the measured data over several files. After a specific event (time or alarm) MCPS generates a new data file with a new identifier. This new file is NOT a new project, it's a new data record, which is handled by the same project. If a project with AutoFiles is opened, MCPS will load the first available data record. Another record is selected by [Project/Select record](#).

MCPS supports two basic mechanism to manage automatic file creation:

### 1. Standard

The file record name is determined by the current time or date. If you select **daily saving**, the record identifier is the current day, e.g. if you start the measurement on March, 12, 2005 then the record number is 00012. At midnight a new file is created with number 00013. **Hourly saving** creates numbers from 00000 to 00023 and **monthly saving** form 00001 to 00012. After a complete cycle the older files are overwritten. So this mode is often used for backup purpose.

Sometimes not all created files are needed for their complete life cycle. **Keep the xxx last files** will delete older files automatically.

## 2. Extended

The extended mode is more complex, but even more flexible. The file names can be from 0 to 99999.

**start number:** MCPS will use this number for the first file

**end number:** End of cycle, if this number+1 is reached

**stop at end:** The measurement completely stops if the end number+1 should be created

**restart with:** The counter of the identifiers is reset to this value, older files will be overwritten

In the extended autofile mode several independent trigger events are possible, which can be combined. If one trigger event occurs, the next data record is created and the measured data is stored in this new file.

Trigger events:

Interval trigger:

The user can define a time interval, which is started with the beginning of the measurement. If this interval is over, a new file is created. This happens everytime the interval is completed.

Fixpoint trigger:

This trigger event occurs at specific (even) time points. For example a time format of 12h will always create a new file at 12:00h and 0:00h, independently of the start time of the measurement. This means, that in most cases the first data record is smaller than the other ones.

Alarm trigger:

A new record file is created, when an alarm event has been detected and this alarm should be used for AutoFile creation. Therefore the button **AF** should be set in the channel's alarm configuration window. See [channel parameters](#).

Data filename definition:

In addition to the record number, some date information can be added to the filename (e.g. 2004\_10\_11\_00002).

## 3. Batchmanager

This is set automatically, when the project is used in a batch session.

See tutorial AutoFiles.

**Autofile setup** **Batchfiles**

Conversion of AutoFiles into batchfiles:

Enable creation of batchfiles

Destination directory:  ...

Database:  ...

Directory in database tree:

Run script:  ...

Delete AutoFile

Signing:

Number of levels:  ▼

Level:	Description:
1:	<input type="text" value="QA"/>
2:	<input type="text"/>
3:	<input type="text"/>

With option /BATCH or /ERES the AutoFiles can be converted into batch files. Since this is independent from a batch session, the batch target directory and database filename must be defined. If no database is available, MCPS will create the file. During this procedure a script can be called (/SCRIPT is required) to add information to the batch file. For example statistic computation can be added.

## Data Reduction



**Data reduction**

Data reduction off  
 No saving  
 Controlled data reduction

Time interval saving every:    Save mean values  
The save time interval should be a multiple of the project's sample rate

Saving on alarm, if at least  channel(s) with active alarm

Save always first scan

Partial saving:  
   of file or AutoFile

**OK** **Help** **Cancel**

MCPS supports data reduction to reduce the amount of saved data. Not every scan is saved to disk, but in specific intervals. MCPS always writes all channels of one scan. It's not possible to save only some channels. Some mathematical functions compute their values by using all data of the saved file (Summation, Integration, mean function, ...). In case of data reduction this might cause problems, because a lot of values are missing and these functions will show other results than during online display.

- Data reduction off

Default setting. All channels will be saved every scan.

- Saving off

No saving at all.

- Controlled data reduction

Time interval saving every: xx seconds

The user can specify a time interval, after which one scan is saved to disk.

Save mean values: xx seconds

The average value of all scans after the last saving is stored for each channel. Invalid scan data is skipped and the average is computed over all valid data scans.

*Saving on alarm* If the **DR** button is set in the **channel's alarm config window** and enough alarms are detected, the data is saved.

See tutorial data reduction.

## Automatic data acquisition



**Automatic data acquisition** [X]

**Time control:**

Start time: 12.09.2012 15:12:20 Change time...

Stop time: 12.09.2012 15:13:20

Stop data acquisition after  hours and  minutes

Stop data acquisition after  scans

Time sync (SR >= 1min):  h  m

**External control:**

Start with input: TESTDRV - 1: DI1

Stop with input: TESTDRV - 1: DI1

**Trigger by channel value:**

#	Channel	Trigger	Value	Action
<input type="checkbox"/> 1	1	H	2	Start
<input type="checkbox"/> 2	2	L	15	Stop

Pretrigger sample rate (s):

### Start time:

The measurement of the project is started at the specified time. The project must be started by user normally by DATA ACQUISITION/START. The default windows are opened but the data collection is not running yet.

### End time:

Stops the measurement at the specified time and removes the project from the data acquisition manager.

### Stop data acquisition after x hours and y minutes:

After the measurement has been started, it will be stopped after the specified time has been reached.

**Stop data acquisition after x scans:**

After the measurement has been started, it will be stopped after the specified number of scans has been reached.

**External control:**

MCPS supports some input cards and devices, which can be used to start or stop the measurement. The action is level triggered, that means if the input has a high signal the data acquisition is started, a low level (0) stops the daq.

**Trigger by channel value:**

The measurement starts normally. All devices are scanned and data is shown in online windows, but not saved to disk. It checks the defined channels for the trigger condition:

H (High): Value must be greater than trigger value to execute action.

L (Low): Value must be lower than trigger value to execute action.

After the start action the data is saved, math functions are reset (counter, registers,...)

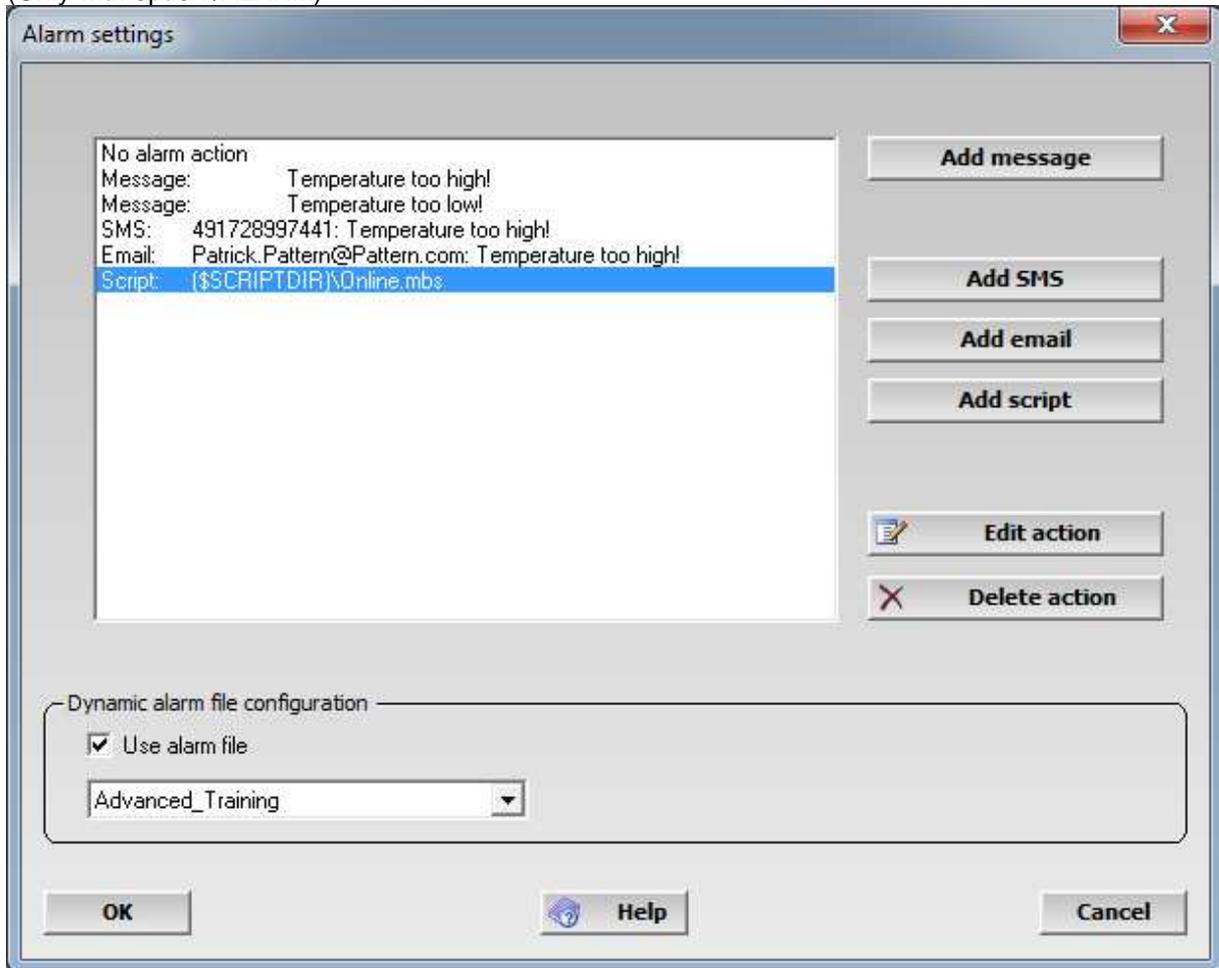
The pretrigger sample rate is used to scan the devices before the start action is executed. Then the normal sample rate is used.

In the data acquisition manager the project is set to WAITING, if the measurement isn't started yet. Start and stop times are also shown.

## Alarm Settings



(Only with option /ALARM)



In this window you have to configure the basic alarm settings for the whole project.

The upper part defines those alarm actions, which can be selected for each channel and each level in the channel alarm configuration (see [CHANNEL CONFIGURATION](#)).

### - Add message

A new message text is added to the project's alarm action list. If this action is selected in the channel alarm configuration, the alarm (with message) is displayed in the [Online Alarm Window](#). Alarms without any message won't be visible in the Online Alarm Window.

### - Add digital output

This button is only available, if output cards are defined in the device manager configuration window. All available outputs are listed. There are several types to handle the output in case of alarm:

a) Follow alarm Turned on/off with alarm on/off

b) Reset with Ack Turned on with alarm; turned off, when ACK button is pressed in the Online Alarm Window.

To use this feature, you need also an message for this alarm.

c) Single pulse To output is set for several seconds and reset independently from the alarm status.

### - Add SMS

This action will send a SMS message to a handy (with option /SMS-E).

The message is sent to the main telephone number. Some SMS providers can tell MCPS, if the message has reached the target person or not. If not MCPS will send the message to the next number. For SMS actions some basic settings are necessary, which are defined in [Configuration/MCPS Settings/SMS-Email](#).

**- Add Email**

This action will send an Email (with option /SMS-E).

The alarm channel, level, type and so on is added automatically. If several alarms occur at the same time, only one email is generated with all alarm channels. See [Configuration/MCPS Settings/SMS-Email](#)

**- Add Skript (with option /SCRIPT)**

This action will run the specified script.

**- Edit action (or double mouse click on action)**

The settings of an existing action can be changed.

**- Delete action**

The action is removed from the alarm action list. MCPS will also set all channel alarms with this action to NO ACTION!

## Project defaults



### - Print layouts:

For printouts you can use [printer layouts](#) as defined in the layout manager. If you make a printout, you have to load a layout. If you are using always the same layouts for specific windows, you can define these layouts as default ones. The printout will automatically use the layout as defined in this list (for the related window).

### - Trend windows:

#	Active	Reference	Value	Color	Split screen
1	<input checked="" type="checkbox"/>	001: Temp 1	10	Red	All
2	<input checked="" type="checkbox"/>	001: Temp 1	2	Blue	All

#### Window layout at start of measurement:

A window layout, that has been saved by the [PROJECT/WINDOW MANAGER](#), can be defined as default start window layout. It will be loaded, when the measurement is started.

#### Default x-axis time span:

The time axis preset value, when an online trend window is opened

#### Window layout for offline graphic:

This layout is loaded automatically, when DISPLAY/GRAPHIC is called. In this case it might be helpful, that other windows are not closed automatically. See [PROJECT/WINDOW MANAGER](#).

#### Default x-axis time span:

The time axis preset value, when an offline trend window is opened. The visible time span can be from the beginning or from the end of the file. So it is possible to setup the offline trend window to show by default only the last 24 hours.

#### Trip lines:

Trip lines are horizontal lines in trend windows. They can be set relative to the window (percentage) or related to a channel to use its scaling. In the second mode, the trip line is also affected by zooming in y-direction.

Active: Trip line is visible or not

Reference: a) Relative to window in % or b) channel related.

Value: a) Value from 0 to 100 or b) a value near to channel value

Color: Color of the trip line

Split screen: When used with several screens in trend windows, the screen can be selected

Add trip line: Adds a trip line to the list

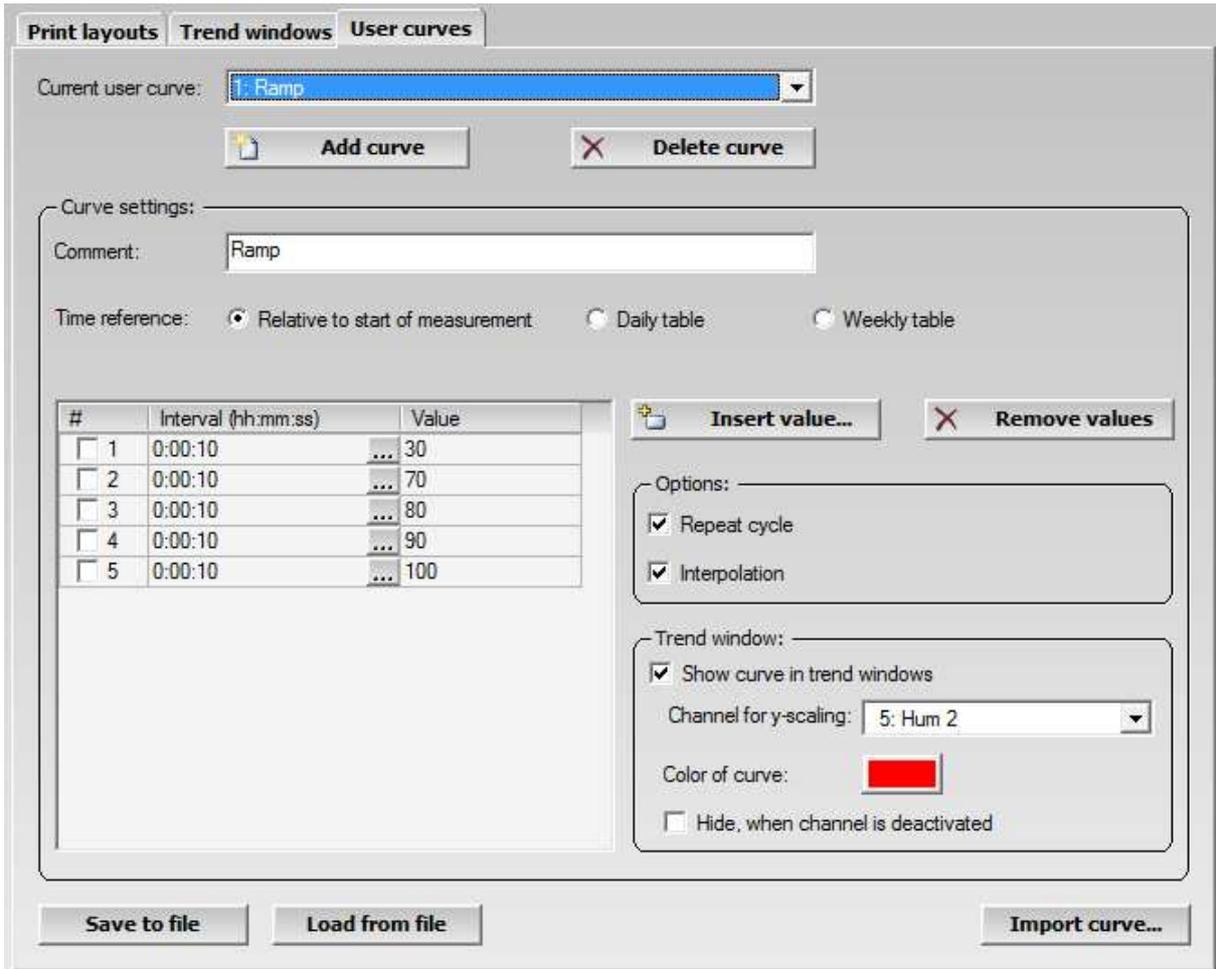
Delete trip line: Removes one or more trip lines from the list

#### - Reference measurement:

(Only available with at least one completed reference measurement)

A window displays all available reference measurements. The highlighted line will be the next active set of offset values (or no correction). All open windows will be updated after saving the project.

- User curves:



(Only with option /MATH2)

A user curve is a time based signal, which is visible in trend windows and also returns a data value in the math formula related to the point of time of the measurement. The curve is defined by several data points. A data point becomes valid, if the relative time of the data acquisition is in the time span of the data point. In the graphic above we have a value of 1 for the first 10 seconds. For the next 2 minutes the curve has a value of 2 and so on.

In the mathematical formula the result of the user curves are provided by  $U_n$  ( $n$  = number of curve:  $U_1, U_2, \dots$ ). E.g.  $X_1 + U_1$

Since the relativ user curve always starts with the first value in the table, the daily and weekly tables are referenced to the current date and time. They are absolute time pattern with a daily or weekly cycle. The weekly user curve contains seven daily pages.

- Insert value

A new data point can be inserted into the list at a selectable position

- Remove values

All marked lines (check box) are deleted from the list

- Repeat cycle

If the measurement takes more time than the list of data points covers, the curve can start with the first data value again. So it is possible to create cyclic signals.

- Interpolation

If active, MCPS will draw a line between the data points and computes for the steps between them the interpolated value. If not active, the value is hold until the new value is valid.

- Show curve in trend window

If active, MCPS will draw the curve for the complete window (preview). The curve must be linked to a channel to get the right scaling. Each curve can have its own color and can be set invisible, if the related channel is deactivated for example by the groupmanager.

- Save to file

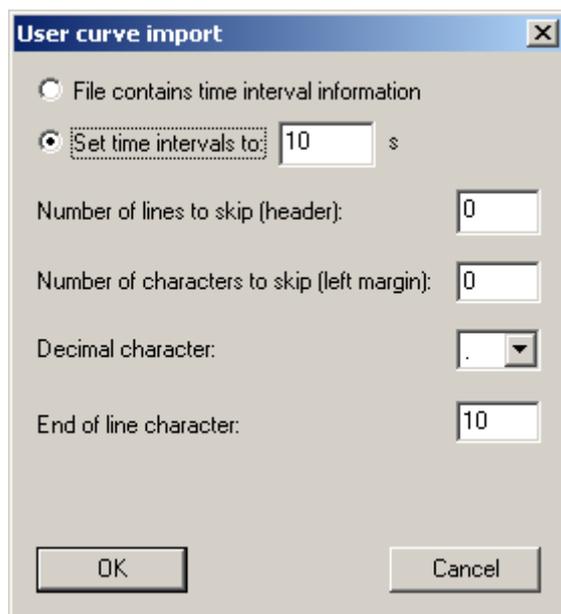
All user curves of a project can be saved to an independent file. So different sets of curves can be handled or even other projects can use (load) these files.

- Load from file

Loads a file containing user curves.

- Import curve

MCPS can import a text file containing a user curve. The file can contain time information, that means a column with the time span in seconds, or the time interval is set for all data points to one value.



Header lines or characters at the beginning of each line can be skipped. Decimal point and end of line character must be specified correctly.

Example:

Import file for user curve

```
001,10,4  
002,20,5  
003,15,4.7  
004,11,1.2
```

File contains time information (10,20,15,11)

First line must be skipped (Comment)

The first four characters must be skipped (line number)

Decimal character is .

EOF character is LineFeed (10)

A simple file can be:

4

5

4.7

1.2

In this case the time information must be set. So it is very easy to create a user function by external programs or data files (e.g. Excel data).

Very useful is this function in conjunction with the channel output button. The user curve value can be sent to an analog output in control or test environments.

## Automation



### Print:

The 'Automation' dialog box has three tabs: 'Print', 'Export', and 'Scripts'. The 'Print' tab is active. It contains three sections:

- Graphic printouts:**
  - Graphic printout on end of data acquisition (Options...)
  - Graphic printout, when AutoFile has been finished (Options...)
- Statistic printouts:**
  - Scheduler 1 (Set... Options...)
  - Scheduler 2 (Set... Options...)
  - Scheduler 3 (Set... Options...)
  - Statistic printout on end of data acquisition (Options...)
  - Statistic printout, when AutoFile has been finished (Options...)
- Alarm printouts:**
  - Scheduler 1 (Set... Options...)
  - Scheduler 2 (Set... Options...)

Buttons at the bottom: OK, Abbrechen, Hilfe.

#### - Graphic printouts:

Automatic printout at the end of the measurement or AutoFile. The button <options> allow you to make the following settings:

All or only specific channels to be printed, distributing the printout over several pages, using a specific common window layout and print layout. Activate the related check box to use the layout.

The 'Print options' dialog box contains the following settings:

- Output channels:**
  - All channels
  - Selected channels (Define channel list)
- Multi page printout:**
  - Fixed number of pages: 1
  - Fixed time range per page: 10 M
  - X-axis autoscaling
- Window layout:**
  - lb
- Print layout:**
  - C:\Program files\MCPs\Layouts\layoutger.ply
- Printer:**
  - Default printer

Buttons at the bottom: OK, Cancel.

- Statistic printouts:

A statistic can be printed at the end of a data acquisition, end of AutoFile or triggered by one of three schedulers:

The 'Scheduler event' dialog box is titled 'Scheduler event' and has a close button (X) in the top right corner. It is divided into two main sections. The first section, 'Fix time intervals:', contains three rows of controls. The first row is for 'Fixpoint trigger 1', which is checked. It has a 'Type' dropdown set to 'daily', a 'Day' field, and 'Hour' and 'Minute' spinners set to 0 and 4 respectively. The second row is for 'Fixpoint trigger 2', which is unchecked, with 'Type' set to 'daily' and 'Hour' and 'Minute' spinners set to 0. The third row is for 'Fixpoint trigger 3', which is unchecked, with 'Type' set to 'daily' and 'Hour' and 'Minute' spinners set to 0. The second section, 'Interval from start of data acquisition:', contains an 'Interval trigger' checkbox which is unchecked. To its right are 'Day(s)', 'Hour(s)', and 'Minute(s)' spinners, with values 0, 0, and 5 respectively. At the bottom of the dialog are 'OK' and 'Cancel' buttons.

Fixpoint trigger is always the same point of time independent from the start of the measurement. Interval trigger is relative to the start of the data acquisition.

The 'Statistic options' dialog box is titled 'Statistic options' and has a close button (X) in the top right corner. It contains several sections. The 'Output channels' section has two radio buttons: 'All channels' (selected) and 'Selected channels', with a 'Define channel list' button to the right. The 'Output types' section has five checkboxes: 'Minimum', 'Mean value', and 'Maximum' are all checked; 'Standard deviation' and 'MKT' are unchecked. The 'Output format' section has a 'Decimal places' spinner set to 2. The 'Time range' section has two radio buttons: 'From last trigger' (selected) and 'Last', followed by a text box containing '365' and a 'Days' dropdown menu. The 'Print layout' section has a text box containing the file path 'C:\Program files\MCP56\_1\Layouts\layoutger.ply' and a browse button (...). The 'Printer' section has a dropdown menu showing 'Acrobat PDFWriter'. At the bottom are 'OK' and 'Cancel' buttons.

The options define the computation to be done, the used channels, format of the output values, time span and print layout. The time range of used data scans can be from the last run (From last trigger) or a selectable time range. This value can even be greater than the interval of the scheduler and forces MCP5 to scan several AutoFiles if necessary. So it is possible to have a statistic report every day over the last 365 days (Important for MKT computations).

- Alarm printouts:  
A scheduler triggers the alarm print out.

**Alarm print options**

Output channels:  
 All channels     Selected channels    Define channel list

Report type:  
 Alarm list     Alarm statistics

Time range:  
 From last trigger     Last    5    Days

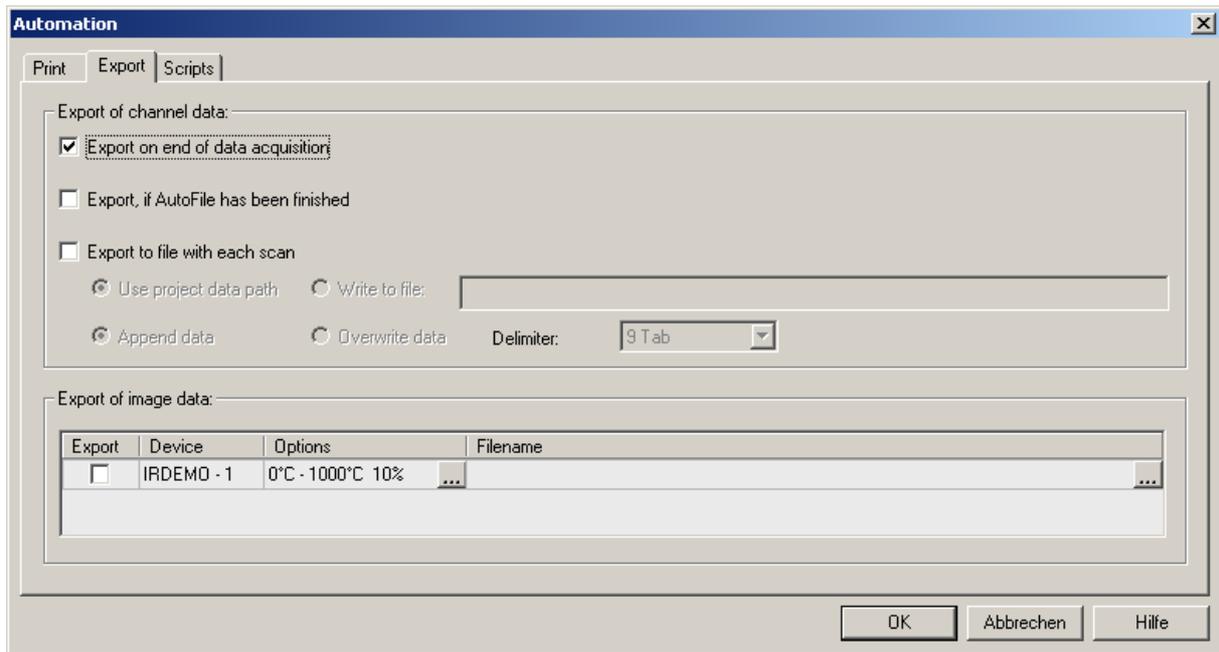
Print layout:  
C:\src\WinMcps\src\Debug\Layouts\cad.ply

Printer:  
Default printer

OK    Cancel

The alarm print options define the channels to be used, report type, time range and print layout. The report type <Alarm list> will print all alarms as in the alarm logfile in the selected time range. The <Alarm statistics> will print a channel list with the number of alarms for each channel. The time range can be from the last printout or a selectable value. MCPS will scan the alarm logfile for the last xxx minutes, hours or days. So it is necessary to activate the LOG-button in the channel alarm settings. Only alarms in the logfile can be printed!

## Export:



### Channel data:

The data file is automatically exported:

At the end of the measurement

After an AutoFile has been finished

With each scan: The export data can be stored in a file located in the project path (with extension XLS) or in a user defined file.

For further export settings see also [PROJECT/EXPORT](#).

### Image data:

All infrared devices of the project are listed. The check box defines, from which device(s) the images are taken and saved to the selected bitmap file. With the ellipsis ... same additional settings can be made.

### Scripts:

For certain actions scripts can be called to customize applications. The DAQ scan script is called with every n-th scan. This script must be very fast and shouldn't contain any type of message box to wait for user actions. The script mode defines in which environment the script should be executed:

Standard: All scripts are started on a DAQ-computer, Online related scripts are not started on clients

Server: Script is started only on a DAQ-computer

Client: Script is started only on a client

All: Script is started always

Example: If a measurement is running with an online script to update excel tables, it is not useful to run the same script on a client, which is connected to the DAQ-PC by the /MSERV option.

### Script start types:

Project opened: Started after the project is loaded

Project closed: Started after the project is closed

Daq preparation: Started after the start button has been pressed. Typically used to request additional parameters and to save them in project variables

Start measurement: Started after the connection to the devices is made. Used to send commands to the instruments (new range, alarm setpoint, ...)

Each scan: With every or every n-th scan the script is started to computed values, transfer data or update the process window.

Stop measurement: Started after the data acquisition has been stopped.

AutoFile completed: Started after an AutoFile has been closed and a new one is created. Typically used to create daily or monthly reports.

Scheduler: Started by the trigger settings of the scheduler during a data acquisition

New project created: Started after PROJECT/NEW was called. The script can set a return value (SetReturnValue) to control further actions:

- 0: Normal procedure; the configuration window is opened
- 1: Abort; no further actions
- 2: Skip; the project was created, but the configuration window will not be opened. The project name and its saving must be handled in the script. This procedure is useful in case of creating a new project by a template and then adding or modifying some customer specific values.

Status window event: Called, if a button or tab control in the process window (also controlled by script) is pressed.

Configuration saved: Called after the configuration is saved.

Project configuration: A button is added to the project configuration, if this script is defined.

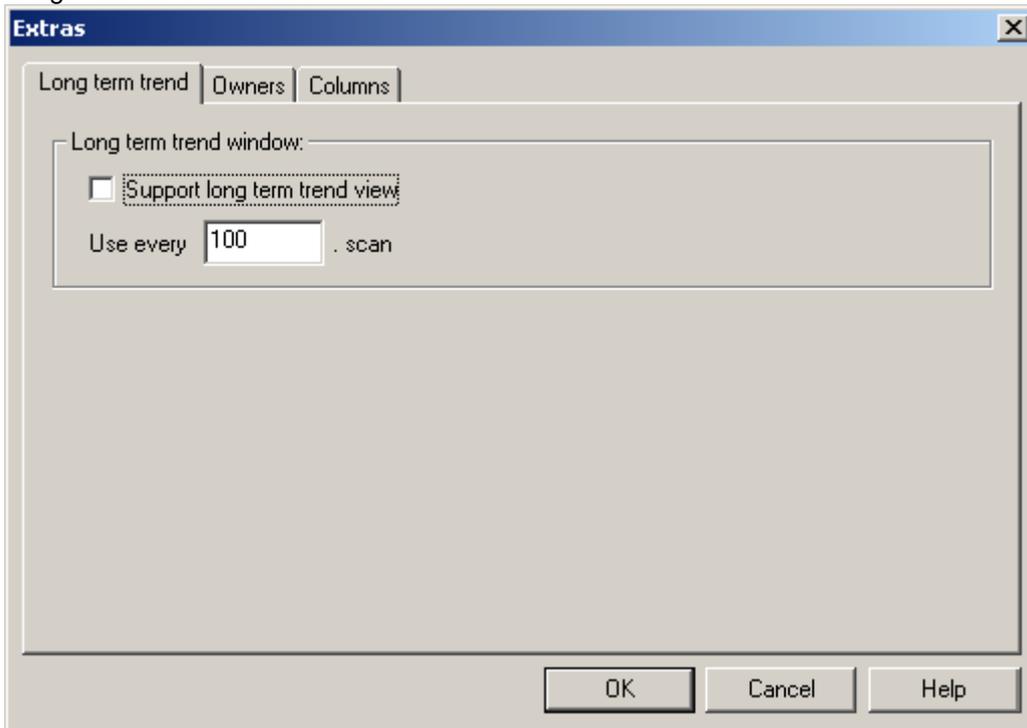
Before printing: Before each print a script is called. To distinguish the calls from different window types the function GetScriptType2 can be used.

See also Concept of scripts.

## Extras



Long term trend window :

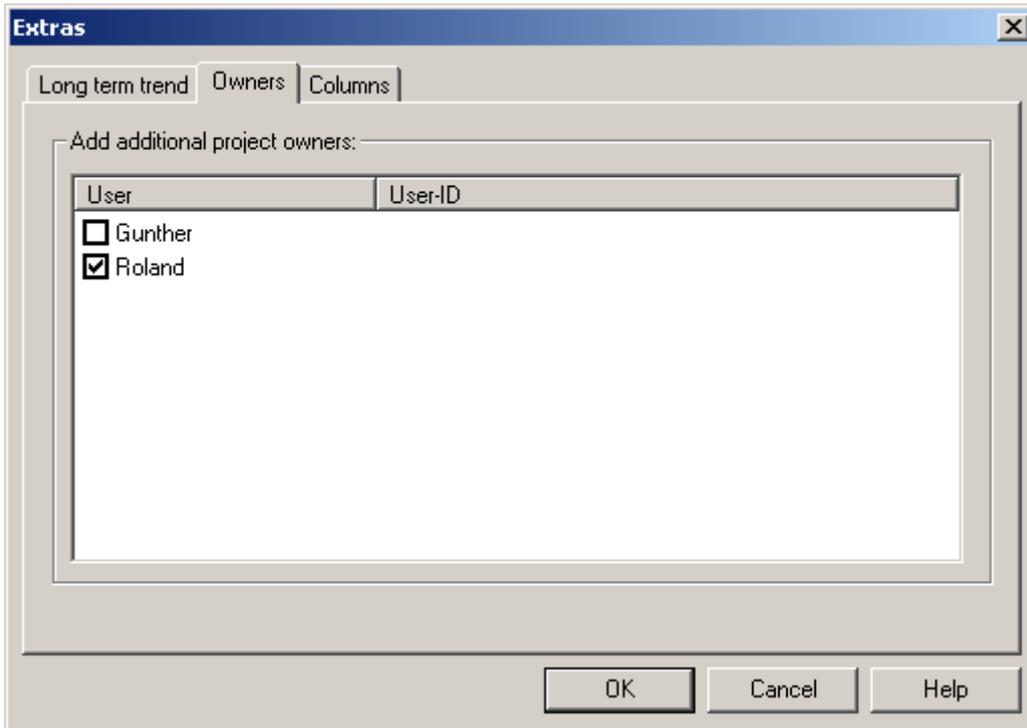


MCPS is used in applications, where a lot of data is gathered and saved to disk. If the online graphic window needs a redraw, because of another group or window resize, MCPS will not use the save data to display the curves. Instead it will recall the data from an internal memory buffer, which is much faster. This buffer is of course limited, so some older data will not be redrawn. If the x-axis shows a great part of the measurement (a long time range), probably most of the window will be left empty.

The long term trend window can avoid this problem by reducing the scan resolution. If turned on, MCPS will allocate a second buffer for data, which is updated for example only every 10<sup>th</sup> scan. The time range you can see now after a redraw will be 10 times the normal one. This feature is used to get a general long term trend view.

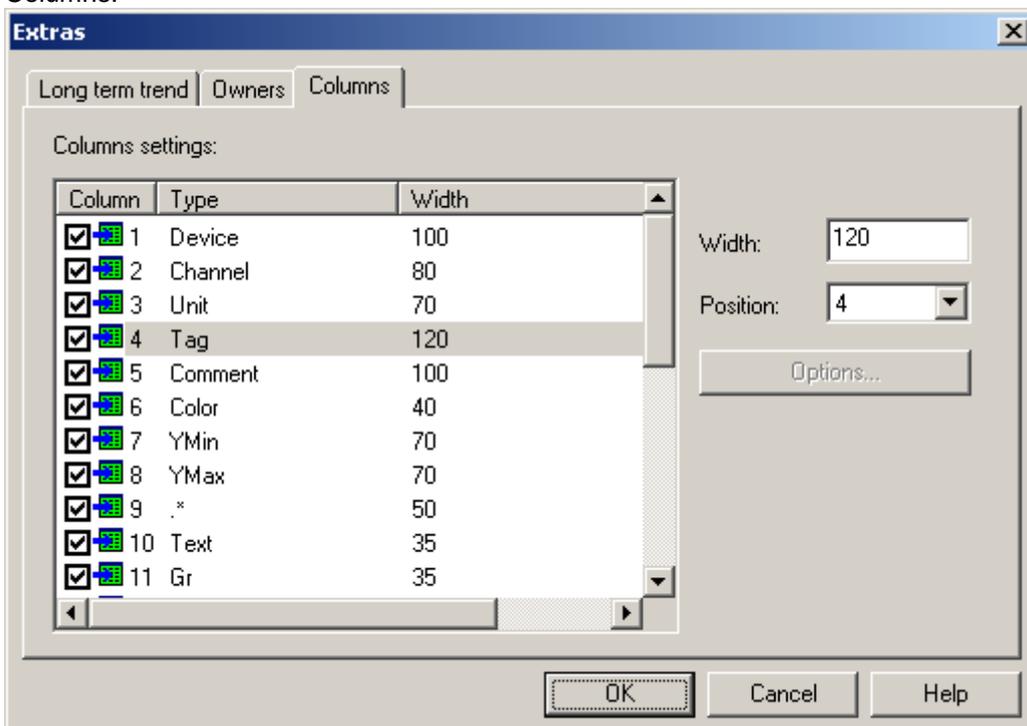
See also MCPS Settings/General

Owners:



This part is only available in case of an active user administration. If a project is created, the current user is defined as the owner of the project. Other users cannot access this project, if they have only have access rights to their own project. Sometimes it is useful to allow other users access to the own project. This is often the case, when an administrator creates the projects for several users, who can not make their own project, but just work with existing ones. Additionally to users access can also be given to groups.

Columns:



The setting columns of the project configuration can be configured in several ways:  
Visible or invisible column  
Horizontal position of column (1 = most left)  
Width of column

## User columns:

In every project up to 15 user columns can be added for additional channel information (comments). These columns appear and work like all other columns. Each user column has its own header text and can have one of the following types:

1. Text                    A normal text field like comment or tag
2. Check box                A check box like Text or graphical output
3. Drop downlist        A list box with predefined texts to select
4. Drop downlist+text    A list box with predefined texts and the possibility to enter free text

These channel information can be used in the print layout as additional documentation. See the [control codes](#) how to use them.

## Get settings from devices



This window is used to get special information from a device, which are inserted into the project. At the moment it is possible to receive tag information from specific devices. The target field can be one of the following columns:

Tag, comment or user

The target field can automatically be updated, if the channel name or device is changed by the user.

In the device manager it is possible to lock a device to a setup. This is used to have the same setup in the device and in the related setup file. In this case it is faster to get the tag information directly from the setup file instead of downloading it.

Actions:

Tag information can be downloaded from all devices in the project configuration or only from those, which are marked or selected. It is enough to select one channel of a device to get all tags of it. The tags are downloaded and put into internal memory for further operations.

To get the downloaded information in the target column you have to press <Set tag information of all selected channels>. So we have a two step sequence: First download, second set columns. After this you can set tag information as often as you want, if channels changes or are added. In case of adding a new device you have make new download.

## Cut, copy and paste channels



Removes the marked channels and saves them to the internal clipboard



Copies the marked channels to the clipboard.



Inserts the channels of the clipboard.

These functions can also be used to copy channels from one project to another one.

## Print project configuration



The channel settings and most of the standard projects parameters can be printed. The user can select, which columns should be printed. To get enough information on a sheet of paper it is recommended to use a print layout with landscape format and a small font size. The font size can be defined in the output frame of the layout. See [print layout manager](#) for more information.

## Open

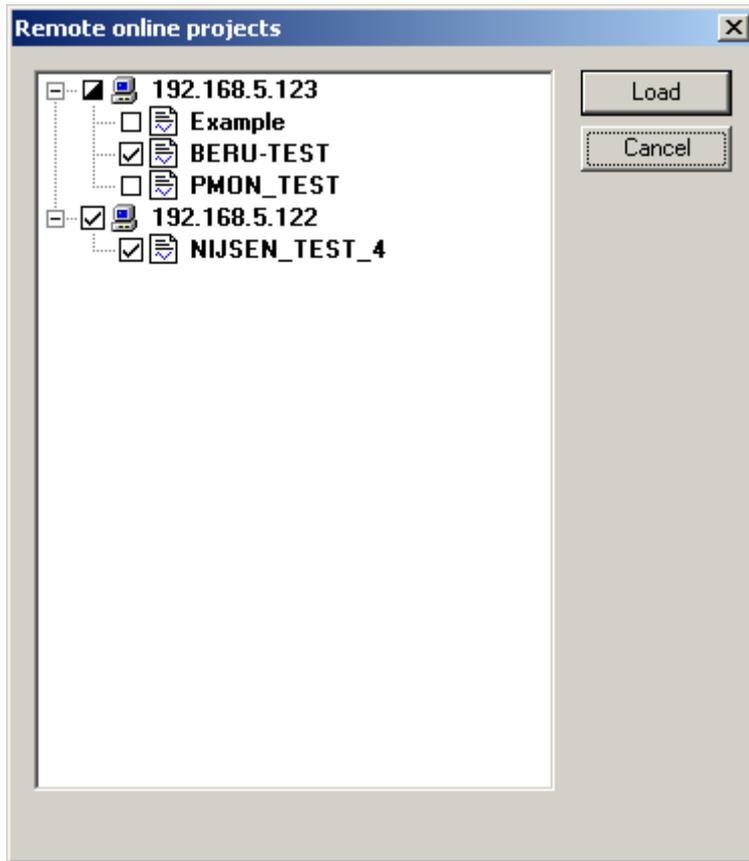
A file request box appears to select and load one or several MCPS projects. A loaded project can not be loaded a second time. MCPS can handle several projects at the same time. For some users this behaviour could be too complex or unuseful. In [CONFIGURATION/MCPS SETTINGS](#) it's possible to select a single project mode. MCPS will only handle one project. If a new project is loaded, the other one is unloaded. This is not possible during an active measurement.

## Open project list

Opens a file containing a list of project to load. This list (\*.prl) can be configured by PROJECT/EXTRAS/MANAGE PROJECT LISTS.

## Open from DA Server

(Only with option /MSERV)



This function makes connections to all MCPS data acquisition servers as defined in [CONFIGURATION/MCPS SETTINGS/MSERV](#). All running online projects of the servers are listed. Several projects can be loaded onto the MCPS client computer by activating the check box. This is not a normal file load, so that only online windows and the alarm window can be opened. Configuration or offline actions are not possible. If the project is reconfigured on the server, it will be closed on the client and reopened again.

After the project is loaded the recent data scans are transmitted to show a history in trend windows. The amount of scans, which should be transferred can be defined in [CONFIGURATION/MCPS SETTINGS/MSERV](#). If the online alarm window is configured to open automatically, it will also be appear, if an alarm occurs on the server.

## Select new active project

All loaded project are shown in a window to select a new active project. The base window of the new active project is brought into the front. This function is useful, if you want to see all projects or no window of the desired project is on the top.

## Select record

Projects with the autofile function enabled may produce a lot of data records. All offline functions will work with the current loaded data record. The number of the active record can be found in the main window's title bar. If another record should be examined, this menu item will select a new one. MCPS will display all available data records with time information about the first and last data scan of each record. Windows (text or graphic), which were opened with a specific record will always belong to it, even if a new record is loaded. So it's possible to have several graphic windows with different records on the screen.

After a project is opened, the latest AutoFile is loaded. During a measurement the currently active offline AutoFile can change automatically.

## Close

The current (active) project, which is displayed in the main title bar, will be closed, if possible. That means the project isn't in any measurement state. If the project is closed, all of its windows will be removed.

## Configuration

Use this menu item to configure an already existing project. The configuration window is the same as of **PROJECT/NEW**. If the project contains already data it's not possible to change all parameters. Device and channel selection is disabled, raw data channels can't be deleted and only math channels can be added.

## Data

### - Copy project

This will create a copy of the project and all related files. You should always use this function instead of copying with the Windows Explorer. Alarm, event and window layout files will be copied and if desired even a part of the measured data.

In the last case it's helpful to define the part to copy in the graphic window by using the cursors. With the CO button you can also call the copy function.

If a project was created, when no user administration was used, the project is public and can be used by everyone. If a user administration is installed later on, it is often useful to give the public project a creator and to prohibit access to it by other users. Since the original public project cannot be modified in the way to have a creator, a copy of this project can be made and a creator can be assigned. MCPS will automatically ask for this before copying the project.

### - Combine projects

Several projects with the same number of channels can be concatenated to one project. The time stamps of the data scans shouldn't overlap. This function is often used with imported projects from devices, which generates cyclic files.

### - Combine AutoFile records

Several records of an AutoFile project can be put together in a new project with only one data file. Use Shift and Ctrl with mouse to select multiple records.

### - Combine with original project

If MCPS had to change to the alternative saving directory, the current project is renamed and saved to disk. If the original destination is available again, the new project can be added to the original one. Therefore the new project must be stopped.

### - Remove project(s) on hard disk

Removes a project and all of its related files from the hard disk.

### - Delete measured data

Deletes a part or all of the measured data in a project. This function can also be called from the cursor operation window.

### - Delete Logfile

Opens a window to delete the alarm logfile and/or event logfile.

- Delete AutoFile records

Removes records from the project. Use Shift and Ctrl with mouse to select multiple records.

- Check data file integrity

This function checks, if any data block has been modified. Sometimes MCPS can detect an error, if the project is loaded. To be absolutely sure, you have to run this function. In some error cases the block length has been changed. MCPS will correct this and mark the block as modified.

- Fill in lost data

If a project had communication problems and some valid data are missing, the gaps can be filled in, if backup data is available. Some device like the MX100 can create backup data, which is stored on memory cards during communication errors. Normally MCPS recovers the data automatically, when the connection is OK again. But if the internal buffer is not large enough or the connection cannot be established again, manual gap filling is required.

- Data file sizes:

Bytes per scan:	Disk space needed, if one scan is saved.
Bytes per hour:	Disk space needed, if project is measuring one hour.
Current file size:	Size of the data file or the current loaded AutoFile.

## Info

Some basic information about the project like target location, creator, batch information and so on are displayed. The button can be used to open the Windows explorer at the project's location.

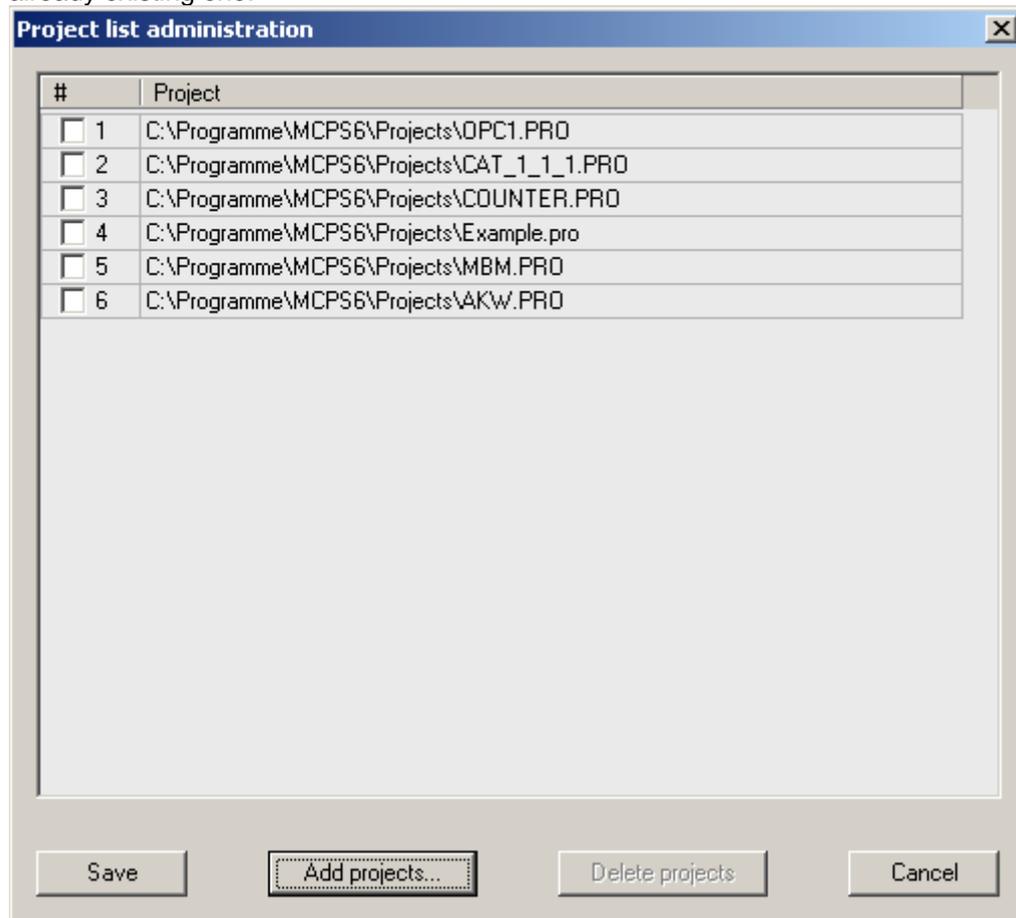
## Extras

### Save project as template

The project will be saved as a template. This means a copy is saved to the template directory. If a new project should be created and the template directory is not empty, all available templates are shown. A click on a template creates a new project with all the settings of the template project.

