An Axiometrix Solutions Brand

imc STUDIO 2023

Manual

Doc. Rev.: 6.4 - 2023-12-12

© 2023 imc Test & Measurement GmbH

imc Test & Measurement GmbH • Voltastr. 5 • 13355 Berlin • Germany

Disclaimer of liability

The contents of this documentation have been carefully checked for consistency with the hardware and software systems described. Nevertheless, it is impossible to completely rule out inconsistencies, so that we decline to offer any guarantee of total conformity.

We reserve the right to make technical modifications of the systems.

Copyright

© 2023 imc Test & Measurement GmbH, Germany

This documentation is the intellectual property of imc Test & Measurement GmbH. imc Test & Measurement GmbH reserves all rights to this documentation. The applicable provisions are stipulated in the "imc Software License Agreement".
The software described in this document may only be used in accordance with the provisions of the "imc Software License Agreement".

Open Source Software Licenses

Some components of imc products use software which is licensed under the GNU General Public License (GPL). Details are available in the About dialog.

A list of the open source software licenses for the imc measurement devices is located on the imc STUDIO/imc WAVE/imc STUDIO Monitor installation medium in the folder "*Products\imc DEVICES\OSS*" or "*Products\imc DEVICEcore\OSS*" or "*Products\imc STUDIO\OSS*". If you wish to receive a copy of the GPL sources used, please contact our tech support.

Notes regarding this document

The manual for imc STUDIO is available both as an e-book and in PDF format. In the menu bar, click on the symbol to open the help.

The program "*imc Help and Documentation* 70" provides you with access to all formats and other documents such as the **device documentation** or the **technical data sheet**.

What is the best way to read the imc STUDIO documents?

Getting Started

Before installing the software, start by reading the document "*Getting Started*". This contains important tips for achieving problem-free installation, for updating the software, as well as for operation and integration of devices in measurement setups.

All information presented in the document "*Getting Started*" is also presented in the imc STUDIO manual.

Manual - complete documentation

The manual serves as reference material. This document describes how to operate the software and how to configure parameters for the devices. In some case, individual device groups have special functions. The documentation for these usually appear exclusively in the device manuals.

Tutorials

Some chapters come with supplemental tutorials 71. We recommend practicing these tutorials as an "accompanying" introduction to the respective topic.

If you are working for the first time with imc STUDIO, we recommend that you work through the guided first steps 37 before beginning any tutorials.

Training programs for introduction to the systems, and advanced workshops

We recommend that before you begin working with imc STUDIO you participate in an extensive training session. Such training will enable you to get started working efficiently much faster. Additionally, you will obtain valuable tips and information on how to use the software more effectively. More information is available on our homepage under "*Service & Training*" > "*imc ACADEMY*".

Special notes

Warning

Warnings contain information that must be observed to protect the user from harm or to prevent damage to property.

Note

Notes denote useful additional information on a particular topic.

Reference

A reference in this document is a reference in the text to another text passage.

Table of contents

1 General introduction ... 8
1.1 Before you start ... 8
1.2 Tech support ... 8
1.3 Service and maintenance ... 9
1.4 Legal notices ... 9
1.5 imc Software License Agreement ... 10
2 Chapter overview ... 13
3 Setting Up - Software ... 15
3.1 System requirements ... 16
3.2 Installation - Preparation ... 17
3.3 Installation - Step by step ... 21
3.4 Product configuration / Licensing ... 29
3.5 Start ... 31
3.6 Training - Guided first steps ... 37
3.7 Info / Version information ... 39
3.8 Information and tips ... 40
4 Setting Up - Connect the device ... 45
4.1 Connecting via LAN in three steps ... 46
4.2 Special options for connecting to the device ... 48
4.3 Information and tips ... 61
4.4 The network ... 62
4.5 Firmware version ... 63
5 Documentation of devices - Help and Documentation ... 70
6 Tutorials ... 71
6.1 Simple measurement - First steps ... 72
6.2 Circular buffer memory for the display ... 79
6.3 Subsequent saving of measured data ... 83
6.4 Simple triggering ... 87
6.5 Simple triggering - Enhanced: Trigger status ... 93
6.6 Triggered measurement ... 96
6.7 Thermocouple measurement ... 102
6.8 Artificial sine - signal generation using imc Online FAMOS ... 106
7 imc STUDIO (general) ... 110
7.1 Experiments, projects and the database ... 110
7.2 Ribbon ... 121
7.3 Navigation pane and Quick access toolbar ... 139
7.4 Tool windows ... 140
7.5 User administration and Access rights ... 146
7.6 Views ... 155
7.7 Information and tips ... 161

© 2023 imc Test & Measurement GmbH

7.8 Placeholders ... 166
8 Setup - Device (general) ... 186
8.1 Device overview ... 187
8.2 Ribbon ... 188
8.3 Tool Windows ... 228
8.4 Operation ... 231
8.5 Performing a measurement: Procedure ... 241
8.6 Metadata ... 244
8.7 Information and tips ... 246
8.8 Troubleshooting ... 286
9 Setup pages - Configuring device ... 288
9.1 Documentation (experiment description) ... 289
9.2 Configuring devices ... 289
9.3 Configuring channels and variables ... 339
9.4 Channel balance ... 377
9.5 Trigger and events ... 384
9.6 Additional pages ... 407
9.7 Storage options and directory structure ... 411
9.8 Device hard disk, removable drive ... 443
9.9 Measurement types ... 454
9.10 Fieldbusses ... 474
9.11 Special modules and Third party devices ... 716
9.12 Sensors, characteristic curves and TEDS - for imc DEVICES devices ... 729
9.13 imc SIMPLEX - Sensors for imc DEVICEcore devices ... 742
9.14 Information and tips ... 751
10 Setup - Advanced Device Functions (imc DEVICES) ... 769
10.1 imc Display editor ... 769
10.2 imc Messaging ... 786
10.3 Configure via FTP ... 809
10.4 imc REMOTE WebServer ... 822
11 Data analysis and signal processing ... 843
11.1 Overview of the tools ... 843
11.2 imc Online FAMOS and imc Inline FAMOS ... 851
11.3 Inline Analysis ... 1064
11.4 imc FAMOS dialog ... 1088
11.5 Data Processing (general) ... 1098
11.6 Bus Decoder - Decoding fieldbus log channels ... 1101
11.7 Powertrain Monitoring - Machine and bearing diagnostics ... 1106
12 Channel types, variable types, data types and data storage formats ... 1130
12.1 Data types ... 1130
12.2 Various variable types - Channel type ... 1134
12.3 Data file format - Saving measured data and export/import formats ... 1152

12.4 imc Data format 1190
12.5 imc Format Converter 1191
13 Panel - Data visualization and controls 1205
13.1 Ribbon 1206
13.2 Tool windows 1211
13.3 Context menu 1231
13.4 Design mode 1234
13.5 Widgets - operation and properties 1235
13.6 Curve window 1265
13.7 Special Widgets 1521
13.8 Pages 1535
13.9 Variable linkage 1550
13.10 Information and tips 1554
14 Automation - Real time test automation 1557
14.1 Getting started 1559
14.2 Operation 1564
14.3 Tool window Automation elements 1569
14.4 Task management 1573
14.5 Editors 1584
14.6 Templates - Elements for the editors 1588
14.7 Functions 1611
14.8 Information and tips 1622
14.9 Tutorial 1628
15 Sequencer, events and commands 1681
15.1 Sequencer page 1682
15.2 Tool window commands 1683
15.3 Sequence table 1684
15.4 Event Dialog - Panel and Automation 1686
15.5 Running and stopping 1688
15.6 Creating a sequence of commands