### Manage project lists

To load and start several projects in one step you can define a set of projects called project list. This list is saved with extension <prl>. By pressing this menu item you can create a new list or configure an already existing one.



Add projects:

The file requester is opened and several files can be selected by using the control (ctrl) key.

Delete projects:

All projects with an active check box will be removed from the list (not physically).

### Dynamic Alarm File

(Only with option /Alarm)

NEW

Creating a new dynamic alarm file.

The dynamic alarm file is only possible with AutoFile project. The following two basic modes are possible:

- One reference curve:

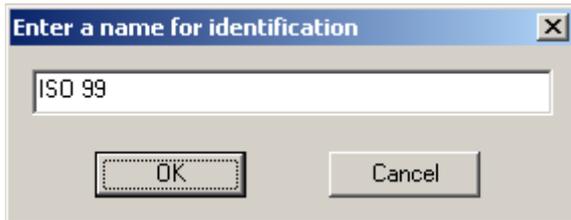
An already existing record is used as a reference for alarm checking of following measurements. Typically these measurements are all of the same course as in production lines or test fields.

The reference file can be separated into several time zones. For each time zone it's possible to define for each channel a high and low deviation. This deviation can be of absolute value (+5) or percentage (10%). If the measurement starts, the reference file is loaded. For every scan the alarm offsets are added to the values of the file. The results are compared with the new measured data and might cause an alarm.

- Envelope curve:

Not one but several AutoFiles are selected to create envelope curves as reference for new AutoFiles. As result an upper curve and a lower curve exists as limits. Additional deviation offsets to these curves can be defined.

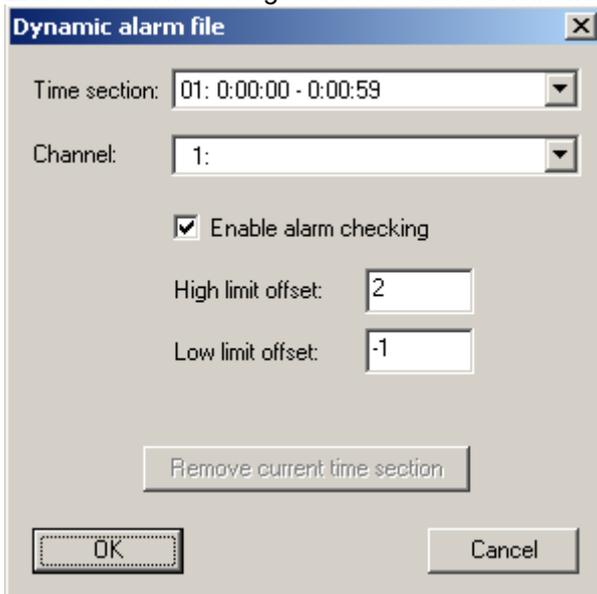
The first step is to give the new alarm file a name, since the project can handle several dynamic alarm files.



The second step is to choose the type of reference file and to select one or more AutoFile records as reference.



The reference file is now displayed in the graphic offline window. This window enables some additional features. The different time zones are visible (initially one) and in the window's option menu the item *Extras/Alarm file configuration* is enabled. This will open the following configuration window:



The top button selects the time zone. The next one specifies the channel. For each time zone and each channel alarm checking can be turned on with independent deviation values. These values are related to the data values of the reference file at a specific point of time. If a new scan is made, MCPS computes the time since the start of the measurement. It determines the responsible time zone and the related data values of the reference file. The deviation values are added and compared with the new scan values.

#### New time zones

The cursor can define new time zones by splitting one. This is done with the function *Split time zone* in the CO (cursor operation) window.

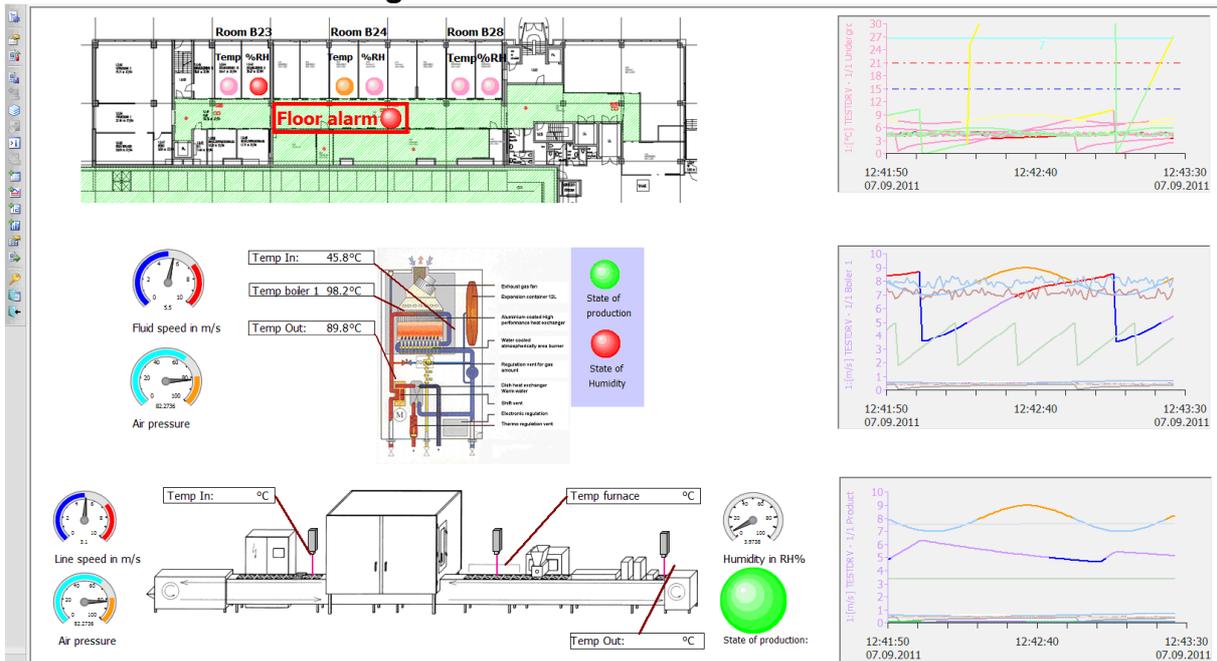
#### Edit

Use this menu item to change settings of an already existing dynamic alarm file.

#### Remove

Removes an alarm file.

## Process monitoring editor



MCPS provides a window (PMON) with several process icons, which is docked to one side of the frame window. The icons can be texts, bar graphs, leds, bitmaps and so on. These items can be linked to channels (measured value or alarm) and positioned by the user. This allows better visualization of important information. The icons can act related to a channel's value or to a channel's alarm. I.e. a led can change its color, when a specific value is reached or an alarm is detected.

A double mouse click on an icon opens its configuration window. Several items can be selected, moved or copied.

Basic settings for all projects are done in [CONFIGURATION/MCPS SETTINGS/PMON](#).



Cancel: Leave editor without saving.



Save: The Configuration is saved into the project. Bitmaps and Videofiles are not included and only referenced, so the original files must be available.



New: All items are deleted



Adds a page

When the mode of MCPS SETTINGS/PMON is set to <Projects selectable by tabs> each project can have several sub pages in the process window.



Deletes a page



Page configuration



Operation mode: Selecting and moving items



Unselect item(s).



Adds a text element:

#	Value	Text	Color
<input checked="" type="checkbox"/>	-100	System stopped	Red
<input checked="" type="checkbox"/>	1	Sytem running	Green

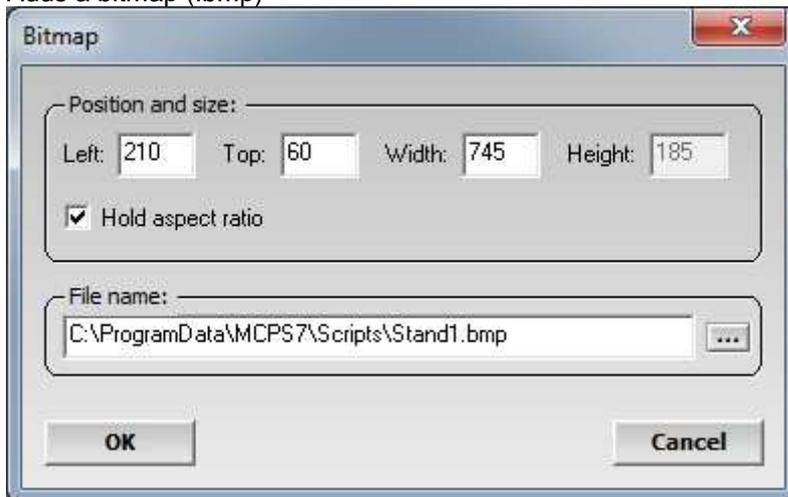
The text field can contain a static text, a measured value of a channel, a register value or a value dependent text. With exception of static text all other texts are updated automatically every time the project has new data. The text can be varied in size, color and font settings. Additionally the text can be adjusted in three different ways to group several text items in the same way:

- Left:** The text is started on the left side of the bounding rectangle
- Decimal point:** The text is always positioned in the way, that the decimal point is on the right border of the bounding rectangle.
- Right:** The text ends on the right border of the bounding rectangle.

For the dynamic text you first select the source: A channel of the project or a register. Then you define a list of thresholds to determine, which text is displayed related to source value.



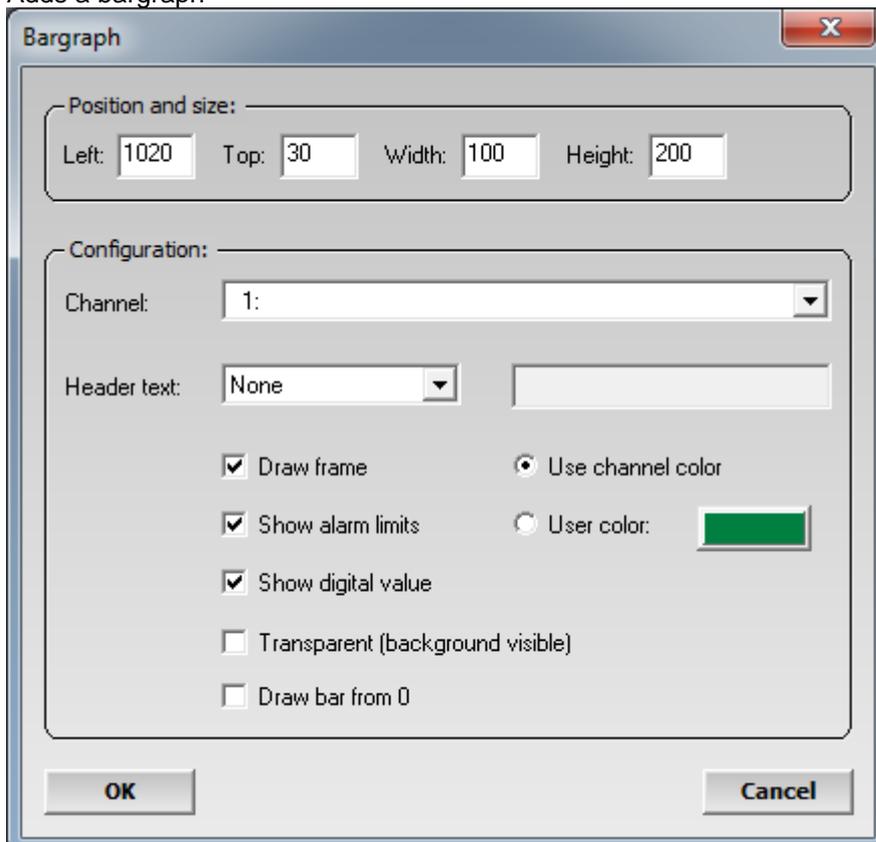
### Adds a bitmap (.bmp)



Hold aspect ratio: The height of the bitmap is automatically adjusted to hold the aspect ratio. The filename can be changed, if the bitmap was moved or the project is on another computer with different paths. If a bitmap cannot be loaded, a crossed rectangle is displayed instead.



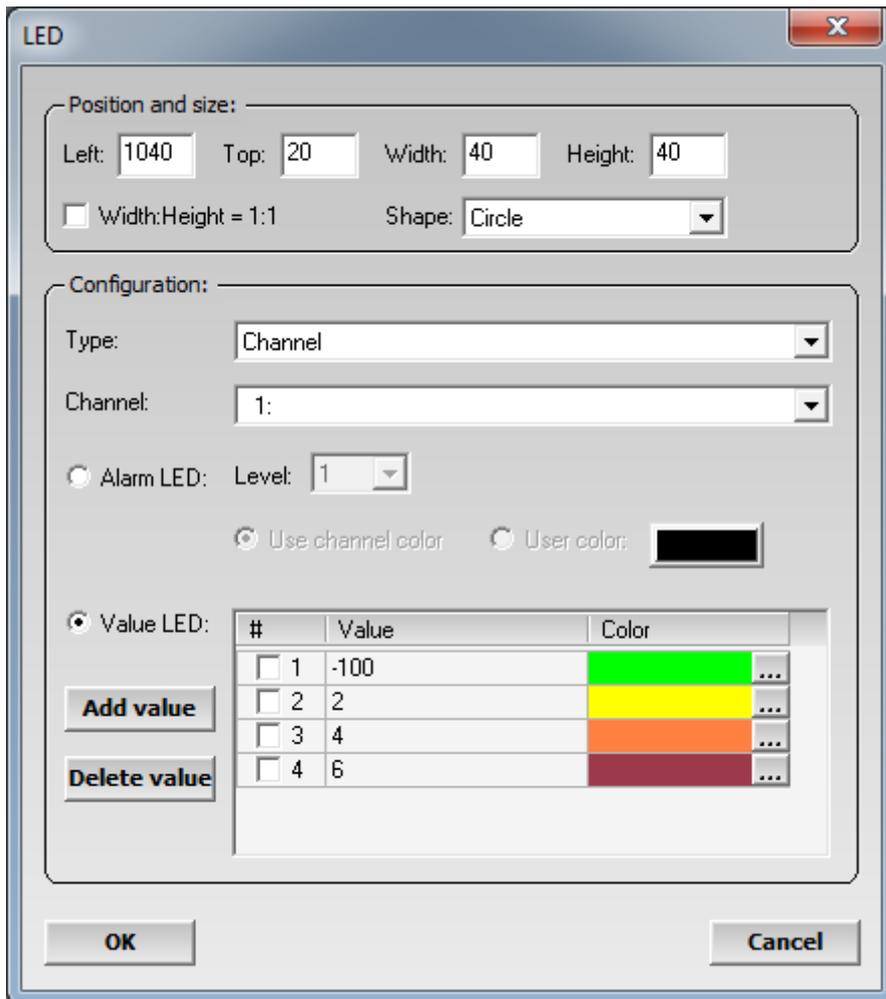
### Adds a bargraph



The bargraph is linked to a channel to represent its value. The minimum and maximum scaling values are taken from the channels YMin and YMax settings in the project configuration. The color and appearance can be set individually.



### Adds a LED



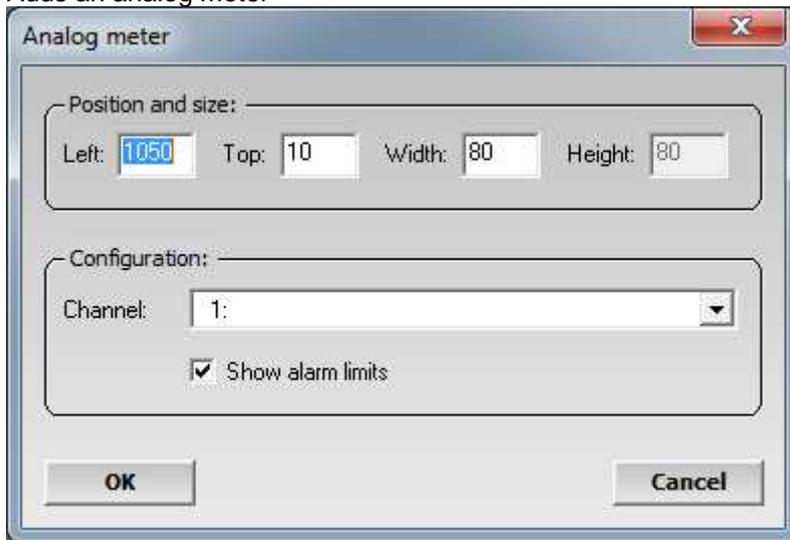
The LED element can be controlled by channel alarm or channel value. An alarm LED uses the color information from the project configuration and will flash, until the alarm is acknowledged. The non alarm color can be the channel color or a user defined one.

In the second case, a list of threshold values with related colors must be defined. The LED will change its color, when the next threshold value is exceeded. If a value is added with <Add value>, the list is sorted. With <Delete value> all marked (checkbox) values are removed from the list.

Changes the color definition



Adds an analog meter



The dialog box is titled "Analog meter" and has a close button (X) in the top right corner. It is divided into two main sections: "Position and size" and "Configuration".

**Position and size:**

- Left: 1050
- Top: 10
- Width: 80
- Height: 80

**Configuration:**

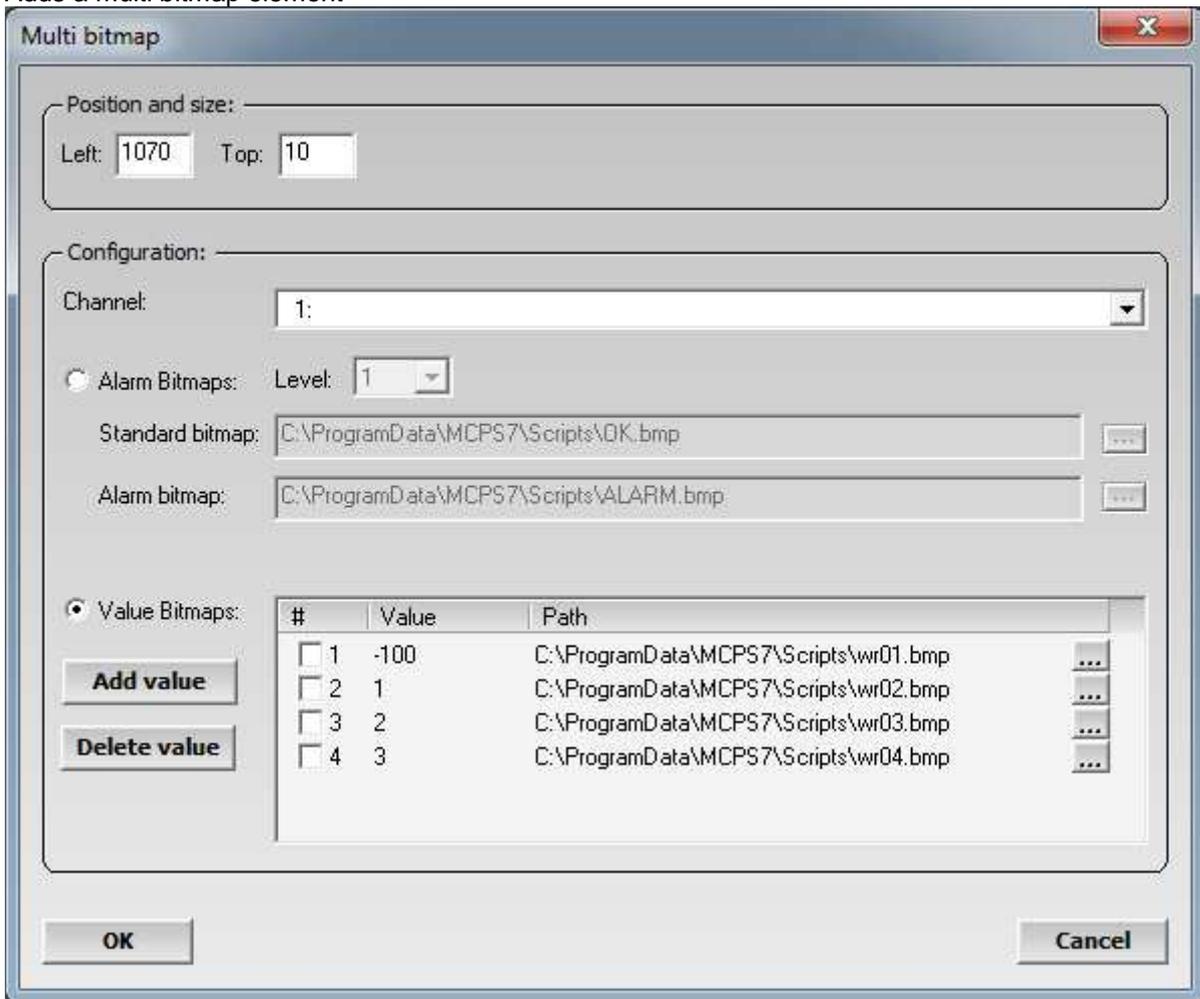
- Channel: 1 (selected in a dropdown menu)
- Show alarm limits

At the bottom of the dialog are two buttons: "OK" and "Cancel".

The analog meter shows the measured value of a channel. Existing alarm levels can be enabled.



### Adds a multi bitmap element



The multibitmap element is very flexible and can be used to simulate a lot of other typical process icons. Related to a channel value or alarm one bitmap out of a list of bitmaps is displayed. For example to show the status of a pressure value or petrol tank, several bitmaps must be defined to represent each status of the element.

#### 1. Alarm bitmaps:

For alarm monitoring between two bitmaps can be selected. The <Standard bitmap> is shown, when no alarm of the channel is active. If the file definition field is empty, no bitmap is shown. If the related alarm becomes active, the <Alarm bitmap> is displayed.

#### 2. Value bitmaps:

A list of threshold values and related bitmap files must be defined. According to the channel value the corresponding bitmap is shown. With <Add value> a new item is added and the list is sorted. All marked items (check box) are removed from the list with <Delete value>. In the process window above we see a wind rose, that consists of 16 bitmap files. A channel value from 0 to 10 represents a wind direction between 0 and 360°.



Adds a trend element.

In the editor there is only a place holder visible. During measurement the channel data is displayed in the same manner as the standard trend window does. Most of the settings are fixed and cannot be changed by the user. For each split screen the visible channels can be set. An active groupmanager allows the user to select the visible channels himself.



Adds a movie file (.avi)

Warning! Not all AVI-files are supported, because of different file types. This is a Windows restriction for this type of element. The start of the movie can be triggered by an alarm or a value of the channel. If the trigger condition is not active, the movie can halt or vanish.



Adds a simple circle, line or rectangle element



Adds a button to perform one or several actions

#	Action	Item	Options
<input type="checkbox"/> 1	Set digital outputs	W750 - 1: D01	...
<input type="checkbox"/> 2	Script	C:\ProgramData\MCP57\Scripts\...	A1 ...
<input type="checkbox"/> 3	Jump page	Title	...

All actions, which have been defined in the action list are executed, when the button is pressed. Since some actions can be used without data acquisition or even control it, the process window must be visible in this case immediately after loading the project. This can be set in [CONFIGURATION/MCP5 SETTINGS/PMON](#).

The following actions are possible:

Digital outputs can be set, reset, toggle or pulsed. Several outputs can be set at the same time

Analog outputs can be set with a constant value or a value can be requested (input window)

A script can be called with optional information, which can be requested by GetScriptData

A register can be set with a constant value or a value can be requested (input window)

Jump page. The specified page becomes the active page.

Start / stop project or batch, this requires a running session.

Configure batch (batch input mask), this requires a running session.

**Access rights:** In the user administration several levels can be defined for PMON access. These levels can be used for each button.

**Client control:** If a PMON is running on a MCP5 client and the client is connected by MSERV to the data acquisition computer, the actions of the buttons are executed on the daq computer. So it is possible to start or stop a project by a client.



Adds a slider element

#	Action	Item	Options
<input type="checkbox"/> 1	Set analog outputs	W750 - 1: A02	...
<input type="checkbox"/> 2	Set register	2	...

The slider values can be sent to an analog output or a project math register. The slider doesn't display minimum, maximum or current value. This must be added manually. But it's recommended to set a math register, which is not used in the project itself. This register value can be displayed with the text item. If you move the slider, the register text item will show the current value of the slider. See button element for access rights and client control.



Element configuration. Same as double mouse click on item.



Copies all marked items.



Sets an element in the z-layer. Background images should be placed on the bottom, buttons in front.



Deletes all marked items



Help

## Convert AutoFile to batchfile

This function is used to convert an AutoFile to a batchfile, which is normally done automatically after an AutoFile has been finished and the next one is generated. If the automatic procedure was not successful due to some circumstances, it can be performed manually. Therefore all settings of PROJECT/AUTOFILE/BATCHFILES are taken to create the batchfile and update the database.

## Create successor project

This function copies the active project and its data to a predecessor project, i.e. TEST.PRO is copied to TEST[1].PRO. The data files of the active project are deleted, so the configuration of the project can be changed in the way to add or remove raw data channels.

## Printer Setup

Configures the default printer. The selected printed is set again, if MCPS is started.

## Import

### Import File

#### Device format:

MCPS can import files, which are created by measurement instruments and stored to floppy or hard disks. The following formats from Yokogawa devices can be imported:

Yokogawa: DARWIN, DX, DL (wvf + rtm, kein Multifile), OR, VR, LR

It is possible to select several files which can be imported in one step and are combined to one final project.

Some instruments provide digital channels. MCPS detects those channels and sets the graphic output for them to **DIG** and not to **GR**. (See [channel parameters](#) )

#### DL708 /DL716:

The values of the channels of the digital input module are combined to one value from 0-65535. To extract the bits for each channel MCPS adds 16 math channels to the project for each digital module. Each math channel contains a formula to represent the right value 0 or 1. To do this the option /MATH1 is necessary.

#### Ascii format:

Number of lines to skip (header): 6

Use Date/Time evaluation      Sample rate: 10 seconds      Decimal character: .

Line format:

Skip:      Skip:      Skip:      Skip:      Skip:      Skip:      Skip:      Delimiter:      EOL:

0    Day    1    Mon    1    Year4    1    Hour    1    Min    1    Sec    1    59    10

OK      Help      Cancel

Import an ascii-file, where each line contains one scan with all channel value. The values are separated by the same delimiter. Each line ends with the end-of-line character. The first lines, which may contain no data, can be skipped. If the lines provide time information for each scan, it can be imported too, otherwise a sample rate must be defined.  
All time types are 2 characters long, except Year4 or --- (no type).

### **Import / Default import**

The user can import a file in two ways:

- Import: A new project is automatically created by MCPS
- Default import: The configuration of an already existing project is used to create the new one.

### **Import Memory Card**

MCPS can read data files from SRAM memory cards by using the MCRCAD232 cardreader. The following instruments from Yokogawa are supported:

- HR
- LR
- uR

### **Memory Download**

Some instruments can store the measured data into internal memory to be read out at a later time in one step. MCPS can download and import those data files to create regular projects. First MCPS asks for the device for download, which must be available in the device manager. MCPS will start communication and show a selection of data blocks or files for download. After the user selects a file of the device and a target project, MCPS receives the data and is assembling a new project. Some instruments will pause during the transmission in case of larger files. But after a while the download continues.

The DX/MV series with external memory can produce a lot of files. So MCPS supports multiple file selection by pressing the Ctrl or Shift key with the left mouse button. If several files are selected, MCPS will only copy them to the hard disk. If only one file is selected, it's automatically imported.

## Export

### **Export Channel Data**

ASCII export into a file or to printer. If AutoFiles are used, the export can be made over several AutoFiles (Set time range) or specific AutoFiles, which can be selected by pressing the **AF** button.

**- Export range:**

Sets start and end time for export.

**- Export type:**

Format of the evaluation program

**- Index type:**

X-axis description format.

**- Output options:**

Additional information like channel comment will be added at the beginning of the file. <Data header> will add the line [Data:] just before the measured values. This is helpful for external software to import the data.

**- Header output format:**

The output options can be organized in columns like the data or in rows (line by line). The second case is necessary for several evaluation programs, because they cannot handle a lot of channel information in one line.

**- Invalid data:**

Defines, how invalid channel data is output:  
As text string or as constant value

**- Data reduction:**

Averaging over: MCPS will output a mean value, which is computed over several scans.  
Export every: MCPS will skip values to reduce output size.

**- Set current settings as default:**

The export window will open with the default settings. Automatic export will use the default settings directly.

### **Export Limit Exceedings**

This function will create a list of time stamps containing the points of time, when a channel value exceeds the upper or lower limit as defined in the project configuration. The time stamps, the corresponding project channel numbers and H(High limit) or L(Low limit) are saved into a file with Excel format.

### **Export reference measurements**

Exports the data of the reference measurement file. In the export dialog the check box <ZM value> must be activated and the channels must be set to export (EXP) in the project configuration.

## Window manager

The window manager can save and restore window placements and related settings. This is possible even during the data acquisition, so that the user can easily change the presentation of his data. MCPS provides three different ways to handle the window properties.

### 1) Project related management

This type handles only all windows of the project. If the window placements are saved, only the windows of the current project are used. The information are stored in a file, that belongs to the project. If the project is duplicated, the window management file is also copied.

- Select new window layout



A window layout, which has been saved with the project, can be selected. All windows are being closed and those of the layout will be opened.

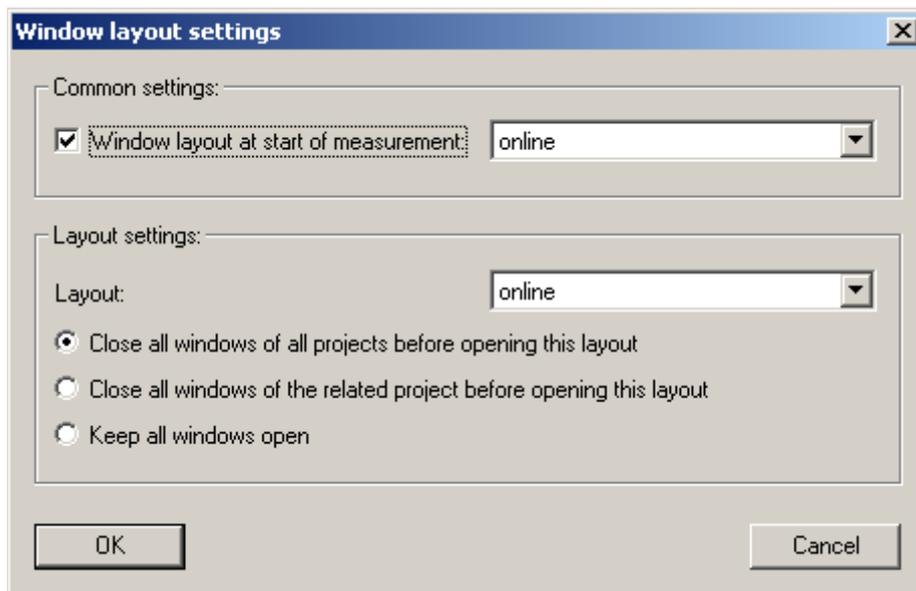
- Save current window layout

All windows of the active project and their settings are saved into the project's window layout file.

- Delete window layout

Deletes a window layout from the window management list.

- Window layout settings



After a layout has been saved some additional settings can be made:

- Window layout at start of measurement

The layout is automatically shown, when the data acquisition is started.

- Close windows

For each layout it can be specified, how to handle already open windows. MCPS can close all other windows before displaying the layout, close only all windows of the project or do nothing.

### 2) Multi project management

The first type is little bit to complex, if a lot of project are loaded and their windows are on the screen. Especially for online windows it's not useful to reload the layout for each project. To avoid this, MCPS can store the window settings of all projects together in one file. This global file belongs to no project, and so cannot be deleted or copied.

- Select multi project window layout

Loads a window layout from the global window layout file. For all projects, which are loaded, the windows will be restored.

- Save multi project window layout  
The windows' properties of all projects are saved to one window layout file.

- Delete multi project window layout  
A window layout is removed from the global window layout list.

### **3) Window layouts for common use**

Project related layouts, which should be used for other projects or for project independent functions, are save here and can be used for the following tasks:

These layouts are automatically available for all other projects.

Only these layouts can be used for batch printing. They are selected in a batch mask in the session configuration

These layouts can be selected, when a batch is loaded from the database. This is an easy way to open all batches with the same layout. See CONFIGURATION/BATCH for more details.

- Save common window layout  
Saves the window layout in a local file

- Delete common window layout  
Deletes a layout

## **MServ Startup Snapshot**

This function is used on NetView clients to save the current MServ condition. That means all connections, loaded MServ projects and the window layout is saved for the next startup of the client. In this case MCPS will automatically connect to the server(s), load the projects and displays the related window layout.

## **Change own login password**

If allowed by the administrator, the user can change his login password. The password is then unknown to everyone even the admin.

## **Logout**

If the **user manager** is active (to disable some menu items), the user should logout after his work is done. All functions will be disabled and a new user has to login again.

## **Smart Quit**

Normally MCPS cannot be terminated, if any data acquisition is in progress. A smart quit will save all measuring projects into a file and quit MCPS. The next time MCPS is started, the projects can be loaded and started again in one step.

## **Quit**

If no measurement is active, MCPS is terminated. Connected online clients will be disconnected.

# Display

## Numerical Offline Display



This window will display the project's data in list form. If the project contains AutoFiles it is possible to display all, only a part or only one of these AutoFiles.

In the window you can move the slider button at the right up and down to see more time scan and the bottom slider left and right to get more channels. In the main menu the item **OPTIONS** appears. This submenu contains the following features:



### Update data

If new data is available, the window is updated and the latest data scans are displayed.



### Jump to

A time dialog box appears to enter a start time. The related data scan is displayed.



### Relative time

The time information is shown zero based. The time display format is based on the absolute time information:

dd.mm.yyyy hh.mm.ss

Days.Months,Years Hours.Minutes,Seconds after the start of measurement. This means after 24 hours you will get 1 day and 0 hours next and not 25 hours.



### Group manager

This item can activate the group manager, if groups are defined. In the project configuration (see [Channel parameters](#)) it's possible to set the output mode for each channel in the way, that a channel is not visible in the text window.



### Font

**Defines the current text font.**



### Print

Printout of the numerical data. Time range and channels to be printed can be selected. If layout printing with 2 pages is used, the first print page is configured by the first layout page. All subsequent print pages are set to layout page 2, if exists. [See layout manager](#) for more details.

### Header

Up to six headers can be activated to give more information about each channel



Consecutive number



Unit



Tag



Comment



Channel name



Device name

**Restore toolbar**

The toolbar of the numerical window can be moved and also be closed. To bring the toolbar back on the screen this function is to be called.

In case of AutoFiles MCPS can merge several AutoFiles in one window or show only one file per window. In the last case a toolbar with arrows is added to easily jump to the next/ previous file.

## Graphic: Standard Offline Display



Graphic display of the channels as y/t or x/y diagram. When opening the window, MCPS displays all the project's measured data. The y-axis for each channels is based on the values in the project configuration (see [Channel parameters](#) ). Only those channels will be displayed, which are set to graphic output. When AutoFiles are used, MCPS can show the current selected AutoFile or all files together.



**Stop drawing**

Two arrows below the y-axis select the next/previous channel.



This button jumps to a channel directly.

This window adds an option menu to the main menu bar with the following items:



**Instant trend redraw**

The graphic is updated with every action the user makes. In case of larger files, this can take some time especially when several action are to proceed. If the instant redraw is turned off, no redraw occurs and all settings (axes, groups, screens, ...) can be made without delay.



**Update data**

If new data is available, the window is updated and the latest data scans are displayed.

ZOOM:



**Reset**

Resets the scaling for x/y axis



**Zoom**

This enables the mouse to mark an area, which can be zoom in x,y or xy-direction.



**XZoom**

Increases the x-axis resolution one step



**XUnzoom**

Decreases the x-axis resolution one step

### - Manual X-Zoom

A dialog window is displayed to enter a new start and end time for the x-axis. If MCPS should exactly scale the axis to these values, X-axis autoscaling should be turned off.

DISPLAY:

Display type

#### - Absolute time

X-axis description with absolute values.

#### - Relative time

X-axis description with relative values related to the first data scan's time stamp.

#### - Relative hours

X-axis description with relative values in hours related to the first data scan's time stamp.

## - XY

XY display. All channels are related to the reference channel.

### - Define x-axis channel

X-axis channel for the XY display

Number of axes

Up to 5 Y-axes are possible

### Y-axis description

Selectable between channel tag and comment (see [channel parameters](#))

### Number of axes

Up to 5 Y-axes are possible

### Number of markers

The frequency the channel's marker is drawn at the curve.

### Number of screens

The graphic display can be divided into several subscreens. All these screens have the same x-axis scaling and an independent group manager. Each additional screen can be set to analog or digital display format (see properties below).

### X-axis autoscaling

This function will scale the x-axis to more even and better readable values. In some cases it's possible, that the shown time window is much larger than desired by the user for example after an x-zoom. If turned off, the x-axis fits better to zoom operations.



### Grid

Turns on/off the graphic grid

### Connect points

MCPS will draw a line between the measurement points, if this feature is enabled.

### Fast display

To get a fast view over large files this feature can speed up the display. MCPS will not load all measurement points from disk, but in greater steps. This method reduces load and display time. It should not be used with math formulas, which contain integration functions or use the last measured value, because only a part of all data values is processed. MCPS will load as many scan as necessary to fill the graphic window without holes. If the user zooms in, MCPS will load the data in smaller steps. In case of large files for example DL708 import projects, it's useful to select the fast display and to get a glance over the file. Then the interesting part should be zoomed and the fast display mode turned off. MCPS will redraw the screen with the current scaling and all measurement points in this time range.

The *Fast Display* can be preset in the [MCPS Settings/Graphic](#) window.

EXTRAS:

### Channel events

**This menu item opens a dialog window to manage the channel related events. An event belongs to a specific channel at a certain time position and can contain a text string, which is placed at the event.**

To add a channel event, you need at least one active cursor. You have to position the cursor at a specific channel event for example an extraordinary peak. Then call this menu item in the options menu. With the button ADD EVENT AT CURSORx you can add a new event at the cursor's position for the selected channel. Some parameters like event text, text color and text adjustment can be changed. The channel events are saved with the project file and will be displayed each time the graphic offline window is used.

A double mouse click on an event text in the graphic window opens the channel event dialog window with the related event.

In case of many events the filter function is useful. Enter a text in the input field and press <Set filter>. Only matching events are shown in the dropdown list. Wildcard \* can be used. After pressing <Reset filter> all events are visible again.

### **Graphic export**

Exports the graphic window to the clipboard or into a file. MCPS uses the Enhanced Metafile Format, standard Windows Bitmap files (bmp) or JPEGs. For clipboard operations with large projects a lot of memory should be available. If it's not possible to get the graphic via clipboard into for example Word, then use the MCPS/file export and Word/Import method. The EMF-Filter must be installed in WORD.

### **Comparison settings**

This menu item is only enabled, if several AutoFile records should be compared (see [Display/Compare/AutoFile records](#)).

Each records display can be turned on/off. One record can be used to be a reference file, which gets its own color.

### **Alarm file configuration**

This menu item is only enabled, if a dynamic alarm file should be created or modified (see [Dynamic alarm file](#)).

### **Show dynamic alarm file limits**

If an alarm file is activated in the project configuration, it's possible to show the curves of the alarm limits for each channel. This can be used in online trend windows and alarm file configuration windows.

### **Restore toolbar**

If the graphic window is active, its toolbar is displayed the main frame. This toolbar can be moved over the screen and even be closed. Use this menu item to restore the closed toolbar.



### **CURSOR OPERATIONS**

If cursors are defined, it's possible to export, delete or copy only those data scans, which are between the cursors.

**PRINT:**

Printout of the graphic window with the current settings. The user can [select a print layout](#), which was created with the [layout manager](#).

Print options:

Multi page printout:

Using a fixed number of pages (sheets of paper) the x-axis is divided into several time spans

Using a fixed time range for the x-axis results in several pages depending on the total x-axis width.

**Speed:**

Not every scan is printed.

**Print range:**

A partial time range can be selected.

### SHOW GROUP MANAGER:

If groups are defined this features enable or disables the group manager in the right part of the graphic window. MCPS will display those channels, which are in the marked group or in related subgroups. The groups or the width of the groupmanager are defined in the project/configuration (See [Group manager](#)).

### PROPERTIES:

Definition of number and type of subscreens

Show full event texts: An event is normally marked with an 'E' at the x-axis. With this option the full event text is display above the graphic

Show alarm ranges: A small bar below the graphic shows all part in red, where at least one alarm is active. A double click on that part will open the offline alarm window and jumps to that time position. Since several alarms can be active the first one, which is in this time range will be shown.

Y-Axis: The number of lines and steps of description can be selected for this window session

### Cursor control bar:



Turn on cursor 1



Move cursor 1 one scan left



Move cursor 1 1/40 of time axis to the left



Move cursor 1 1/40 of time axis to the right



Move cursor 2 one scan right



Öffnet einen Dialog mit verschiedenen Operationen. Einige sind nur mit beiden Cursors möglich und beziehen sich nur auf den definierten Zeitbereich (z.B. ausschnittsweise kopieren).



Turn on cursor 2



Move cursor 2 one scan left



Move cursor 2 1/40 of time axis to the left



Move cursor 2 1/40 of time axis to the right



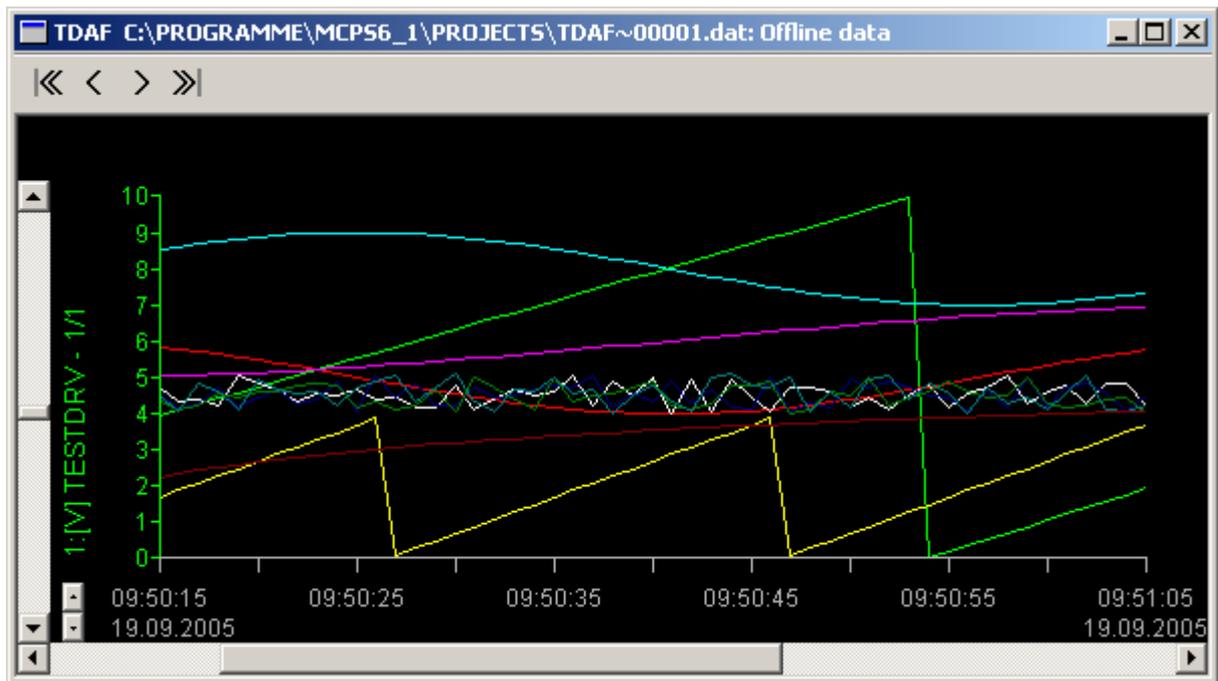
Move cursor 2 one scan right

To turn on cursor 2 you have to activate cursor 1.

Smart cursors

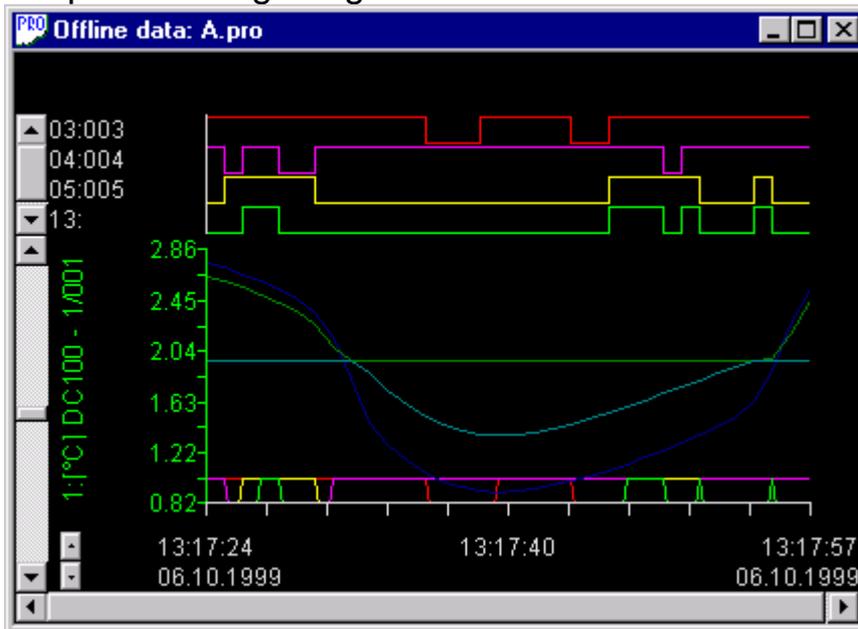
By pressing the CTRL-button and the left mouse button you can define a rectangle in the trend window to directly set both cursors at the edges of the rectangle.

AutoFile selection



In case of projects with AutoFiles the offline trend window has an additional control bar to step through the file or to jump to the first or last file of all records.

## Graphic: Analog + digital trend



This window shows two subscreens; one digital and one analog data display. All channels, which **DIG**-button is set in the project configuration, are visible in the upper part of the window. All channels with the **GR**-button set are shown in the bottom part. See also [channel parameters](#).

## Graphic: Trend of current alarm channels

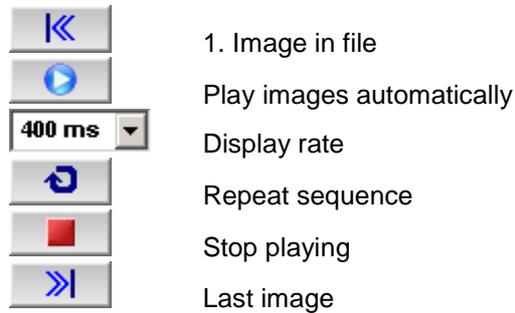
Opens an offline trend window with all channels, which have an active alarm in the moment the window is opened.

## Profile display

This window type will show all channels of a scan in profile view. An additional horizontal slider at the right side of the main slider is used to step through each data scan.

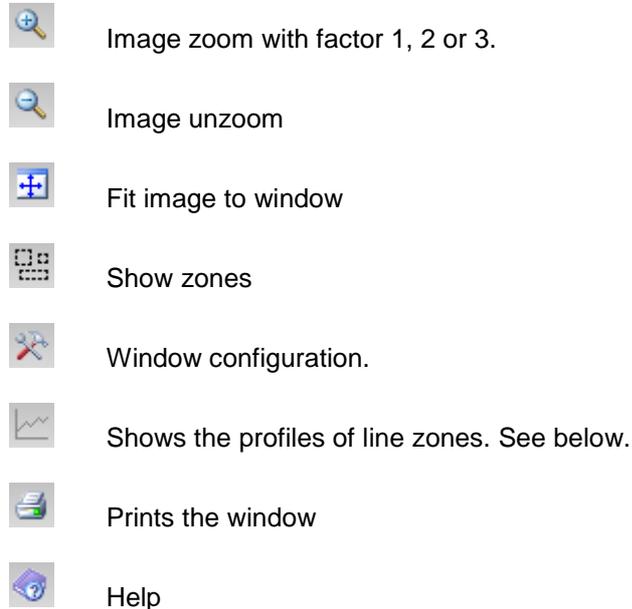
## Thermo display

Projects with thermographic device save the image data in special data files. This menu item is used to display the images of those files:



A double click on the color bar opens the window settings:

Toolbar:



Profile window:

If one or more line zones are defined, the data profile over the lines can be displayed in an extra profile window. Press the  button to select the lines. You can configure the profile window by pressing the right mouse button or by clicking on the y-axis. In addition to the profiles, MCPS can compute and show three statistic curves: Minimum values of all curves, average values and maximum values. For better documentation it is possible to define color, line width and comment for each curve. These information are stored with a window layout.

## Reference measurement file

Displays the values of all reference measurements. The active set is marked with an asterisk after the date and time information.

## Logfiles

### Event Logfile

This menu will show the project's event logfile. This file is created by the user by adding event texts during the data acquisition or by importing data files with event information (VR200). In the graphic window each time event is marked with an **E** at the x-axis. The logfile can be printed, exported or deleted under [Project/Data](#).

### Alarm Logfile

Alarms, which have been logged into the project's alarm file, will be displayed with this menu function. See also [channel parameters/alarms](#) to enable alarm logging. The alarm can be saved to the project's alarm file and/or to the global alarm logfile, where all alarms from all projects are being saved. See [Configuration/MCPS Settings/General](#).

With the mouse button or the menu option/mark several lines can be highlighted. These lines can be printed or exported. If no lines are marked, MCPS will output all lines.

### Global Alarm Logfile

This file contains all alarms with alarm logging enabled. The project's name is added to the alarm line. All other settings are the same as for the project [ALARM LOGFILE](#).

### Global email and sms logfile

SMS and email alarm actions are logged into this file to see, if an action has been initiated or if any error has occurred. With every month a new file is created.



Loads a different month



Exports list to Excel



Deletes list of current month

### Administrator logfile

This logfile shows administrator related information like invalid login attempts.

## Audittrail

Time	User	Action	Project	Special information	Comment
22.10.2012 16:23:35	Administrator	Login			
22.10.2012 16:26:39	Administrator	Logout			
22.10.2012 16:33:07	Administrator	Login			
22.10.2012 16:33:14	Administrator	MCPS settings modified		Auto logout: Off, 60s (On, 60s)	
22.10.2012 16:50:33	Administrator	MCPS settings modified		(Directories) Project directory: C:\Users\Public\Documents\MCP57\Projects	
22.10.2012 16:50:37	Administrator	Project loaded	C:\Users\Public\Documents\MCP57\Projects\Example.pro		
22.10.2012 16:50:49	Administrator	Project loaded	C:\Users\Public\Documents\MCP57\Projects\EXAMPLE_ADVANCED.PRO		
22.10.2012 16:59:20	Administrator	Quit MCPS			
23.10.2012 10:03:51	Administrator	MCPS Start			
23.10.2012 10:04:11	Administrator	Login			
23.10.2012 10:05:12	Administrator	Project loaded	C:\Users\Public\Documents\MCP57\Projects\EXAMPLE_ADVANCED.PRO		

Only accessible for administrators.

All user actions are logged into monthly audittrails, which are handled as one file. MCPS starts with the last 30 days and all actions. Old values before modification are shown in ().

To see a greater time span or certain actions you have to define a new filter condition.



A new filter condition is set, to find specific information.

**Audit trail filter** [X]

Start time:

End time:

Action:  ▼

User:

Project:

Special info:

For text search you can use wildcards (\*!). Empty fields means all elements of this type.



Export list to Excel.



When data was deleted or a setup was sent, these information must be hold in the audittrail. To easilly review these data, the related line must be highlighted and the load button must be pressed. In case of deleted data a temporary project is loaded into MCPS containing the removed data. In case of a setup, the button opens the setup editor of the devic to show the settings sent to the device.



Output to printer

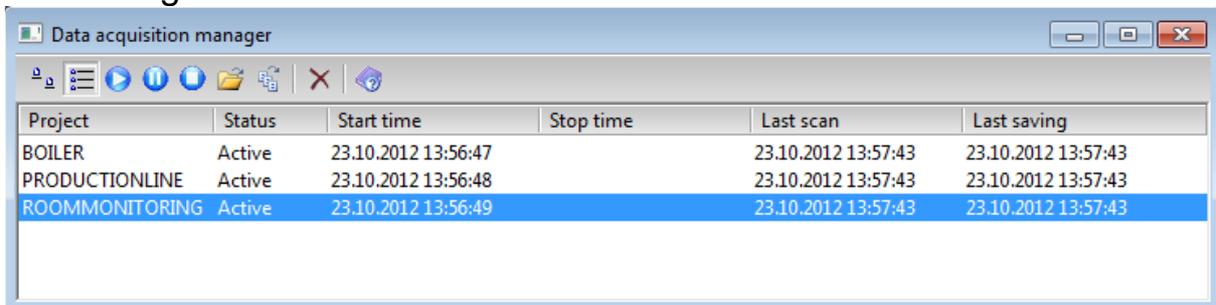
## Comparisons

### **Compare AutoFile Records**

Some records of a project with AutoFile records can be compared. MCPS will overlay the data curves within a relative time graphic display. In this mode the menu item [Options/Extras/Comparison settings](#) is enabled to modify some parameters.

# Data acquisition

## DA Manager



The screenshot shows the 'Data acquisition manager' window with a table of active projects. The table has six columns: Project, Status, Start time, Stop time, Last scan, and Last saving. Three projects are listed: BOILER, PRODUCTIONLINE, and ROOMMONITORING, all with a status of 'Active'. The ROOMMONITORING row is highlighted in blue.

Project	Status	Start time	Stop time	Last scan	Last saving
BOILER	Active	23.10.2012 13:56:47		23.10.2012 13:57:43	23.10.2012 13:57:43
PRODUCTIONLINE	Active	23.10.2012 13:56:48		23.10.2012 13:57:43	23.10.2012 13:57:43
ROOMMONITORING	Active	23.10.2012 13:56:49		23.10.2012 13:57:43	23.10.2012 13:57:43

The data acquisition manager (DA manager) handles all measurements running in MCPS. It organizes the start and stop of each data acquisition and shows a list of all measuring projects. If a project is loaded into the DA manager, MCPS first receives necessary device information to initialize the complete measurement. Then the data acquisition is started and processed as defined in the project configuration (data reduction, autofile, ...).



### Large icons

Displays project symbols and names.



### Details

All measuring projects are listed with additional information (e.g. start time, last scan, last saving, ...)



### Add project

A project, that is already loaded into MCPS and isn't not yet measuring, can be added to the DA manager.



### Pause project

The project is stopped, but not removed. If pressed again, the data acquisition continues.



### Remove project

A selected (marked) project will be removed from the DA manager. The measurement for this project will be **stopped**.



### Load and add project

This button allows to load a project from disk and start it.



### Start project in project list

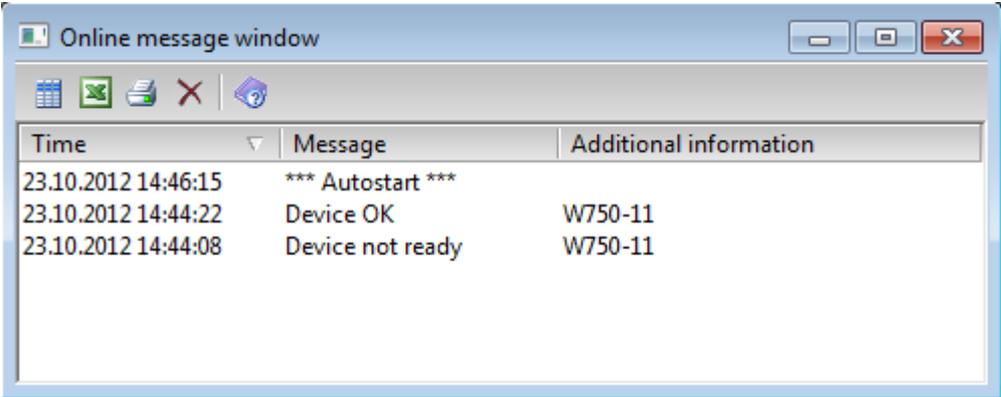
All projects in a selectable project list are loaded and started.



### Remove all projects

This will stop all measurements and remove all projects in one single step.

# Message Window



During the measurement MCPS will create messages for communication problems or other events. These messages are saved into a file and can be displayed everytime by this menu item.



Marks lines for export



Exports list to Excel



Output to printer

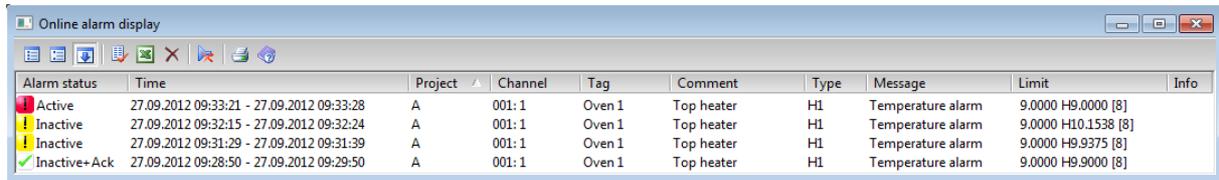


Removes all messages.

The number of messages kept by MCPS and specific actions like emails can be set in [Configuration/MCPS Settings/General](#).

## Alarm Window

(Only with option /ALARM)



Alarm status	Time	Project	Channel	Tag	Comment	Type	Message	Limit	Info
Active	27.09.2012 09:33:21 - 27.09.2012 09:33:28	A	001:1	Oven 1	Top heater	H1	Temperature alarm	9.0000 H9.0000 [8]	
Inactive	27.09.2012 09:32:15 - 27.09.2012 09:32:24	A	001:1	Oven 1	Top heater	H1	Temperature alarm	9.0000 H10.1538 [8]	
Inactive	27.09.2012 09:31:29 - 27.09.2012 09:31:39	A	001:1	Oven 1	Top heater	H1	Temperature alarm	9.0000 H9.9375 [8]	
Inactive+Ack	27.09.2012 09:28:50 - 27.09.2012 09:29:50	A	001:1	Oven 1	Top heater	H1	Temperature alarm	9.0000 H9.9000 [8]	

All alarms which are active and have a **message action** (see channel parameters) are listed in this online window. An alarm can be acknowledged by pressing the left mouse button on the alarm status image. All inactive and acknowledged alarms can be removed. By pressing the right mouse button on the alarm status field MCPS opens the log event window to enter a message. This window can be opened automatically with acknowledging an alarm. See [CONFIGURATION/MCPS SETTINGS/MESSAGES](#).



Shows all **active** alarms



Shows all monitored alarms, even inactive alarms are listed.



Shows all alarms of each channel separately. Each new alarm is a new entry in the list. MCPS will hold up a certain number of alarms in memory. This number can be set in CONFIGURATION\MCPS SETTINGS\PROJECT\ALARM BUFFER SIZE. This value should be set in relation to number of projects, sample rate, channels and system performance.

The headers of columns <alarm status>, <Time> and <Project> can be used to sort the list.



Acknowledgement of all alarms



Exports list to Excel



Removes all inactive **AND** acknowledged alarms.



Deactivate current alarm sound



Print alarms

The online alarms can also be transmitted to MCPS clients with the option /MSERV. If the clients opens the alarm window, MCPS is establishing a connection to the measurement PC. See chapter [online data and alarms on network clients](#). A client can also acknowledge or remove alarms.

MCPS can save the current alarm conditions into a file to be reloaded after power fail or MCPS shutdown and restart. This must be set in CONFIGURATION\MCPS SETTINGS\PROJECT\BACKUP ALARM CONDITIONS

## Log Event

During the measurement the user can enter text strings, which are saved into a logfile with the current time stamp. So it's possible to comment certain events. With [DISPLAY/LOGFILES/EVENT LOGFILE](#) the file can be shown on the screen. In the graphic windows each event is marked with an **E** at the x-axis. A client can also enter text, if an online project (/MSERV) is used.

## KEYDRV value

Changes the value of KEYDRV channel. Used during a measurement to enter a new value by keyboard. The DEVICES/KEYDRV for more details.

## Update device status

The project configuration can be setup to sent the channel alarm limits to the device, when they are changed by the user. If the device is changed manually or setup is sent, this function sends the current alarm limits again to the devices.

## Manual data saving

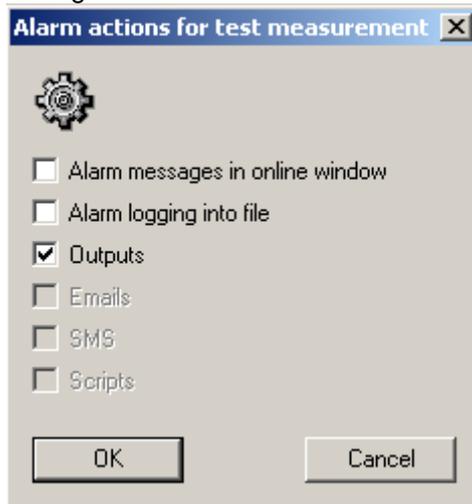
With this menu item the next scan(s) can be saved into the data file. The user can select how many of the next scans will be saved. It is recommended to disable regular data saving (see [data reduction](#)).

## Test



This menu is used to test the communication to all devices and to run a test measurement under the same conditions as a normal data acquisition will do. The project's samplerate is ignored and saving is disabled. MCPS computes the measurement time to see, if the samplerate of the project is possible. The test measurement can be stopped by closing the window. The sample rate of the project should be higher than the measured time for one complete scan cycle!

If the project will start specific actions on alarms, they are disabled by default, but can be enabled for testing.



## Reference measurement

The reference measurement or calibration will make one or several scans to define offset values, which can be subtracted during a normal data acquisition. It is possible to run several reference measurements, even after a data acquisition. But only one set of values can be used for calibration. All ZM-values can be displayed under **DISPLAY/REFERENCE MEASUREMENT**. The active set of offset values can be changed in the project configuration: **PROJECT DEFAULTS/REFERENCE MEASUREMENT**. The offset values are also available as math variables for computations. Use Z1 ... Zn to have access to the current set of calibration values.

## Start



The current active project (listed in the main title bar) is added to the [data acquisition manager](#). This has the same functionality as the DA manager's *Add Project*.

Additionally the window layout can be defined and special project settings are shown.



It's possible to define a [window layout](#), which is automatically shown at startup. See [project configuration/defaults/acquisition](#).

Before the measurement starts, MCPS has to receive the channel information from the devices. All devices, which are not responding, are listed. MCPS can start the data acquisition even without a connection to a device, if this instrument supports the suspend mode. See [DEVICES/DEVICE INFO](#).

## Pause



The project is stopped, but not removed. If pressed again, the data acquisition continues. There is no start marked in the data file and the math functions are not reset.

## Stop



The measurement of the current active project is stopped and removed from the [data acquisition manager](#). This has the same functionality as the DA manager's *Remove Project*.

## Text Window



#	Chn	Tag	Value	Unit
1:	1	Generator	1	V
2:	2	Pressure	8.9518	bar
3:	3	Coil	4.6931	V
4:	4	Door	6.5887	V
5:	5	Sensor	3.7654	V
6:	6	Flow	24.7179	V
7:	7	Current	4.5373	mA
8:	8	Power	4.0694	W

The text window is the numerical window for online data. If not yet open, this menu item will open the text window for the active project. Otherwise MCPS will bring up the window into the front. If another text window should be opened, the menu item **Window/New acquisition window/Text window** must be selected.

 No tags

 Short tags

 Full tags

 No comments

 Short comments

 Full comments

 Device name

 Text font

 Draw measured values with channel color

 Group manager

## Trend Window

The trend window is the graphic window for online data. If not yet open, this menu item will open the trend window for the active project. Otherwise MCPS will bring up the window to the front. If another trend window should be opened, the menu item [Window/New acquisition window/Trend window](#) must be selected.

The trend window provides some functions of the [Graphic offline display](#) .

A trend window specific menu item is

### - Lock window

The window's size cannot be changed and prevents graphic redraws.

Properties / Online

- Do not clear window in case of relative time and new AutoFile:

Normally the screen is cleared, when a new AutoFile is created and data is displayed with relative time format. With this button active, the new data is drawn over the old one.

- Do not shift window

The time axis is not shifted

For each project MCPS defines a history buffer of several hundred scans. If the trend window is redrawn, the data is not loaded from disk, but from this buffer. If a trend window is time scaled to show a lot of data, it's possible to lose part of the display, when the window must be updated. This happens, if the window is resized, a group is selected or the scaling has been changed. If the trend window is only overlapped by another window of any Windows application or minimized by the user, the display is not affected. This is achieved by a special virtual screen technique.

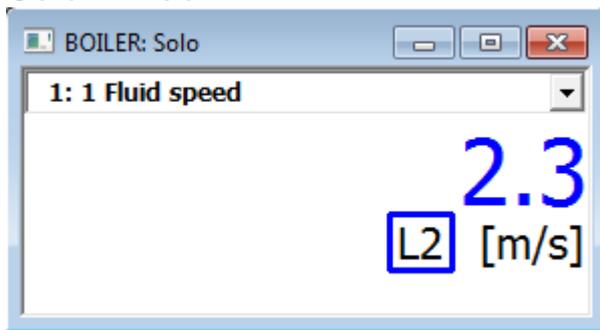
The size of the redraw buffer can be changed. See [Configuration/MCPS Settings/General](#).

## Long Term Trend Window

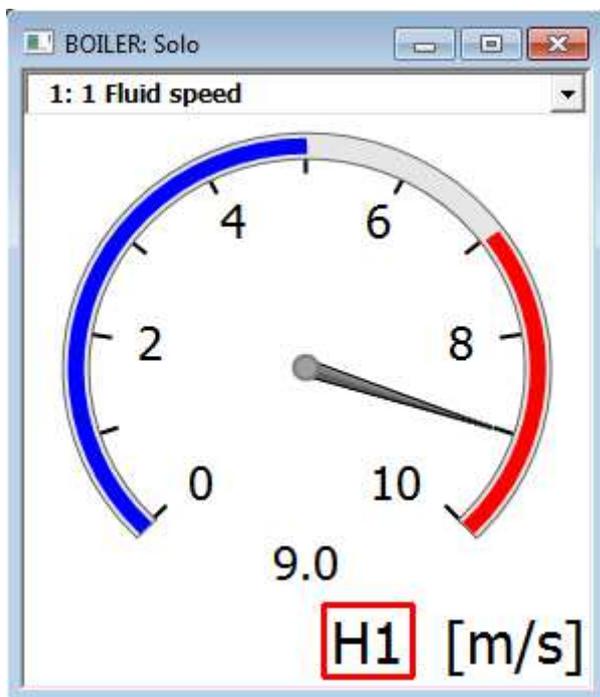
This feature is only enabled, if the long term window functionality is activated in the project configuration ([options/extras](#)).

This special window will skip some scans and won't show every scan. Since the long term trend window has its own redraw buffer of the same size as the regular trend window, it can hold redraw data for a longer time. On the other hand this means, that a data peak might be not shown. The long term trend window is used generally to get a view of the measurement in the past.

## Solo Window

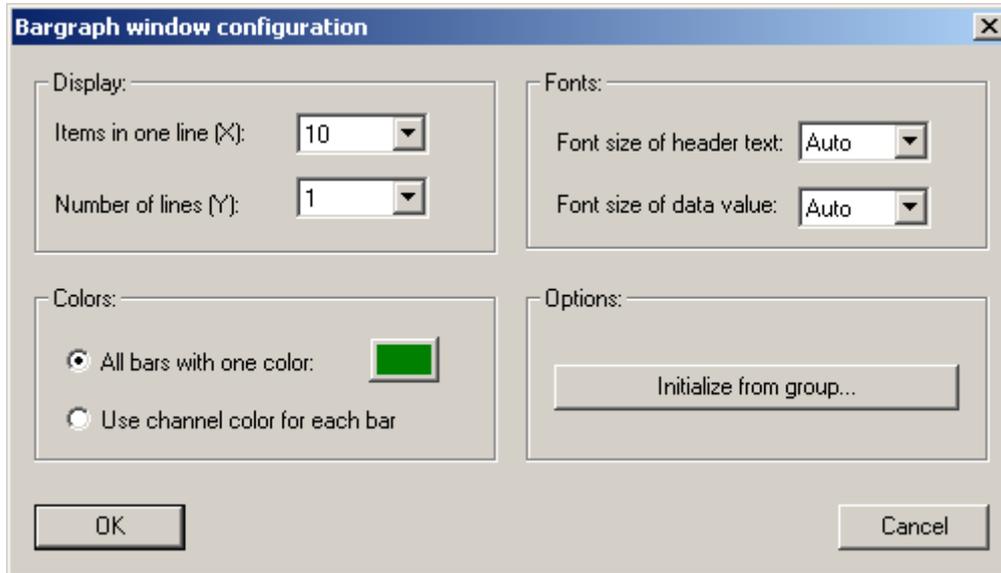


The solo window displays only one channel. It's possible to size the window to get a large display as digital number or analog meter. Use the options menu or right mouse click in the window to select type and size of window. If not yet open, this menu item will open the solo window for the active project. Otherwise MCPS will bring up the window to the front. If another solo window should be opened, the menu item [WINDOW/NEW ACQUISITION WINDOW/SOLO WINDOW](#) must be selected. Active alarms are displayed with type and level.



## Bargraph

### Window configuration

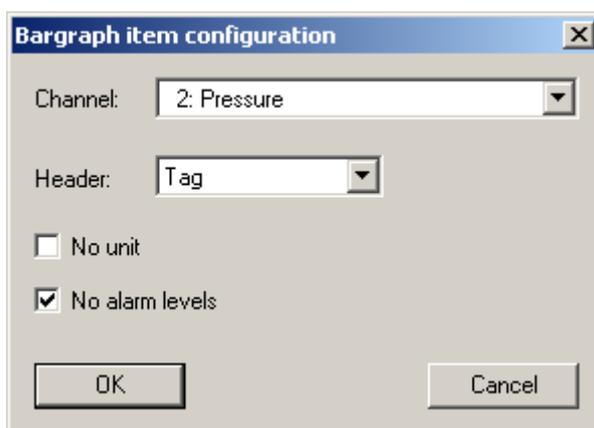


The bargraph window can be organized in 50 columns and 10 rows. All bars can have one color or the color of the channel.

A bar graph window can be initialized by a group. MCPS will compute the best row and column values and set the items to the members of the group. After the OK button is pressed, MCPS asks for item settings, that should be used for all bars. So you can change for example the header directly for all elements.

### Bargraph item configuration

You have to double click on the element to change settings.



- Channel:  
The channel with tag of the selected item is displayed. It's possible to choose another one.

- Header:  
Text, which is placed in the item's header (channel number, name, tag,...).

- Unit and alarm levels can be disabled, if not necessary.

## Profile window

Each scan is displayed in profile form. If the window already exists it is put to the top of all windows, otherwise a new window is opened. To have more than one profile window open at the same time you have to select WINDOW/NEW ACQUISITION WINDOW/PROFILE.

## Image window

If the project contains thermographic devices, the images can be displayed in this window. If more than one instrument belongs to the project, the desired device must be selected. A double click on the color bar opens the window settings;

Setting of upper and lower temperature for y-axis

Color system

Display of y-axis with pixel/user values

Alarm frame: In case of an alarm of the zone the related frame is flashing. The alarm limits are set in the project configuration/channel configuration

Toolbar:



Image zoom with factor 1, 2 or 3.



Image unzoom



Fit image to window



Show zones



Pause. The image display is hold. Temperature value at mouse point is shown.



Window configuration.



Shows the profiles of line zones. See below.



Prints the window



Help

Profile window:

If one or more line zones are defined, the data profile over the lines can be displayed in an extra profile window. Press the  button to select the lines. You can configure the profile window by pressing the right mouse button or by clicking on the y-axis. In addition to the profiles, MCPS can compute and show three statistic curves: Minimum values of all curves, average values and maximum values. For better documentation it is possible to define color, line width and comment for each curve. These information are stored with a window layout.

# Mathematics

## Statistics

MCPS computes the statistical values (minimum value, mean value, maximum value, standard deviation and **Mean Kinetic Temperature**) for the current project. A time range can be set to evaluate all or only partial data. The result can be printed or exported. This function can also be called by the cursor operations from the graphic offline window. In case of AutoFiles the statistic can be run over several files.



Display and output configuration. Defines the channels to be output, the computations to be made and the number of decimal places.



Export of statistic results



Printout of statistic results

The standard deviation is at least divided by  $n-1$  values (not by  $n$  values).

The MKT is computed over all values (in the specified time range).

## User Functions

User functions are stored in project independent user function files (see chapter [concept of user functions](#)). Each file contains up to 500 functions. First you have to load a function file. If no file is available a new one must be created with [Mathematics/New files](#). After a file is loaded the first user function is displayed. For each function a comment and a 200 character formula can be entered. The parameters used are named with **A**, **B**, **C** and so on. It's possible to use standard operators and functions. Examples:

1.  $A+B-C*A+5.32$
2.  $SQRT(A^2 + B^2)$
3.  $SIN(B) - COS(A)$

A syntax check is performed, if a new function is selected or the file is saved. In a project the first example can be called in this way:

$F1(X1, 22.5, X3-2)$

It's important to use the right number of parameters. The value of X1 will be transferred to parameter A, 22.5 to B and X3-2 to C.

Since the user function file is project independent, it should be connected to a project, which uses user functions. This is done in the project configuration (see [options/path](#)).

## Polynomials

For detailed information about polynomials see also chapter [use of polynomials](#). First you have to load a polynomial file. If no file exists, one can be created with [Mathematics/New files](#). Up to 500 polynomials are handled in one file. After the file is loaded the first polynomial is displayed. Up to 20 xy-pairs can be entered to compute the coefficients of the polynomial. Important xy-pairs can get a higher priority by changing the weight to another value (0-999). MCPS can create polynomials with a 9<sup>th</sup> power. The DISPLAY button will show a curve based on the current polynomial.

Since the polynomial file is project independent, it should be connected to a project, which uses polynomials. This is done in the project configuration (see [options/path](#)).

MCPS can also import polynomial files, where the coefficients are saved in ascii-format. Each line is one polynomial and can overwrite one polynomial of the current file. The import filter defines, how MCPS should understand the parameters in the line.

## New Files

This menu item will create an empty polynomial or user function file.

## Time measurements

### Concept

This function is intended to automate time measurements between two events. It's possible to define up to 10 start and stop conditions to determine the time between these occurrences. E.g. an oven is powered up and the time from 100°C up to 250°C should be found out. The temperature is available in channel X6:

Start condition:

X6 >= 100 (timer is started)

End condition

X6 > 250 (timer is stopped)

MCPS will examine the complete offline data file and check for all time measurements. The results are displayed in a window.

### Run

This menu item will load a time measurement list from disk. This file is project independent and can be used even for other projects. Important, the channel numbers used in the list should exist in the project. After a file is selected and loaded, MCPS examines all offline data. The results are shown in a window.

## Edit

C:\PROGRAMME\MCP56\_1\MATHFILES\VENTILTEST\_E.TMF

	Timer start condition:			Timer stop condition:			Comment:
<input checked="" type="checkbox"/> 1:	X1	>=	100	X1	>=	150	Temperature
<input checked="" type="checkbox"/> 2:	X2	>=	1	X3	>=	1	Power up delay
<input checked="" type="checkbox"/> 3:	M7	<=	X4	M7	<	2.5	Weight reduction
<input type="checkbox"/> 4:	X1	=	1	X1	!=	1	
<input type="checkbox"/> 5:	X1	=	1	X1	!=	1	
<input type="checkbox"/> 6:	X1	=	1	X1	!=	1	
<input type="checkbox"/> 7:	X1	=	1	X1	!=	1	
<input type="checkbox"/> 8:	X1	=	1	X1	!=	1	
<input type="checkbox"/> 9:	X1	=	1	X1	!=	1	
<input type="checkbox"/> 10:	X1	=	1	X1	!=	1	

In this window the user defines the start and stop conditions for the time measurements. Each condition consists of three parts:

Variable to compare. Must be a project channel Xn or Mn

Type of comparison

Value to compare. Must be Xn, Mn or any constant

The list is saved into a project independent file and can be used for every project, which contains the channels used in the list. Only those lines with the check box at the left set to active will be used for evaluation.

# Configuration

## MCPS settings

### Passwords

#### - Passwords (for the hardware protection module/Dongle)

MCPS is protected by a combination of hardware protection (Dongle) and a related password. Each customer gets his own dongle and his own password, that is created for this dongle. Without a dongle or the correct password MCPS is started in demo mode. In this case only one physical channel of one device can be measured. More channels in a project are only possible with the TESTDRV's channels. This driver generates data to test the functionality of MCPS even without having any instrument. Therefore there is only one MCPS version necessary, which can be used for installations with dongles, for demonstration purpose or even for updates. We don't deliver any special demo disk of MCPS, because it's always the same program. This means also, that all features are included in the program. Which of them are available to the user depends on the related password. Only those functions will be accessible, which are encrypted in the password.

If the user wants to expand the functionality of MCPS, he can order the appropriate option and will receive a new password.

Often several MCPS packages are used in one company. Normally you have to connect the right dongle to the computer with the related password. To make the handling easier, it's possible to enter up to ten passwords even from other dongles. So you need only one of these dongles to go to work. MCPS will use the first password, which fits to the connected dongle. To enter several password press the button **<password table>**. This is also useful to enter a password, which is in the clipboard. Press the button ... right of the password to enter or copy a new one in one line.

#### Search for network dongle

If a MCPS package has no local dongle and should be run with the network dongle (as a NetView-MCPS), this button must be activated. To access a computer with the Netview-Dongle, which is not part of the client's sub net, you have to specify the name or IP address of the Netview server.

Since MCPS supports several kinds of dongles and manufacturers please read also the chapter [Dongle](#) to get all information about installing and configuring the Netview-Dongle.

#### - Show all instruments, when adding a device

If you want to add an instrument to the device manager, MCPS will normally only show those devices which are free to be used concerning your password. Nevertheless in some cases it's useful to see all devices. For example to setup a configuration and a new project, which should be used with another MCPS package.

- Enable MSERV-Client features

You have to activate this button, if you want to see data and alarms from a MCPS package running on another PC, which is connected by network. You have to restart MCPS to get the data acquisition menu and the **MSERV** tab in this configuration sheet.

- Enable batch manager menu

The menu *batch manager* is not visible by default and must be turned on.

## General

Workspace mode:

MCPS supports more than one project at the same time in the workspace. For some applications or customers it might be easier to have only one project. The single project mode will remove the current project, if a new one is loaded (compatible to the MCPS DOS version).

*Smartmenu*: Several menu items are expanded by icons

*Startup size*: Size of the mainframe window, when MCPS is started.

Extras:

Send PC time to instruments:

The computer's time information is sent to all devices of a project the next time the project is started for data acquisition or MCPS detects a deviation of more than 2 seconds between device and computer during a measurement. Once the time is set, the next update is not made before the next 60 seconds to avoid permanent settings in case of configuration problems.

Cardreader:

Selects the RS232 port the MRCAD cardreader is connected to.

Allow device suspension at start of measurement:

Normally all devices of a project must have a valid connection before the project can start its data acquisition. If this check box is enable, all devices without access can be suspended. The project will start anyway and the suspended devices will deliver no data. In the device manager there is for most of the device types a check box, where you can set the device suspension manually. It is also possible to unsuspend a device during the measurement. If the connection test is OK, MCPS will receive the device information and than data is taken, displayed and saved normally. See also [DEVICE MANAGER](#) and [DEVICE INFORMATION](#).

Alarm logging mode:

If an alarm should be logged into a file (see channel parameters), MCPS offers two destinations to stored the information. Normally the alarm is saved to the project's alarm file. This one will be copied with the project and shows only alarms of that project. The project's alarm file can be seen with [DISPLAY/LOGFILES/ALARM LOGFILE](#). Additionally the alarm can be logged to a global alarm logfile. There all alarms of all projects are stored. The function [DISPLAY/LOGFILES/GLOBAL ALARM LOGFILE](#) will show this file, which gives a view over all logged alarms. The projects' names are also listed.

Hide login window automatically:

After a manual or automatic logout the login window is visible. No other action but a login can be performed. To have a full view to the monitoring windows, the login window can disappear after a while automatically. This check box must be set to do so. Window appears again, after the user takes any action.

Startup:

Enable autostart feature

If a project is started for data acquisition MCPS is logging this information into a specific **autostart** file. After a power fail and automatic restart of the software, all projects in this file will be loaded again and started. We recommend to put a link of MCPS into the **AutoStart folder** of Windows. Otherwise MCPS is not restarted automatically.

Wait for dongle driver

WindowsXP often loads drivers with delay. If MCPS is in the AutoStart folder of Windows, it can happen, that MCPS is started, but the dongle driver not yet loaded. If this check box is on, MCPS will wait at least 10 seconds before continuing.

#### Load project list

Projects in the specified list will be loaded at startup of MCPS.

#### Start projects

The loaded projects of the project list will be started for data acquisition.

#### Use multi-project window layout

After the start of all projects, the window layout is shown (See [PROJECT/WINDOW LAYOUT](#)).

#### Start script

A script is started at startup (option /SCRIPT is required).

#### Hardware watchdog

MCPS supports a specific hardware watchdog to reboot the computer in case of software problems. In energized mode the on board relay will be closed for 2 seconds, if the wathdog becomes active. In deenergized mode, the relay is closed with power on of the computer and opened for 2 seconds. This behaviour is useful to see, if the computer itself has power.

## Schedulers

MCPS can handle some global schedulers to perform tasks, which are not related to any project or data acquisition. The task is run, when the time conditions defined under <Cycle> are reached.

The options are type dependent:

- Script: Filename of script
- Print message window: Filename of print layout

## ECS

The External Control Server ECS is used to control MCPS by an external device such a PLC or a programmable display panel. With the MODBUS protocol batch information can be read or set and batches can be started and stopped. Additionally a script can be called to do further actions. In ASCII mode the command handling is completely done by the script.

For Modbus the following registers are used to exchange data:

### Writeregisters:

- 400001: 1=Start / 2 = Stop of projects. 400004-400007 are used for identification
- 400002: 1=Start / 2 = Stop of batch. The slave address of the PLC or panel is used to define the batch unit. Slave address 1 = First batch.
- 400003: If several input masks are available, this register specifies the mask to be used: 0 = first mask
- 400004-400007: 4 registers (8Bytes) for project identification. MCPS searches these bytes in the project name (not path). If not all bytes are used the rest must be set to 0. Typical IDs are 0001, 0002 etc. The related project name can be Test\_0001, Test\_0002 and so on.
- 400008: 1 = Use batch configuration, but do not start the batch
- 400009: 1 = Batch is configure by the panel. This blocks the MCPS batch configuration.
- 400010-400019: Login name
- 400020-400029: User ID
- 400030-400039: Password
- 400040-400044: 5 Wordregister = 80 Bits. For each batch unit the panel wants to control a bit must be set. A write to this registers will start the login test. Before this is done, the next registers must be set.
- 400045-400046: 2 Wordregister = 32 Bits. The panel/PLC has to define, which batch manager operations are requested to control.
  - Bit 0 is set: Batch start
  - Bit 1 is set: Batch stop
  - Bit 2 is set: Batch setupThe requested functions are check in the user administration for the logged in user.

### Write-/Read registers:

The following registers are preset by MCPS and can be modified by the PLC/panel:

- 400101-400120: Batch text
- 400121-400140: Batch
- 400141-400160: Lot number text
- 400161-400180: Lot number
- 400201-400220: Comment for first input field of batch information
- 400221-400240: First input field of batch information
- 400241-400260: Comment 2
- 400261-400280: Input 2
- 400281-400300: Comment 3
- 400301-400320: Input 3
- 400321-400340: Comment 4
- 400341-400360: Input 4
- 400361-400380: Comment 5
- 400381-400400: Input 5
- 400401-400420: Comment 6
- 400421-400440: Input 6

400441-400460: Comment 7  
400461-400480: Input 7  
400481-400500: Comment 8  
400501-400520: Input 8

Status read registers (Bits):

100001: 0 = No batch is running in the unit, 1 = Batch is active

## Graphic

Sets the default settings for the graphic windows:

Background color: Color of the graphic window's background. Depending on this color the project configuration uses two different color schemes for channel color. The first one has dark colors for a lighter background the second vice versa.

X-axis color: Color of the x-axis. The y-axis is drawn with the channel color.

Channel line width: Width of all lines in pixels

Fast display: MCPS can display the offline data much faster by skipping some values. Faster means automatically less values are read. When zooming the display becomes more accurate, because a smaller time range is displayed and less values are skipped.

Event texts: Normally an E is displayed at the x-axis to show, that an event text is available at this point of time. With the check box on, the complete text is shown at the top of the graphic.

Alarm ranges: A small gray bar is drawn below the x-axis. For the time an alarm is active, the bar is colored in red. A double click on a red part opens the project alarm window.

Curve info: When the mouse is moved on a curve, some information can be displayed to identify the channel.

Every graphic window provides a fast display mode. This one can be preset here.

(See [DISPLAY/GRAPHIC](#))

## Project

Online settings:

Project online history:

For graphic windows the measured data is also stored in a memory based ring buffer during data acquisition. This allows fast redraw if a group has been changed or some scaling has been done. If the graphic window is scaled in that way to show a great range of time, it might be, that after a redraw not all data points are redrawn again, because of the limited buffer. The number of scans hold in this buffer can be set up to 100000 scans. In case of many channels, a slow computer or less free memory we recommend to use a lower value.

Data saving buffer for write errors

If the data file can not be accessed, because of network problems or backup program, MCPS can hold the scans in memory. If the file can be written again, before the buffer overflows, the data are appended normally. Otherwise the [alternative saving directory](#) is used to create a child file for further data saving.

Warning, if free disk space is below <n> MB

MCPS checks every 5 minutes (or if sample rate is lower then with every scan), if enough free disk space is available. If the free disk space is below this threshold MCPS places a warning in the message window.

Alarm buffer size

MCPS holds alarms in an internal memory buffer to show active and older inactive alarms of the same channel and level in the alarm online window. If alarms are not removed, after a while the buffer gets full. In case of many projects, fast sample rate and a lot of channels, this memory buffer should not be too great, otherwise the system performance is going down.

Backup alarm conditions

The online alarms are saved with each change to be reloaded again after power fail or MCPS restart.

Automatic gap filling:

TCP/IP gap filling:

After an online communication problem, MCPS will receive the missing data from the internal buffers of the dataloggers after the connection is working again. This feature is supported for MX100/MW100 and DX series. Depending on the size of the gap, the internal buffer might be not large enough to provide all missing data. In this case offline gap filling must be executed. Especially the DX series have only a small buffer for gap filling. See below file gap filling.

Fill gap after restart of MCPS with <no data>

After a power fail or smart quit, data and even time stamps are normally missing in the data file. This behaviour prevents a manual gap filling. By enabling this check box, MCPS will fill up all projects with timestamps and <no data> at the next start of the measurement. According to the last data scan and the project sample rate the missing time stamps are computed and written into the data file. If a project is stopped for a longer time, this can lead to some time of waiting for the next operation or even MCPS shutdown, because a lot of data must be written.

Gap filing with device file

This function will search for device files in the import directory and make an automatic gap filling. After this the original file can be deleted or moved to the backup directory. This feature is best used with FTP file transfer of device files, which are sent regularly to a server system. MCPS checks, if gap filling is needed, imports the data file and fills up existing gaps.

**Important:** The feature is started ONE hour AFTER MCPS has been started. So the user has enough time to start or restart projects after a shutdown or smartquit. Since this function can also be used be manually transferring files into the import directory, you have to wait at least this startup time to see any effect!

Offline settings:

Maximum column width for numerical display

The offline text window is organized in rows and columns with a specific width. In case of larger comments or tags it can be useful to increase the size of column with this selection box.

## New / Configuration

### Show first n columns all the time

The project configuration is normally handled over several horizontal pages. If the second page is selected, only the channel's project number is seen at the beginning of the line. Sometimes this is not enough to identify the channel, especially if several instruments are used. If this value is changed, more columns of the first page are also displayed on the following pages. Device name, channel name, unit and even tag can be set to be permanently available.

### Hide template for empty projects

In case of defined templates a window with all available templates and a basic one are listed, when a new project is to be created. The basic template can be removed from the template list, so a user can only create a new project by using a predefined project. See [PROJECT/EXTRAS/SAVE PROJECT AS TEMPLATE](#).

### Manual text input for channel names

In the project configuration the channel name is normally selected out of a drop down list. With this check box it is possible to enter the name of the channel manually by keyboard. A verification is done, when the project is saved.

### Hide template explorer

Before creating a new project, the user can select a template. By default an explorer icon is available to select templates from any path and not only from the default template directory. The check box can hide the explorer icon

### Mark active line

A blue rectangle is drawn around the active line in the project configuration to see which channel is currently being modified.

### Common settings:

#### Display sort mode

In several windows the channel are sorted by number. It is also possible to sort the channels by the tag text of the channel configuration. This affects for example the group manager, text windows, statistic window and so on.

## Admin

Before you can use the user administration, you have to define an administrator. The admin has all rights. Some functions like user access definition, audittrail or administrator logfile are limited to him. His password must be at least 8 characters long. The general settings are also valid for the users. The administrator can define two additional administrators with all rights, but changing the passwords of the main administrator.

Minimum length of password:

The password of the user (not admin) must have at least the specified number of characters.

Password renewal:

The password can be valid all the time or the user must define a new password after the specified duration. The next time the user wants to login, MCPS requires a password change.

Automatic logout:

After the specified time MCPS will call the logout function, if the user has not made any mouse or keyboard inputs. Configuration windows are closed, so that an other user with lower rights cannot modify open windows.

*Disconnect NV-Server* is used to disconnect from the Netview dongle and to free login accounts. After new login the connection is made again to the HL-Server.

Inadmissible passwords:

To increase security a set of typical passwords can be blocked. They must be separated by semicolons.

Disable user, when login fails:

After three invalid login tries the login window is blocked for 1 minute. After this the user can login again or can be totally disabled. Only the administrator can reactivate the account.

Do not show user names on login

If not active, a dropdown list with all users is available for faster login.

More options

User administration and audittrail for multiple MCPS:

To run several MCPS packages with the same user administration, you have to enter a network directory for each client to share the user file. After the directory is set active, all user and administrator information are loaded from there. If there is not user file found, a new one can be created. All local settings are ineffective. All files created in the user directory should be saved sometimes to a backup directory!

In case of the /ERES option all audittrail information of all clients are logged to this directory.

The administrator logfile is placed in the same directory as the user administration file and is also available on clients.

This user directory can also be handled by a service program on a computer in the network. The connection is made directly by ethernet and not on files. Since in the first case the user must have write access to store password changes or audittrail events, the file access can be completely blocked when using the service. Only the computer name must be defined, where to find the service. A special setup program (ncs\_setup.exe) can be found on the MCPS-CD in the directory <installation>.

Administrator messages

Some events generate an administrator message like invalid logins or corrupt files. These messages can be written to the admin logfile or sent by email. Therefore you have to setup the SMTP server (see [SMS/EMAIL](#)). To see the admin logfile select [DISPLAY/ LOGFILES/ ADMINISTRATOR LOGFILE](#).

Check for same user administration on online clients and data acquisition computers

If active, the measurement computer checks, that the client is using the same user administration (file) before loading a remote online project. This prohibits remote actions (e.g. alarm acknowledge) by a client without correct authentication.

Login verification:

MCPS: Login and passwords are configured completely in MCPS  
Windows server: For all users in MCPS the same logins must be used as for Windows logons. The passwords are not defined in MCPS, they are defined in the global company network. MCPS can verify login and password by calling a company server with **LDAP** support. So the users don't need to keep additional MCPS passwords in mind.  
Nevertheless an administrator must first define a MCPS password. When the CTRL-key is pressed during login (OK button), the MCPS password is verified instead LDAP is used. This is useful, if communication to the LDAP server fails.

## Users

MCPS provides a user password security system to disable certain functions for specific users. For each user a password, id and the allowed functionality can be defined. The user login manager operates basically like the group manager and can organize the users in groups. Users without a password cannot login. Same user names or passwords are not possible. Use the **right mouse** button on the tree items to add groups or users.

If a user creates a new project, his identification is saved with the project. So MCPS can check to which projects the user can have access. The access can be limited to only his own projects or those of users of the same group. If a user or administrator creates a new project and a different user should have access to it, but only has rights for his own projects, the user must be added to the owner list of the new project. See [PROJECT CONFIGURATION/EXTRAS/OWNERS](#).

Account is deactivated:

The user cannot login and is not available in the login dropdown list.

Account expires:

After the specified time the account will be automatically deactivated and can only be enabled by the administrator.

After a user has done his work, he should logout himself (menu: [PROJECT/LOGOUT](#))!

## Directories

- Default directories

For specific file operations MCPS can be told, where to open the file request box. This can even be a network server. You can enter the directory manually or by pressing the button .... This will open a directory selection window.

- External setting software

The device manager allows you to receive and send setups from and to many instruments. For some devices MCPS supports own channel configuration. For Yokogawa DARWIN and DX/MV series the software can call the original setup software from the manufacturer. MCPS needs to know, where this software is installed.

To make it very easy for the customer we have added the installation program Ysetup.exe to the MCPS CD, which installs those external setup programs. This should be done after the installation of MCPS, so that the setup programs are automatically installed in the MCPS directory and all links are made by MCPS. If you install it at a different location, you have to set the pathes manually.

- Alternative directory for data saving

If MCPS is not able to save the measured data, because of a hard disk failure or network problem, this alternative path is used to stored the information. MCPS will create a new project with the same settings at the new location. Afterwards all measured data is saved into the new project. In the MCPS environment the new project gets a leading ! in its name.

Later on the new and the old project can be [Combined](#).

## Messages

### Alarm messages:

Open alarm window automatically:

In case of a new alarm, MCPS will open the alarm online window automatically. This can also be activated on a client, which is connected to a measurement computer by the [MSERV](#) interface to see remote alarms.

Disable sceen saver in case of alarms:

If an alarm message is present, the screen blanker is stopped and disabled.

Log event with alarm acknowledge:

The event log window is opened, when an alarm is acknowlegded. This feature is also available on client computers, which are connected by the MServ interface.

Alarm sound:

Several sounds can be played with alarm occurence. In case of playing sound until alarm acknowledge, the sound stops, when the alarm is acknowledge in the [alarm online window](#).

### System and device messages:

Open message window automatically:

In case of any message in the online message list, MCPS will open the message window automatically.

Maximum of online messages:

Defines how many online messages (device error, autostart) are stored and hold in the message file. Older messages are removed.

Create device error message after n transmission errors:

In some communication environments (especially networks) the instruments often produce communication errors for a short time. To create not every time a message and a related action, the errors can be collected before something happens.

- Message actions

The messages are grouped and for each group an output can be set, an email sent or a script started. The output can be any output provided by an instrument in the device manager. If a script is called the message can be taken with <GetScriptData>.

**Device OK:** Successful reconnection to device

**Device error:** No answer, checksum error, incomplete data or device specific error message.

**Autostart:** Restart of MCPS

**File (load/write) error:** File cannot be read or written, disk full

**System error:** Low on memory, NCS-error, email error usw.

**Skript error:** Error during script execution

- Output pulse length

Defines how long the output should be active (in seconds).

- Common output for device errors

This output is set, if at least one device has communication problems. When all devices have the status OK, the output is reset.

## Network

### Local project configuration

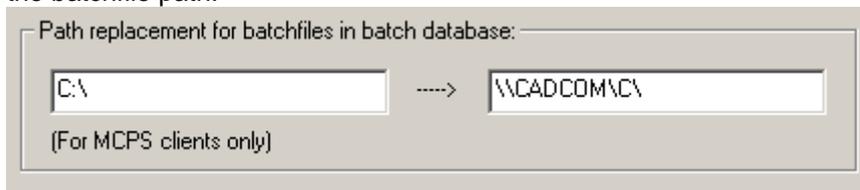
Enables / disables the **local project configuration** for network applications. The user can specify a path to his local data.

### String replacements in path settings

In network sessions the MCPS client versions (e.g. NETVIEW) often see the original measured data on another drive than the MCPS data acquisition package. For example if the project is saved to F:\MCPS the client package find it on X:\MCPS. If the project contains a math file (e.g. F:\MCPS\MATHFILES\POLY.POL), it cannot be evaluated by the client package, because the drive F: is maybe unkown. To avoid this conflict the drive names can be replaced, e.g. from F: to X:. Even a path can be replaced (F:\data ---> X:)

### Path replacement for batchfiles in batch database.

If the batch database is loaded on a MCPS client computer, the pathes of the batchfiles as saved in the database are different than seen from the client side. Only if the measurement computer saves the batchfiles on a server, which is accessable in the same way from the client (exactly the same drive and path), then there is no problem, otherwise the client MCPS must change the drive and directory of the batchfile path.



The image shows a dialog box titled "Path replacement for batchfiles in batch database:". It contains two text input fields. The first field contains "C:\", and the second field contains "\\CADCOM\C\". A double-headed arrow points from the first field to the second. Below the fields, the text "(For MCPS clients only)" is displayed.

The example will exchange the drive in the original path to the drive and path seen from the client.

## MServ

(Only with option /MSERV)

One of the most interesting and powerful features of MCPS is the capability to distribute measured data and alarms over the network to listening clients. If a MCPS measurement version (with device drivers) contains the option /MSERV, other MCPS packages can be connected to it and can display its online data and alarms. A client can be connected to five measurement PCs.

To establish a connection, the remote computer's name or ip address must be filled in.

If the alarm window is opened or the menu item **PROJECT/OPEN FROM DA SERVER**, MCPS tries to get data from the remote computer. If the connection is established, MCPS will also receive a number of history data from the online project.

Number of data scans for trend history:

Defines how many historical data is transmitted to the client after opening a remote online project. If not necessary, this number should not be too high to reduce network traffic or block slow client machines. This is also influenced by the online buffer size of the project on the remote computer. Only data which is stored in memory can be transmitted.

Fastest update time:

To reduce network traffic the data update can be made in greater time steps. Therefore the update time must be defined in seconds. A zero means update time equals sample time of project.

Connection method:

Until MCPS 6.2 the client – server communication was done by the DCOM interface of Windows. Since XP SP2 a lot of security settings must be done to enable DCOM communication. Therefore since MCPS 6.3 the data transmission can also be done by a specific TCP/IP communication. In this case only the firewall must be setup for MCPS.

Search pathes to original projects:

The projects, which are loaded by the client from the data acquisition computer, are transmitted without any file access directly over the network. An internal (DCOM-) communication channel is used. Therefore normally only the online data can be made visible, since the client itself does not know, where the data files are located, because the pathes on the measurement PC are in general different to the pathes the client sees.

By entering the pathes, where to find the data files, MCPS can link these information to the remote projects and the user can see online and offline data, without loading the offline projects manually.

MCPS packages without any device driver (View version) have to activate the button *Enable MSERV-Client features* in register tab **PASSWORDS**.

## **SMS/Email**

(Only with option /SMS-E)

With this feature MCPS can send SMS messages to handies or email clients, if an alarm is detected. In this window you have to configure basic settings, which are valid for all projects.

- Enable SMS messages

This turns the SMS feature basically on/off.

a) SMS via service center

- Telephone number of SMS service center

MCPS should be connected to an analog modem or ISDN line. The SMS message must be sent to a provider, who will transmit the message to the handy. The provider's telephone number has to be filled in here.

- Protocol type:

The providers use different protocols. The UCP protocol can tell MCPS, if the message could be delivered or not. In case of non delivery it's possible to call other handy numbers.

- Connection device

A windows based driver to make the connection to the provider; analog modem or ISDN.

- Retries:

Number of retries, if the provider couldn't be connected.

- Seconds to wait

MCPS will wait the amount of time before calling again.

b) Connection via GSM modem

A GSM modem with a SIM card is connected to the computer by comport or Ethernet.

- Own hostname

The domain to which the system belongs. Rarely used.

- Enable Email messages

This turns the email feature basically on/off.

- Sender name or address

Name or account of the sender. Some email servers are setup to send emails only from a valid account.

- SMTP server address and port

The (Simple Mail Transfer Protocol) server must be available in the network.

- Authentication

If login and password are filled in, MCPS will send these information to the server to authenticate the sender.

Emails can also be sent, if a **system** or **administrator message** has been created.

See also PROJECT/ALARM SETTINGS.

## **Batch**

A batch session file can be loaded and run at startup of MCPS. See BATCH MANAGER for more information. The related database can be loaded with the session whenever a session is opened.

Independent from a session a database can be loaded at startup of MCPS. This is useful for clients without any session.

A double click on the batch in the database can result in the following actions:

The batch is loaded. The project bar is visible at the bottom of the main window

The the window layout, which is saved in the batch and was defined in the batch mask, is opened

A specific common window layout is opened. See PROJECT/WINDOW MANAGER

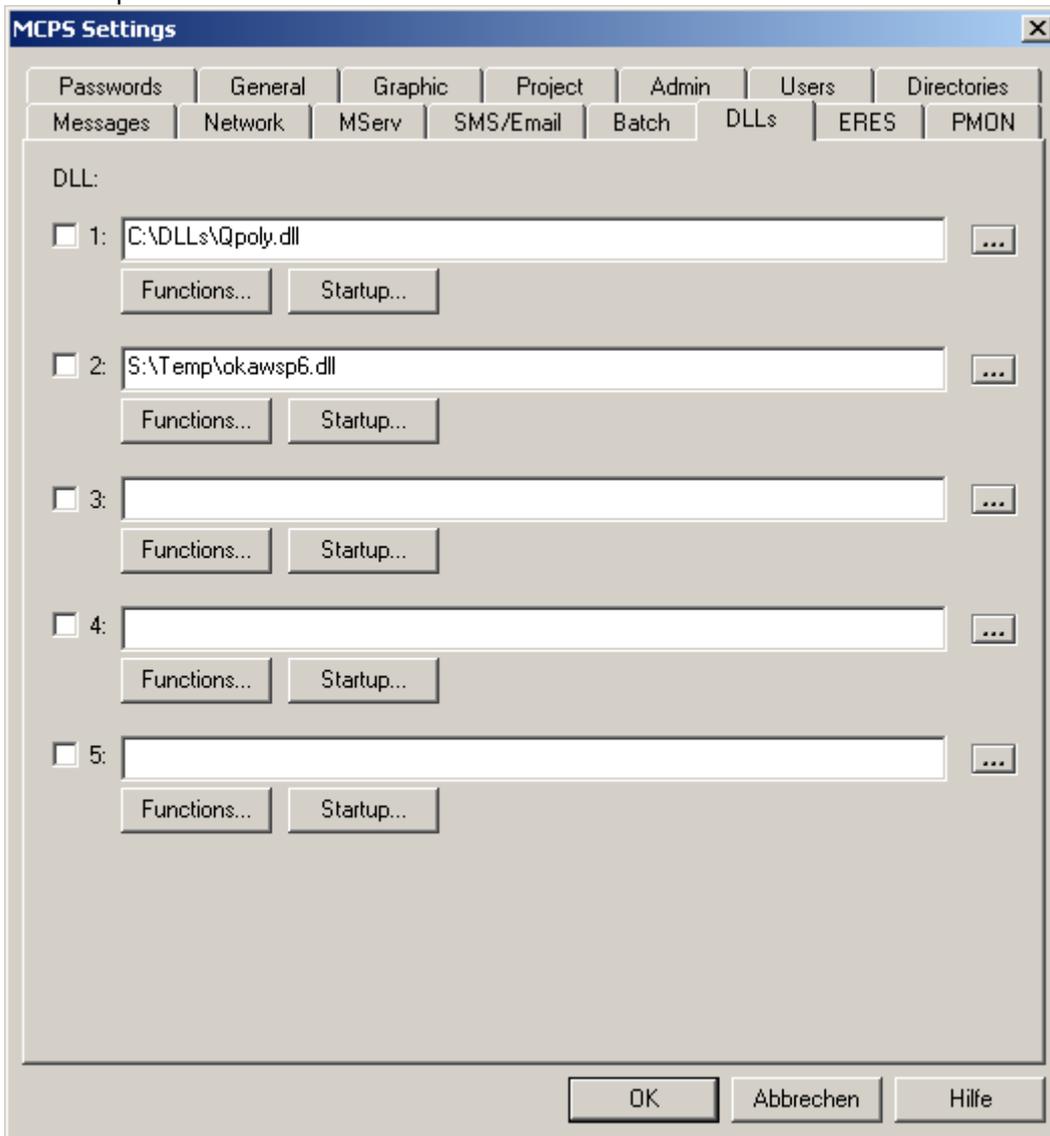
A common window layout can be selected.

A batch project can be closed automatically, if the last of its windows (numerical, trend, ..) is closed.

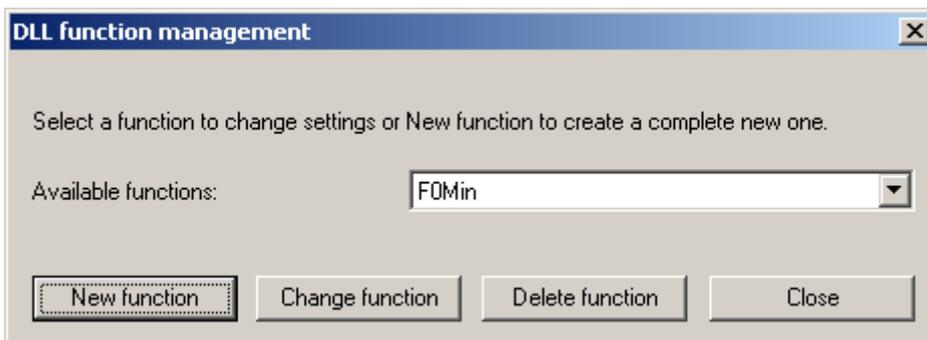
## DLL

(Only with option /MATH2)

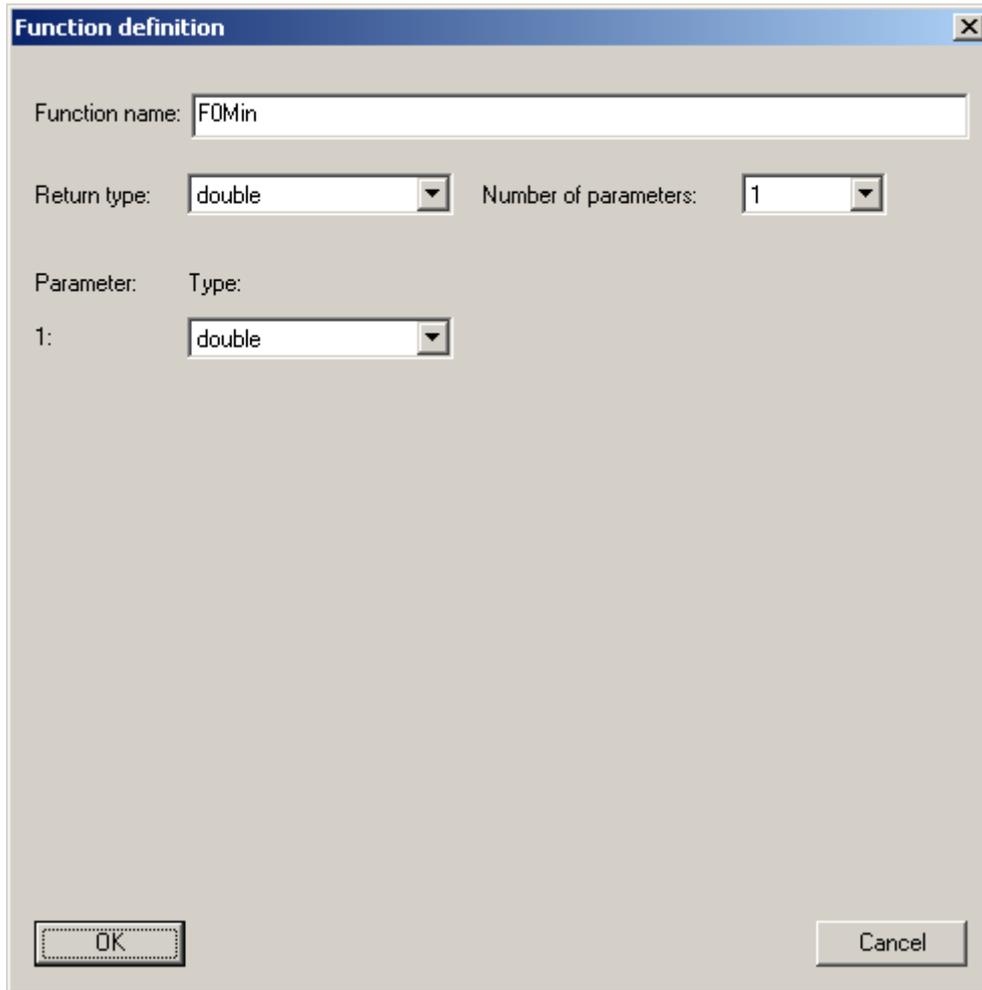
This windows allows you to integrate external DLLs into MCPS for usage in math formulas. Since a DLL gives no information about parameters and their types, all these information must be defined in MCPS. Up to 5 different DLLs can be included.



With an active checkbox MCPS will load the specified DLL at startup and tries to find the functions, which are defined by the function button:



In this window an already existing function can be modified or deleted or a new function (first step) must be created.



The image shows a dialog box titled "Function definition" with a close button (X) in the top right corner. The dialog contains the following fields:

- Function name: F0Min
- Return type: double
- Number of parameters: 1
- Parameter: 1, Type: double

At the bottom of the dialog, there are two buttons: "OK" and "Cancel".

Here the function name as implemented in the DLL must be specified. As well as the return type of the function, the number of parameters and their types.

In the math formula you can call the function by the following syntax:  
DLL(Number of DLL, "function name", parameter 1, parameter 2, ... , parameter 15)

Example:

X1 + DLL(1,"SUM",X2,X3,X4)

The function name must be in quotation marks. If the function is called with text parameters, they also must be enclosed in quotation marks: X1 - DLL(2,"GetCalibrationOffset","ID-2534","SN-1234")

Sometimes it is necessary to run a function at startup to initialize the DLL or related data fields. This can be done with the STARTUP button. Up to three startup functions with default parameters can be called. These functions must be defined with the FUNCTION button first.

## **PMON**

(Only with option /PMON)

Settings of the process monitoring window. It can be docked to each side and is not overlapped by other windows. The size is with or height depending on docking position. The window can be [hidden](#) or its size can be decreased by dragging the margin. See also [PROJECT/EXTRAS/EDITOR FOR PROCESS MONITORING](#)

Process window settings:

Full screen mode: The process monitor uses the complete MCPS screen. If the project configuration or other windows are opened, the PMON is hidden completely until the window is closed again. This mode is recommended for applications, where the users should have less influence on the system.

Maximum window size: Size of process window in pixel.

Docking position: In case of Top or Bottom the process window uses the entire width of MCPS, in case of Left and Right the entire height is used.

Display PMON: START: The PMON window can be displayed (opened), when the measurement is started and disappears, after project is stopped.

OPEN: The PMON window appears immediately after loading a project with PMON items. This is necessary, if the PMON window contains items (buttons) to start and stop the data acquisition.

Project handling:

All projects together: All process window items of all running projects are put into the process window together. This mode is useful, if the window is divided into parts for the projects, which are configured to use only a specific part of the window. E.g project 1 uses the left part, project 2 the middle part and project 3 the right part of the process window. Different pages of a project are not supported.

Projects selectable by tabs: On the top of the process window all running projects with PMON items are listed in tabs. Only one project at the same time is displayed in the window. On the bottom all pages of that project are listed, if at least 2 pages are defined.

Editor:

Grid resolution: Snap grid resolution to position items easier.

## **Webserver**

(Only with option /WEB)

This page defines basic settings of the integrated [WebServer](#). Some changes may need a restart of MCPS to take effect, e.g. port settings.

*Root directory* is the path the location, where all pages, subpages and images are located.

*First page* is loaded from the root directory, when a browser starts a new session.

*Refresh dynamic items* defines the time interval to update images and texts. With this value you can reduce network traffic.

*Display login screen* opens a login window the first time a browser connects to MCPS. The user administration of MCPS is used.

## **ERES**

(Only with option /ERES)

This page defines some special settings for ERES applications

- Force comment four audittrail after sending device setup:

After a setup is sent to an instrument, a window opens to enter a text, which is added as a comment to the audittrail.

- User can sign only one level

Normally a user can sign all levels of a batch, if he has the related access rights. Sometimes it is necessary, that a user can sign only one level, but it can be one of all three ones. If it should be a specific level then the access rights should be used.

### **- Default import only with same serial number**

Device serial number of default project and import project must be equal. This avoids, that project default settings of one device are used for a different device.

### **- Run only encoded scripts**

Scripts are usually regular text files. CAD Computer can encode these files to avoid manipulations. Since MCPS can read both types of files this checkbox forces MCPS to read only encoded script files.

## Interface configuration

The serial communication interface should be set identically to the parameters of the device. This is necessary for RS232 and RS422A interfaces. The following parameters are often the reason for data transmission problems. See also [RS232](#).

Baudrate:	Transmission speed
Parity:	Error check bit
Data bits:	Number of data bits
Stop bits:	Number of stop bits after data bits
Handshake:	Communication synchronization
	None (default for MCPS)
	Software (with Xon/Xoff character)
	Hardware (Cable with DTR/CTS lines)

## Device configuration



All devices for data acquisition are handled by the device manager of MCPS. A project can only control instruments, which are listed in the device manager and set to active. A double mouse click on an already installed instrument opens its configuration window to set interface type, channels or options. Several instruments of the same type are distinguished by a unique number called device specifier, which appears after the device name (e.g. HR1300 – 2).



Large icons

Device symbol with device name and specifier



Details

List view with additional information. By pressing the header button <device> the instruments are sorted alphabetically. Devices, which are currently used in a data acquisition are marked with a green LED. In case of communication problems the LED is red. A blue one is used for suspended devices.



Add device

Opens a window to select an instrument to be added to the device manager. Only those devices are listed, which are accessible by the users password and the dongle. If a device is missing please ask your dealer for the correct device driver. If you want to add a device, which is not listed anyway, you have to turn on the switch Show all instruments, when adding a device under [Configuration/MCPS Settings/Passwords](#).

After the device is added the configuration window appears. This is instrument dependent, but the following buttons are common:

a) Active

Only active devices are listed in the project configuration editor and can be used for data acquisition. Because several devices can not use the same interface (or same address) it's sometimes useful to set an instrument to inactive. This will skip the interface and device specifier check.

b) Specifier

Often several instruments of the same device type should be used. MCPS needs this unique number to exactly identify each device. In many windows and the project configuration devices are listed as DEVICE – SPECIFIER (e.g. DA100 – 1, DA100 – 2, HR2300 –5, VR200 – 1, VR200 – 2, VR200 – 3,...).

c) Interface

Each device should be connected to an interface (RS232, GPIB, RS422A or ETHERNET) and if necessary a device specific address. This button opens a dialog box to set the interface parameters. See also [Interface selection](#) .

d) Comment

Description of the device for better identification

#### e) Suspend

If this button is available, a device can be suspended. This can be done before or even during a data acquisition. MCPS will stop communication to this instrument and display and save <invalid data>. This is useful, if a device is not connected, turned off or should be used for calibration.

Before the start of a measurement MCPS checks the communication to all devices of the project. If a device is not responding, the user can now instruct MCPS to automatically suspend this device. The data acquisition can start even one or several devices are not connected. During the measurement a device can be unsuspended again. MCPS will check the communication and if necessary download the device information. If everything is OK, data are shown and saved normally.

#### f) Test device

If available, this button starts a simple connection test. If the test fails, please check the interface parameters like interface type, address, baud rate, parity, ... Otherwise, if the test is OK, you can not be sure, that the communication and data transfer will work correctly in any case. This depends also on the parameters of the device configuration for example the correct number of channels, subunits, module types and so on.



#### Remove device

The highlighted device is removed. This can not be done during an active measurement.



#### Copy device

This button can make a copy of a device and its settings. Not all instruments are supported.



#### Receive setup

For some instruments MCPS supports a setup download. The parameter settings are transferred to the PC and can be saved into a file. For each device type MCPS will add its own file extension (e.g. .d10 for DA100).



#### Send setup

This function uploads a setup file back to the instrument.



#### Edit setup

For some devices it's possible to edit the setup files. Some parameters like range settings and scalings can be changed to setup the instrument with MCPS. This is very important for data loggers, which have no user interface.

The Yokogawa DARWIN and DX/MV series come with an own powerful setup program. MCPS can call this setup programs directly. Sending and receiving is done by MCPS.

On the installation CD there is the program Ysetup.exe. This will install the external setup programs in the MCPS directory.

See Configuration/MCPS Settings/Directories for more information.



Lock setup file to device. Only one specific file can be used for download or uploading a setup.



#### Test device

Basic device communication test



#### Send command

Send direct commands to device. For Darwin and DX series MCPS can start/stop measurement or computation or acknowledge alarm at the recorder.

# Extras

## Print Layout Manager

Normally text and graphic data is printed using the entire sheet of paper. MCPS includes a layout manager, that can define printout masks (layouts) to add text comments and to resize the graphic area. A layout is defined before a printout is made. It is saved to disk and can be loaded, if the graphic should be sent to the printer.

Commonly each layout contains a frame, which can be placed on the layout sheet. The size and position is user defined. This frame will contain the data (graphic or text) after the printout. The rest of the sheet can be filled with static texts or request field. The last one will ask the user for a comment, when the layout is loaded. This features allows individual printouts for example to enter a serial number each time. A layout item will be marked with a mouse click. This allows further operations.

Some printout functions of MCPS can produce several or a lot of sheets. In this case it's useful to use the 2 page mode of the layouter. A second page can be defined for example with less standard information and a greater output frame. MCPS will use the first layout page for the first printout page. All following printout pages will use the second page of the layout.

The following buttons are available:



**New layout**

A new empty layout is created. The user has to select the paper size and page orientation. During printout the printer is set to the layout's orientation.



**Load layout**

Loads an already existing layout.



**Save layout**

Saves the current layout in the same file.



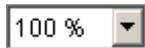
**Save as**

Saves the current layout in a new file.



**Select page 1/2**

Selects the current page. The second page can be empty. If the second page contains any item, it must also contain an output frame.



**Zoom**

Changes the zoom factor of the sheet



**Layout settings**

It's possible to define a grid, to which all items are automatically positioned. Paper size and page orientation can be changed.



**Select item**

The standard mouse pointer to select items already placed on the layout.



**Output frame**

After clicking on this button it's possible to define the output frame. In this frame MCPS will place the graphic or other data (e.g. statistic values). Only one output frame per page can be defined. In addition a font can be set for text outputs like numeric, alarm or configuration printouts. A rectangle can be drawn around the output frame.



**Text field**

With this button you can define a static text field. This field can contain printer variables. A double click on a text field will edit the text.



#### Input field

This field will generate a request, when the layout is loaded before a printout session. The text of the request field is displayed in the request dialog window and an empty string field is available to enter a text. This text will be placed at the position of the request field. If the operator should define a serial number before a printout, the layout can contain a request field with the text S/N?. Before the printout a dialog window will appear with the text S/N? and an empty text field. The operator enters the serial number, which is printed at the position of the request field.



#### Graphic field (bitmap)

This button will import a color bitmap (\*.bmp file) or JPEG file (\*.jpeg). The property button can be used to resize the graphic. The bitmap is embedded into the layout.



#### Legend

Defines an area, where a legend is placed for graphic output. The property button allows some additional settings.



#### Properties

Used to change the properties of existing items (text, graphic size).



Font button

Changes font and color of text and request fields



Copy

Makes a copies of an item, which is placed below the original one.



Z-Order

Changes the Z-Order of an item. Moves it into the front or into the background.



Remove

Removes the marked item.

For printouts MCPS can convert specific control codes. This codes are often defined in the layout manager and converted during the print process. The following key words can be entered as text strings and will be replaced:

\$PN	Project name
\$RN	AutoFile record
\$CO	Company name
\$GR	Group
\$OP	Operator
\$DATE	Date of printout
\$TIME	Time of printout
\$POS	Start time of printout data
\$POE	End time of printout data
\$PI(1)	1. Line of comment rows in <a href="#">Project information..</a>
\$PI(10)	10. Line
\$PX(1)	1. Line of additional information, which is sometimes set during import or by scripting and also visible, when maximizing the base window of a project.
\$PV(xxx)	Value of the project variable xxx.
\$PS	Start time of project data (1. data scan)
\$PE	End time of project data (last data scan)
\$PD	Duration of measurement (End time – Start time)
\$UCn(chn)	Contents of user column <b>n</b> of channel <b>chn</b> in the project configuration
\$LUN	Current login user name
\$LUI	Current login user id
\$LUC	Comment of current login user

After a statistical computation the following variables are set and can be used for the printout:

\$SnMIN	Minimum value of channel n
\$SnMAX	Maximum value of channel n
\$SnMEAN	Mean value of channel n
\$SnSTD	Standard deviation
\$SnMKT	MKT

Example:

\$S12MAX: Maximum value of channel 12

Formating statistic data:

\$SnFxMIN-> F = Format, n = 0-6 decimal places or A for using the channel format of the project configuration (E.g. \$S25F2MAX: Use 2 decimal places for maximum value of channel 25)

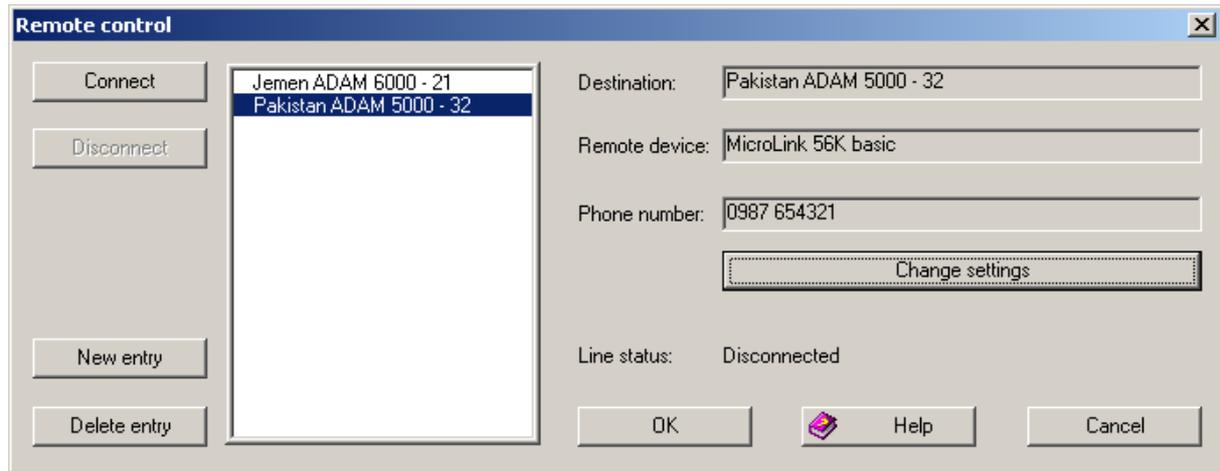
\$SSD	Start date of statistic run
\$SED	End date of statistic run
\$SST	Start time of statistic run
\$SET	End time of statistic run

\$SBC            Number of evaluated scans

If a batchfile is loaded, the following commands are available:

\$BA            batch (base name as shampoo)  
\$BN            batch name (full name as shampoo-001-020602)  
\$BL            lot number  
\$BP            related project  
\$BS            batch start time  
\$BE            batch end time  
\$BD            batch duration  
\$BU            batch unit  
\$BI(n)        additional input info **n** of the batch mask (e.g. serial number)  
\$BV(n)        related value (input by user)  
\$BXSn        level n(1-3): signed / not signed  
\$BXTn        level n: time of signing  
\$BXCn        level n: comment for signing  
\$BXUn        level n: user, who has signed  
\$BXIn        level n: user-id  
\$BXDn        level n: description

## Remote Connection (Only with option /REMOTE)



MCPS can have access to devices by using a modem connection. Therefore you have first to install a modem in the Windows system configuration (control panel/modems). In the remote connection dialog window of MCPS you can enter several destination addresses, which are handled in a list like a phone book. For each entry you have to define a destination name, a related modem or other connection device and the telephone number. The destination, which is currently selected in the list, can be called by pressing the CONNECT button. After the line is established, you can leave this window.

To have access to the device at the other end of the line, you have to add it into device manager. The interface type of this instrument should be set to REMOTE CONNECTION and the same modem must be selected, which is used for the connection (see also [interface selection](#)).

Now you can use the device in the same way as usual (Online, Setups, Download,...).

After all operations you have to disconnect the line. Therefore you have to call this menu item again, select the connected destination (=>) and press DISCONNECT.

The remote connection is only working, if a modem at the other end of the line is working automatically. This modem is connected to the instrument and gets no setting information. The modem must be programmed in that way, that all settings are right, if the modem is powered on.

This can be done with the hyperterminal program of Windows:

Command to modem:	Comment
ATS0=1	Modem catches the line with the first ring
AT&B1	Constant baud rate
AT&D0	Ignore DTR line
ATE0	Echo off
ATQ1	No reply
AT&W0	Save settings in profil 0 (loaded at power on)

## Run Script

(Only with option /SCRIPT)

This function can run a VisualBasic-Script. Some of the MCPS features are available to those scripts, so that they can control the complete software package. It's used for customer specific applications. See [CONCEPT OF SCRIPTS](#).

# Window

## Cascade

All windows will be overlapped with a small offset to each other.

## Tile

All window will be displayed in a tiled format.

## Close All Windows

Closes all windows instead of the project base windows.

## User toolbar

The user toolbar contains buttons for menu or script functions, which can be added or removed by the user.



### **Activate toolbar**

The toolbar is not visible by default. This menu item activates the toolbar at the left side of the mainframe window.

### **Customize toolbar**

Opens the toolbar's configuration window. In the left part all unused icons are listed and on the right side all visible icons. By drag and drop you can move the icons. The separator can be added several times.

### **Define toolbar scripts**

There are 10 icons for scripts. This window defines the relationship between an icon and the related script. If the script button on the toolbar is pressed, MCPS will start the script. The default icons can be replaced by small user bitmaps with a size of 16x16 pixels.

## Restore Main Toolbar

The main toolbar can be moved and closed with the mouse. This menu item will restore the toolbar.

## Lock mainframe

Locks the main frame to prohibit resizing. In addition several Windows keys are locked.

## Hide process window

Hides the process window to have more space for other windows. This status is hold even MCPS is quit.

## Print process monitor

Prints the process monitor window

## Open working directory

Opens the Windows-Explorer with the MCPS working directory. By default scripts, web server, setups and so on are placed in the working directory, which is not always visible on all Windows systems.

# ? Help

## Info About MCPS

### - Info about MCPS

Shows version number, build type and installed drivers and options.

### - Help topics

Starts the online help

### - Support

Contact and email address

### - Save logging files

In case of support CAD Computer may need some logging and information files. This function will pack all files together. An Explorer window is opened to save the resulting file to a location, from where you can send it easily by email.

### - Run MSInfo32.exe

Windows has a powerful program to collect system information and error messages. The resulting report is also packed by <Save logging files>. Since this program can take several minutes, you have to run it manually. Normally it's enough to start it once or after system has been changed (hardware, operation system,...)

# Devices

## Device information

This chapter describes several instruments in a more detailed way. This is necessary, because some devices need a specific configuration to communicate with MCPS. Please read the related pages carefully to avoid problems. For image devices like infrared cameras or matrix instruments (DV2D) please see also the chapter [<Image devices>](#).

Some devices can be suspended before or during the measurement. From a suspended device MCPS is not taking any data, there is at least no communication to that device. MCPS saves the data status <Invalid data> for all channels of such an instrument. A suspended device can be resumed by pressing the related button in the device configuration of that instrument. If the device was suspended before the start of the measurement, MCPS will test the connection and receive the channel settings, which is necessary to get the data. In addition a measurement can be started without a connection to the device. For those instruments, which support the suspend mode, MCPS will set this check box and start the data acquisition.

### Sampling:

Normally all devices are triggered by the project that means according to the project's sample rate. Some drivers support project independent sampling to handle slower devices or sampling outside a measurement. Those drivers have the button **Sampling settings** with the following options:

Triggered by project:	Standard case
Autorun:	Driver operates as fast as possible. Useful for devices slower than project sample rate
Fix sample rate:	Independent sample rate
Run device with project:	Device is started and stop with project measurement
Run device from start of MCPS:	Sampling starts immediately

# A6000

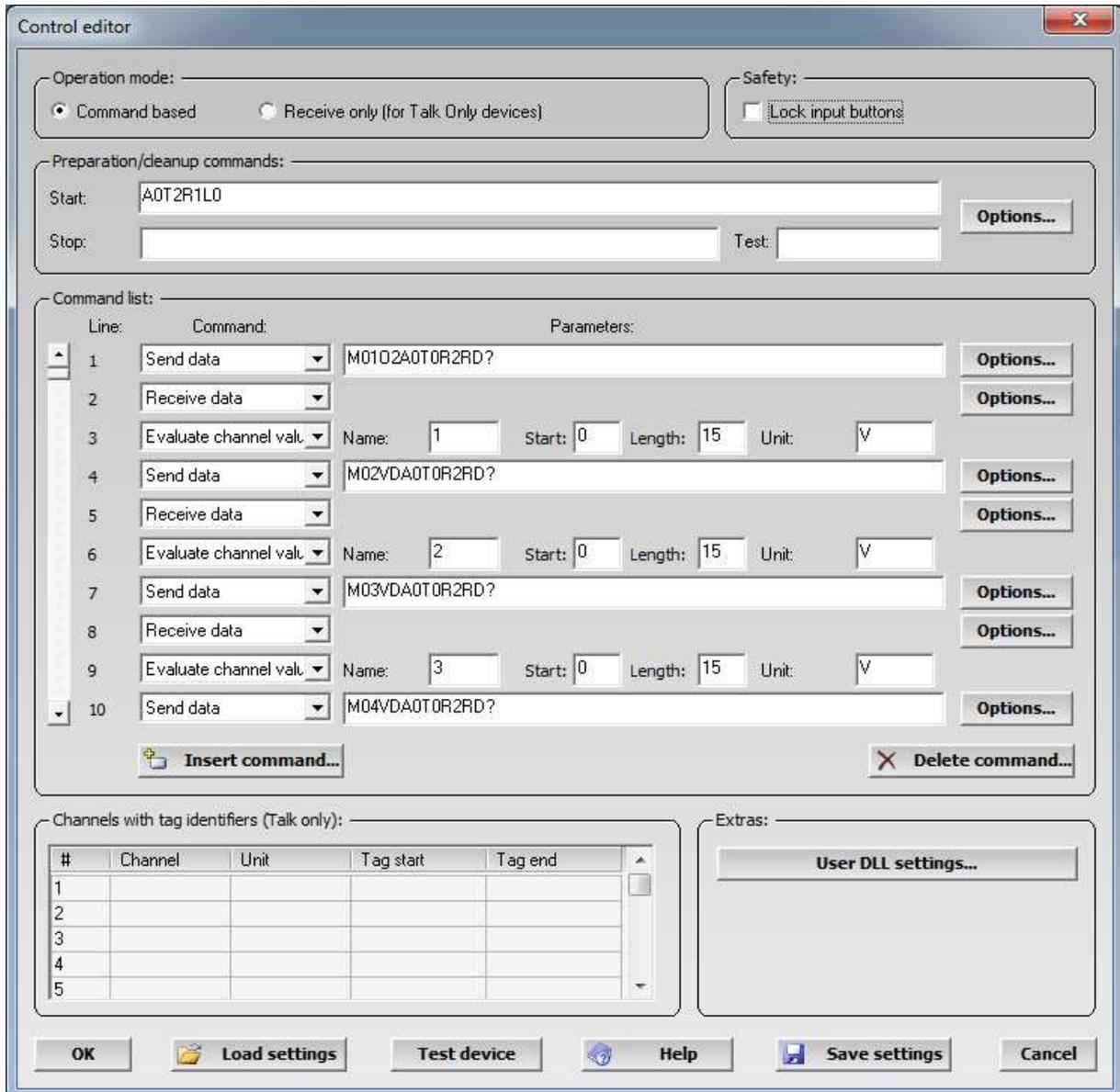
## **VERY IMPORTANT:**

Some firmware versions use the HOST IDLE timeout to automatically disconnect. This value must be set to 300 seconds, especially if the Adam is only used for alarm output.

The Adam6050/51/60 contain digital inputs, which can be used to start or stop a project or batch.

# ALLDRV

The ALLDRV driver is designed to access instruments with ASCII-Protocol. Several commands can be used to communicate with the device and to evaluate the data values.



## Commands:

### No operation

No action is performed. Use this function to have some space for later extensions.

### Send data

The string is sent to the instrument. The option button opens a dialog, which specifies the end of line character(s). Extraordinary characters in the command string can be set with [nnn], where nnn is the ASCII-code. E.g. [27] is used for the Escape-character. Hex data can be sent by using the following format: #842DF211#. Only pairs of characters are allowed.

### Receive data

Data is received until the end of line character is received. The EOL character can be specified with the SET button. The received data is stored in the driver's buffer for further evaluation. This buffer is valid, until a new RECEIVE DATA command is executed.

If the device isn't answering in the specified timeout time, the driver will report an error.

### Evaluate channel value

After a line has been received from the instruments, the channel data can be extracted. For each channel 4 parameters must be defined:

- a) Name            User name of the channel (1,2,3; U1, V1, P1, ...)
- b) Start           Start position of data in the string
- c) Length        Length of data part
- d) Unit            User definable unit

If a received string contains several values, you can call this function several times.

### Wait

Waits the specified time in seconds before processing the next command.

### Read field

Reads several lines and related channels in one command assuming that all lines have the same format. *Start chn* and *End chn* define the number of channels to read and are automatically the channel names in MCPS. If *Start chn* is 1 and *End chn* is 20 the driver will receive 20 lines from the device, evaluate each line and add 20 new channels to its channel list.

This command combines several calls of *Receive data* and *Evaluate channel value*. Thus the parameters *Start* and *Length* and the *Options* have the same meaning as for *Evaluate channel value*.

If a received string contains several values, you can call this function several times.

### **Channels with tag identifiers:**

Some devices just send their data without any request. Such so called talk only devices can be supported by the AllDrv, if the data values are sent with start and end identifiers.

### User DLL:

MCPS can call functions in a DLL, which is to be developed by the customer. Following a specific interface, it is possible to send and receive binary data, which is then evaluated by the DLL. For more information please contact CAD Computer.

### **Important:**

All data, which is sent by the instrument must be received by the ALLDRV driver. Otherwise unhandled data is stored in the Windows interface input buffer and will be transferred to MCPS the next time. So this might be a problem, if you see invalid or extremely changing data.

# Darwin Series

The DARWIN instruments are very powerful and flexible. A lot of module combinations and configurations are possible, so it's very important to reflect the exact DARWIN configuration in the MCPS device manager.

The DARWIN device configuration:

The DARWIN series provide two basic systems, standalone and expandable with up to five subunits. Each subunit contains an address switch, which determines the channel numbers. An address of 3 will produce channel numbers from 301 up to 360 for this subunit. In many cases wrong address switches cause unexpected errors. The channel numbers also depend on the slot, where a module is plugged in.

For each DARWIN several options are possible. Very important is the MATH option, because it will provide additional channels. If these channels are not necessary, the option can be turned off to speed up data transmission.

For standalone systems the button including the text **Module configuration** will open a window with several slots, where the modules can be configured. For expandable instruments the **Subunit** buttons should be pressed for module configuration.

If the DARWIN device is connected to your computer it is easier to use the **AUTODETECT** button. The configuration is received from the instruments and the device manager is updated.

If the configuration of the DARWIN has been changed (new module or interface) it's necessary to reconstruct the device. This should also be done, if you are not completely sure, if the last reconstruction has been made correctly. MCPS can start a reconstruction with the button **RECONSTRUCT**.

Ethernet module:

If you are using the ethernet communication module, you have to **set the interface** to ETHERNET and enter an IP address for the module. If this address is unknown, leave the address untouched. Press OK to close the window. The button **Program module ip address...** appears. It will open a window to read out or write the ip address of the ethernet module. For this operation it's necessary to set the first dip switch of the module to **on** to enter the configuration mode.

*GET ADDRESS* will receive the current ip address.

*SET ADDRESS* will program the addresses set in the window.

*USE ADDRESS* will transfer the address from this window to the device manager.

In most cases the ip address of your computer and of the ethernet module must be equal for the first three number. The best way is to program the module to the computers ip address. The last number **must be different** from any number used in your network!

External alarms:

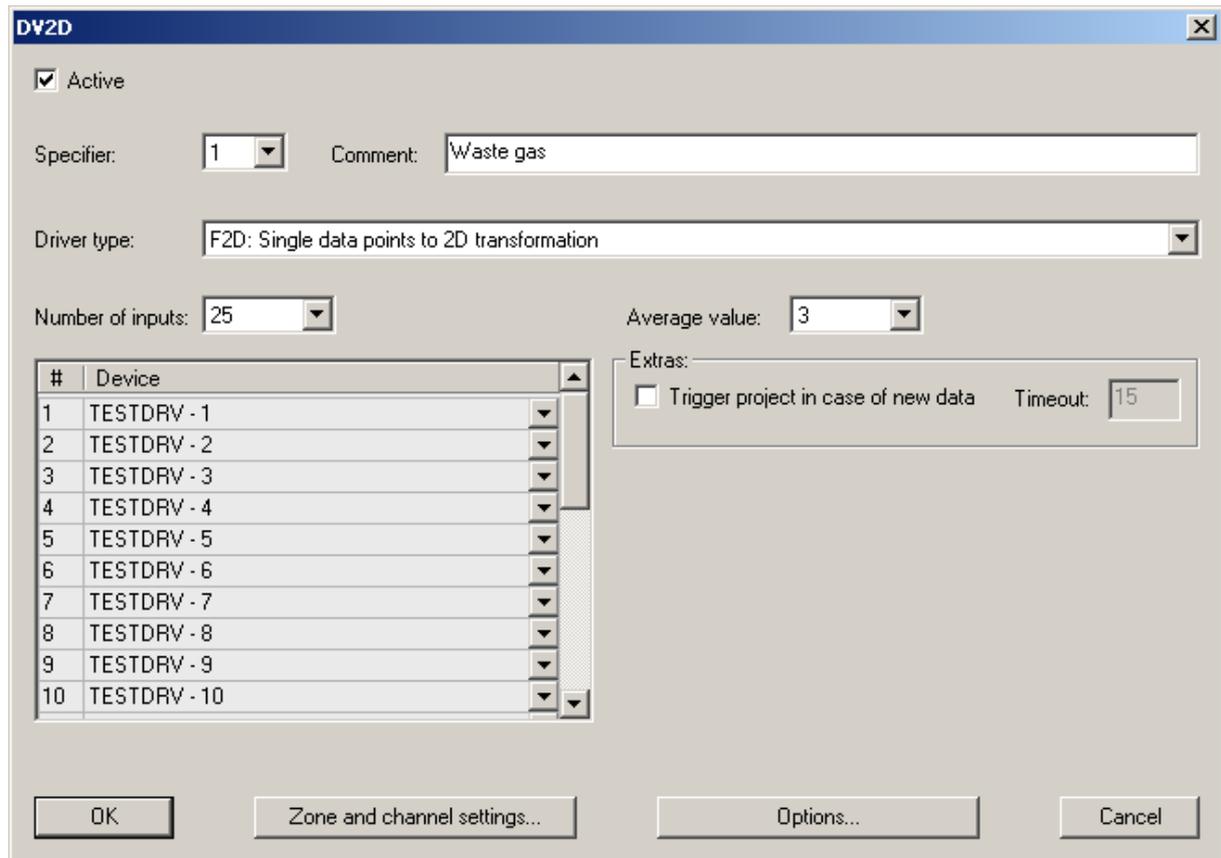
If external alarms should be detected with the alarm option, it's necessary to set the button **TRANSMIT ALARMS**. Otherwise alarms of the darwin device won't be recognized.

Yokogawa provides a powerful setup program for these instruments. It can be called directly from MCPS, if the installation path of the software is known. This can be done in **Configuration/MCPS settings/Directories**.

This setup program is also available on the MCPS installation CD (Ysetup.exe). The best way is to install MCPS first and then install the setup program in the default path (MCPS subdirectory).

After that you can download a setup from the instrument in the device manager. By pressing the button Edit setup the external Yokogawa setup program is called. Make your modifications, **save** the setup (NOT Save as) and leave the setup program. Now it can be sent back to the device by MCPS.

# DV2D



The DV2D is a multifunctional driver to display 2D thermographical data with different source. Three completely different modes are possible:

## 1. F2D

Single data points taken from temperature channels like pyrometers or thermocouples must be organized in a 5x5 matrix. MCPS will interpolate the matrix to show a 2D graphic. In addition the values can be averaged over several scans.

## 2. MIV

In this mode the images of two or more infrared cameras are combined side by side to one big image.

## 3. DLL

For this setting the customer needs a specific DLL, which provides the image data. E.g. if the data is taken from another system and must be converted to MCPS.

## DX/MV Series

The Yokogawa DX/MV instruments provide a multi user mode with several users and one administrator. The regular users cannot access all functions like setup settings or memory downloads. You have to pay attention to this, if you want to you these functions in MCPS.

### Import:

With the Import/Download function MCPS can receive files from internal or external memory. First the directories are shown in a explorer style display. With Ctrl and Shift key in addition to the mousebutton it's possible to select more than one file in one step. MCPS will copy these files to a directory, that you can specify. After that you have to call the Import/File menu function to import the data. If only one file is downloaded, MCPS will import it immediatly.

Yokogawa provides a powerful setup program for these instruments. It can be called directly from MCPS, if the installation path of the software is known. This can be done in [Configuration/MCPS settings/Directories](#).

This setup program is also available on the MCPS installation CD (Ysetup.exe). The best way is to install MCPS first and then install the setup program in the default path (MCPS subdirectory).

After that you can download a setup from the instrument in the device manager. By pressing the button Edit setup the external Yokogawa setup program is called. Make your modifications, **save** the setup (NOT Save as) and leave the setup program. Now it can be sent back to the device by MCPS.

### Communication input channels:

The DX/MV series support 30 input channels, which can be set by external software. These channels can be used in math channels with C01 – C30 to reflect the transmitted data. MCPS can use these channels as an digital output (value 0 or 1 is set) or an analog output. With an alarm set to the math channel in the DX/MV it is possible to set an relay by MCPS (directly not possible).

### Send alarm acknowledge to device:

All hardware alarms are acknowledged, when one of the insturment's channels is acknowledged in the MCPS online alarm window.

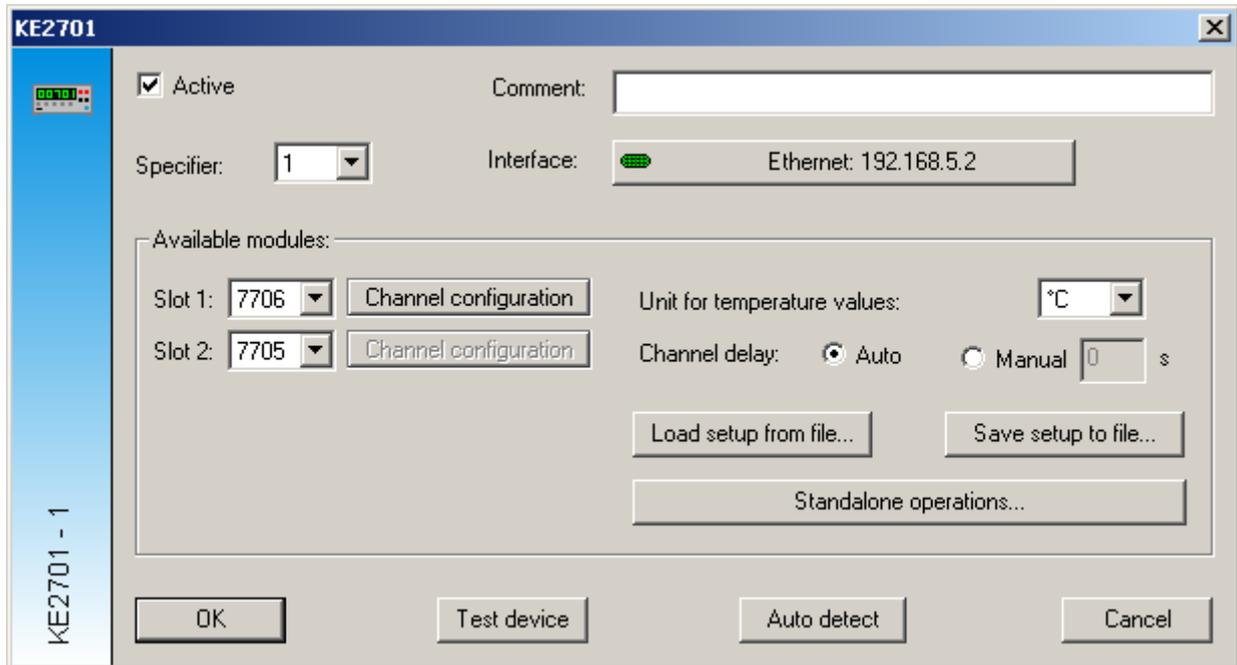
### DXP specific:

#### Suspend device measurement to set time:

For automatic time adjustment a running measurement at the DXP must be stopped and started again. Otherwise the time could not be set.

# Integra series

2700,2750, 2701



The instrument settings of the integra device are completely handled and set in MCPS. First the correct device configuration should be read (Auto detect). After that the channels can be configured. With OK the setup is saved in the device manager and sent to the instrument when starting a measurement.

Different setups can be handled by <Save setup> or <Load setup>.

The <standalone operations> window is used to setup the device for PC independent data acquisition. The internal memory data can be downloaded with Project/Import/memory file.

Channel setup

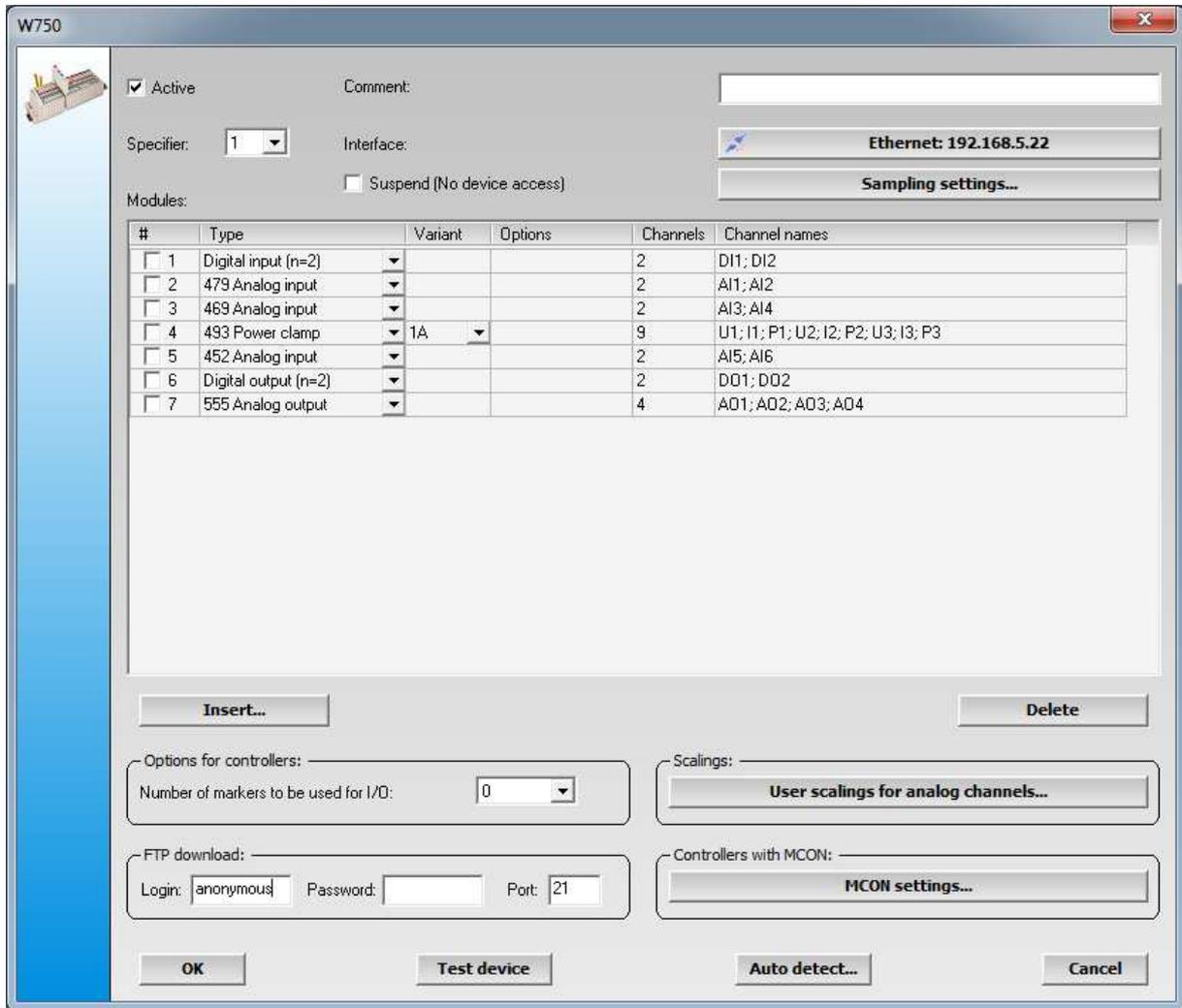
#	Channel	Mode	Range	CYCLES	OC	
<input type="checkbox"/>	1	101	DC-Volt	Auto	0.2	
<input type="checkbox"/>	2	102	DC-Volt	100mV	0.2	
<input type="checkbox"/>	3	103	DC-Volt	Auto	0.2	
<input type="checkbox"/>	4	104	DC-Volt	10V	0.2	
<input type="checkbox"/>	5	105	DC-Volt	Auto	0.2	
<input type="checkbox"/>	6	106	DC-Volt	Auto	0.2	
<input type="checkbox"/>	7	107	DC-Volt	Auto	0.2	
<input type="checkbox"/>	8	108	DC-Volt	Auto	0.2	
<input type="checkbox"/>	9	109	TC	J	0.2	
<input type="checkbox"/>	10	110	DC-Volt	Auto	0.2	
<input type="checkbox"/>	11	111	DC-Volt	Auto	0.2	
<input type="checkbox"/>	12	112	Ohm-2	Auto	0.2	
<input type="checkbox"/>	13	113	DC-Volt	Auto	0.2	
<input type="checkbox"/>	14	114	DC-Volt	Auto	0.2	
<input type="checkbox"/>	15	115	AC-Volt	Auto	BW: Slow	
<input type="checkbox"/>	16	116	DC-Volt	Auto	0.2	
<input type="checkbox"/>	17	117	DC-Volt	Auto	0.2	
<input type="checkbox"/>	18	118	DC-Volt	Auto	0.2	
<input type="checkbox"/>	19	119	SKIP			
<input type="checkbox"/>	20	120	DC-Volt	Auto	0.2	

OK Totalizer... Cancel

During a measurement MCPS is scanning always all channels, which are active (not set to SKIP) even if a project contains only a part of the channels. Channels, which are definitely not by a project, should be set to SKIP in the column MODE.

The first column is for selecting several channels to use the block operations. In addition the # button can be pressed to mark lines. If a header button is pressed (Channel, Mode, Range, ...) all selected channels can be setup in one step (if there is no type conflict between new value and marked line).

# I/O system 750



For communication with the WAGO I/O system 750 the correct clamp configuration must be known. Therefore you have to press <Auto detect>. MCPS will read the configuration and list all clamps. If a specific clamp couldn't be recognized, it is recommended to put this clamp behind all other clamps. Digital inputs and outputs do not provide an identification code. Some clamps are available in different variants, which cannot be detected automatically and must be set in MCPS manually. The channel names are those, which are used in the project configuration.

## Sampling settings:

Project independent sampling. See chapter DEVICES/DEVICE INFORMATION.

## Options for controllers:

Controllers support so called markers. These registers can be read and written by the PLC and MCPS. MCPS can read status information from the PLC or transmit specific parameters to control the program in the PLC.

By default digital and analog outputs are routed to the PLC and cannot be set by MCPS. In CoDeSys you have to change the clamp configuration from PLC to FIELDBUS1 to have access by our software.

## Scalings:

For analog inputs and markers the values can be scaled to physical values, e.g. for 4-20mA to 0-100bar. This scaling is also used, when downloading / importing data files from memory card.

**FTP download:**

MCON data files, which are stored on SD card can be downloaded (PROJECT/IMPORT/MEMORY DOWNLOAD) by using the FTP transmission. MCPS contains a small FTP client to perform downloads.

**MCON:**

MCON is a PLC program developed by CAD Computer to be run on 880-controllers with SD cards. MCON provides the following features:

Saving input values and markers to SD card for backup purpose. These files can be downloaded by any FTP client or directly by MCPS. The file can be imported into MCPS projects.

Complete handling of outputs and intelligent clamps in conjunction with MCPS. Normally a controller gains access to all outputs or special clamps as the power clamp. MCON communicates with MCPS, so that data are shared between PLC and MCPS. As a result MCPS get the data from all clamps and can set each output.

Smart interpreter to perform custom specific controlling. Several elements are implemented by default (PID controller, timers, comparisons and so on). A command list tells MCON, which inputs and outputs are to be used and routed to application specific functions.

## **IRDemo**

On the MCPS installation CD in the folder INSTALLATION there is a program IRDemo.exe. This program must be run after the MCPS installation to provide the demo driver IRDEMO for thermographic test purposes. Several sequences are copied, which can be selected, after the IRDEMO was added to the device manager. If a sequence was defined, several zones can be added, which are available as channels in the project configuration.

## **KEYDRV**

The KEYDRV is an internal MCPS driver and is similar to the TESTDRV. It is used to measure data, which are not taken from a datalogger but from the keyboard. A KEYDRV can handle several channels, which are configured with user defined values. These values are shown and saved as regular input values. The KEYDRV channel values can be modified during the measurement to reflect a new situation or the change of an external data value. See DATA ACQUISITION/KEYDRV VALUE.

# Modbus Devices

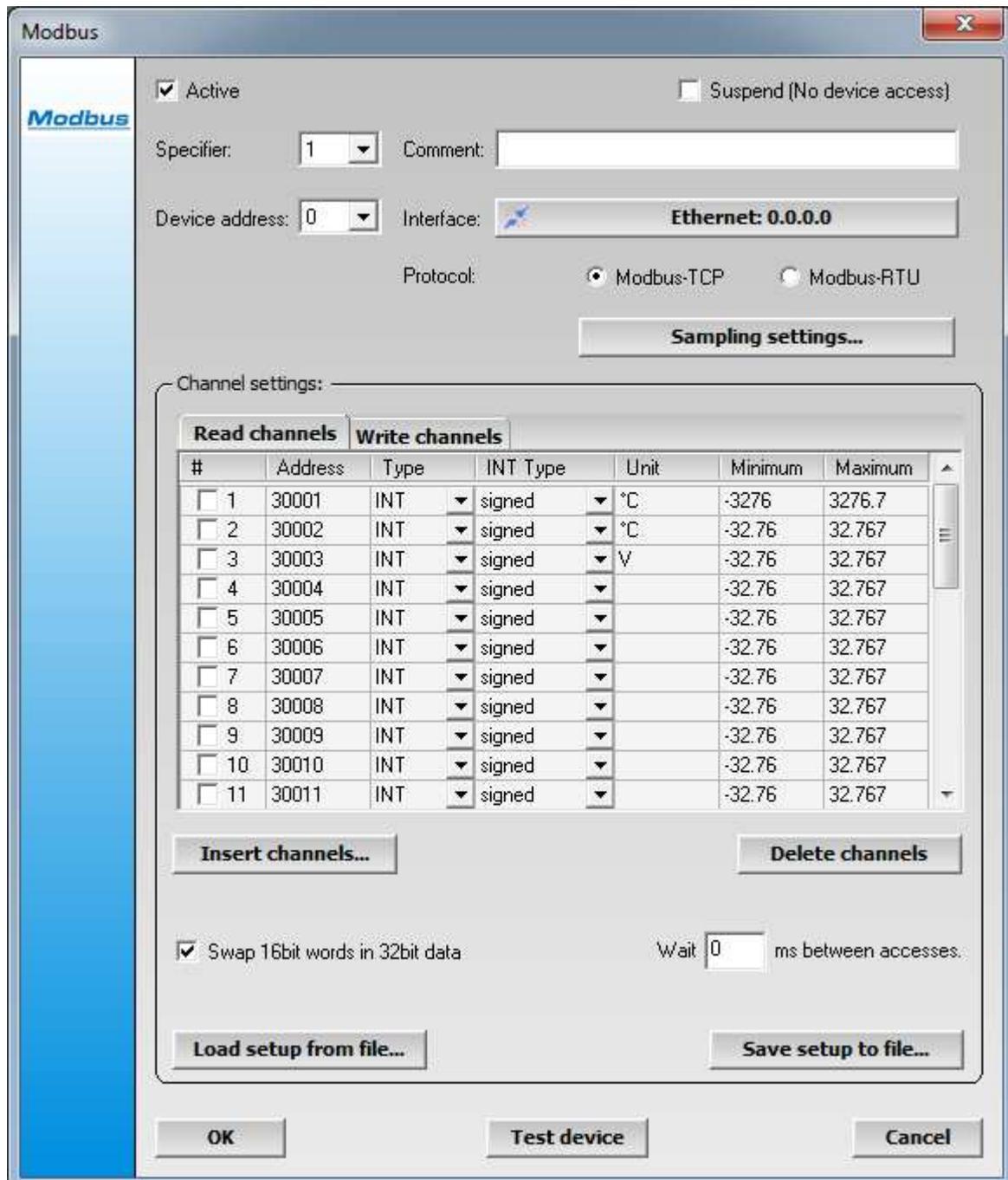
A Modbus instrument can be used in MCPS by adding a Modbus device in the device manager. In the configuration dialog window it's necessary to set the interface, device address and number of channels.

Protocol:

If ethernet is used for communication, MCPS will normally use the specific Modbus-TCP protocol. In case of an ethernet/serial converter the underlying protocol must be set to Modbus-RTU.

Data transmission:

Each device receives the data when it is triggered by the project according to the samplerate. If an instrument is too slow to transmit data with every scan the device can be set to <Autorun>. The driver will start itself again, when the data transmission is completed. The project will not wait until the device is ready but will take those data, which is available from the last transmission.



**Sampling settings:**

Project independent sampling. See chapter DEVICES/DEVICE INFORMATION.

MCPS supports several Modbus data types:

**1. 16 Bit Integer format**

This is the native Modbus format, where values are stored into 2 Bytes. Since these are unscaled values, you have to enter the low and high value as defined in the instrument. This must be done for each channel. If the instrument's settings are changed, you have to modify MCPS settings.

**2. 32 Bit Float format**

These are scaled values and high/low parameters are not needed. Since this format uses 4 Bytes, the modbus addresses are increased by 2.

Some devices changes high and low word for the transmission. MCPS can handle even this behaviour.

**3. 32 Bit Integer format**

As 1. but with 2 Registers

**- Insert channels**

One or more channels are added to the channel list. The format and start address can be specified, so channels can be added easily.

**- Delete channels**

All channels, which are marked (#) are removed from the list.

**- Load / save setup file**

The settings can be saved into a file and reloaded again. This is useful in case of different settings or several devices in MCPS with the same configuration.

If the addresses cannot be read out from MCPS in one step and several transmissions are necessary (for example to get status information) it's possible to enter a wait time. This is sometimes necessary, because some (few) instruments cannot handle fast following accesses.

**Write channels**

MCPS can also write data to a Modbus device. The first address is 1. The scaling is done in the project configuration for each output and not in the driver settings. These analog outputs are also available in digital output selection in MCPS e.g. alarm actions. The Modbus register is set in this case to 0 or 1.

# MX100 / MW100

## Basic settings:

If you connect a MX100/MW100 the first time to MCPS or modules are changed, you have to make a <reconstruction> to update the internal settings of the device. After this an <Auto detect> should be performed to update MCPS.

!!! After a reconstruction all channel settings are reset to default by the instrument itself !!!

## Memory card:

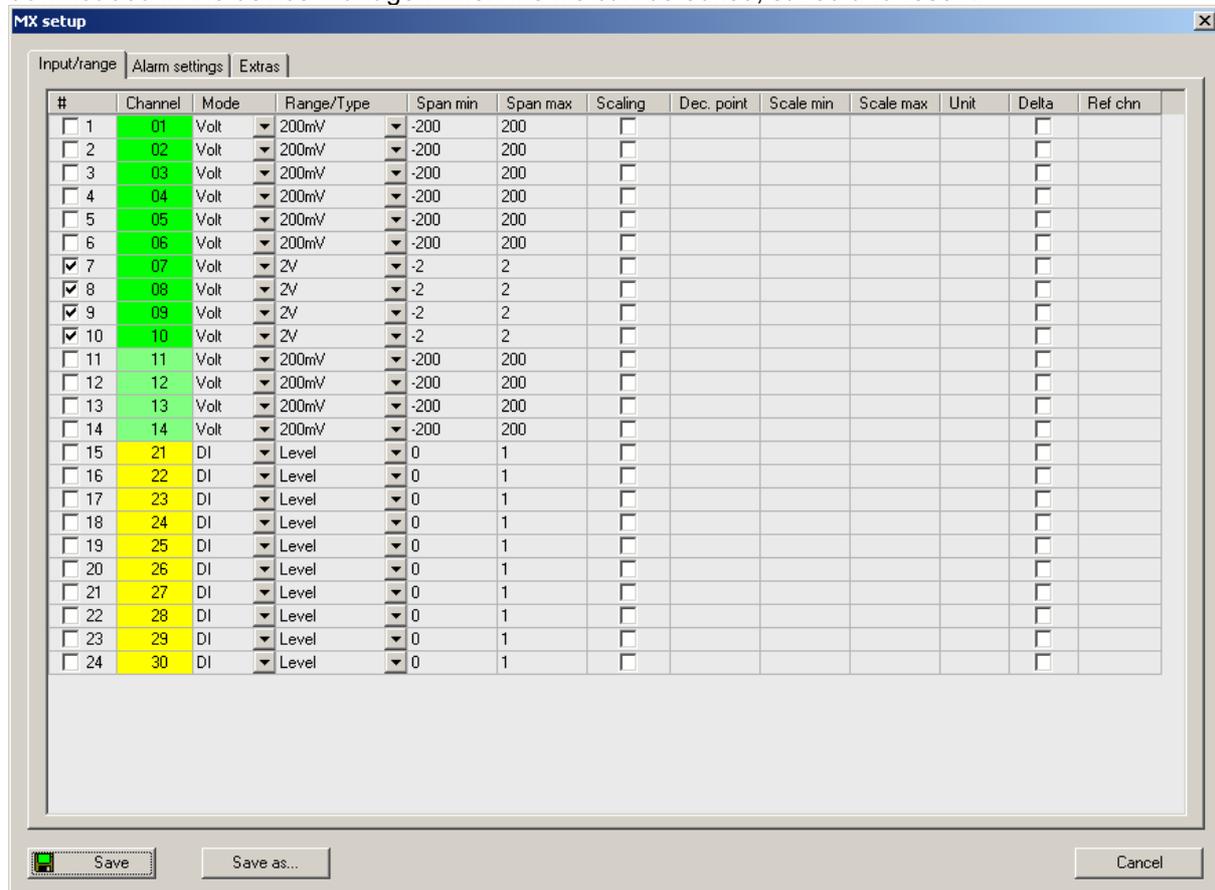
If backup is selected, MCPS will set the datalogger to DIRECT/ROTATE. So the latest data is always available. According to device sample rate and number of channels, the file size must be selected properly (not too big). Otherwise the datalogger can not be started.

## Relay module:

If the MX100/MW100 contains relay outputs and they should be controlled by MCPS, the relays must be set accordingly in the setup editor. After setting the relays to <MCPS> and sending the setup back to the instrument, an <Auto detect> must be made. Otherwise MCPS cannot recognize the new settings.

## Setup editor:

The setup configuration of the MX100/MW100 is completely made in MCPS. First a setup must be downloaded in the device manager. Then the file can be edited, saved and resent:



Channels can be marked with the check box or the <#> in the header line. All marked channels can be set in one step by pressing the header button of a column.

# OPC

This industrial software standard interface is used to connect instruments with OPC server capabilities to MCPS. MCPS contains an OPC client, which can read data from OPC server systems. These software programs are provided by the hardware manufacturer and make it unnecessary to develop specific drivers.

Since every data system with an OPC server can be connected to MCPS, it's possible to get data from dataloggers, PLCs, Profibus, SPS or even other software packages.

MCPS can connect to those servers, which are registered on the MCPS system. Mostly the OPC server is installed on the MCPS computer or is registered by a special program.

When adding an OPC device, the device manager will list all servers, which are known to the system. The channel configuration defines, which channels of the OPC server should be used in MCPS. The tag names can be entered manually or, if supported read out from the server (Add from server).

Each channel used in MCPS gets a channel number for identification in the project configuration. If a channel is deleted, the other ones are renumbered!!! This is important for the project configuration. There you have to change the channel numbers. If a channel is of type string, MCPS tries to convert it to a real value. String data is used for controlling MCPS (batch start or serial number). See triggering below.

For each channel a **trigger** monitoring can be activated. If the value of channel changes, the script is called. With GetScriptData the script can received additional trigger information.

Some OPC-Servers also support channels to be written. These write channels can be used in MCPS in the <OUTPUT> column of the project configuration. This column must be enabled by **PROJECT/CONFIGURATION/EXTRAS/COLUMNS**. The measured or computed value of this channel is sent to the OPC server write channel.

The OPC channel list can be imported or exported to ASCII files. This can be used to integrate tags from other sources or simply modify a large number tag identifiers.

If a server is shutdown, the event is sent to MCPS. MCPS will gather no more data. If the server is restarted again and MCPS should continue data acquisition, you have to press the button <reconnect server> in the device manager (OPC device).

## **Sampling settings:**

Project independent sampling. See chapter DEVICES/DEVICE INFORMATION.

# Opus

With the start of a measurement MCPS activates only the online / current values of the device. Maximum, Minimum or average values are turned off, because statistical computation can be done in MCPS. Settings from Smartgraph Software are overwritten to store those channels in memory, which are used in MCPS. This guarantees, that in case of a gap filling all needed channels and data are available.

For security issues you can set a password, which is written into the Opus20 device. So it is not possible to access the device from another MCPS system or SmartGraph without knowing the access parameters.

To get the internal memory data use PROJECT/IMPORT/DOWNLOAD.

# S7

Driver for Siemens® S7-300, S7-400 or S7-1200. It's possible to read inputs, data blocks or markers and to set outputs.

Several types of communications are possible:

- RS485: Direct MPI communication
- USB: MCPS uses an installed SimaticNet to communicate. USB/MPI converters can be used
- Ethernet: Protocol:
  - a) TCP/IP: Direct Ethernet communication to CP-module (Set CPU slot)
  - b) Netlink: Communication to MPI via Netlink-Ethernet-MPI converter
  - c) NetlinkPro: Communication to MPI via NetlinkPro Compact-Ethernet-MPI converter

For b) and c) you have to define the MPI address. For Baudrate you can choose <Auto>, if communication speed is unknown, otherwise we recommend to use default setting (\*) speed. Some PLCs do not support automatic speed detection.

Columns:

- # Consecutive number
- Group: I = input, O = output, M = marker, DB = datablock
- DB: In case of group <DB> the number of the data block
- Address: Address or offset of the channel
- Type: Data type
- N: Number of bit in case of data type <BIT>
- L,H,SL,SH: Scaling of input values. Values from L to H are scaled from SL to SH. E.g. 4-20mA to 0-10bar
- Tag: Tag string, which can be used in the project configuration
- Unit: Unit string
- .\*: Number of decimal places
- Trigger: If the check box is activated, MCPS will call the script below, if the current value differs from the last one.
  
- Insert: Inserts one or several channels
- Delete: Deletes all marked channels
- Import: With Step7® you can define tag names for a channel, which can be exported into an ASCII-file. MCPS can import those files.
- Load setup: Reloads a setup, which was saved with <Save setup>
- Save setup: Saves the current channel list into a file. This can be useful for backup purpose or to transfer settings to other drivers or even computers.
  
- Trigger actions: A script is started, when a trigger condition of a channel is detected. In the script a list of the channels, which cause the script call can be received by calling <GetScriptData>

## Sampling settings:

Project independent sampling. See chapter DEVICES/DEVICE INFORMATION.

## VR Series

VR100 / VR200

For internal saving the View Recorder has two basic memory section. One is the event file to store each measured scan and the second is the display file, where a minimum and a maximum value for each display interval is stored. Depending on the memory settings the instrument creates one or two files. The event file name starts with e the display file name with a d.

In case of a powerfail or event the VR creates an additional info file. If one of the data files is copied to your harddrive, make sure, that the related xxx.inf file is also available in the same directory for the MCPS import function.

# WT Series

WT1x0, WT2x0, WT10x0, WT20x0.

GPIB:

If you are using one of these devices it's absolutely necessary for GPIB communication to set the interface mode at the instrument to **ADDRESS MODE A!**

RS232:

Communication setting: nor (normal)

WT1x0, WT2x0, 2531:

These instruments have 14 channels. Each channel can be set to a specific input value. Please check the channel output setting (CO-OUT) at your powermeter. MCPS can change the channel settings by editing a downloaded setup.

WT10x0/20x0

These powermeters can transmit all values, that are available in the instrument. But it's possible to transmit only a part of all channels to speed up the data transfer. Often the devices are set to send only a few basic values and the others are missing in MCPS. Please check the channel output setting (CO-OUT) at your powermeter.

For the great powermeters MCPS uses channel names like V1 and P2 instead of channel numbers.

WT1600/WT500/WT3000/WT1800:

MCPS has a setup editor to set input type and element of up to 255 channels. First a setup must be downloaded in the device manager. Then this file can be edited, saved and resent again. To use the harmonic analysis, you have to activate the related button in the device configuration of the device manager. MCPS will turn the device automatically in harmonic analysis for setup operations and measurements. WT1600 only: For normal data acquisition the check box <Harmonic analysis> must be turned off.

USB interface: To use USB communication a specific driver from Yokogawa must be installed. It is available on MCPS-CD. See directory Special/Yokogawa/YKMUSBxxx.zip. In MCPS you have to enter the serial number of the device into the field <USB-ID> in the device manager. The correct USB serial number is displayed at the WT in the communication section.

# Infrared systems

## Infrared device options

### Overlay image:

The bitmap file is displayed over the thermographical image. All pixels with the transparency color are not visible.

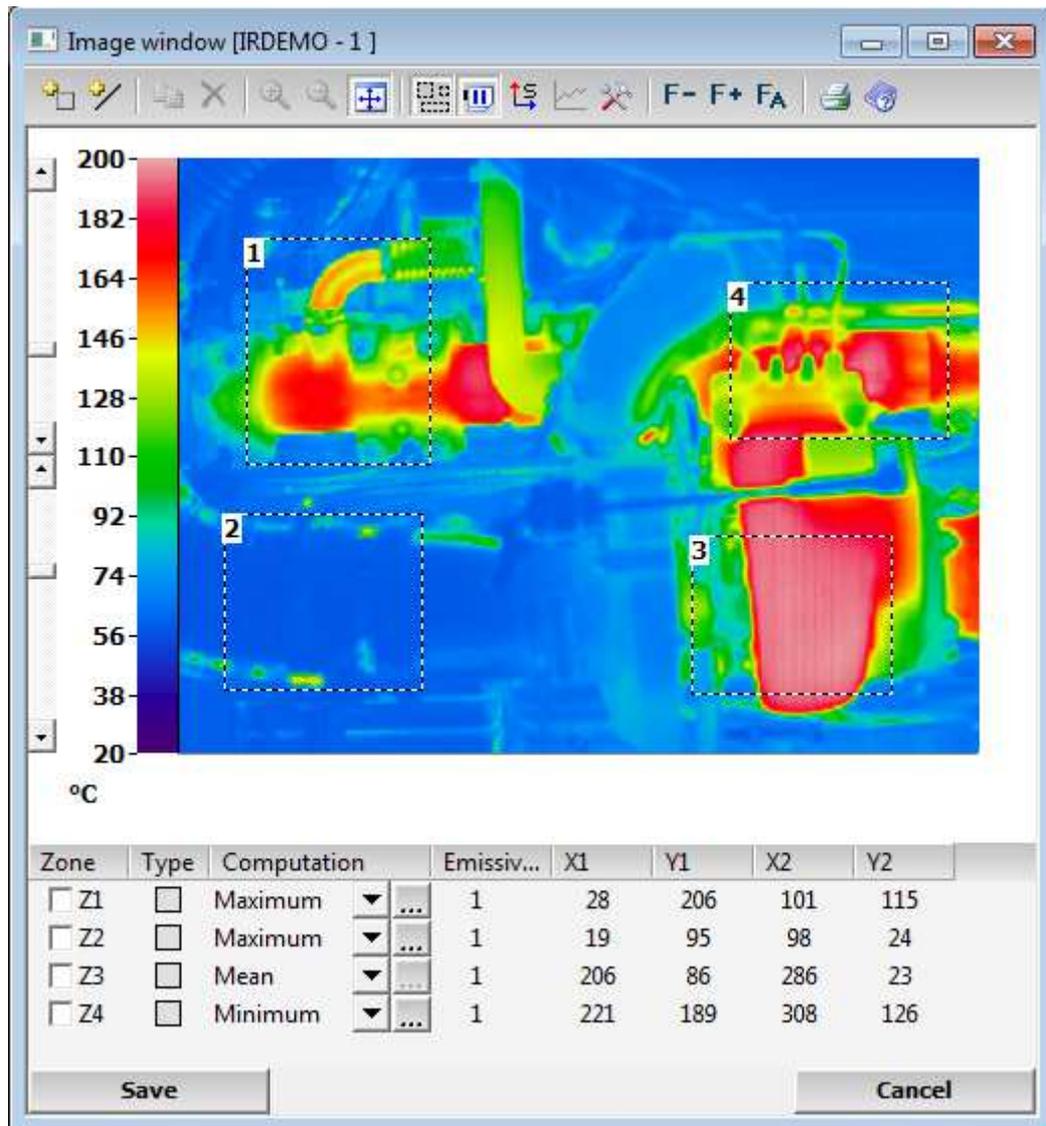
### Image settings:

The image can be flipped and/or rotated. This is done before the zone data is computed.

### Scaling of axes:

Normally the width and height of the image in pixels are shown. This can be replaced by user scalings, which are also used to define the zones.

## Configuration of zones



Device drivers for infrared cameras can define zones or regions of interests (ROI) in the thermographical image. This can be used to compute the minimum, maximum or average values of a specific region. The mathematical result is available in the project configuration as a hardware channel and can be used in all other windows as a regular data logger input.

The toolbar:

-  Inserts a rectangle zone
-  Inserts a line zone. If line zones are available, the button to view the profiles of the line regions is enabled.
-  Copies a zone
-  Deletes the marked zone. A zone is marked, if the related check box in the table is active.



Pause. The image display is hold. Temperature value at mouse point is shown.



If scaled values have been selected in the device options of the camera, this button can change the table data from pixel values to user values.



Image zoom with factor 1, 2 or 3.



Image unzoom



Fit image to window



Show/hide zones



Window configuration.



Shows the profiles of line zones. See below.



Prints the window



Help

The table:

#### 1. Zone

This column displays the zone number and contains a check box. If a zone is selected by pressing the mouse button in the zone frame or one the zone line, the check box of the related zone becomes active. A zone can also be selected by pressing the check box. This check box is only for selecting a zone and does not mean the activation of a channel.

The delete operation removes only the selected zone.

#### 2. Type

Kind of zone: 1. rectangle, 2. line

#### 3. Computation

Type of computation of the values inside the frame or on the line. The result of this computation is available as a channel value in the project configuration with the names Z1 (zone 1), Z2, Z3 and so on. The type <Ignore> delivers no value and is so not available as a channel. The ignore zone is used to skip values in the image for computation.

In case of maximum or minimum computation the ellipsis button ... is enabled and allows further settings:

Maximum / minimum value: The highest/lowest value in the zone is returned.

Average value of the upper/lower **n** values: The values are sorted and the upper/lower **n** values are taken to compute an average value, which is returned.

Average value of the upper/lower **n** percent. Same as 2., but **n%** values are used, e.g. 10% of all values in the zone.

#### 4. Emissivity

Since nearly all materials do not have an emissivity of 100% the measured temperature is not the right one. To correct the values an emissivity factor can be set.

#### 5. X1, Y1, X2, Y2 :

Coordinates of the zone frame/line. For line cameras only X1 and Y1 are visible. These values change, if the zone is moved or sized. The table can be modified directly by entering new coordinates and the zones are updated.

# Interfaces

## RS232

### RS232 cable

For any kind of transmission from or to an instrument MCPS doesn't need any handshake. Be sure, that the device is also set to no handshake. Otherwise it maybe won't respond. So MCPS only needs a very simple communication cable:

PC (25pins)	Device (25pins)
2	3
3	2
7	7

PC (9pins)	Device (25polig)
2	2
3	3
5	7

**MCPS** supports up to 36 serial interfaces. The user must take care, that the interface selected in MCPS is really available. In most cases system drivers must be installed for serial interface cards, which provide additional communication ports. These drivers should come with the serial cards. See also [Interface configuration](#) .

## RS422A

**MCPS** supports up to 36 serial interfaces. The user must take care, that the interface selected in MCPS is really available. In most cases system drivers must be installed for serial interface cards, which provide additional communication ports. These drivers should come with the serial cards. See also [Interface configuration](#) . To communicate with RS422A devices your computer should use a RS422 interface card or an external RS232/RS422 converter. The company CAD Computer recommends to use the WT-86001 converter, which can also be found in the CAD Computer's pricelist.

## RS422A Interface Connection Diagram

WT Converter	RS422A Instrument
11	SDA
10	RDA
23	SDB
22	RDB
14	SG

## GPIB or IEEE-488

At this time MCPS supports GPIB cards from National Instruments. The card and the drivers must be installed as described in the NI's manual. In case of problems please use the diagnostic programs, which are delivered with the card.

## Ethernet

The communication with ethernet instrument is running with TCP/IP. So it's necessary, that this service is installed on your computer. In many cases the TCP/IP address of the instrument must be adapted to the computers IP address. The first three addresses should be equal for computer and device. If the instrument should be connected by DNS rather than IP address, the check box <DNS> must be activated to show the DNS text input field.

## Interface selection

Interface selection

Interface type:

RS232  GPIB  RS422A  Ethernet  Remote connection

RS232:

Port: COM1

GPIB:

Address: 1

RS422A:

Port: COM1

Address: 0

Ethernet:

Address: 192 . 168 . 005 . 8

Port: 34318  DNS

Device for remote connection:

U-1496E+ external LED,19k2,Fax + Voi

Protocol: (Modem / Serial-Converter)

RS232  RS422A 1

Error handling:

Timeout: 1 second(s)

Number of retries: 1

Options:

Pre-command:

Pause after send: 0 ms

IP Computer: 192.168.5.121

OK Help Cancel

For most of the devices the user must select an interface. The right settings are important for communication. For serial interfaces (RS232/RS422A) it's also necessary to set some interface parameters (See also [INTERFACE CONFIGURATION](#)).

Using the REMOTE CONNECTION interface type is only possible, if you have installed at least one modem in the system configuration of Windows. Additionally a protocol type should be selected depending on the instrument, which is connected at the other end of the line.

For some devices special settings are very important for a valid connection. See chapter DEVICES for more information.

### Error handling:

In some cases it's necessary to modify the standard error handling. Especially in conjunction with modem connections the timeout or retry values need to be changed.

### Pre-command:

This hex string (e.g. 03BA) is sent as binary data everytime before a command is sent to a device. This can be used for addressable radio modems.

### Pause after send:

After a command is sent, MCPS will wait the specified time (in milli seconds).

# Mathematics

## Basic Math Concept

**MCPS** provides several features to compute new values from raw data or already computed data. Basically MCPS distinguishes between the following situations in the project configuration:

Mathematical operations in a raw data channel

In this case the project channel contains a device and channel identifier. The math formula is added and if set active the result will replace the raw data value in every output window of MCPS. The original value is modified by a mathematical operation for example to scale or convert the raw data. If the math formula is turned off, to raw data value will be displayed.

Pure math channel

The device name is set to MATH and no raw data is configured for this project channel. In this case MCPS will generate values in addition to the raw data channels. If a project already contains data, it's only possible to add pure math channels. This is even during a measurement allowed. If a pure math channel is turned off, it will be removed from any output window.

Computations with raw data:

In the math formulas of the project configuration raw data values are represented by **Xn**. X is the raw data identifier and n the consecutive (line) number of the project. This can be any row, which contains raw data input (Device and channel).

$X1 + 1$
$X1 + X2 - X3$

Examples:

$X1 + X2$       The result is raw data from line 1 added to raw data of line 2

$X2 * (X3 - X5) / X7$

A special case is the variable **X** without a following number. In this case MCPS will use the line number, in which the X was found. This makes it easier, if a lot of channels are modified by the same formula, which can be set by using the **block operations** :

$X * 1000$

The example also shows the usage of constants: 1.23, -5e3, 7.456778

The X variable without line number can not be used in a pure math channel.

Computations with math values

Each value of a project channel, that is a mathematical result can be used for further math operations. The variable **Mn** specifies the project channel n, which result is used for the computation:

$X1 + X2 + M1 * M3$

MCPS provides a lot of variable and functions to compute new math values. Each math result of a function can be used for further operations even in the same formula:

$MAX(X3, X1 + 5 + SIN(COS(X) - 1))$

- Standard functions (sin(X1), cos, tan, sqrt, exp, ...)
- Compare functions EQ, GT
- Moving average functions
- Logical functions (AND, OR, ...)
- Special functions
  - Integrations
- User functions
- Polynomials

# Operators and Functions

## Standard operators

+ - \* / ^

$X1 \wedge X2$ ,  $(X1 - X3) + x5$

## Constants

$X1 + 2$

$X5 * 1.23456$

$1e-3$

A math formula can contain only a constant. In the graphic window MCPS will draw a horizontal line. Sometimes this is useful to have a reference line or an alarm limit.

## Standard functions

SIN(X), COS(X), TAN(X), SINH(X), COSH(X), TANH(X), ASIN(X), ACOS(X), ATAN(X), LN(X), LOG(X)

**EXP(X)** ( $=e^X$ ), **SQRT(X)** ( square root), **ABS(X)** (Absolute function)

# Integrations

Integration with timed reset

INxxxxxU(X)

Each scan the parameter X is added. After the time xxxxxU the value is reset, with xxxxx the time number (1-99999) and U the unit S, M, H.

IN30S(X1): Add X1 for 30 seconds, then restart with 0.

Summation

SUM (X)

Each scan the parameter x is added.

Summation with comparison reset

SUM (p1, p2, p3)

The parameter p1 is added so long as p2 is greater then p3. Otherwise the summation value is reset to zero.

SUM(X1, X2 , 100)      X1 is added until X2 is not greater then 100.

# F-Value

MCPS contains a function to compute the F-Value for sterilization (e.g. F0).

## 1. Standard F-Value:

FV(T,Tb,Z)

T: Temperature value from data acquisition system  
Tb: Reference temperature (121°C in case of steam)  
Z: Z-Value (10°C in case of steam)

The F-Value is computed as long as  $T \geq T_b$ . Otherwise the value is reset to 0.

## 2. Extended F-Value:

FV(T,Tb,Z,m, n\_low, t\_low, av)

T: Temperature value from data acquisition system  
Tb: Reference temperature (121°C in case of steam)  
Z: Z-Value (10 in case of steam)  
m: Operation mode of function  
m = 0: F-Value is reset to 0  
m = 1: F-Value is computed normally  
m = 2: F-Value computation is paused

m = 3: T can be  $n\_low$  times below  $T_b$ , but not longer in total than  $t\_low$  before reset.

If the sterilization time is above the acceptance value (av), the F-Value will not be reset by an underrun and is frozen even if T is above  $T_b$  again.

m = 4: Equal to 3, but the F-Value itself is used for comparison to the acceptance value instead of the sterilization time.

FV(X1,121,10,3,60,1800)

Channel 1 is used for F-Value calculation with a reference temperature of 121°C and a Z-Value of 10°C. The value is reset, if more than 3 times X1 is below 121°C or the total time of underrun is more than 60 seconds. After half an hour the required sterilization time is reached and a new underrun will not reset the F-Value.

In case of operation mode = 0,1 or 2 the parameters  $n\_low$ ,  $t\_low$  and  $av$  can be omitted.

FV(X1,121,10,2)

Time of sterilization

In parallel to the F-Value computation the related time is available. Use FT instead of FV.

FT(T,Tb,Z) Standard function  
FT(T,Tb,Z,m, n\_low, t\_low, ak) Expanded funktion

The return value is the time of sterilization in seconds.

## Compare Functions

The functions **GT** and **EQ** can compare two parameters. As a result one of the two additional parameters is returned:

**GT(p1, p2, p3, p4)** results p3, if p1 > p2, otherwise p4.

**EQ(p1, p2, p3, p4)** results p3, if p1 = p2, otherwise p4.

The EQ function should be used carefully, because the compared values are often similar, but not identical.

Examples:

GT(X1, 100, X2, 2) if X1 is greater 100 then the result is X2 otherwise 2.

X5 / EQ( X1, 0, 1, X1) Avoids division by zero: if X1 is 0 then the result is 1 otherwise X1

## Logical Functions

**AND, OR, XOR and NOT** are logical or boolean functions. All values different from zero are treated as 1.

AND, OR and XOR can work with two or more parameters.

AND: Result = 1, if all parameters are not zero

OR: Result = 1, if one parameter is not zero

XOR: Result = 0, if all parameters are zero or all non zero. (0,0,0) / (1,1,1)

NOT: Result = 1, if parameter is zero, otherwise 0

Example:

AND(X1, X2, OR( X3, X4, X5), NOT( X5))

**BIT** Bittest of bit 0-31; result = 1, if bit is set otherwise 0

BIT(X3,0)

## Moving Average Functions

MCPS can collect values over a certain range of time and computes their mean value. If the time range is defined for one hour, then MCPS will deliver each scan the mean value of the last hour. This is a moving average, which changes every scan and is not hold for one hour.

The mean function's format is SMxxxU:

SM is the identifier for mean function, xxx is the time range and U is the time unit. E.g. moving average over 10 minutes of raw data X1:

SM10M(X1).

MCPS supports up to **100** mean functions. Depending on the time range these functions need a lot of memory.

MCPS tries to average the value in the specified time window. In case of **data reduction** the offline value can be different from online values!

## Special Functions

### ITG, FRAC

ITG(x) / FRAC(x)                      Results the integer/fracture part of a value

### MIN, MEAN, MAX

MIN(x1,x2,...) / MEAN(x1,x2,...) / MAX(x1,x2,...):

Minimum value, mean value or maximum value of all parameters.

### RXMIN / RXMEAN / RXMAX / RMMIN / RMMEAN / RMMAX:

Minimum value, mean value or maximum value of a range of raw data or math channels.

RX = Raw data of channel, RM = Math value of channel.

Format: RXMIN(Start channel, End channel)

RXMIN(1, 77)    ---> Minimum value of all raw data values from channel 1 to 77

### ID

IDxxU(x,n)

Interval difference between the current value and the value xx seconds/minutes/hours ago. Normally used with (slow increasing) counter values to see the increase for a specific time period. The result is static for this period. If a second parameter is in the formula, MCPS will use this as the counter maximum value. If the counter is starting at 0 again, MCPS will calculate the correct difference.

ID10M(x1,65535)

Difference channel x1 between current value and value 10 minutes ago.

### SD

SDxxU(x,p2p)

Stability function. Checks, if for a specified range of time the difference between maximum and minimum is below or equal to the given value p2p.

SD10M(x1,3): Returns 1, if the peak to peak value of channel 1 is lower than or equal to 3.

SD function with additional control parameter:

SD10M(x1,1,modes)

mode = 0: As above

mode = 1: As above, but check is performed with first scan (without waiting until the specified time window is valid).

mode = 2: Maximum of data in time window

mode = 3: Minimum of data in time window

mode = 4: Difference of maximum and minimum value

mode = 5: Data value 10 minutes before (in case of 10M)

### RV

Replacement value for invalid data or invalid math operation

RV(Original value, Replacement value)

RV(x1,5)

x1+x2+RV(m1/m2,1)

## **A**

Alarm status of a channel/level. Important: The result is delivered with the NEXT scan, because math formulas are computed before alarm conditions are checked. The function returns 1 in case of alarm, otherwise 0.

A(Channel number, Level)

A(2,1): Alarm of channel 2, level 1

A(5,0): Channel 5, all levels

## **PID**

PID-Controller can be used to control slow systems. Since the sample rate of the project is important for the control algorithm, the sampling must be fast enough to work properly. The controller function should not be used for faster or safety relevant systems. It is intended to be used for example in test applications.

Computation formula:

result =  $kp * (\text{error} + \text{integral}/tn + tv * \text{derivative}) + y\text{offset}$  ;

error = Setpoint – Process value

integral = integral + error \* dt (Sample rate)

derivative = error / dt

kp: Proportional part for scaling

tn: Time constant for integration

tv: Time constant for derivation

Dependent on the input signal the parameters can be smaller than 1.

Parameters:

PID(Measured value,Set point,kp,tn,tv,low limit, high limit,y-offset,manual value, manual mode)

Limit parameters should be set to define valid output range.

The controller can be stopped by setting the parameter <manual mode> to 1. In this case <manual value> is output.

PID(x2, 25.2, 0.1, 0.3, 0, 10, 30, 2, 22, x3)

# Extras

## System variables and constants

SR (Sample Rate) Sample rate of the project  
SC (Sample Count) Number of scans since the start of the measurement  
SS (Sample start) Only 1 for the first scan, otherwise 0  
PI Circle constant  
INV Invalid value

Example:

SC \* SR

This will show the passed time of measurement in seconds.

## Values of the last scan

Like X1, X3, X of the actual scan the variable Y1, Y3, Y delivers the values of the last scan. This is useful to see important changes or gradients:

X1 - Y1 : Difference to the last value

X and Y are the the row data variables!

For math values you can use M1, M2,... for the current scan and N1, N2, ... for the values of the last scan.

Important:

For the first scan Y and N deliver \*\*\*\* (NO DATA). In some cases this is not useful and can be avoided with the SS (sample start) variable. SS is internally set to 1 for the first scan, otherwise it's always 0.

EQ(SS,1,100,Y1):

EQ is the Equal function (see below) and means: If SS is equal 1 then deliver 100 otherwise Y1.

## Register

Registers are channel independent memory blocks, where values can be stored and readout again. They can be set several times in the math functions, so that even more complex computations are possible. There are 100 register called R1, R2, ... R100. They are set with the RS (Register Set) function. The format is:

RSn(p) n = Register, p = value or parameter

Because every function must return a value the RS function delivers the parameter value. Example:

X1 + X2 + RS1(X3)

X1 \* R1

The math formula RS1(MAX(R1,X1)) will hold the maximum value of channel 1 during the complete measurement.

Note:

The registers are initialized with 0. If you want to use an other start value, you have to enter the following math line before a register is used.

RS1(EQ(SS,1,-5,R1))

For the first sample the Register is set to -5 otherwise it holds its content.

## Values of user columns

A user column can be used to define values. For example an offset correction. To use these values in a mathematical function the following syntax is necessary:

UCV(column,channel) or UCV(column)

If channel is defined the related channel line of the project configuration is used, otherwise the line of the math formula is evaluated. Column defines one of the user columns. The text of the user column field must be a real value.

Example:

X1 \* UCV(5) → X1 is multiplied with the value defined in user column 5

## RM Values

The current active set of reference measurement is also available as variables to be used in math formulas. The Z-identifier and the related channel number return the offset value.

Zn or ZXn	Offset value of raw data channel <b>n</b>
ZMn	Offset value of math formula of channel <b>n</b>

Example:  
ZM1 – ZX1

# Concept of User Functions

MCPS supports so called user functions. These are universal and often used math formulas, which are stored project independently in own files. One user function file can contain up to 500 user functions. Each file can be used in several projects. A user function is referenced in a project channel's math formula by  $F_n(x,y,z,\dots)$ , with  $n$  the function number and  $x, y, z, \dots$  the parameters. There are up to 26 parameters possible. The user function will name the parameters as A, B, C, D and so on.

A typical user function can be the following:

$$A + B - C$$

In the project configuration a math formula can call this user function by

$$F1(X1, 5, X3 - X5)$$

Translated to the user function:

$$A = X1$$

$$B = 5$$

$$C = \text{result of } X3 - X5$$

The user function will compute the result, which is returned for F1. This return value can be used for other functions:

$$X2 + F1(F2(F3(X1), X2, X3) + 5)$$

The number of parameters for each user function is defined by the its number of different variables:

1 parameter:  $A$  or  $A+1$  or  $A+2*A-2/A$

2 parameters:  $A+B$  or  $\text{SQRT}(A^2 + B^2)$  or  $A+A+A+B+B+B-A/2$

3 parameters  $A+C+B-2*A/5$

The user must specify, which user function file the project should use. This is made in the project configuration/global settings/[Project path](#).

To define user functions see [Mathematic/User functions](#).

To create a new empty user function file see [Mathematics/New files](#)

## Use of Polynomials

MCPS supports polynomials to adjust measured data (e.g. linearization, ...). These polynomials (power 1-9) are created by up to 20 xy- pairs, which are defined by the user. MCPS can handle 500 polynomials in one specific polynomial file. Several files are possible.

In the math formula of a project channel a polynomial is called by  $P_n(x)$ , with n the number from 1-500 and x the argument:

$P_1(X_1)$

$P_{233}(X_1 + X_2 - P_2(X_3))$

To define a polynomial see [Mathematics/Polynomial](#).

The project **must** know, which polynomial file is used. The reference is made in the project configuration/global settings/[Project path](#) . Each polynomial file can be used for several projects. So it's only necessary to define basic polynomials once.

## DLL function calls

MCPS can call functions in DLLs for customer specific applications. The communication to the DLLs is defined in [CONFIGURATION/MCPS SETTINGS/DLLS](#). In the math formula the function is called in the following way:

DLL( DLL number, "function name", parameter 1, parameter2, ... parameter 15)

The function name or text parameters must be enclosed in quotation marks.

Example:

$X1 + X2 + \text{DLL}(1, \text{"TOTAL"}, X3, X4, X5)$

The return value of the function is converted to a float value to be used for further arithmetics.

Important:

Since DLL function calls are integrated in the complete data acquisition and display cycle, the functions should not take too much time to operate. Otherwise this can influence the measurement.

# Tutorials

## First steps

MCPS has a modular design with several options and a lot of combinations are possible. We distinguish two basic application types:

### 1. Measurement version

The package includes one or more device drivers for online data acquisition

### 2. View version

This package cannot be used for online data acquisition. It can only import floppy files or evaluate projects, which are created with MCPS online packages.

If several view packages are handled by one network dongle, we call it a MCPS-NetView version.

The basic element of MCPS is the PROJECT. It describes the measurement, the channels and devices, colors, sample rate, tags, units and so on. Most of the operations in MCPS are project related.

So the first step is to load one of the example projects. Please select menu item *Project/Open*.

In the project directory you will find:

example.pro	Project without any math
example_math1.pro	Project with basic math (Loadable in demo mode or with option /Math1)
example_math2.pro	Project with extensive math (Loadable in demo mode or with option /Math2)

After loading the project's name is shown in the mainframe's titlebar. Since it's possible to have several projects in MCPS at the same time, you should always be aware, which one is the active one. Some windows like the device manager are not project related and no project name is visible. To make one project active, you have to click on any window, that belongs to that project.

To remove a project from MCPS call *Project/Close*.

Typical offline functions:

- Please select the menu *Display* and then *Numerical* or *Graphic*. You will see the measured data of the project.

The icon  or the F1 key will start the context help for more information.

- With *Project/Export/Channel* data you can export the measured data into an Ascii-file, which is readable by MicroSoft Excel.

- In the demo version or with option /MATH1 you can easily try the statistic functions with *Mathematics/Statistic*.

- To change settings of the project you have to call *Project/Configuration*.

## Running a Data Acquisition

Remark:

Even the demo version can receive data from a device. The only restriction is, that a project should have only one channel. This does not affect the settings of the instrument in the device manager. Even a system with 100 channels can be connected, but only one channel can be displayed.

Before a measurement can be started, you have to add the instrument to the device manager. There all settings are made concerning the type of instrument, interface and its configuration.

Please call *CONFIGURATION/DEVICE CONFIGURATION*.

If you have no device to connect, you can use the preinstalled TESTDRV instrument. It will deliver some demo values.

To add an instrument to the device manager please press



You will get a list of devices. If MCPS is running in demo mode, you will see all possible instruments. In end user mode (with dongle and password) you will see all devices, which drivers are enabled by the password.

Now select the device you want to add. For some instruments please see the chapter *Devices*. There are some additional device specific information listed.

For all systems you have to specify the type of interface. Please click on the button right to the text *Interface*. You will get a window to **SELECT THE RELATED INTERFACE**. If you are using serial interfaces (RS232/422A), you have to define the interface settings too (Configuration/Interface configuration).

**Important:**

The instrument should be configured in MCPS very exactly. This means to setup the right number of channels and options, otherwise this will cause transmission errors! For Yokogawa Darwin/DX/MV devices it's a good idea to use the AutoDetect button, which will setup the instrument's configuration automatically.

For a lot of devices the button *Test device* should be used, after all settings are made. This will test the basic connection to the instrument.

Now you can close all dialog windows by pressing the OK button and leave the device manager. The next step is to make a project with the new device. Therefore you have to call **PROJECT/NEW**.

You see the project configuration window with one channel. If your device is not listed, press the select button under *Device*. In the demo version you should only have one channel of an instrument, but you can add several channels with the TESTDRV device. The button +1 adds one channel to the bottom of the channel list.

Change the sample rate to 2 seconds and save the project.

After that call **DATA ACQUISITION/TEST**. MCPS will run a test measurement (sample rate is ignored) to check the transmission and to show the values in numerical form.

To run a normal data acquisition you have to select **DATA ACQUISITION/START**. A numerical and graphic window can be opened and the data is received and saved according to the sample rate of the project. After 10 seconds it's possible to see the data in the offline windows.

**DATA ACQUISITION/STOP** will stop the measurement.

## AutoFiles

Normally the measured data are save into one file. In case of long or permanent data acquisition the data file becomes very big. So MCPS can distribute the data in several so called AutoFiles. Several events can cause the creation of a new data file. There will be only one project file, which controls all data files. The AutoFiles can be used to make backup files, which are deleted after a while or copied to a server. Another application is to generate a new data file each time an alarm occurs. For example an external trigger can be used, if a new batch is coming in.

A project, that has been configured for AutoFiles and that contains data, cannot be set back to normal data acquisition (one file method).

The AutoFile parameters are set in the project configuration under [OPTIONS/AUTOFILE](#).

With [PROJECT/SELECT RECORD](#) you can select one of the already created data files. The currently active record is shown in the main title bar. All offline operations are now related to this record.

### Compare AutoFiles

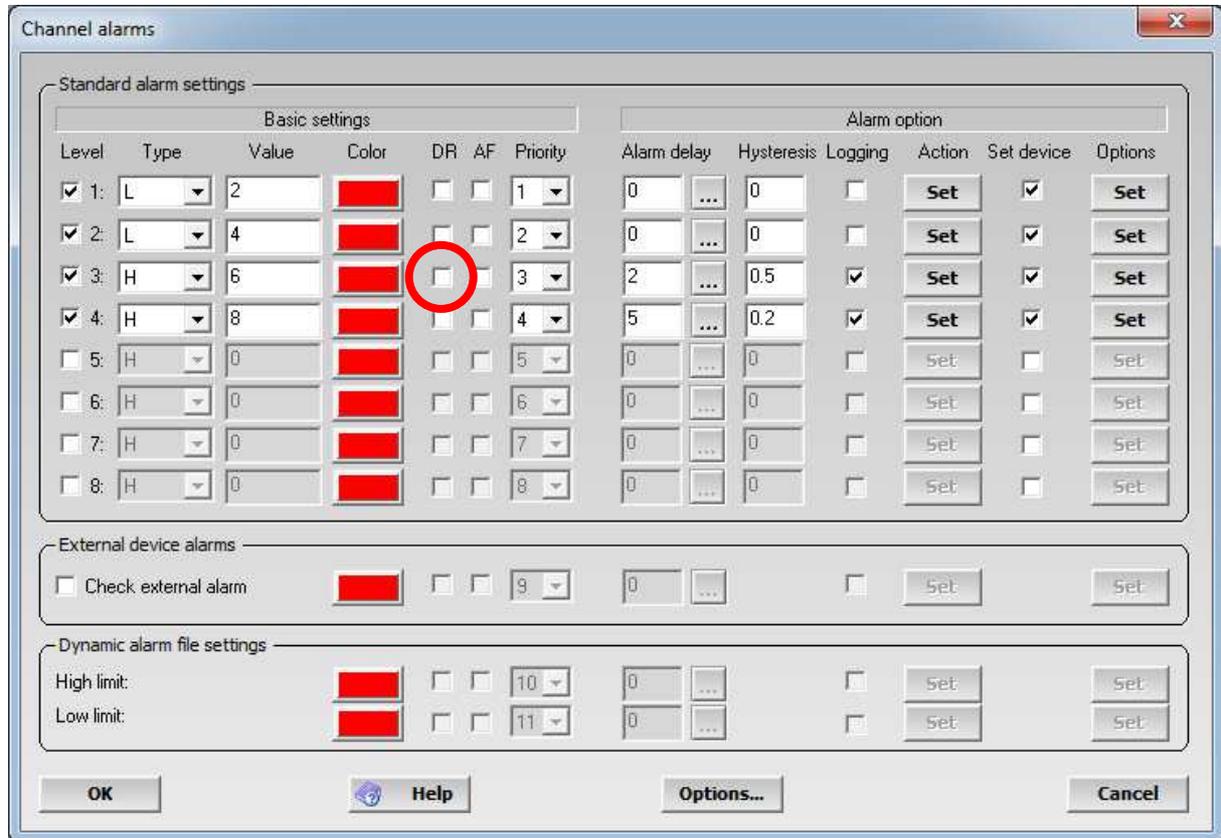
MCPS can compare AutoFiles. Several records are layered and displayed in relative mode. This is useful for repeated production processes, which should be equal. The user can immediately see, if the curves of all processes are similar or not. To do this call [DISPLAY/COMPARE/AUTOFILE RECORDS](#).

### Combine AutoFiles

MCPS can combine AutoFiles again to a new project with one data file. [SEE PROJECT/DATA](#).

# Data reduction

To reduce the amount of data stored to disk MCPS contains a dynamic data reduction function. In the project configuration under **OPTIONS/DATA REDUCTION** the general settings are made. MCPS can save data in time intervals other than the scan rate or in case of alarms. Therefore the button DR should be set in the the channel alarm configuration.



As long as the related alarm is active MCPS saves the data. A special case are the alarm types DHS and DHL, which are only meaningful in conjunction with the data reduction. DH and DL are referenced to the measured value one scan before, DHS and DHL are related to the last saved values.

Typical applications:

## 1. System surveillance

The data are gathered fast but are saved in only a great time interval. In case of alarm (something goes wrong) the data is saved every scan for later evaluation.

## 2. Production recording

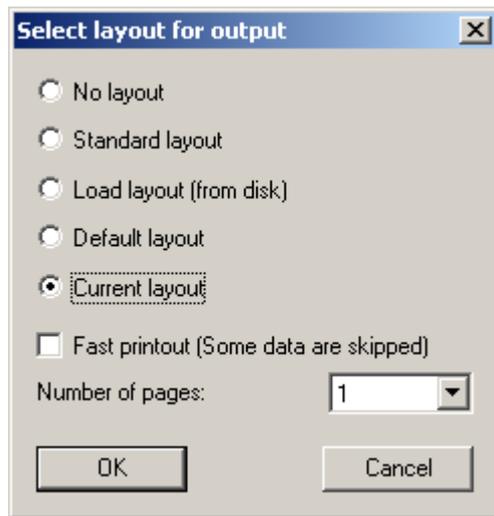
A project is measuring continuously, but the production system is not used all the time. Only if a new batch or product comes in the data is really needed. An external signal informs MCPS, if data is important or not. With an alarm set MCPS will only save data as long as the production system is in use.

## Printing With Layout

Normally if (numerical, graphic, statistics, export, ...) data are printed, the complete paper sheet is used. For better presentation MCPS can handle print layouts with predefined static texts, bitmaps, legends and so on. Use the **PRINT LAYOUT MANAGER** to define a layouts, which is saved to disk and can be used for every printout.

The layout contains a dynamic part, where the data are output (defined by the output frame) and a static part, where company logo, comments and other texts are printed.

Before a printout MCPS asks, howto print the data.



- Without layout

Printout without any layout over the whole page.

- Standard Layout

Uses a predefined layout.

- Layout laden

User can load a layout from disk.

- Default Layout

If you are using mostly the same layout for a printout, you can define default layouts in the project configuration (options/**DEFAULT SETTINGS**). So you do not need to load it everytime.

- Current layout

If you want to make a second printout, you can use the last layout again.

# Local Project Configuration

The network option of MCPS provides a feature called *Local Project Configuration*. Often the original projects and their data are only writable by the MCPS package, which is running the data acquisition. Other MCPS packages in the network often configured only for data evaluation (e.g. Netview) have normally no write access to the data directory. If a project is loaded by a client package, it can't be changed for example to set another color or scaling range. In this case (project is write protected) the user can save the project configuration to a local path, where he has write access. MCPS generates a link to the original data, so that it's not necessary to copy the whole project. The next time the project is loaded, MCPS asks to load the original or local one.

The local project configuration is enabled and the local path is set under [Mcps settings/Network](#). The local path can also be a directory on the server. The administrator can define different directories for each user and connect it to a common drive, which is entered in MCPS.

Example:

Server:

C:\USER1 -----> Connected to X:

C:\USER2 -----> Connected to X:

C:\USER3 -----> Connected to X:

Local path in MCPS:

X:

If user 1 logs into the network his X: drive will be connected to C:\USER1, user 2 to C:\USER2 and so on. The local path setting is equal for all user. This is necessary, if all user will start MCPS from the server. If MCPS is installed on each PC locally, each user can define it's own local path.

## Displaying online data and alarms on network clients

The option /MSERV can be used to send the online data values and alarms of a measurement PC to other MCPS packages connected to the network. These client computers, which only need a MCPS View-Version can then display the online data and alarms. It's possible to see the data from several measurement computers at the same time.

What to do:

First all computers have to be connected to a network. TCP/IP is recommended. The computer with the MCPS package for data acquisition needs the option /MSERV.

For the client computers the following settings are necessary:

- **CONFIGURATION/MCPS SETTINGS/PASSWORDS:** *Enable MSERV-Client features* should be set to on. Restart MCPS.
- **CONFIGURATION/MCPS SETTINGS/MSERV:** Enter the daq computer name or its ip address

Start a measurement on the DAQ computer. On the client you have to call **PROJECT/OPEN FROM DA SERVER.**

After the connection has been established, you should see all online projects of the measurement computer(s). Select one to load. It will be marked with the ~ character. Afterwards open some online windows (Data acquisition/text window) to see the data.

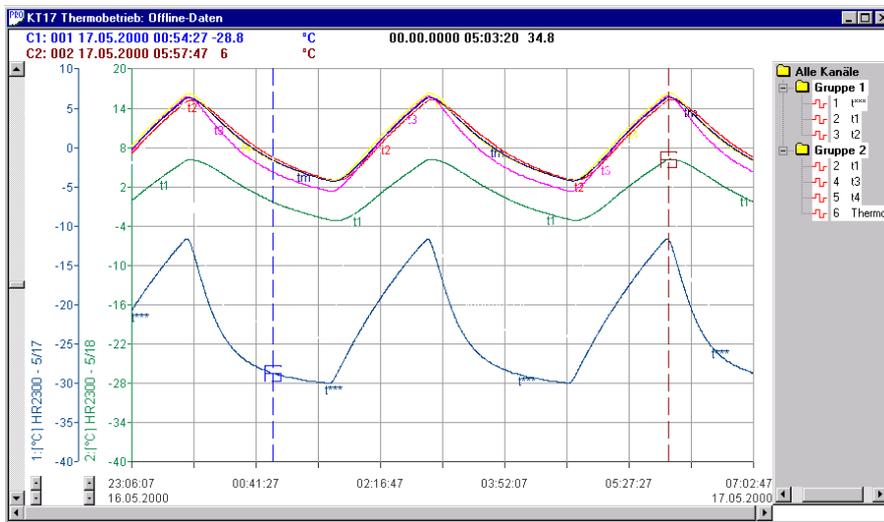
If the project is stopped or the configuration is modified at the DAQ computer, it is automatically closed on all clients!

The alarm window shows the alarm of all connected servers. It's useful to set the button *Open alarm window automatically* in **CONFIGURATION/MCPS SETTINGS/GENERAL.** The alarm window will be opened, if an alarm is available.

# Dynamic Alarm File

The dynamic alarm file is a very powerful but also complex feature. It is used to check repeated measurements for example in production processes. If always the same product is created or tested, the measured data should be the same for each measurement. So it's useful to have a reference file with alarm limits, which can follow even a dynamic process.

The dynamic alarm file is used in MCPS with AutoFiles. First at least one AutoFile records should be available to create an alarm reference file. Since the trigger point is important to compare the measured data with the reference data, a new AutoFile record should be created with an alarm depending on an external event. I. e. if the new process or test sequence is started, a signal should be set, so that MCPS creates a new AutoFile record. This guarantees, that there is no time offset between new and reference data.



To create a dynamic alarm file you have to call **PROJECT/EXTRAS/DYNAMIC ALARM FILE/NEW**. Afterwards the project should be informed to use an alarm file. In the project configuration under **OPTIONS/ALARM SETTINGS** you must activate the button *Use alarm file* and select one. The next time the data acquisition is started, MCPS will load the alarm file and check the alarm limits. If a new record is created, the alarm file is reset. The **CHANNEL PARAMETERS** in the project configuration can define colors or actions as for regular alarms. It's even possible to combine all alarm levels.

If an alarm file is activated in the project configuration, the deviation curves can also be shown in graphic windows. The menu item **OPTIONS/EXTRAS/SHOW DYNAMIC ALARM FILE LIMITS** is activated and can be used for offline and online windows. If editing an alarm file (**PROJECT/EXTRAS/DYNAMIC ALARM FILE/EDIT**) this feature is useful to see the changes immediately.

For the online trend window the alarm limits are also partially shown for the next scans as a preview.

## Tips and tricks

### Create a new project with already made settings

Often you have configured a project very carefully and made your data acquisition. For a new measurement you need again the same or a similar project. Then you should easily copy the old one without any measured data. See [PROJECT/DATA/COPY PROJECT](#).

You shouldn't copy a project by the windows explorer, because there are other important file related to the project and a copied project is not exactly the same as the old one. It will get a new internal identifier.

### Import by using an already existing project

When importing files MCPS creates a new standard project. Often this is done with data from the same device with the same configuration. Before import MCPS asks you for the type of import. If **Default-Import** is selected, MCPS will copy the settings of an already existing project (for example another imported one) into the new one. This saves a lot of time. See [PROJECT/IMPORT/FILE](#).

### Combining several projects

MCPS can combine several projects with the same number of channels to a project with only one big data file. There should be no time overlap in the time stamps of the data. This feature is useful for instruments, which generate a lot of files to floppy or ZIP-drives. See [PROJECT/DATA/COMBINE PROJECTS](#).

### Quit MCPS and Restart With Measuring Projects

With [PROJECT/SMART QUIT](#) you can quit MCPS even a measurement is still active. The next time MCPS is started, all measuring projects are reloaded and start again.

### Number of Data Scans in Online Graphic Windows

After scaling or resizing the online trend window MCPS has to redraw the data. The values are not loaded from disk again, but from an internal buffer. Because the buffer is limited in size, maybe not all of the values are shown after a redraw. This happens particularly then, if the x-axis is scaled to show a great time window. The size of the redraw buffer can be changed in [CONFIGURATION/MCPS SETTINGS/GENERAL/PROJECT ONLINE HISTORY](#).

### Project Configuration With Better Channel Identification

If you change one page right in the project configuration, only the consecutive number is hold to identify the channel. To have the device or even the tag on all configuration pages see [CONFIGURATION/MCPS SETTINGS/GENERAL](#):

Show first 0 columns all the time.

## Fast Display

MCPS provides a fast display mode for graphic offline windows. In case of big files not all data values are shown. MCPS will skip some data to give a general view. The fast display can be activated in the graphic window under [OPTIONS/DISPLAY/FAST DISPLAY](#). The setting can be preset for all graphic windows in [CONFIGURATION/MCPS SETTINGS/GRAPHIC](#).

## Stop Drawing

By opening a graphic offline window all data of the file is drawn. In case of big files this can take some time. The following icon will stop the drawing:



## Autostart

After a powerfail MCPS can automatically restart the measurement projects, when it is started again. The user should take care, that MCPS is started after system reboot. See autostart directory of your Windows system. Computers with network login should use an autologin feature. Contact your network administrator.

The autostart feature can be disabled in [Configuration/MCPS Settings/General](#).

# Webserver

(Only with option /WEB)

MCPS provides an integrated Webserver, which can generate custom specific pages. There are no static pages, all pages can be created or modified by customer (or designed by CAD Computer). To create HTML pages you need at least a text editor or a more professional program like Frontpage or Web expressions. There is no editor inside MCPS. You can create a page with all features of this external program with Bitmaps, texts, tables layers and so on. To display MCPS data like values or trends you have to implement special commands or placeholders. When a browser wants to download a page, MCPS replaces the command strings or placeholders by real data or images.

## 1. Text information:

Text information like channel values or units are replaced directly by control strings as for example `$CV(2;TD)`. MCPS will replace this code by measured value of channel 2 of project td. We recommend to use tables to handle text replacement due to variable length of text outputs.

Command list:

**\$CV(Channel;Project)** ChannelValue Current measured value of <Channel> of <Project>. If only channel is defined, MCPS will take the first started project.

**\$CU(Channel;Project)** ChannelUnit Unit of <Channel> of <Project>

**\$CT(Channel;Project)** ChannelTag Tag of <Channel> of <Project>

**\$CC(Channel;Project)** ChannelComment Comment of <Channel> of <Project>

**\$TC(Project)** TimeCounter Time counter of a running project or batch (hh:mm:ss)

**\$DATE** Date Current date

**\$TIME** Time Current time

## 2. Images (Trends, LEDs, Bars, Analog meters):

To display a dynamic image you have to place a placeholder bitmap with desired width and height in your HTML page. You can use any graphic you want, but MCPS provides a placeholder for each type of graphic in the subdirectory <WebServer> which is located by default in the working directory of MCPS. For each bitmap you can define an alternate text, which is displayed in a browser when an image could not be loaded. In Frontpage and other programs you can find this alternate text in the properties of the bitmaps. A specific control string as alternate text is detected by MCPS, a related image is created and inserted instead of the placeholder bitmap.

Command list:

`$LED;Project;AChannel;Level;RGB color`

Alarm LED of <Project> and <Channel> and <Level>. The alarm color is used from the channel alarm settings of the project configuration. <RGB color> is the non alarm color. If <Level> is 0, then all alarm levels are checked and that one with the highest priority is used.

`$LED;BATCH2;A51,3,008000` → Alarm LED of project BATCH2, channel51, level3. Standard color is dark green.

`$LED;Digital output;D; Off color; Off type; On color; On type`

LED, which represents the status of a digital output as set in MCPS. If an output is set by an alarm action, PMON or script, the related status can be made visible. Off/On color defines the color of the LED when inactive/active. Type can be 0 (Circle) or 1 (rectangle).

`$LED;W750 - 2: DO1;D;00FF00;1;FF0000;0`

Output <W750 – 2: DO1” with rectangle-LED in green when off and circle-LED in blue when on.

`$ANALOG;Project;Channel;Min;Max;Flags;Options`

Analogmeter with value of <Project> and <Channel>. Span range is from <Min> to <Max>. If <Min> is equal to <Max> the project settings of the channel (YMIN.YMAX) are used.

Cumulative values of <Flags>:

1 = Do not display alarm ranges

2 = Logarithmic display

<Options> are 0

`$ANALOG;P1;2;7;9;0;` → Project P1 with channel 2 from 7 to 9

`$BAR;Project;Channel;Min;Max;Bar color; Background color;Flags;Options`

Bar graph with value of <Project> and <Channel>. Span range is from <Min> to <Max>. If <Min> is equal to <Max> the project settings of the channel (YMIN.YMAX) are used. <Bar color> is the color of the bar itself, <Background color> is the color of the whole element.

Cumulative Hex-values of <Flags>:

01 = No title

02 = No value in text form

04 = No frame

08 = No alarm limits

10 = Transparent

20 = Logarithmic display

40 = Draw bar from zero and not from bottom.

<Options> are 0

\$BAR;;1;0;10;00FF00;CACACA;0F,0 → Channel 1 with range from 0 to 10, green bar and light gray background without title, text value, frame and alarm limits.

\$TREND;Project;Time span for x-axis;Channels;Options

Trend display of <Project>. <Time span> defines the time difference between start and end time of the trend display. <Channels> is list of channels in several split screens, which are separated by character # (1-5#7,9#23-28).

Cumulative Hex-values of <Options>:

1 = Activate autoscaling for x-axis

\$TREND;TEST;90;1-4#5-7;1

Important

MCPS will create Java scripts for dynamic values, which are implemented in the HTML page. The browser must support Java to update text and images.

Dynamic pages:

Often it is necessary to create the whole HTML page at runtime to handle different projects or variable number of channels. These pages are created with scripts, which can gather information from MCPS about the current situation. Such dynamic pages are called ActiveServerPages and are supported by MCPS. Please contact CAD Computer to get more information...

# Scripts

## Concept of scripts

Scripts are small programs written in VisualBasic-Script to customize MCPS. Scripts can be used for offline evaluation or online data transfer to Excel. To run scripts you need the VB script engine from MicroSoft. Normally this is installed at least by the Internet Explorer or other MicroSoft components. If not, it is available on the MCPS installation CD/Special/Microsoft. There you can also find the VB script documentation.

MCPS provides a lot of function to get information, which can be used for evaluation or other programs like Excel or Word. These programs have a standard automation interface, which means, that they can also be accessed by the script program. MCPS related functions consists of the namespace MCPS.xxx and the function xxx.

A script can be started manually by calling **EXTRAS/RUN SCRIPT** (see also **USER TOOLBAR**) or automatically related to certain actions (loading or starting a project, with every scan, ...). Automatic scripts must be defined in the project configuration (see **OPTIONS/AUTOMATION/SCRIPTS**).

It is possible to use encrypted scripts. They cannot be modified by any user. The encryption is done by CAD Computer.

All Parameterss have VisualBasic compatible types. The following abbreviations are used:

int	Fixed numbers or integers (123)
str	Strings ("This is a text")
float	Floating point values (1.23)
date	VB date and time format
array<type>	Field with elements of type
variant	Any type of VB type

## The first script

To write a script you can use a simple text editor like notepad. Change the file extension to .mbs and start the script by calling **EXTRAS/RUN SCRIPT**.

Example:

```
MsgBox MCPS.GetActiveProjectName
```

This will show the name of the active project in MCPS or nothing if there is no project loaded or a nonproject window is active.

The MCPS setup program installs some example scripts in the default script directory.

## Writing offline scripts

An offline script is reading the saved data scan by scan to make customer specific calculation and evaluation. Often the results are transferred to Excel as a report or presentation.

```
Dim objXL
Set objXL = GetObject("Excel.Application") 'Excel must be started
handle = MCPS.OfflineStartRead("") ' In case of a specific AutoFile you need to enter its name
"00001"
if handle then
for x = 1 to 1000
objXL.Cells(x+1,1).Value = MCPS.OfflineGetTime
for n = 1 to MCPS.GetProjectChannelCount
objXL.Cells(x+1,n+1).Value = MCPS.OfflineGetData(n)
next
if MCPS.OfflineNextBlock(handle) = 0 then
exit for ' no more scans
end if
next
MCPS.OfflineEndRead(handle)
end if
```

This example will transfer 1000 scans of the data file to Excel

## Writing online scripts

Online scripts are defined in the project configuration, [OPTIONS/AUTOMATION/SCRIPTS/DAQ SCAN](#). The online script should be held small and fast and is not allowed to show any message box or to wait for any user action, because it should be run with (nearly) every scan.

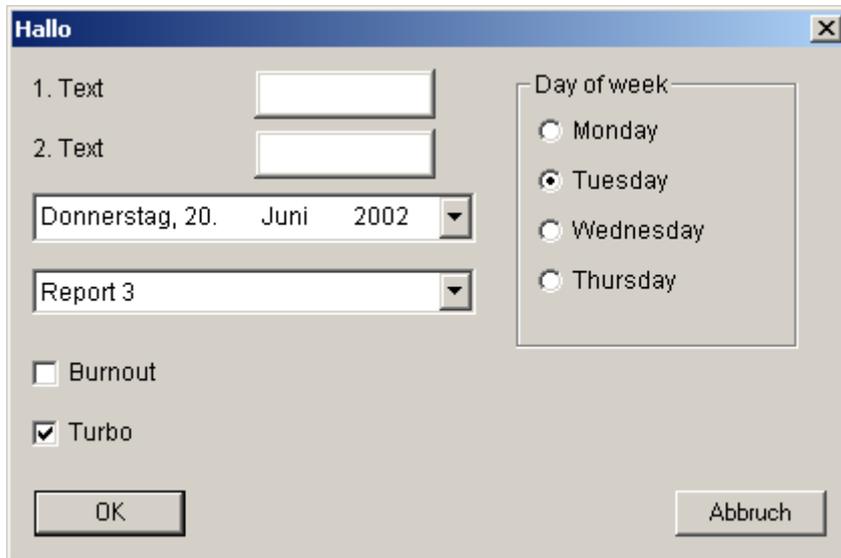
```
Dim objXL
Set objXL = GetObject(,"Excel.Application") 'Excel must be running

field = MCPS.GetOnlineDataField()
for n = 1 to MCPS.GetProjectChannelCount
  objXL.Cells(n,2).Value = field(n-1) ' zero based index in VB script
next
```

With every scan the online values are displayed in Excel. With a complex sheet you can even make a process visualization.

## Request window

For offline scripts and reports it's often necessary to enter some information, select between different reports or define a time span. For those cases MCPS provides a request window, where texts, input fields, radio and check buttons or combo boxes can be defined by the script.



First you have to create the window and you will get a reference handle. For each item (button, field, box, ...) you add, you will get also a specific identifier. For further operations the window handle and the item identifier are always necessary. After all items are placed, the window is activated. The script is waiting until the user presses OK or CANCEL. The settings of each item can now be read out and after all the window is deleted.

```
h = 25
```

```
Dim choice(5)
```

```
dialoghandle = MCPS.DialogCreate("Hallo",420,280,0) 'Creates the dialog with width 420 and height 280
```

```
if handle >= 0 then
```

```
  t1 = MCPS.DialogAddText(dialoghandle,"1. Text",10,12,110,h,0) 'Adds a static text x1,y1,width,height
```

```
  t2 = MCPS.DialogAddText(dialoghandle,"2. Text",10,42,110,h,0)
```

```
  e1 = MCPS.DialogAddEdit(dialoghandle,120,10,90,h,0) ' Text input field
```

```
  e2 = MCPS.DialogAddEdit(dialoghandle,120,40,90,h,0)
```

```
  d1 = MCPS.DialogAddDateTime(dialoghandle,Date,10,72,220,h,1)
```

```
  s1 = MCPS.DialogAddSelection(dialoghandle,10,110,220,h,0) ' Combo box
```

```
  MCPS.DialogAddSelectionString dialoghandle,s1,"Report 1" ' Adds the items of the box
```

```
  MCPS.DialogAddSelectionString dialoghandle,s1,"Report 2"
```

```
  MCPS.DialogAddSelectionString dialoghandle,s1,"Report 3"
```

```
  MCPS.DialogAddSelectionString dialoghandle,s1,"Report 4"
```

```
  MCPS.DialogAddSelectionString dialoghandle,s1,"Report 5"
```

```
  MCPS.DialogSetSelection dialoghandle,s1,3
```

```
  c1 = MCPS.DialogAddCheckBox(dialoghandle,10,150,100,h,"Burnout",0)
```

```
  c2 = MCPS.DialogAddCheckBox(dialoghandle,10,180,100,h,"Turbo",1)
```

```
  x2 = 250
```

```
  y2 = 10
```

```
  g1 = MCPS.DialogAddGroupBox(dialoghandle,x2,y2,140,140,"Day of week",0)
```

```
  x2 = x2+10
```

```
  y2 = y2+20
```

```
  r1 = MCPS.DialogAddRadioButton(dialoghandle,x2,y2,100,h-2,"Monday",2)
```

```
  y2 = y2+h
```

```
  r2 = MCPS.DialogAddRadioButton(dialoghandle,x2,y2,100,h-2,"Tuesday",1)
```

```
y2 = y2+h
r3 = MCPS.DialogAddRadioButton(dialoghandle,x2,y2,100,h-2,"Wednesday",0)
y2 = y2+h
r4 = MCPS.DialogAddRadioButton(dialoghandle,x2,y2,100,h-2,"Thursday",0)

status = MCPS.DialogRun(dialoghandle) 'Show window and wait until OK or CANCEL

if status = 1 then 'OK Button has been pressed
  MsgBox MCPS.DialogGetRadioStatus(dialoghandle,r1)
  MsgBox MCPS.DialogGetEditText(dialoghandle,e1)
  MsgBox MCPS.DialogGetSelection(dialoghandle,s1)
end if
MCPS.DialogRemove(dialoghandle)
end if
```

## System script commands

### GetScriptType

Description:	Source of script call
Return value:	Source of call (int)
0	= Project loaded
1	= Project closed
2	= DAQ started
3	= DAQ scan
4	= DAQ stopped
5	= AutoFile completed
6	= DAQ preparation (Immediately called after start of DAQ)
7	= Creating a batchfile by normal AutoFile function (not session)
8	= Scheduler
9	= Before project creation
10	= Event in script controlled status window (button pressed)
11	= After the project configuration has been saved
12	= When the script button in the project configuration is pressed
13	= Before the project configuration is saved
238	= Script was called by another script
239	= Script was called from client via MSERV
240	= Script at the end of a dialog window
241	= OPC
242	= Process monitor
243	= Webserver
244	= Batch automatic import
245	= ECS
246	= Event by OpenRequestWindow2
247	= Global scheduler (MCPS SETTINGS/GENERAL/SCHEDULERS)
248	= Start of MCPS (MCPS SETTINGS/GENERAL)
249	= After removing a waiting batch
250	= Before opening the batch input mask
251	= After closing the batch input mask
252	= Batch start
253	= Batch stop
254	= Alarm
255	= Manually started
Parameters:	None

### GetScriptType2

Description:	Additional information to type of script call
Return value:	Source of call in case of pre print script (int)
1	= Numerical window
2	= Offline trend window
3	= Online trend window
4	= Statistics window
5	= Alarm window
6	= Event window

#### Automatic printouts:

7	= Offline trend
8	= Online trend
9	= Statistics
10	= Alarms
11	= Events

Parameters: None

**SetReturnValue**

Description: Sets a return value, which is in some cases evaluated after the script is called. A value of 1 means to abort further actions in MCPS, when script type 2, 6, or 9 is called.

Return value: None

Parameters: Return value (int)

**GetSystemDateChar**

Return value: Delimiter in Windows for date strings (str)

Parameters: None

**GetSystemTimeChar**

Return value: Delimiter in Windows for time strings (str)

Parameters: None

**GetSystemDecimalChar**

Return value: Decimal character as defined in Windows for values (str)

Parameters: None

**GetSystemDateType**

Return value: Windows date format type (int)

0 = Month, Day, Year

1 = Day, Month, Year

2 = Year, Month, Day

Parameters: None

**GetLoginName**

Return value: Returns the name of the current login user

Parameters: None

**GetLoginID**

Return value: Returns the id of the current login user

Parameters: None

**GetExeDirectory**

Return value: Path to MCPS directory (str)

Parameters: None

**GetProjectDirectory**

Return value: Path to current project default directory (str)

Parameters: None

**GetTemplateDirectory**

Return value: Path to current template default directory (str)

Parameters: None

**GetExportDirectory**

Return value: Path to current export default directory (str)

Parameters: None

**GetSetupDirectory**

Return value: Path to current setup default directory (str)

Parameters: None

**GetLayoutDirectory**

Return value: Path to current layout default directory (str)

Parameters: None

**GetMathDirectory**

Return value: Path to current math files default directory (str)

Parameters: None

### **GetImportDirectory**

Return value: Path to current import default directory (str)  
Parameters: None

### **GetBatchSessionDirectory**

Return value: Path to current batch session default directory (str)  
Parameters: None

### **GetBatchDatabaseDirectory**

Return value: Path to current database default directory (str)  
Parameters: None

### **GetExternalStorageDirectory**

Return value: Path to current default directory for external storage (str)  
Parameters: None

### **GetSystemMessage**

Return value: Returns an index based message from the system message window (str)  
Parameters: Index (int), Options (int)  
Index: 1 = First entry, 2 = second entry, ...  
Index: -1 = Latest entry, -2 = penultimate entry, ...  
Options = 0

### **SetGlobalVariable**

Description: Sets a project independent global variable  
Return value: None  
Parameters: Name of variable (str), Value (variant)

### **GetGlobalVariable**

Description: Returns the global variable set by SetGlobalVariable  
Return value: Value (variant)  
Parameters: Name of variable (str)

### **IsAutoStartActive**

Description: Status (int) = 1, when the script is called during the autostart procedure  
Parameters: None

### **RunScript**

Description: Starts a script at a specific time. For example to set an output or data reduction 10 minutes after start of data acquisition you just call a delayed script.

Return value: Status (int) = 1

Parameters: Script file (str), execution time (DATE), additional information (str), Options (int) = 0

The delayed script has access to the <additional info string> by calling <GetScriptData>

### **RunServerScript**

Description: Starts a script on the server (measurement computer) initiated by a client

Return value: Status (int) = 1, if connection is OK

Parameters: Connection index (int), script file (str), additional information (str), Options (int) = 0

If connection index is -1, then MCPS uses the connection of the related project. Otherwise the zero based index as defined in the MServ tab of MCPS settings must be used.

### **GetClientIP**

Description: Scripts, which are started due to external actions (ECS or Webserver) can get the ip address of the remote client to handle different answers. E.g. client specific HTML pages.

Return value: IP address (str)  
Parameters: None

### **SelectFile**

Description: A file request window is opened to select a filename for loading or saving  
Return value: Filename (str): "" = Cancel  
Parameters: Directory (str), Description of files (str), File extension mask (str), Options (int)  
Options: 0 = Load text, 1 = save text  
Example: fn = MCPS.SelectFile("C:\WINNT","Text files","txt",1)

### **Global variables**

Global identifiers are stored in MCPS even after stop of the script. The identifier are available to store conditions until the next script start. This allows you to write sequencer scripts with several steps.

MCPS.I1 to MCPS.I10 (Int)

**MCPS.D1 to MCPS.D10 (Double)**

## FTP script commands

### FTPOpen

Description: Opens a connection to a FTP server for uploading files.

Return value: Handle for further operation (int); 0 = error

Parameters: Server name or IP (str), Server port (int), Login (str), Password (str), Optionsstring (str) = ""

Example: `h = FTPOpen("ftp.mcps.de",21,"Gunther","12345","")`

### FTPClose

Return value: None

Parameters: FTP handle (int)

### FTPUpload

Description: Sends a file to the FTP server

Return value: Status (int); 1 = OK

Parameters: FTP handle (int), File name (str), File name on FTP server (str), Optionsstring (str) = ""

Example: `status = FTPUpload(h,"c:\test.txt","test.txt","")`

## Batch script commands

The following batch information are available, when a batch is started or stopped in the batchmanager.

### **BatchGetMaskName**

Return value: Name of the input mask of the batch (str)  
Parameters: None

### **BatchGetName**

Return value: Batchname (str)  
Parameters: None

### **BatchGetFileName**

Return value: Complete batch filename (str)  
Parameters: None

### **BatchGetInfo**

Description: Additional infos of the batch input mask.  
Return value: Info identifier or Info data (str)  
Parameters: Info index (int), ID(1) or data(2)  
Example: sn = MCPS.BatchGetInfo(1,2)

### **BatchGetDXInfo**

Return value: Comment name or user input  
Parameters: Index (1-3), Type 1 or 2  
Example: c1 = MCPS.BatchGetDXInfo(1,2)

### **BatchSetBatch**

Return value: None  
Parameters: Batch (str)

### **BatchSetName**

Return value: None  
Parameters: Name of batch (str)

### **BatchSetFileName**

Return value: None  
Parameters: Name of batch filename(str)

### **BatchSetInfo**

Description: Sets the batch information  
Return value: Status OK =1 (int)  
Parameters: InfoIndex (int), InfoID(1)/Data(2), Text (str)  
Example: MCPS.BatchSetInfo(1,1,"Material")  
MCPS.BatchSetInfo(1,2,"Iron")

### **BatchGetUnitID**

Return value: Number of unit (int)  
Parameters: None  
Example: unit = MCPS.BatchGetUnitID()

### **BatchGetUnitName**

Return value: Name of unit (str)  
Parameters: None

### **BatchGetLotNumber**

Return value: Lot number (str)  
Parameters: None

**BatchGetBatch**

Return value: Base batch (str)  
Parameters: None

**BatchGetStartTime**

Description: Returns the start time of the batch  
Return value: Start time (date)  
Parameters: None

**BatchGetEndTime**

Description: Returns the end time of the batch  
Return value: End time (date)  
Parameters: None

**BatchSetUnitAction**

Description: Sets specific batch actions or operations  
Return value: Status (int): 1=OK  
Parameters: Command (int), Options (int) = 0  
Command: 1 = Ignore batch after stop, do not create batchfile

**BatchSetUnitText**

Description: Sets additional text for unit window  
Return value: Status (int): 1=OK  
Parameters: Text (str), Options (int)  
Options:  
1 = Add text to the latest batch  
2 = Insert text before other batch information  
4 = Show text, if no batch is available

Commands available during a session:

**BatchGetUnitProject**

Description: Returns the complete project name including path  
Return value: Project name (str)  
Parameters: Unit (int)

**BatchStart**

Description: Starts a batch  
Return value: None  
Parameters: Unit (int), Options (int) = 0

**BatchStop**

Description: Stops a batch  
Return value: None  
Parameters: Unit (int), Options (int) = 0

**BatchGetUnitBatchName**

Description: Returns the batch name of the unit  
Return value: Batch name (str)  
Parameters: Unit (int), Options (int) = 0

**BatchGetUnitBatchFileName**

Return value: Batch filename of unit (str)  
Parameters: Unit (int), Options (int) = 0

**BatchGetUnitLotNumber**

Return value: Lot number of unit (str)  
Parameters: Unit (int), Options (int) = 0

**BatchGetUnitInfo**

Return value: Batch information of unit (str)

Parameters: Unit (int), Index of batch information (int), Type (int), Options (int) = 0  
Type: 1= Identifier, 2 = User input

**BatchGetUnitStatus**

Description: Returns status of one or two batches in the unit. In case of two batches the return values are separated by <;>.

Return value: Status (str): 0=empty, 1=Waiting; 2 = running; 3 = stopped

Parameters: Unit (int), Options (int) = 0

Batchdatabase operations:

**BDBChangeBatchName**

Description: Changes the batchname in a database, the related batch filename and the internal link to the batchfile.

Return value: Status (int) 1 = OK

Parameters: Database name (str), old batch name (str), new batch name(str), options (int)

Example: BDBChangeBatchName("C:\test.bdb","POWERBOOK13","POWERBOOK15",0)

**BDBOpen**

Description: Opens a batch database

Return value: Status (int) 1 = OK

Parameters: Database name (str), options = 0 (int)

Example: MCPS.BDBOpen("C:\test.bdb",0)

## Additional script commands

### GetProjectField

Description: Returns a field with all loaded or running projects  
Return value: Field (VARIANT)  
Parameters: Options (int): 0 = all projects; 1 = running projects; 2 = MServ projects  
example: field = MCPS.GetProjectField(0)  
msgbox field(0)

### LoadProject

Return value: Status OK =1 (int)  
Parameters: Project name (str)  
Example: MCPS.LoadProject("C:\projects\test.pro")

### CloseProject

Description: Closes the project  
Return value: Status OK =1 (int)  
Parameter: Complete filename of project (str)  
Example: status = MCPS.CloseProject("C:\projects\test.pro")

### SetDataReduction

Description: Controls the data reduction  
Return value: Status OK =1 (int)  
Parameters: Mode (int), save interval (int), control Parameterss (int), minimum alarms (int),  
Options (int)  
Mode:  
1 = Turn off data reduction (Normal saving)  
2 = Turn off saving( Full reduction)  
3 = Controlled reduction  
Save interval:  
Value in seconds for saving in regular intervals  
Control Parameterss:  
1 = Interval saving  
2 = Mean value saving (Only if also interval saving is active)  
4 = Save in case of alarms  
8 = Save always first scan  
The values can be added  
Minimum alarms:  
Number of active alarms before storing data, when saving on alarm is enabled  
Options:  
0  
Example: status = MCPS.SetDataReduction (3,60,11,0,0)  
Controlled data reduction, 1 minute mean value saving, save 1. scan

### GetProjectChannelCount

Return value: Number of project channels (int)  
Parameters: None  
Example: count = MCPS.GetProjectChannelCount

### GetActiveProjectName

Return value: Name of project (str)  
Parameters: None  
Example: MsgBox MCPS.GetActiveProjectName

### GetActiveProjectPath

Return value: Complete path of project (str)  
Parameters: None  
Example: MsgBox MCPS.GetActiveProjectPath

### **SetActiveScriptProject**

Description: During a script all functions are referenced to the calling project. If the script is not called by a project, then the active project in MCPS is used. With this function the script gains access to a specific project for example to set some parameters. This redirection is only valid in this one script.

Return value: Status (int) 1=OK

Parameters: Complete name of project (str)

### **ProjectDataAvailable**

Description: Checks, if the project contains any data in any file.

Return value: Data available=1 (int)

Parameters: None

Example: if MCPS.ProjectDataAvailable then

### **GetProjectInfoLine**

Description: Returns a text line of the project's info window

Return value: Text (str)

Parameters: Line (int)

### **GetProjectAdditionalInfo**

Description: Returns a text line of the project's additional info list, which is created during imports or other by other functions and can be seen by maximizing the base window of project.

Return value: Text (str)

Parameters: Line (int)

### **GetProjectOwnerField**

Description: Returns a field with all owners and owner groups of a project

Return value: Owners (Array of str)

Parameters: Options (int): 1 = Return no users; 2 = Return no groups

### **GetProjectDataBase**

Description: Returns the database, which is used for conversion from AutoFiles to batchfiles

Return value: Database (str)

Parameters: None

### **AFLastCompletedFile**

Return value: Name of the last finished AutoFile

Parameters: None

### **GMGetActiveChannels**

Description: Returns a field of active group channels. For each group the channel numbers are returned. The end of a group is marked as -1. Then the channels of the next group are following.

Return value: Field with channel numbers (VARIANT)

Parameters: Options (0)

### **GMGetRootGroup**

Description: Returns a handle to the main group of the group manager, which can be used for further operations.

Return value: Handle (int); 0 in case of error

Parameters: None

Example: h = MCPS.GMGetRootGroup()

### **GMGetRootGroup**

Description: Returns a handle to the main group of the group manager, which can be used for further operations.

Return value: Handle (int); 0 in case of error

Parameters: None

Example: h = MCPS.GMGetRootGroup()

### **GMGetChildGroup**

Description: Returns a handle of the child group of the given handle.  
Return value: Handle (int); 0 = no childs  
Parameters: Handle (int)

### **GMGetNextGroup**

Description: Returns a handle of the next sibling group of the given handle.  
Return value: Handle (int); 0 = no further group  
Parameters: Handle (int)

### **GMGetGroupChannels**

Description: Returns an array of all channels linked to the given handle.  
Return value: Channel field (Variant array of integers)  
Parameters: Handle (int), Options (int) = 0

### **GMGetGroupName**

Description: Returns an array of all channels linked to the given handle.  
Return value: Channel field (Variant array of integers)  
Parameters: Handle (int), Options (int) = 0

### **SetProjectVariable**

Description: Defines or sets a variable for a project. Each variable gets a user defined name and can hold any type of VB script data. Since the lifetime of the variable is project dependent and not script related, the variable can be used in several scripts.  
Return value: None  
Parameters: Name (str), Data (any type)  
Example: MCPS.SetProjectVariable "cycles", 5

### **SetProjectVariableEx**

Description: Similar to SetProjectVariable, but with additional Parameters  
Return value: None  
Parameters: Name of variable (str), Parameters (variant), Options (int)  
Example: MCPS.SetProjectVariable "Material", "Iron", 1  
Options:  
1: The variable will be saved with the project the next time the project configuration is saved. When the project is loaded again, the variable and its content is available again.  
2: Defines, that the variable should be available during project configuration. At the moment used, when a new project is created (even from a template).

### **GetProjectVariable**

Return value: Value of a project variable.  
Parameters: Name of variable(str)  
Example: for i = 1 to MCPS.GetProjectVariable("cycles")

### **RemoveAllProjectVariables**

Description: Removes all project variables  
Return value: None  
Parameters: None

### **SetProjectInfoLine**

Description: Writes a text into the project info lines.  
Return value: None  
Parameters: Line (int), Text (str),  
Example: MCPS. SetProjectInfoLine 1, "Serial no.: AXC-22"

### **SetProjectAdditionalInfo**

Description: Writes a text into the additional project info lines. These lines are visible, when maximizing the project's base window. In addition these lines are also available in the print layout by specific control codes. See [PRINT LAYOUT MANAGER](#).

Return value: None

Parameters: Line (int), Text (str),

### **GetTripLineValue**

Return value: Value (float)

Parameters: Number of trip line (int)

### **SetTripLine**

Description: Sets the Parameterss of a trip line (must be defined in the project configuration).

Return value: Status (int): OK = 1

Parameters: Number of trip line (int), Active (int), Value (float), Options (int) = 0

### **UCSetTrendDisplay**

Description: Sets the display Parameterss of a user curve

Return value: Status (int): OK = 1

Parameters: Number of user curve (int), Active (int), Optionen (int) = 0

Active: 1 = Curve is visible

### **SetOnlineDefaultTimeSpan**

Description: Sets the default x-axis time span for online trend windows

Return value: Status (int): OK = 1

Parameters: Time range (int), Time unit (int), Options (int)

Time range: Value of time range

Unit: Unit of time (0=seconds, 1 = minutes, 2 = hours)

Options: 1 = Update open online windows

### **SetOfflineDefaultTimeSpan**

Description: Sets the default x-axis time span for offline windows

Return value: Status (int): OK = 1

Parameters: Time range (int), Time unit (int), Options (int)

Time range: Value of time range

Unit: Unit of time (0=seconds, 1 = minutes, 2 = hours)

Options: 0 = Time range from start of measurement

1 = Time range from end of measurement

**SaveProjectConfiguration**

Description: Saves the project configuration. Cannot be used, when the project configuration window is open.

Return value: None

Parameters: None

**SaveProjectConfigurationEx**

Description: Saves the project configuration with a different filename (Makes a copy).

Return value: Status (int): OK = 1

Parameters: Filename (str), Options (int) = 0

**PCAddChannels**

Description: Adds one or more channels to the project configuration. This command is only executed in a script, which has been started inside the project configuration. In the script list (AUTOMATION) a script can be defined to be run during the configuration. ipt aufruft.

Return value: Status (int): OK = 1

Parameters: Number of channels (int), Device type (int), Device specifier (int), Channel name (str), Options (int)

**SetAFBatchDirectory**

Description: AutoFiles, which are automatically converted into batchfiles, can be assigned a subdirectory in the database. This function sets this parameter of the project configuration.

Return value: Status (int): OK = 1

Parameters: Directory name (str)

**GetChannelDevice**

Description: Returns the device of the specified channel in the project configuration

Return value: Device name (str)

Parameters: Channel number # in project (int)

### **GetChannelName**

Description: Returns the channel name  
Return value: Name (str)  
Parameters: Channel number # in project (int)

### **GetChannelUnit**

Return value: Channel unit (str)  
Parameters: Channel number # in project (int)

### **GetChannelTag**

Return value: Channel tag (str)  
Parameters: Channel number # in project (int)

### **GetChannelComment**

Return value: Channel comment (str)  
Parameters: Channel number # in project (int)  
Example: MsgBox MCPS.GetChannelComment(11)

### **GetChannelColor**

Return value: Color as RGB value (int)  
Parameters: Channel number # in project (int)

### **GetChannelMaxScale**

Return value: YMax (float)  
Parameters: Channel number # in project (int)

### **GetChannelMinScale**

Return value: YMin (float)  
Parameters: Channel number # in project (int)

### **GetChannelTextOutput**

Return value: Status (int) of the column **TEXT**  
Parameters: Channel number # in project (int)

### **GetChannelGraphicOutput**

Return value: Status (int) of the column **GR**  
Parameters: Channel number # in project (int)

### **GetChannelDigitalGraphicOutput**

Return value: Status (int) of the column **DIG**  
Parameters: Channel number # in project (int)

### **GetChannelExportOutput**

Return value: Status (int) of the column **GR**  
Parameters: Channel number # in project (int)

### **GetChannelMathFormula**

Description: Returns the math formula of the channel  
Return value: Formula (str)  
Parameters: Channel number # in project (int)

### **GetChannelMathFormulaOn**

Description: Returns, if the math formula is activated  
Return value: Status (int)  
Parameters: Channel number # in project (int)

### **GetChannelAlarmOn**

Return value: Alarm checking on=1, off=0  
Parameters: Channel number # in project (int), alarm level (int)  
Example: if MCPS.GetAlarmOn(1,2) then

**GetChannelAlarmType**

Return value: Alarm type setting 0-5 (H, L, DH, DL,...)  
Parameters: Channel number # in project (int), alarm level (int)  
Example: if MCPS.GetAlarmType(2,1) = 1 then

**GetChannelAlarmValue**

Return value: Alarm value setting  
Parameters: Channel number # in project (int), alarm level (int)  
Example: if MCPS.GetChannelAlarmValue(1,1) > 8 then

**GetChannelAlarmDelay**

Return value: Alarm delay value or math channel (str)  
Parameters: Channel number # in project (int), alarm level (int)

**GetChannelAlarmExtendedDelay**

Return value: String with all settings of the extended delay (str)  
Parameters: Channel number # in project (int), alarm level (int)

**GetChannelAlarmHysteresis**

Return value: Alarm hysteresis value (int)  
Parameters: Channel number # in project (int), alarm level (int)

**GetChannelAnalogOutput**

Return value: Name of output (str)  
Parameters: Channel number # in project (int), output index (int)

**GetChannelAlarmStatus**

Description: Returns during a measurement the current alarm condition of a channel  
Return value: Status (int): 0 = No alarm, 1 = Alarm, 2 = Acknowledged alarm  
Parameters: Channel number (int), alarm level (int)

**GetChannelUserColumnText**

Return value: Text of a user column (str)  
Parameters: Channel number # in project (int), Index of user column (int)

**GetChannelZM**

Return value: Mode of channel reference measurement (int)  
Parameters: Channel number # in project (int)

**GetChannelValue**

Return value: Current channel offset value of reference measurement (float)  
Parameters: Channel number # in project (int)

**GetZMTime**

Return value: Time of active reference measurement (int)  
Parameters: Options (int) = 0

**GetZMComment**

Return value: Comment of active reference measurement (int)  
Parameters: Options (int) = 0

**SetChannelUnit**

Description: Sets the channel unit  
Return value: OK=1 (int)  
Parameters: Channel number (int), unit (str)

**SetChannelComment**

Description: Sets the channel comment  
Return value: OK=1 (int)  
Parameters: Channel number (int), comment (str)

**SetChannelTag**

Description: Sets the tag  
Return value: OK=1 (int)  
Parameters: Channel number (int), tag (str)

**SetChannelMinScale**

Return value: Status (int) 1 = OK  
Parameters: Channel number # in project (int), YMin (float)

**SetChannelMaxScale**

Return value: Status (int) 1 = OK  
Parameters: Channel number # in project (int), YMax (float)

**SetChannelDP**

Description: Sets the number of decimal places  
Return value: OK=1 (int)  
Parameters: Channel number (int), dp (int)

**SetChannelTextOutput**

Description: Sets the output (on/off) for text windows  
Return value: OK=1 (int)  
Parameters: Channel number (int), status (int)

**SetChannelGraphicOutput**

Description: Sets the output (on/off) for trend windows  
Return value: OK=1 (int)  
Parameters: Channel number (int), status (int)

**SetChannelDigitalGraphicOutput**

Description: Sets the output (on/off) for trend windows with digital screen  
Return value: OK=1 (int)  
Parameters: Channel number (int), status (int)

**SetChannelExportOutput**

Description: Sets the output (on/off) during export  
Return value: OK=1 (int)  
Parameters: Channel number (int), status (int)

**SetChannelMathFormula**

Return value: Status OK = 1 (int).  
Parameters: Channel (int), Formula (str)  
Example: MCPS.SetChannelMathFormula 2, "X1+X2"

**SetChannelMathOn**

Return value: None  
Parameters: Channel (int)

**SetChannelAlarmValue**

Description: Sets the alarm Parameterss of a channel  
Return value: OK=1 (int)  
Parameters: Channel number (int), alarm level (int), On (int), Type (int), Limit (float)  
Example: MCPS.SetChannelAlarmValue 2,3,1,1,5

### **SetChannelAlarmValue2**

Description: Sets the alarm Parameters of a channel with string alarm value, because of dynamic alarm value settings (X, M or U).

Return value: OK=1 (int)

Parameters: Channel number (int), alarm level (int), On (int), Type (int), Limit (str)

Example: MCPS.SetChannelAlarmValue 2,3,1,1,U2

MCPS.SetChannelAlarmValue2 2,3,-1,-1,"1.5" Only alarm value is set

MCPS.SetChannelAlarmValue2 2,3,0,-1,"" Alarm is turned off

### **SetChannelEvent**

Description: Inserts a channel event

Return value: Status OK=1 (int)

Parameters: Point of time (date), channel number (int), event text (str), orientation (int), RGB-color (int)

See cursor operations/channel events for possible orientation indexes (1-10).

### **SetChannelUserColumnText**

Description: Sets the text of a user column with text types

Return value: Status OK=1 (int)

Parameters: Channel (int), User column (int), Text (str)

### **SetChannelUserColumnInteger**

Description: Sets the numeric value of a user column (checkbox or index of drop down list)

Return value: Status OK=1 (int)

Parameters: Channel (int), User column (int), Value (int)

### **SetChannelLog**

Description: Sets the log function box (on/off)

Return value: OK=1 (int)

Parameters: Channel number (int), status (int)

### **SetChannelZM**

Description: Sets the mode of the reference measurement computation for a channel

Return value: OK=1 (int)

Parameters: Channel number (int), Off=0, Raw data = 1, Math data = 2 (int)

### **SimulateChannelAlarm**

Description: Simulates a channel alarm to the alarm action like emails and outputs.

Return value: OK=1 (int)

Parameters: Channel number (int), On/Off (int)

### **PCSetChannelDevice**

Description: This script command can only be used in an open project configuration to set the device of a channel (line) Return value:OK=1 (int)

Parameters: Line number (int), Device type (int), Device specifier (int), Options = 0 (int)

### **PCSetChannelName**

Description: This script command can only be used in an open project configuration to set the channel name line

Return value: OK=1 (int)

Parameters: Line number (int), Channel name (int), Options = 0 (int)

### **GetChannelAlarmStatus**

Description: Returns during a measurement the current alarm condition of a channel

Return value: Status (int): 0 = No alarm, 1 = Alarm, 2 = Acknowledged alarm

Parameters: Channel number (int), alarm level (int)

### **GetProjectOnlineStatus**

Description: Returns the status of the measurement

Return value: Status: 0 = No measurement active, 1 = Measurement active, 2 = Pause

Parameters: None

### **AckChannelAlarm**

Description: Acknowledges an alarm, if script is started on data acquisition computer  
Return value: Status (int): 1 = OK  
Parameters: Channel number (int), Level (int), Time of alarm (DATE), Optionsstring (str), Options (int)  
Channel, Level and / or time can be zero.  
Example: AckChannelAlarm 4,1,d,"",0 Channel 4, Level 1 of time d is acknowledged  
AckChannelAlarm 5,0,0,"",0 All Alarms of channel 5  
AckChannelAlarm 0,0,0,"",0 All Alarms

### **GetCursorDataCount**

Return value: Number of channels at cursor (int)  
Parameters: Cursor number 1 or 2 (int)  
Example: n = MCPS.GetCursorDataCount(1)

### **GetCursorData**

Return value: Channel value at cursor (float)  
Parameters: Cursor (int), Channel (int)  
Example: value = MCPS.GetCursorData(1,15)

### **GetCursorDataStatus**

Return value: Data status OK=1 (int)  
Parameters: Cursor (int), Channel (int)  
Example: status = MCPS.GetCursorDataStatus(2,22)

### **GetCursorTime**

Return value: Cursor time (date)  
Parameters: Cursor (int)  
Example: MsgBox MCPS.GetCursorTime

### **GetCursorTimeString**

Return value: Cursor time (str)  
Parameters: Cursor (int)  
Example: MsgBox MCPS.GetCursorTimeString

### **GetCursorChannel**

Return value: Selected channel at cursor (int)  
Parameters: Cursor (int)  
Example: MsgBox MCPS.GetCursorChannel(1)

### **OpenRequestWindow**

Return value: None  
Parameters: Title, Text, Button\_left, Button\_middle, Button\_right (all str)  
Example: MCPS.OpenRequestWindow("Request?","Ouput in file","1","2","3")

### **OpenRequestWindow2**

Return value: None  
Parameters: Title, Text, Button\_left, Button\_middle, Button\_right (all str), options (int), script (str)  
Example: MCPS.OpenRequestWindow("Request?","Ouput in file","1","2","",1,"c:\script.mbs")  
Options:  
1 = No cancel button

### **GetRequestStatus**

Return value: Button status (int)  
Parameters: None  
Example: button = MCPS.GetRequestStatus

### **CloseRequestWindow**

Return value: None  
Parameters: None  
Example: MCPS.CloseRequestWindow

### **SetRequestWindowText**

Return value: None  
Parameters: Text (str)  
Example: MCPS.SetRequestWindowText("New text")

### **MessageRequest**

Description: Opens a request window with up to three buttons  
Return value: Index of pressed key (0=Cancel, 1=Left, 2=Middle, 3=Right)  
Parameters: Title (str), Text (str), left text, middle text, right text, options (int)  
Example: MCPS.MessageRequest "Script message", "Report type?","Daily","Monthly","",0

### **GetOnlineData**

Return value: Online channel value (float)  
Parameters: Channel number # in project (int)  
Example: sum = MCPS.GetOnlineData(1) + MCPS.GetOnlineData(2)

### **GetOnlineDataStatus**

Return value: Online channel data status (int) ; 0=OK, >0 = ERROR  
Parameters: Channel number # in project (int)  
Example: if MCPS.GetOnlineDataStatus = 1 then

### **GetOnlineTime**

Return value: Time of scan (date)  
Parameters: None  
Example: excel.Cells(1,1).Value = MCPS.GetOnlineTime

### **GetOnlineSampleCount**

Return value: Number of scans after the last start; reset with new AutoFile (int)  
Parameters: None  
Example: if MCPS.GetOnlineSampleCount > 100 then

### **GetOnlineTotalSampleCount**

Return value: Number of scans after the last start (int)  
Parameters: None

### **GetOnlineTotalRuntime**

Description: Returns the difference between now and the start of the measurement  
Return value: Time difference (float)  
Parameters: None

### **GetSampleRate**

Return value: Sample rate (int)

Parameters: None

Example: `t = MCPS.GetSampleRate * MCPS.GetOnlineTotalSampleCount`

### **GetOnlineDataField**

Return value: Online channel values (Array)

Parameters: None

Example: `field = MCPS.GetOnlineDataField()`

### **GetOnlineDataStatusField**

Return value: Online channel status (Array)

Parameters: None

### **GetOnlineSavedData**

Description: Returns the online value, when saved the last time. Useful in case of data reduction

Return value: Online channel value, when saved the last time(float)

Parameters: Channel number # in project (int)

Example: `sum = MCPS.GetOnlineSavedData(1) + MCPS.GetOnlineSavedData(2)`

### **GetOnlineSavedDataStatus**

Return value: Online channel data status, when saved the last time (int)

Parameters: Channel number # in project (int)

Example: `if MCPS.GetOnlineDataStatus = 1 then`

### **GetOnlineSavedTime**

Return value: Time of saved scan (date)

Parameters: None

Example: `excel.Cells(1,1).Value = MCPS.GetOnlineTime`

### **GetAlarmField**

Return value: Array of all channel with (active) alarms

Parameters: Options (int)

The option parameter defines, which types of alarms are returned:

1: Active and not acknowledged

2: Active and acknowledged

4: Inactive and not acknowledged

8: Inactive and acknowledged

To get several types at once the values must be added.

### **StartMeasurement**

Description: Starts the measurement of a specific project

Return value: Status (int): 1 = OK

Parameter: Complete filename of project (str), options (int)

Options:

0 = start without window layout dialog

1 = start with window layout dialog. Be careful! If user does not close the window any way, the system can be blocked.

### **StopMeasurement**

Description: Stops the data acquisition of the project

Return value: None

Parameters: None

### **LogEvent**

Description: Adds a text into the project log file

Return value: None

Parameters: Date/time (date), Text (str), Options (int)

**OfflineStartRead**

Description: Prepares offline data reading  
Return value: File handle (int)  
Parameters: AutoFile name (str) or empty string "", if no AutoFiles are used  
Example: handle = MCPS.OfflineStartRead("00001")

**OfflineEndRead**

Return value: Status OK =1 (int)  
Parameters: File handle (int)  
Example: MCPS.OfflineEndRead handle

**OfflineNextBlock**

Description: Reads the next scan from the file into an internal buffer.  
Return value: Status OK =1 (int), Error/End = 0  
Parameters: File handle (int)  
Example: if MCPS.OfflineNextBlock(handle) <> 1 then

**OfflineGetData**

Return value: Offline channel value (float)  
Parameters: Channel number # (int)  
Example: value = MCPS.OfflineGetData(1)

**OfflineGetDataStatus**

Return value: Channel status OK=1 (int)  
Parameters: Channel number # (int)  
Example: if MCPS.OfflineGetDataStatus(23) <> 1 then

**OfflineGetTime**

Return value: Scan time (date)  
Parameters: None  
Example: t = MCPS.OfflineGetTime()

**OfflineGetTime2**

Return value: Decimal place of scan time (float), Resolution of OfflineGetTime is only 1 second  
Parameters: None

**OfflineSetPosition**

Description: Sets the file read pointer to the specified time  
Return value: Status OK =1 (int)  
Parameters: File handle (int)  
Example: MCPS.OfflineSetPosition(handle,peektime)

**OfflineGetFirstScanTime**

Description: First scan of all data (even AutoFiles)  
Return value: 1.Scan (date)  
Parameters: None  
Example: t = MCPS.OfflineGetFirstScanTime()

**OfflineGetLastScanTime**

Return value: Last scan (date)  
Parameters: None

**OfflineGetAFStartTime**

Return value: First scan of AutoFile (date)  
Parameters: AutoFile path

**OfflineGetAFEndTime**

Return value: Last scan of AutoFile (date)  
Parameters: AutoFile path

**OfflineLinkAutoFiles**

Description: Tells MCPS to operate over all AutoFiles, which belongs to the specified time range. This allows continuous scanning over several AutoFiles. The file read pointer is set to the first scan of the first AutoFile in the group and not to the starttime. This must additionally be done with OfflineSetPosition.

Return value: Status OK =1 (int)

Parameters: File handle (int), Start time (date), End time (date)

Example: if MCPS.OfflineLinkAutoFiles(handle,starttime,endtime) then

**OfflineGetLastData**

Return value: Channel value of scan before the current scan(int)

Parameters: Channel number (int)

**OfflineGetLastDataStatus**

Return value: Channel value status of scan before the current scan(int)

Parameters: Channel number (int)

**OfflineGetDataField**

Return value: All channel values (Array)

Parameters: None

**OfflineGetDataStatusField**

Return value: Status of all channels (Array)

Parameters: None

### PrintOfflineTrend

Description: Prints an offline graphic with window and print layout

Return value: Status (int): OK = 1

Parameters: Window layout in project (str), AutoFile (str), Print layout (str), Options (int)  
All strings can be empty ""

Options:

1 = If no print layout is specified, then the default print layout for offline printouts in the project configuration is used.

2 = If the file to be printed is a batchfile and no print layout is specified, then the integrated print layout is used. The integrated print layout is the layout defined in the batch mask for printing and is stored with the batchfile.

4 = Opens the printer dialog. Should not be used for automated printouts.

8 = Delete AutoFile after printing

16 = The specified window layout is common window layout.

All values can be combined (added)

### PrintOfflineTrend2

Description: As above, but the first parameter is used as a list to set several different values. The info list is a string, which contains several text lines, which are separated by a linefeed (vblf).

Return value: Status (int): OK = 1

Parameters: Info list(str), Window layout in project (str), AutoFile (str), Print layout (str), Options (int)

Assembly of the infolist

1. line: Channellist with the format "1,2,4-6"

2. line: Start time of data to print. Important! This value is not a normal time string, it is a double value representing the time information. But it is also put into the info list string.

3. line: End time of printout

4. line: 5 parameters separated by comma:

Automatic X-axis scaling (1=on;0=off)

Multi page mode (0=fixed number of pages, 1=fixed time base per page)

Number of pages

Time value

Time unit (0=seconds, 1=minutes, 2=hours)

5. line: Name of printer

Example: st = CDate("17.07.2009 12:15:00")

et = CDate("17.07.2009 12:17:00")

s = "2,5-8" & vblf & cdbl(st) & vblf & cdbl(et) & vblf & "0,1,0,0.5,1"

PrintOfflineTrend2 s,"windowlayout1","00003","c:\programme\mcps6\_3\test.ply",8

St and et are defined here directly. The time string may differ dependent on country settings. Normally these values are calculated or entered by the used in an input window.

Important is the conversion (cdbl) of the time values into double values.

Simple example with mostly empty parameters:

s = "" & vblf & "" & vblf & "" & vblf & "0,0,1,0,0"

PrintOfflineTrend2 s,"", "", "", "", 8

### **PrintLayout**

Description: Prints a layout without other standard data (graphics, statistics, ...). This can be used for reports in conjunction with project variables.

Return value: Status (int): OK = 1

Parameters: Print layout (str), Options (int)

Options = 0

### **PrintAlarms**

Description: Prints the alarm file of the project

Return value: Status (int): OK = 1

Parameters: Start time (date), End time (date), Channel list (str), Print layout (str), Options (int)

Options = 0

If start time or/and end time are 0, then the time of the first/last scan is used.

The channel list defines the channels to be output in the following format: "1,3,5-9"

### **PrintEvents**

Description: Prints the user event file of the project

Return value: Status (int): OK = 1

Parameters: Start time (date), End time (date), Print layout (str), Options (int)

Options = 0

If start time or/and end time are 0, then the complete file is output.

### **GetPrintStatus**

Return value: Status (int): 1 = Print job is in print queue

Parameters: None

### **GetPrintStartTime**

Return value: Start time of data to be printed e.g. time of first scan in numerical printout (date)

Parameters: None

### **GetPrintEndTime**

Return value: End time of data to be printed (date)

Parameters: None

### **SetPrintID**

Description: When a script initiates several print outs, these print outs are executed after the script has been finished. If project variables are set and used in the print layout, always the last values set by the script are available during all the printing. To avoid this, you can define a print id in the script before each individual print out. To use only one layout for several printouts you have to define project variables in the print layout with a special ending character (#). For each print out this character is replaced by the according print id. In the script you have to define the same variable several times with different endings.

Example:

You want to make 3 print outs with same print layout. In the print layout you define \$PV(TEST#). In the script you set for the first printout the project variable TEST1 to "AAA" and the print id to 1 and call a printing function, then set project variable TEST2 to "BBB" and print id to 2 and print again and so on. Now you will get three printouts with AAA on the first page, BBB on the second....

Return value: None

Parameters: ID (int), Options (int) = 0

### **DialogCreate**

Return value: Dialoghandle (int)

Parameters: Title,Width, Height, Options (Int)

Example: d = MCPS.DialogCreate("Hallo",300,200,0)

Options:

1 = Modeless window. Normally the input window blocks further script operations and further scripts to run until the OK or CANCEL button has been pressed. To avoid this the dialog can be modeless. That means the window is still open, but the function returns immediately and the script can finish. To get the parameters, when the user closes the dialog a so called end script can be set by <DialogSetEndScript>, which is called after the dialog has been closed.

### **DialogRun**

Return value: Returncode (int) 1=OK 0=CANCEL

Parameters: Dialog handle (int)

Example: status = DialogRun(d)

### **DialogRemove**

Return value: None

Parameters: Dialog handle (int)

Example: MCPS.DialogRemove d

### **DialogAddText**

Return value: ID (int)

Parameters: Dialog handle,text,x,y,xl,yl,Options (4096 = Bold,8192=disabled)

Example: id = MCPS.DialogAddText(d,"Text",10,10,150,20,4096)

### **DialogAddEdit**

Return value: ID (int)

Parameters: Dialog handle,x,y,xl,yl,Options (4096 = Bold,8192=disabled)

Example: id = MCPS.DialogAddEdit(d,10,50,150,20,0)

### **DialogGetEditText**

Return value: Text of edit field (str)

Parameters: Dialoghandle (int), ID (int)

### **DialogSetEditText**

Description Preset text for edit field

Return value: Status (int): 1 = OK

Parameters: Dialoghandle (int), ID (int), Text (str)

### **DialogSetEditTextLength**

Description Defines length of text in edit field

Return value: Status (int): 1 = OK

Parameters: Dialoghandle (int), ID (int), length (int)

### **DialogAddDateTime**

Return value: ID (int)

Parameters: Dialoghandle, date,x,y,xl,yl,Options (1=Long date format, 2=time format,8192=disabled)

Example: id = MCPS.DialogAddDateTime(d,10,90,150,20,1)

### **DialogGetDate**

Return value: Date (date)

Parameters: Dialoghandle, ID

Example: datum = MCPS.GetDate(d,id)

### **DialogAddSelection**

Return value: ID (int)

Parameters: Dialoghandle,x,y,xl,yl,Options (4096 = bold,8192=disabled)

Example: id = MCPS.DialogAddSelection(d,10,50,150,20,0)

### **DialogAddSelectionString**

Return value: Handle (int)  
Parameters: Dialoghandle, ID, Text

### **DialogGetSelection**

Return value: Index of selected item  
Parameters: Dialoghandle, ID  
Example: choice = MCPS.GetSelection(d,id)

### **DialogGetSelectionText**

Return value: Text of selected item (str)  
Parameters: Dialoghandle, ID  
Example: text = MCPS.GetSelectionText(d,id)

### **DialogSetSelection**

Return value: Status (int) 1=OK  
Parameters: DialogHandle, ID, selected item

### **DialogSetSelectionText**

Return value: Status (int) 1=OK  
Parameters: DialogHandle, ID, text

### **DialogSetSelectionTextLength**

Return value: Status (int) 1=OK  
Parameters: DialogHandle, ID, length

### **DialogAddCheckBox**

Return value: ID (int)  
Parameters: Dialoghandle,x,y,xl,yl,text, options (4096 = bold,8192=disabled)

### **DialogGetCheckStatus**

Return value: Status (int) 1=On  
Parameters: DialogHandle, ID

### **DialogSetCheckStatus**

Return value: Status (int): 1=OK  
Parameters: DialogHandle, ID, Status (int)

### **DialogAddRadioButton**

Return value: ID (int) 1=On  
Parameters: Dialoghandle,x,y,xl,yl,text,optionen (1=active, 2=start of group, 4096 = bold)

### **DialogGetRadioStatus**

Return value: Status (int) 1=On  
Parameters: DialogHandle, ID

### **DialogAddGroupBox**

Return value: ID (int)  
Parameters: Dialoghandle,x,y,xl,yl,text,optionen (4096 = bold)

### **DialogDependency**

Description: Defines a specific dependency between two elements.  
Return value: Status OK=1 (int)  
Parameters: DialogHandle, Handle to monitor, Handle to control, Selection, Action, Options  
*Handle for monitoring* is an element or button, which is checked for certain action, e.g. a checkbox is turned on or an item is selected in a dropdown listbox.  
*Handle to control* is the item, which is influenced by the action above for example disabled or hidden.  
*Selection*: The status of the item to monitor. For selection buttons the index of the selected item, for check boxes 0 or 1 to see, if they are on or off.  
Action  
1: Make the button to control visible otherwise invisible

2: Enable or disable item  
3: Set checkbox  
Options: 0

Example: status = DialogDependency(dialoghandle, selectionhandle, texthandle, 5, 1, 0)  
If the fifth element of the selection box (selectionhandle) is chosen, a text (texthandle) appears.

### **DialogDependencyText**

Description: Defines a specific dependency between a text or selection element and another item. The text is monitored and an action is performed, when the specified text is entered.  
Return value: Status OK=1 (int)  
Parameters: DialogHandle, Handle to monitor, Handle to control, Text to monitor, Action, Options

### **DialogSetEndScript**

Description: Defines a script, which is called when a modeless dialog is closed. See <DialogCreate> for more information.  
Return value: Status OK=1 (int)  
Parameters: DialogHandle, Script file (str), Free text (str), Optionsstring (str) = "", Options (int) = 0  
With <GetScriptData> the end script can get some additional information and the free text.

### **GetDeviceID**

Description: Gets the device index of an instrument in the device manager for further operations.  
Return value: Device id (int)  
Parameters: Device name(str), device specifier (int)  
Example: deviceid = MCPS.GetDeviceID("DA100",1)

### **OpenDevice**

Description: Makes a connection to the device  
Return value: Status OK =1 (int)  
Parameters: Device id (int)  
Example: status = MCPS.OpenDevice(deviceid)

### **StringToDevice**

Description: A text string is sent to the device  
Return value: Status OK =1 (int)  
Parameters: Device id (int), String  
Example: MCPS.StringToDevice(deviceid,"UD0")

### **StringFromDevice**

Description: Receives a string from the device  
Return value: Returned string (str)  
Parameters: Device id (int)  
Example: response = MCPS.StringFromDevice(deviceid)

### **CloseDevice**

Return value: Status OK =1 (int)  
Parameters: Device id (int)  
Example: MCPS.CloseDevice(deviceid)

### **GetDeviceComment**

Description: Returns the comment of the device as defined in the device manager  
Return value: Comment (str)  
Parameters: Device id (int)

### **SetDeviceTerminator**

Description: Sets the end of line character for receiving data (StringFromDevice)  
Return value: Status OK =1 (int)  
Parameter: Device id (int), end of line character (int), Options (int)

Example: MCPS.SetDeviceTerminator(deviceid, 13, 0) EOL = Carriage Return

### **SetDeviceStatus**

Description: Devices can be suspended from measurement. No communication is done to this instrument and MCPS will show and save all channel with <Invalid data>. See also chapter [DEVICES/DEVICE INFO](#).

Return value: Status OK =1 (int)

Parameters: Device id (int), Status (int)  
Status = 1: Device in normal communication state  
Status = 0: Device suspended. No communication.

### **WriteSetPoint**

Description: Sets the setpoint of a controller

Return value: Status OK =1 (int)

Parameter: Device id (int), number of set point (int), Value (float), Options (int)

### **SetKEYDRVValue**

Description: Sets the channel value of a KEYDRV

Return value: Status OK =1 (int)

Parameter: Device id (int), channel number (int), Value (float), Options (int)  
Options = 1: The device configuration is saved to make the changes permanent

### **WriteSetPoint**

Description: Sets the setpoint value of a device (controller)

Return value: Status OK =1 (int)

Parameters: Device id (int), Setpoint number (int), Value (float), Options (int)

### **DeviceCommand**

Description: Executes device specific commands.

Return value: Error code of device (int), common error =-1

Parameters: Device id (int), Command (int), Optionsstring (str), Options (int)

Commands:

0: Stop measurement of device

1: Start measurement of device

2: Initialize driver and device

3: Stop keep-alive

4: Start keep-alive again

### **SetDigitalOutput**

Description: Sets the internal status of a digital output as available in MCPS.

Return value: Status OK = 1 (int)

Parameters: Output name(str), Status (int)

Example: status = MCPS.SetDigitalOutput("DA100 – 1: R011",1)

### **UpdateDigitalOutputs**

Description: Sets all digital outputs physically to their status.

Return value: None

Parameters: None

### **GetDigitalOutputStatus**

Description: Returns the status of a digital output as set in MCPS. The information is not taken from the instrument, it's the last setting of a MCPS action (alarm output, script output).

Return value: Status 1=ON (int)

Parameters: Name of output (str)

Example: status = MCPS.GetDigitalOutputStatus("DA100 – 1: R011")

### **SetAnalogOutput**

Description: Sets an analog output value.

Return value: Status OK = 1 (int)

Parameters: Name of output (str), Value (float), Options = 0 (int)

Example: status = MCPS.SetAnalogOutput("W750 – 1: AO2",7.5,0)

### **SetRegister**

Description: Sets a register to the specified value. Since register are not saved to datafile, these modifications are not available in historical data. Usually this function is used for online control for example as a step counter in a test stand.

Return value: Status 1=OK (int)

Parameters: Register (int), value (float), option string (str), options (int)  
Option string is at the moment empty = ""  
Options: 1 = Set offline registers during evaluation of historical data

### **GetRegister**

Description: Gets a register value. Since register are not saved to datafile, these modifications are not available in historical data. Usually this function is used for online control for example as a step counter in a test stand. There are online and offline registers.

Return value: Value (float)

Parameters: Register (int), options (int)  
Options: 1 = Request offline register

### **Export**

Description: Exports the data of the project's datafile or the specified AutoFile.

Return value: Status 1=OK (int)

Parameters: Destination file (str), start time (date), end time (date), AutoFilename or ""

Example: MCPS.Export("C:\export.xls",starttime, endtime, "00001")

### **Export2**

Description: As <Export>, but also supports a list of channels to be exported and a skip value in the infolist parameter. Both strings must be separated by a line feed (vblf)

Return value: Status 1=OK (int)

Parameters: Destination file (str), start time (date), end time (date), AutoFilename or "", Infolist (str) or ""

Example: MCPS.Export2("C:\export.xls",starttime, endtime, "00001", "1,2-4,7" & vblf & "5")  
Exports the channels 1,2,3,4 and 7. Only every fifth scan is used.

### **EmailSend**

Description: Sends an email

Return value: Status 1=OK (int)

Parameters: Sender (str), Subject (str), Emailtext (str), Filename of attachment or ""

Example status = MCPS.EmailSend("info@cad-computer.de", "TEST", "Hello", "")

### **StatisticRun**

Description: Computes the statistics in a given range of time.

Return value: Status OK=1 (int)

Parameters: Starttime (date), Endtime (date), Computation flags (int), Channel list (str), Options (int)  
If starttime or endtime 0, then first or last scan time is used.  
The computation flags the type of statistics to be made.  
1 = Minimum value  
2 = Mean value  
4 = Maximum value  
8 = Standard deviation  
16 = MKT  
The values can be combined by summation.

The channel list defines the channels to use. Example: "1,5,7-12"

Options = 0  
Example: status = MCPS.StatisticRun(0,0,7, "1-20", 0)  
Min-, Max-, Meanvalue over all data for channels 1-20

### **StatisticGetMinValue**

Return value: Minimum value after script statistics (float)

Parameters: Channel (int)

**StatisticGetMeanValue**

Return value: Mean value after script statistics (float)

Parameters: Channel (int)

**StatisticGetMaximumValue**

Return value: Maximum value after script statistics (float)

Parameters: Channel (int)

**StatisticGetStdValue**

Return value: Standard deviation value after script statistics (float)

Parameters: Channel (int)

**StatisticGetMktValue**

Return value: MKT value after script statistics (float)

Parameters: Channel (int)

**StatisticRealStartTime**

Description: Returns the time of the first scan, which is really used in the statistical computation.

Return value: Time of first scan (date)

Parameters: None

**StatisticRealEndTime**

Description: Returns the time of the last scan, which is really used in the statistical computation.

Return value: Time of last scan (date)

Parameters: None

**StatusWindowOpen**

Description: Opens a status window, that is docked to the mainframe to show text information, bar graphs, LEDs or bitmaps.

Return value: None

Parameters: None

**StatusWindowClose**

Return value: None

Parameters: None

**SWAddCommandList**

Description: Adds a new list to hold drawing commands

Return value: Handle to command list (int)

Parameters: None

**SWRemoveCommandList**

Description: Removes the list and all drawing commands in the list

Return value: None

Parameters: Handle to command list

**SWEnableCommandList**

Description: Activates or deactivates a command list. All elements in the list are turned off (invisible).

Return value: Status (int): 1 = OK

Parameters: Handle of command list (int), Mode (int), Options (int)

Mode 1 = on, 0 = off

Options 0

**SWRedraw**

Description: Redraws the entire status window. Must be used after Update commands.

Return value: None

Parameters: None

**SWSetSize**

Description: Sets the current and maximum size of the status window and if it should be resizable.

Return value: None

Parameters: Current size (int), Maximum size (int), Sizeable (int)

**SWSetBKColor**

Description: Sets the background color of the status window

Return value: None

Parameters: Color (int)

**SWAddText**

Description: Adds a text command to the command list to be drawn in the status window

Return value: Handle to text command

Parameters: Handle to command list (int), Text (str), x (int), y (int), Color (int), Options (int)

**SWUpdateText**

Description: Changes text of an existing command

Return value: None

Parameters: Handle to command list (int), Handle to text command (int), New text (str)

**SWSetFont**

Description: Sets the font of a text item

Return value: Status (int): 1 = OK

Parameters: Handle of command list (int), Handle of text item (int), Fontname (str), Character height (int), Orientation (int), Options (int)

Fontname: Name of font as defined in Windows

Orientation: Rotation value in degree (0-360)

Options: 1 = Bold

2 = Italic  
3 = Underline  
4 = Strike out  
The values can be added.

### **SWSetTextBackground**

Description: Changes the background of a text item  
Return value: Status (int): 1 = OK  
Parameters: Handle of command list (int), Handle of text item (int), RGB-Color (int), Mode (int),  
Options = 0 (int)  
Mode: 0 = Transparent 1 = OPAQUE (Solid)

### **SWAddLED**

Description: Adds a LED  
Return value: Handle to LED command  
Parameters: Handle to command list (int), x (int), y (int), xl (int), yl (int), Color (int), Options (int)

### **SWUpdateLED**

Description: Changes the color of a LED  
Return value: None  
Parameters: Handle to command list (int), Handle to LED command (int), New color (int)

### **SWAddLEDThreshold**

Description: Adds a treshold value to an LED item. The color changes, if the defined limit is exceeded. Several tresholds can be set. See SWLinkChannel.  
Return value: Status (int): 1 = OK  
Parameters: Handle of command list (int), Handle of LED (int), treshold value (float), RGB-Color (int), Options (int) = 0

### **SWAddBargraph**

Description: Adds a bar graph command to the command list to be drawn in the status window  
Return value: Handle  
Parameters: Handle to command list (int), Header text (str), x (int), y (int), xl (int), yl (int), Max value, (float), Min value (float), Color (int), Options (int)

### **SWUpdateBargraph**

Description: Updates the bar graph  
Return value: None  
Parameters: Handle to command list (int), Handle to bar graph (int), New value (float), Status (int)

### **SWAddAnalogMeter**

Description: Adds an analog meter  
Return value: Handle of analog meter  
Parameters: Handle of command list (int), x (int), y (int), xl (int), yl (int), Maximum value (float), Minimum value (float), Options (int)

### **SWUpdateAnalogMeter**

Description: Update of analog meter  
Return value: None  
Parameters: Handle of command list (int), Handle of analog meter (int), New value (float), Data status OK=0, Error = 1 (int)

### **SWAddLine**

Description: Adds a line  
Return value: Handle to line command  
Parameters: Handle to command list (int), x1 (int), y1 (int), x2 (int), y2 (int), Color (int), Options (int)

### **SWUpdateLine**

Description: Updates the position of a line  
Return value: Status (int): 1 = OK  
Parameters: Handle of command list (int), Handle of line (int), x1 (int), y1 (int), x2 (int), y2 (int)

### **SWAddRectangle**

Return value: Handle

Parameters: Handle to command list (int), x (int), y (int), xl (int), yl (int), Color (int), Options (int)

### **SWAddCircle**

Return value: Handle

Parameters: Handle to command list (int), x (int), y (int), xl (int), yl (int), Color (int), Options (int)

### **SWAddBitmap**

Description: Adds a bitmap

Return value: Handle to bitmap command

Parameters: Handle to command list (int), Path to file (str), x (int), y (int), Options (int)

### **SWAddMultiBitmap**

Description: Adds a multi bitmap. This element contains several bitmaps, but only one is displayed. Which one is active is defined by a treshold value for each image. If the control value is set by SWUPdateMultiBitmap or SWLinkChannel, the related image is shown. The multi bitmap element can be used to visualize nearly all possible process icen such as fans, valves, tanks, transport belt and so on.

Return value: Handle

Parameters: Handle of command list (int), x (int), y (int), Optionen (int)

### **SWAddMImage**

Description: Adds a bitmap file (\*.bmp) and its threshold to a multi bitmap item..

Return value: Status (int): 1 = OK

Parameters: Handle of command list (int), Handle of multi bitmap item (int), Bitmap file (str), Threshold (float), Options (int)

### **SWUpdateMultiBitmap**

Description: Sets a new control value and perhaps displays a new image depending on treshold.

Return value: Status (int): 1 = OK

Parameters: Handle of command list (int), Handle of multi bitmap (int), Control value (float), Options = 0 (int)

### **SWAddVideoCommand**

Description: Adds a small video file (.avi). Not all movie file types are supported, so it is possible, that a video is not running.

Return value: Handle

Parameters: Handle of command list (int), Filename (str), x (int), y (int), Options (int)

Optionen: 1 = Transparent background (if supported by movie)  
2 = Invisible, if not played (Control value is below treshold)

### **SWSetVideoTreshold**

Description: Sets the treshold value for a video item to be automatically controlled in case of SWLinkChannel.

Return value: Status (int): 1 = OK

Parameters: Handle of command list (int), Handle of video item (int), Treshold (float), Options (int) = 0

### **SWRemoveCommand**

Description: Remove a command from a command list

Return value: None

Parameters: Handle of command list (int), Handle of item (int)

### **SWLinkChannel**

Description: Links a channel of the project to an item in the status window. After a scan is made during data acquisition, the item is updated by evaluation the new channel value. Several items have one or more treshold values to change their appearance related to the channel value. A linked text item will display the value in form.

Return value: Status (int): 1 = OK

Parameters: Handle of command list (int), Handle of item to link (int), Related channel (int)

### **SWAddButton**

Description: Adds a push button. If the button is pressed, the script <Status window event> of the project configuration is executed (See also SWSetEventHandling).

Return value: Handle (int)

Parameters: Handle of command list (int), Text of button (str), x (int), y (int), xl (int), yl (int), Options (int) = 0

### **SWAddTabControl**

Description: Adds a tab control. With SWAddTab several tabs can be added to the tab control. If a tab is pressed, the script <Status window event> of the project configuration is executed (See also SWSetEventHandling). Other items, which are added after the tab control can be placed on it. Several items can be grouped in an own command list and made visible or invisible by enabling or disabling the command list, when the current tabs has changed. So only those items are shown, which tab is active.

Return value: Handle (int)

Parameters: Handle of command list (int), x (int), y (int), xl (int), yl (int), Options (int) = 0

### **SWAddTab**

Description: Adds a tab to a tab control

Return value: Status (int): 1 = OK

Parameters: Handle of command list (int), Handle of tab control (int), Text of tab, Position of tab (int), Options (int) = 0

### **SWSetEventHandling**

Description: Defines the manner of handling button actions and when to call an event script.

Return value: Status (int): 1 = OK

Parameters: Mode (int), Scriptfile or "" (str)

Mode:

1 = Call <Status window event script> of all measuring (online) projects

2 = Call <Status window event script> of all offline projects

4 = Call the specified script file to handle actions without any project

The values can be combined. Default setting is 1.

### **SWGetEventHandle**

Description: Returns the Handle of the button, which caused an event.

Return value: Handle (int)

Parameters: None

### **SWGetEventAction**

Description: Returns the item specific action code

Return value: Action (int)

Parameters: None

Actions:

Push button: 1 = Button pressed

Tab control: 1 = Tab was selected

Additional information are set and can be taken by SWGetEventInfo1 and SWGetEventInfo2

### **SWGetEventInfo1**

Description: Action specific additional information

Return value: Info (int)

Parameters: None

Type of item:

Tab control: Number of selected tab

### **SWGetEventInfo2**

Description: Action specific additional information

Return value: Info (int)

Parameters: None

### **SWGetActiveProjectName**

Description: If the process window (PMON) is set to tabbed mode, all running projects with process window output are listed on the top of the window. Only one projects can be selected and its items are displayed. This function returns the name of the selected project.

Return value: Name of project (str)

Parameters: None

### **SWGetActivePageName**

Description: If the process window (PMON) is set to tabbed mode, all pages of a project with process window output are listed on the bottom of the window. Only one page can be selected and its items are displayed. This function returns the name of the selected page.

Return value: Name of page (str)

Parameters: None

### **ToolsFormatTime**

Description: Formats a date into a string

Return value: Date string (str)

Parameters: Date (date), Format string (str), Options (int)

The format string can contain several control codes, which are evaluated and replaced by MCPS:

\$Y2 Year with 2 digits

\$Y Year with 4 digits

\$M Month with 2 digits

\$D Day with 2 digits

\$H Hours 2 digits

\$I Minutes with 2 digits

\$S Seconds with 2 digits

Example text = MCPS.ToolsFormatTime(now(),"H:\$I:\$S \$D.\$M.\$Y")

### **ToolsStringToList**

Description: Converts an enumeration text to an array  
Return value: Field (VARIANT)  
Parameters: List (str), Maximum value to consider (int), Options (int)  
Example field = MCPS.ToolsStringToList("1,3,5-7,9,11", 10,0)  
field(0): 1  
field(1): 3  
field(2): 5  
field(3): 6  
field(4): 7  
field(5): 9  
Rest is not converted, because it is greater than the allowed maximum value of 10.

### **ECSReply**

Description: Returns a string back to a PLC, which has sent a command to the ECS. This function can only be called in the script, which has been called by the ECS according to the PLC message.  
Return value: None  
Parameters: Text (str), Options (int)  
Example ECSReply "Hello PLC", 0

### **DLLCommand**

Description: This function can be called, if a user specific DLL (Custom DLL) is available and supports the function <DLLScriptCommand>. The script command sends a text string to the DLL function, which again returns a text string.  
Return value: Text (str)  
Parameters: Text (str)  
Example answer = DLLCommand("Hello DLL")

### **WSSetSessionVariable**

Description: Sets in a web server script a variable, which can be used during the complete session, that means even if the browser changes to other HTML pages of MCPS.  
Return value: None  
Parameters: Text (str)  
Example Name of variable (str), Value (VARIANT)

### **WSGetSessionVariable**

Description: Gets the value of a session variable set by <WSSetSessionVariable>.  
Return value: Variable (VARIANT)  
Example Name of variable (str)

# Important information

## Important

This chapter and its subchapters contain basic and important information about the MCPS environment and client / server mode:

- Directories: Since Windows Vista MCPS can't save its configuration data, projects, layouts and so on in the MCPS installation path (C:\program files\mcpsxx). As a result the files of MCPS are distributed over several directories:
  - a) C:\Program files\MCP7: This is the default installation path, where MCPS program files, DLLs and other runtime files are installed.
  - b) C:\Programdata\MCP7: This directory is the so called application data directory for all users. MCPS saves its configuration here. Some other files, which are handled by administrators (scripts, webserver,...) are also put here. Since this directory is often hidden by default, it can easily be opened by MCPS ([WINDOW/OPEN WORKING DIRECTORY](#))
  - c) Explorer libraries\Documents\Public documents: By default MCPS will install and save regular user files like projects, print layouts and exports to this directory.The separation of files is intended by Windows. To keep all MCPS files in one folder, you have to install MCPS in a completely different directory. The working directory and documents directory can be redirected in the special configuration file <mcps.ini>. This can automatically be done during installation by selecting the **expert mode**. MCPS will write the correct #mcps.ini file. For more information see next chapter.
- Dongle (Protection module): Each MCPS package is protected by a small hardware usb key and its related password. For optional MCPS clients it is possible to install one dongle for all clients. A special service program handles the access to this client dongle. This combination is defined as NetView-clients.
- Netview-Clients A client MCPS or NetView-Clients cannot collect data. They are used for data evaluation, statistics, reports and printouts. MCPS clients can be installed on as many computers as you want. During startup the MCPS client connects to the NetView dongle (or better its service) and requests a free login. According to the NetView password 2,5,10,20 or 50 clients can login at the same time. The NetView dongle should be connected to a computer, which is permanently available. It can also be connected to the data acquisition system.
- Mserv Mserv is an option for the measurement computer to distribute online data, alarm and messages to the clients. So historical data and realtime data can be visualized on the clients. Alarm acknowledgement is also possible.
- NCS NetworkControlService: This is a special service to handle audittrail and user administration on a secure path. The users have no direct access to these files. The measurement MCPS and the clients communicate via TCP/IP interface with this service.
- Firewall For Dongle, MServ or NCS the firewall must be adjusted.
- DCOM There are two different types of network transmissions between server and clients available. DCOM is the older type, which is hard to configure since Windows XP SP2. The other and commonly used type is the TCP/IP transmission, which is very simple and only needs one firewall modification. The chapter DCOM describes all necessary DCOM settings.

# MCPS.INI

Since Windows Vista MCPS must use three different directories during installation and runtime:

Program folder. Typically c:\program files\mcps7

Application data folder: Typically c:\programdata\mcps7 (Often hidden)

Document folder: Typically Library\Documents\PublicDocuments.

Since this is often confusing and more complex for backup issues, all MCPS files can be put into one directory by using the #mcps.ini file. This one directory cannot be placed under program files. It must be a completely new location or another drive.

The configuration file #mcps.ini must be saved to the root MCPS folder (mcps.exe). The following entries can redirect the working directory and documents directory:

```
#WORKDIR:  
D:\MCPS7  
#DOCUMENTDIR:  
D:\MCPS7
```

Or just empty lines to set all folders to the MCPS program directory.

```
#WORKDIR:  
  
#DOCUMENTDIR:  
  

```

# Dongle

## Common

MCPS is delivered with a USB dongle. If this hardware protection module cannot be found, MCPS is shut down. Each MCPS measurement package with drivers has its own dongle. You cannot run several MCPS programs on the same computer. If you are using special environments like virtual machines it might be necessary to use an ethernet-usb-converter to have access to an usb port.

If you are using clients, each client can have its own dongle or you can use one *network dongle* to handle all clients. This special dongle is normally plugged into a computer, which is permanently available (e.g. the measurement pc). A specific software must be installed on that computer and the clients can access the remote computer. This kind of use is called Netview. A password for the clients defines their options and how many users can log in at the same time.

Since MCPS V7 MCPS new packages are delivered with a dongle from Wibu-Systems called [Codemeter](#). This module is described in the next chapter.

Older versions of MCPS use the Hardlock dongle from Aladdin Knowledge Systems. This dongle is described in chapter [Hardlock](#).

## Codemeter (Wibu)

Data acquisition computer:

When installing MCPS some drivers for Codemeter are copied (not installed) to the MCPS directory. Normally this works fine. If MCPS cannot find the local dongle, please install the Codemeter's runtime kit. This is available on the CD in the directory Special\Wibu.

Clients (NetView):

If you have one NetView-dongle to handle all your clients, you need to install the Codemeter's runtime kit on at least on that computer, where the NetView-dongle is connected to. The runtime kit is available on the CD in the directory Special\Wibu. After installation you will find an icon in the task bar. You can press this icon or select START/PROGRAMS/CODEMETER to run the CODEMETER CONTROL CENTER.

Press the button **WebAdmin** at the right bottom of the control center. Select the page below and activate the button <Run Network Server>. Press <Apply>. The Codemeter service will be installed and started.



The screenshot shows the CodeMeter WebAdmin interface. At the top, there is a navigation bar with tabs: Home, Content, Server, Configuration (selected), Diagnosis, Info, and Help. Below the navigation bar, there is a sub-menu: Network | Proxy | Access Control | Certified Time | WebAdmin | Backup | Borrowing. The main content area is titled "Network" and contains the following configuration options:

- Bind Address \*: All (Default) (dropdown menu)
- Network Port \*: 22350 (text input)
- UDP Waiting Time \*: 1000 ms (text input)
- Run Network Server:
- Server Search List: (empty text area)

Below the Server Search List, there are four buttons: add, remove, up, and down. At the bottom of the configuration area, there are two buttons: Apply and Default. A note at the bottom of the page states: (\*) Changes only take effect after restarting CodeMeter.

Codemeter is using the registered IP port 22350. This port must be excluded from a firewall. During the installation of MCPS this is done, if possible, automatically with the Windows firewall. Other firewalls must be set manually. See chapter [Firewall](#) for more information.

## HardLock (Aladdin)

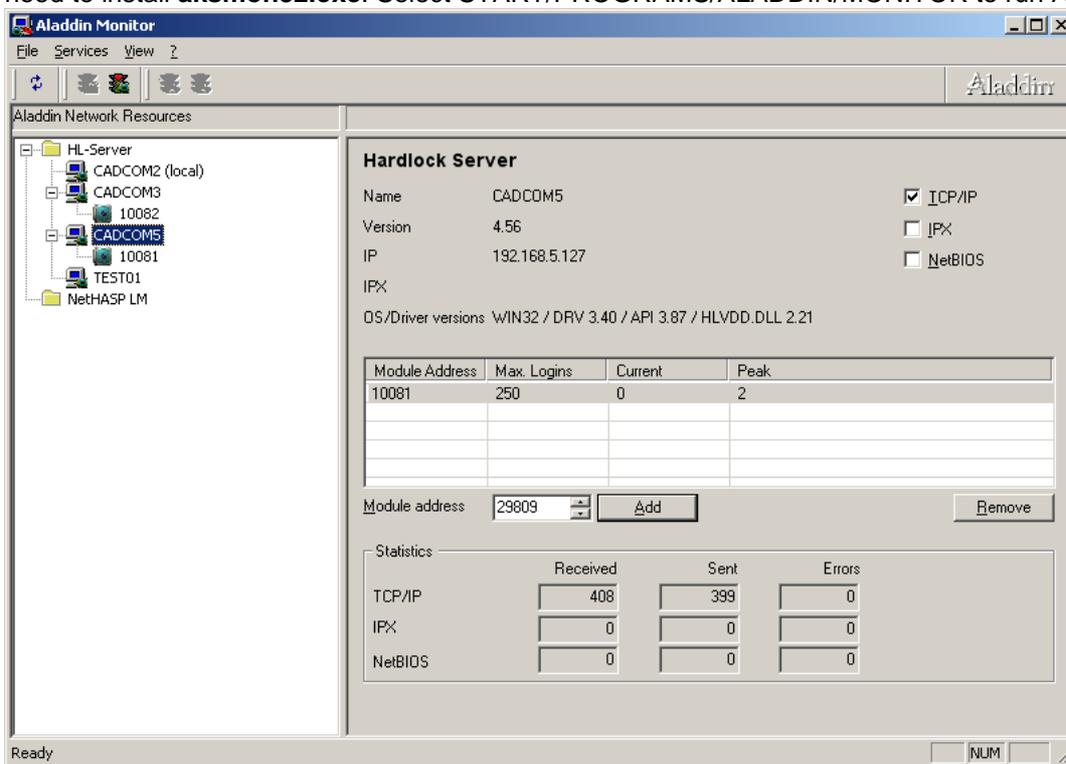
Before version 7 of MCPS the dongle called HardLock from Aladdin Knowledge Systems was delivered.

Data acquisition computer:

During the installation of MCPS all necessary drivers are installed. Therefore you need administrator rights. If the dongle cannot be found at startup of MCPS, please use the driver installation from Aladdin called **HLDRV32.exe**. It is available on the installation-CD in the directory Special\HardLock.

Clients (NetView):

To you the NetView-Dongle for MCPS clients you have to install the server software on that computer, where the dongle is connected to. This setup program is called **HLSW32.exe** and is available on the installation-CD in the directory Special\HardLock. This installs the server. To configure the server you need to install **aksmon32.exe**. Select START/PROGRAMS/ALADDIN/MONITOR to run AksMonitor:



The network will be searched and one or several computer names will be displayed in the tree view on the left side. Select your local machine.

On the right side you have to add the network dongle with module address 10081.

If you are working in different network segments it is possible that the HL-Server is not found. On the MCPS-installation-CD is the program Diagnostix.exe. With this program you can check the basic functionality of the dongle.

# Firewall

A firewall can block several functions of MCPS and has to be set up correctly.

## 1. Dongle

The communication between MCPS and the dongle (local or network) can be blocked. Please see chapter Dongle for more information regarding which type of dongle is used.

## 2. MServ

The communication between server and clients to transmit data, alarms or messages is blocked by a firewall. Please see chapter Mserv to find the correct settings depending on the transmission protocol used.

## 3. NCS

The communication between MCPS and the network control service, which can handle user administration and audittrail is blocked by a firewall. Please see chapter NCS to find the correct settings depending on the transmission protocol used.

## 4. Webserver and ECS

These functions are also using TCP/IP connections and ports, which can be blocked.

The setup of the Windows firewall can be found in the control panel. Select the window <Exceptions> to add MCPS as a program and the ports, which are used by MCPS.

# MSERV

If you have the option /Mserv MCPS can transmit online data, alarms and messages to the clients ([NetView](#)). To use this feature you have to activate the <Use MSERV> button in CONFIGURATION/ MCPS SETTINGS/ PASSWORDS. After pressing OK and calling the MCPS settings again an additional tab called <MSERV> is available. A client can define here the measurement computer to connect to. For clients and servers <ou have to define the communication type:

DCOM: This is a specific Windows communication method and was typically used in older versions of MCPS. Because of some modifications in Windows security settings since XP SP2 you have to configure many parts in Windows itself to get this type of communication working. See chapter [DCOM](#) for more information.

TCP/IP: The connection between client and server is only done by TCP/IP communication with a MCPS specific protocol. Only the firewall can block this communication. So please see chapter [Firewall](#) to create an exception for MCPS and the port used for MSERV connection.

See also CONFIGURATION / MCPS SETTINGS / MSERV.

## Workflow:

After all settings are done you have to start a project at the measurement computer. On the client you have to select [PROJECT / SELECT FROM DA SERVER](#). Choose the project to be loaded into the client (directly by the MSERV connection). Now you can open the online windows (text, trend, bar graph, alarm, messages). If a modification is made to the server project, the client project is updated (closed and reloaded again). On the MSERV configuration page you can also define search pathes to find the related historical data. Since a client may have a different path to the data than the DAQ server, you need to specify the data directory. MCPS will look for the project file, data file, alarm file and so on. If available you can also access the offline data with the MSERV project.

The alarm and message window can be set to open automatically in case of a new message / alarm. See chapter [CONFIGURATION/ MCPS SETTINGS/ MESSAGES](#).

## Netview clients

You can run additional MCPS packages on your office computers as clients to see the data of the measurement PC. Normally each MCPS has its own dongle (hardware protection module). To simplify the use of clients it is possible to have only one dongle for all clients. This is a specific network dongle, which is connected to a computer, which is always available to all clients. It is possible to use the data acquisition PC. A specific software is installed to handle this dongle and to give the clients access to it. A client password defines, how many users can log in at the same time. This mechanism is called NetView.

See chapter [Dongle](#) to get more information about your dongle and how to install the drivers.

The software driver for the network dongle is typically running as a service in Windows.

Basically the NetView-dongle is a replacement of several local dongles. A (NetView-) client can have access to the files of the server. To see online data or alarms the MCPS on the measurement computer needs the [MSERV](#) option.

# NCS

The network control service is used to handle user administration and audittrail on a server, where a client has no file access to.

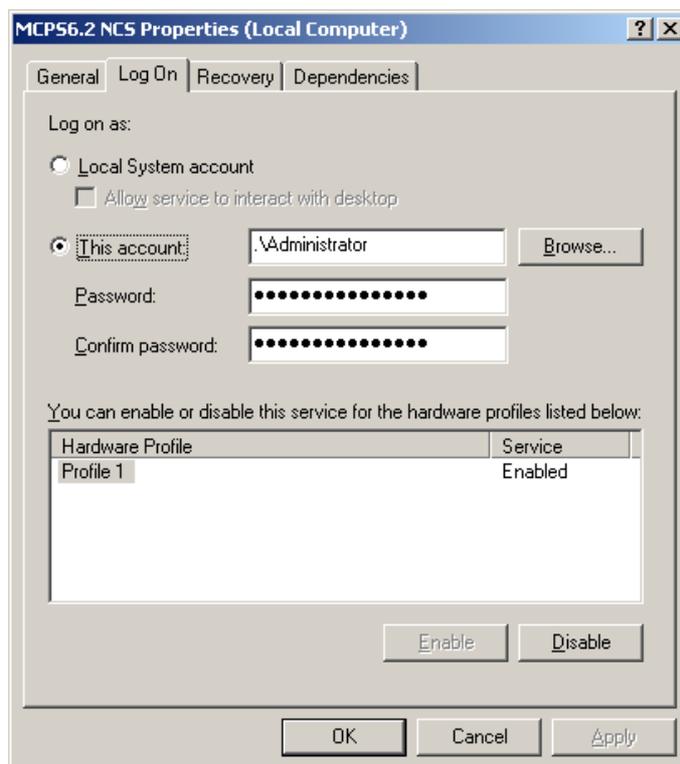
To use the service you have to start the program mcps7\_ncs\_setup.exe located on the MCPS CD in the folder INSTALLATION. After the setup has been finished the NCS is copied to your computer and a DOS box is opened. This is done to make some basic settings before running the service. The following commands are possible:

NCS -? Shows all commands and current settings.  
NCS -install This install the service with the current configuration  
NCS -uninstall This stops the service  
NCS -path:<DIR> Tells the NCS, where to find or place the user administration and audittrail files.  
NCS -TCP/IP:xxxxx The communication type is set to TCP/IP via port xxxxx  
NCS -DCOM The communication type is set to DCOM.

After the service is installed all MCPS programs must be set to NCS communication. Select CONFIGURATION/MCPS SETTINGS/ ADMIN/ MORE OPTIONS to select NCS connection.

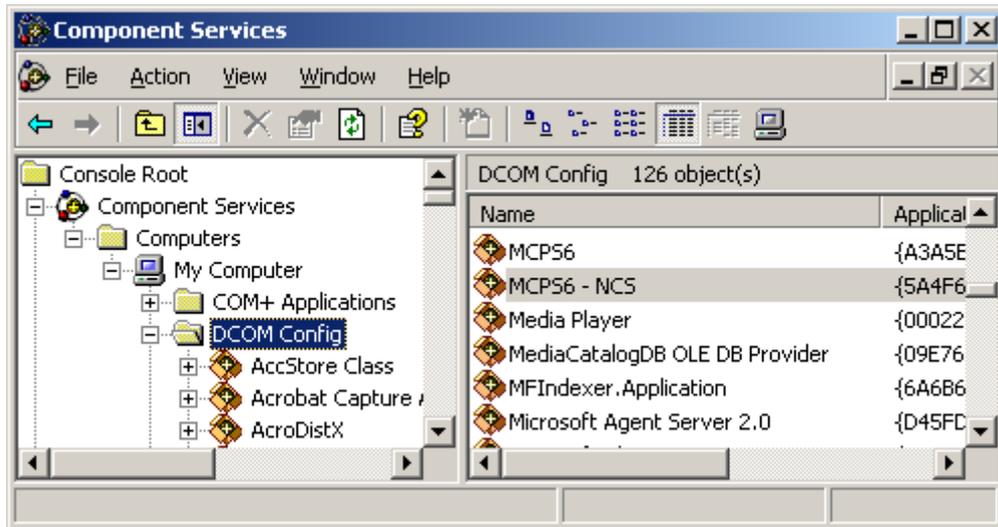
If you want to use the DCOM communication, several Windows settings must be modified, when using XP SP2 or higher:

As the first Step a security option has to be changed. To do this, please open <Control Panel>, <Administrative Tools>, <Local security Policy>, <Local Policies>, <Security Options>, <Network Access>, <Model for common exploitation and security model for local accounts>. Please change these settings to "Classic mode".



The NCS properties must be adapted too. Please click <Start>, <Settings>, <Control panel>, <Administrative tools>, <Services>. Open the NCS properties and switch to register card <Log On>. Please enable "This account" and add a user who is allowed to start DCOM – components (Minimum "Power user").

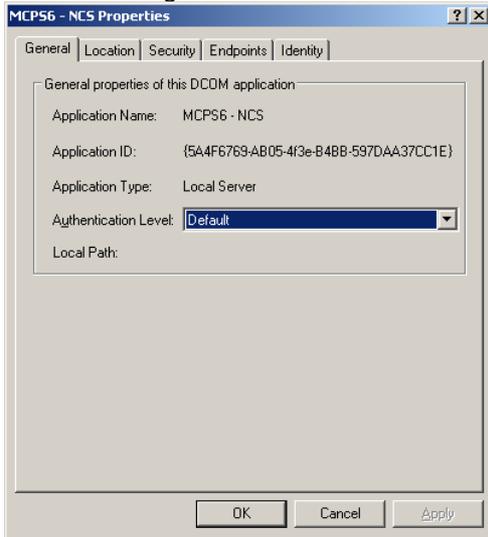
Lastly execute <Start>, <Run...>, type in „DCOMCnfg“ into the „Open“ - Line and confirm by clicking „OK“. Open <Component services>, <Computers>, <My Computer>, <DCOM Config>. Open the NCS properties and switch to register card **“Identities”**.



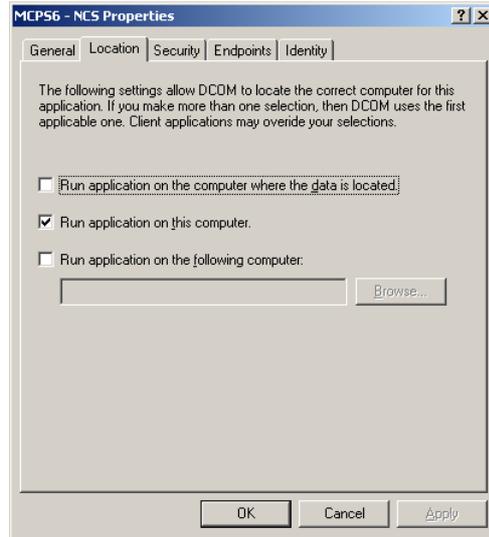
Please enable “This user” and add a user who is allowed to start DCOM – components (Minimum “Power user”).



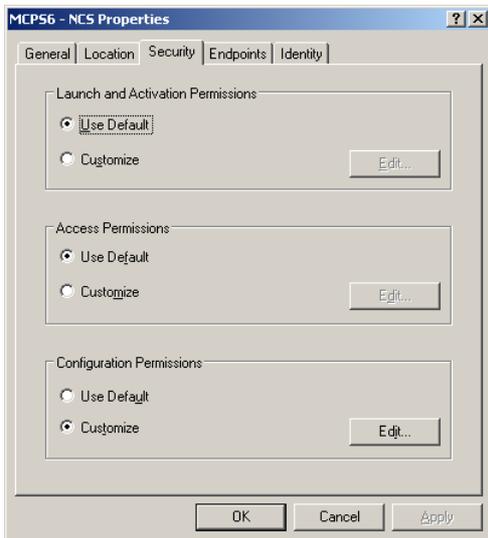
Further settings:



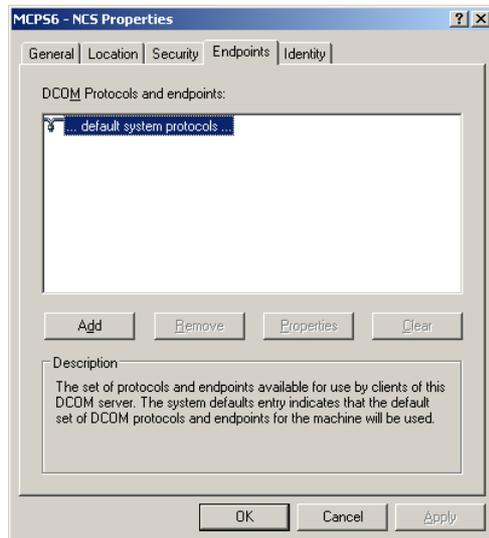
Register card "General"



Register card "Location"



Register card "Security"



Register card "Endpoints"

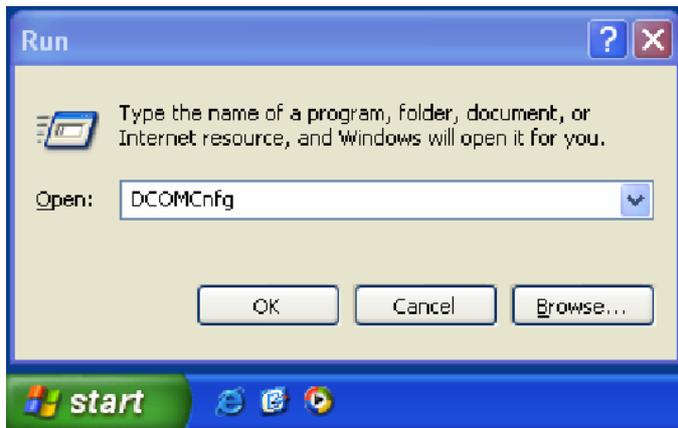
See also chapter [DCOM!](#)

# DCOM

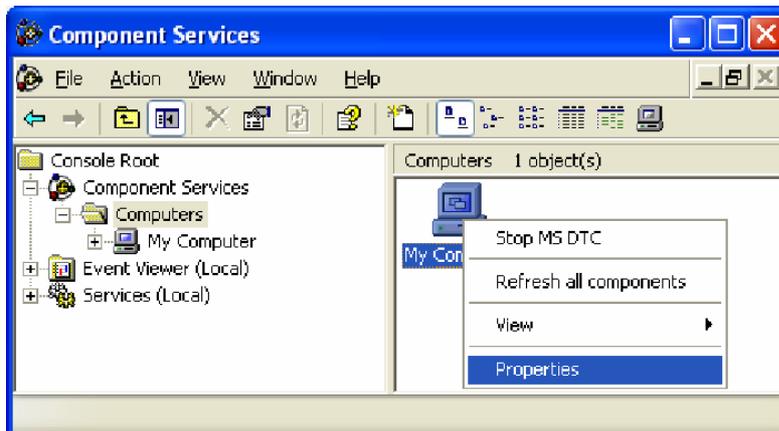
Since Windows XP SP2 and Windows 2003 Server Microsoft has changed some default settings. The result of these changes are, that some applications using DCOM (Distributed Component Object Model) do not work or do not work correct without some adaptations. These adaptations are the following:

As the first Step a security option has to be changed. To do this, please open <Control Panel>, <Administrative Tools>, <Local security Policy>, <Local Policies>, <Security Options>, <Network Access>, <Model for common exploitation and security model for local accounts>. Please change these settings to "Classic mode".

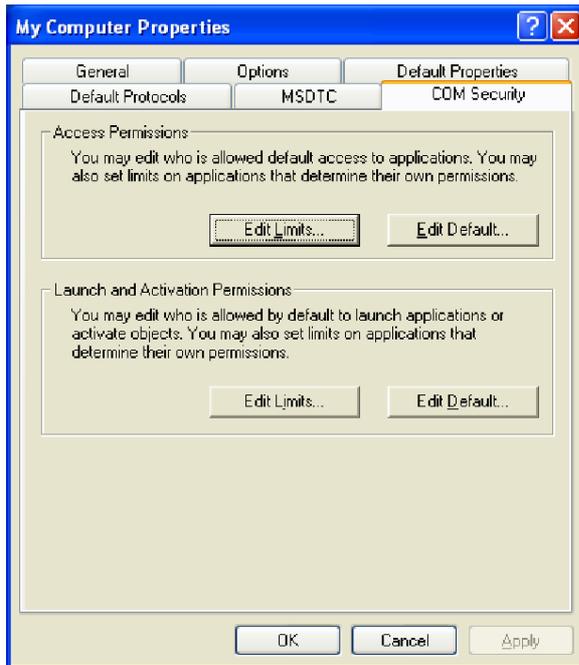
Execute <Start>, <Run...>, type in „DCOMCnfg“ into the „Open“ - Line and confirm by clicking „OK“



Open „Component services“, „Computers“.

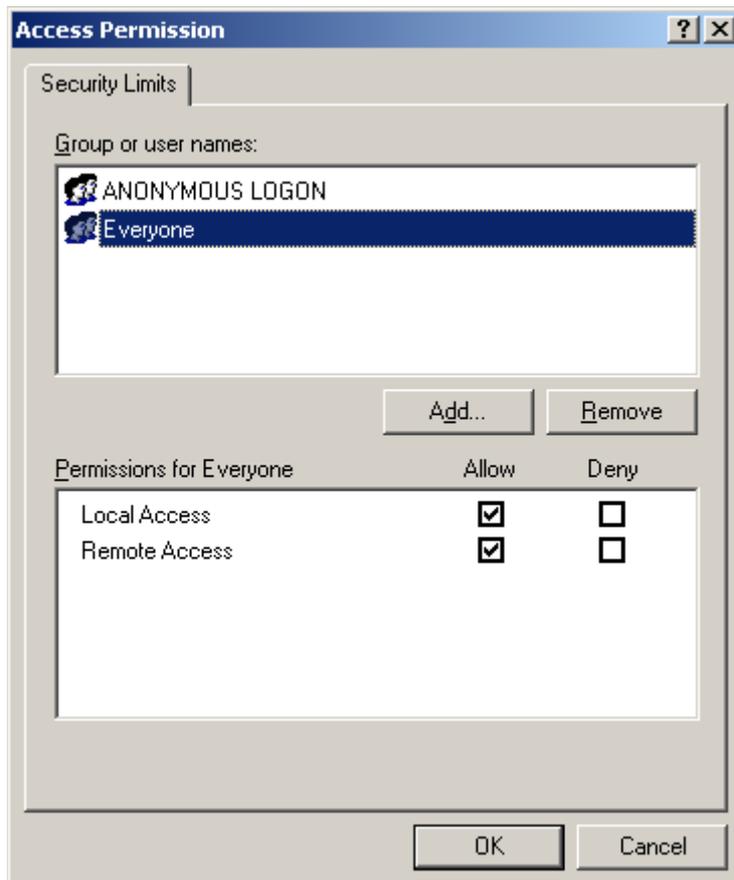
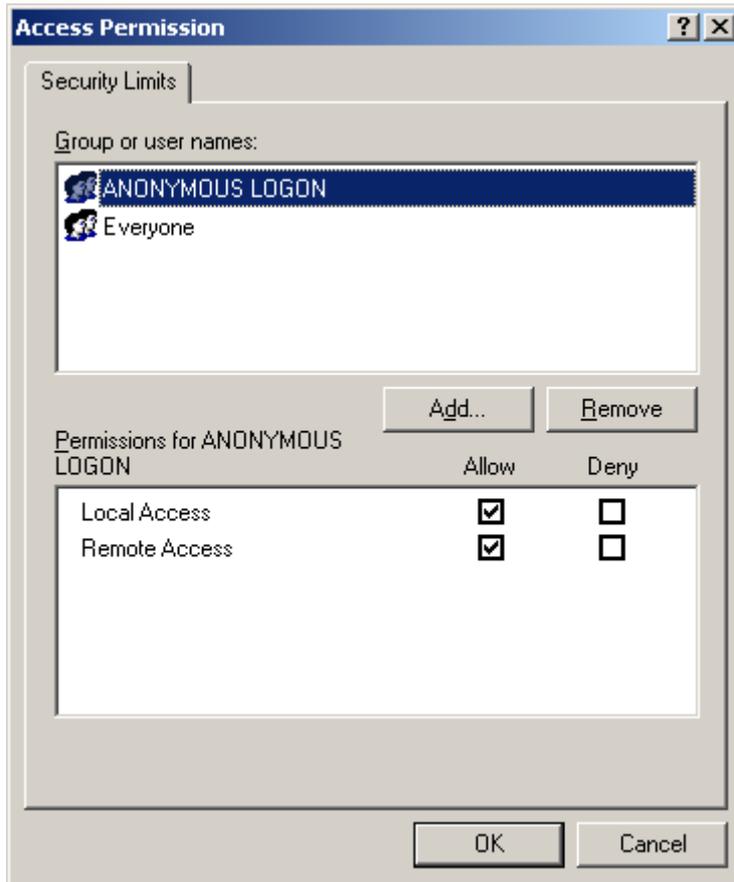


Right click on „My Computer“ and open properties. Select register card „COM security“

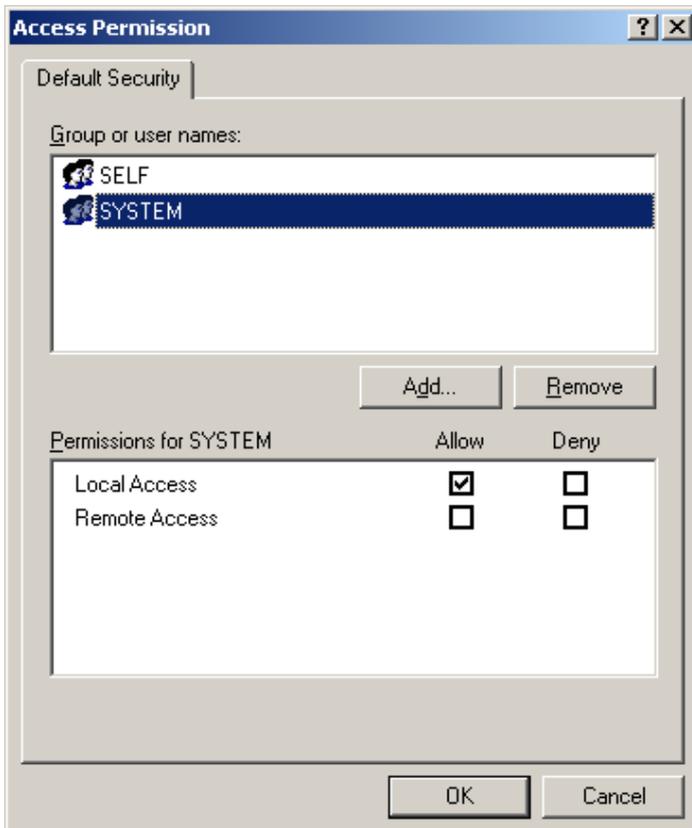
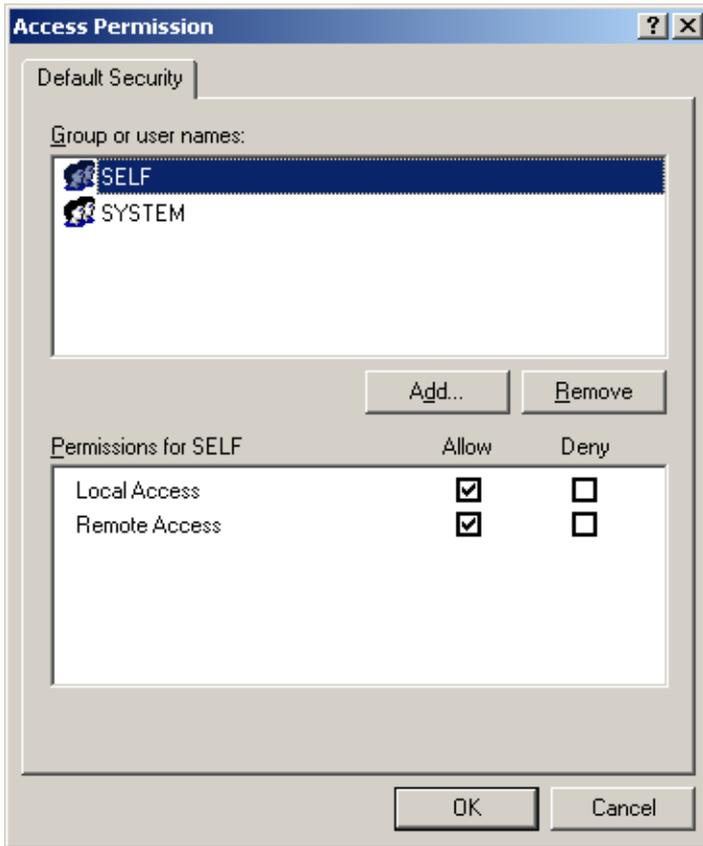


### Adaption of Access Permissions

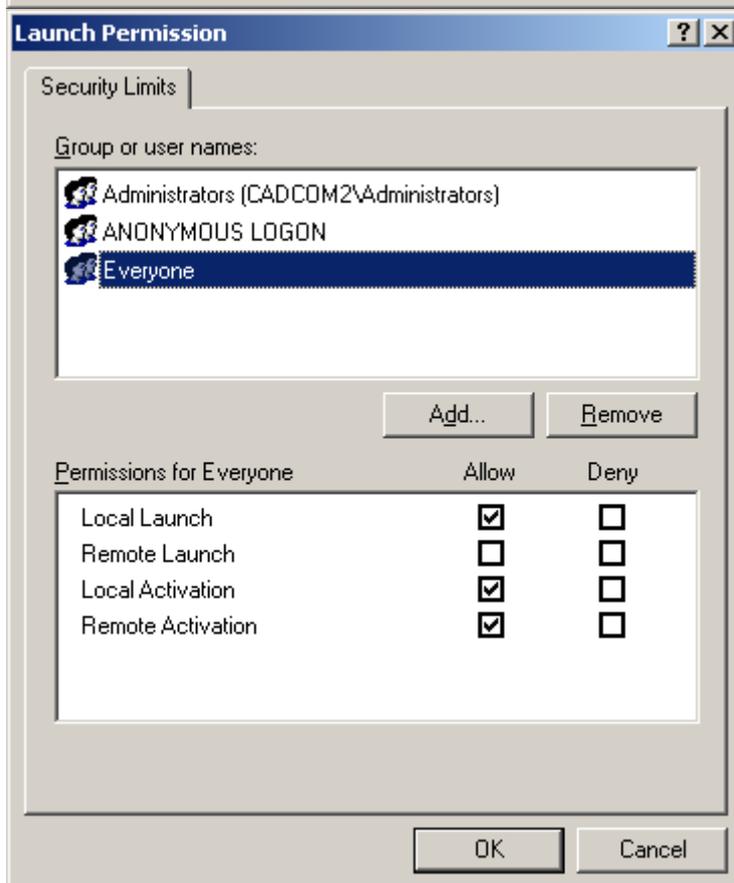
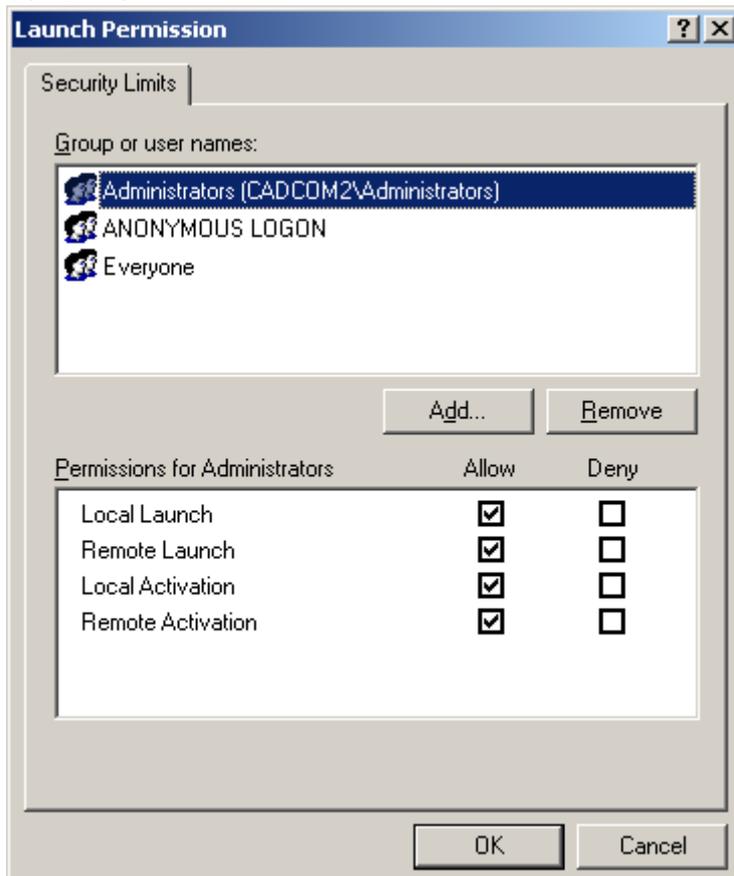
## Edit Limits

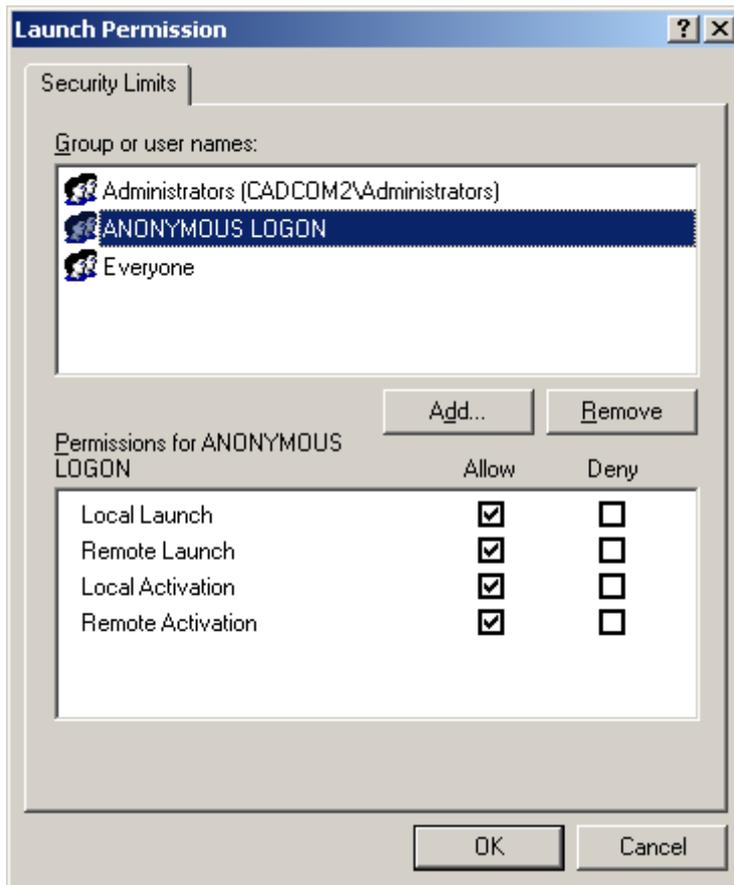


## Edit Defaults

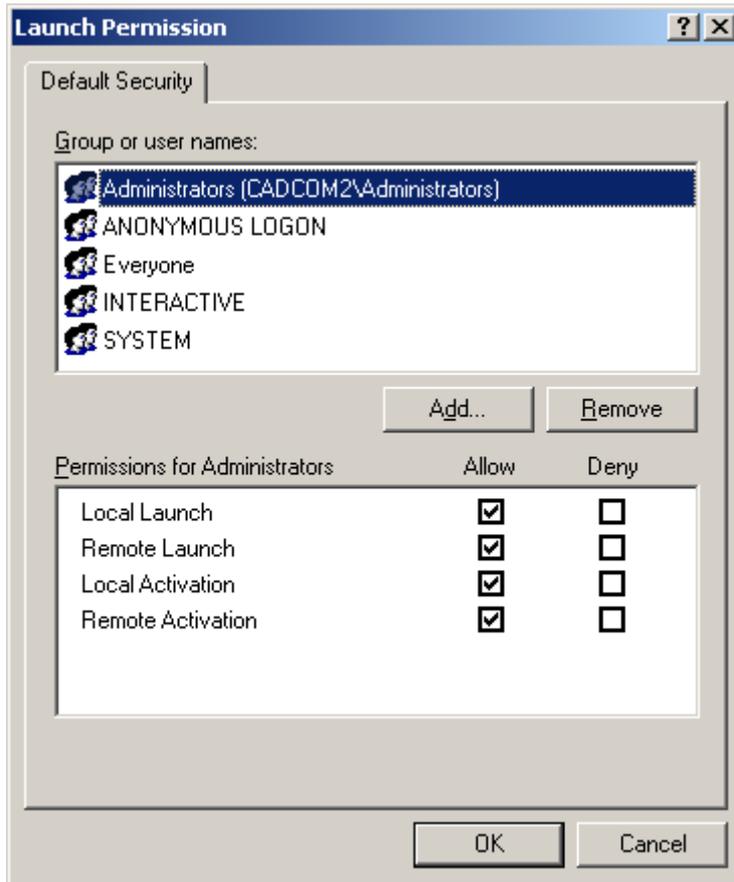


Adaption of Launch and Activation Permissions  
Edit Limits





## Edit Defaults



# Index

## A

additional project info lines 218  
Admin 129  
Alarm 131  
alarm groups 37  
alarm online window 37, 131  
Alarm settings 49  
alarm window 131  
ALLDRV 157, 158  
Autofiledatensätze verbinden 70  
AutoFiles 195  
Automation 56

## B

BACKUP ALARM CONDITIONS 107  
Batch 137  
batch input mask 11, 12  
Block operations 39

## C

calibration 109  
Channel parameters 31  
Compare AutoFile Records 104  
Create a new project 201

## D

DA manager 105  
Darwin series 159  
Data reduction 46, 196  
database 11, 27  
Database operations 27  
Directories 131  
DLL 138, 139, 140  
DLL function calls 192  
DX/MV 11  
Dynamic Alarm File 74, 200

## E

ERES 143  
expert mode 246  
Export 88  
external storage 27

## F

Fast display 202  
F-Value 184

## G

Group manager 40

## I

Import 86, 87, 201  
index saving 38  
Index saving 30  
Info about MCPS 154

Info and Signature 27  
Interface selection 180

## L

Layout 147, 148, 149  
LDAP 130  
List of functions 7  
Load batchfile 27  
Local project configuration 198  
Logfile 102  
Logout 91  
Long term trend window 61, 112  
Lotnumber 12

## M

Mathematics 181  
MAX 186  
MCON 165  
MEAN 186  
Messages 131, 132  
MIN 186  
MX100 169

## N

Network 133  
NewTopic 1 205

## O

OPC 170  
Open from DA server 67  
Owners 62

## P

password table 121  
Passwords 121  
PID-Controller 187  
PLC 165  
PMON 75, 141  
Polynomials 41, 118, 191  
Print project configuration 65  
Project data 69  
project info lines 220, 221  
project list 123  
project variable 220

## R

reference measurement 101, 109  
Remote connection 151  
Remove batchfile 27  
Request window 208

## S

S7 172  
Save project as template 72  
Script command summary 218  
Search results 27  
Sign batch 26  
Smart Quit 91  
SMS 135

Solo window 113  
Standard parameters (Project) 30  
Stop session 26  
Suspend 146  
suspend mode 110

## T

Test measurement 109  
Text window 111  
Trend window 112  
trip line 221

## U

User 130, 131  
user columns 63  
User functions 41, 118, 190  
User toolbar 153

## V

VR series 173

## W

Welcome 6  
Window manager 90  
WT series 174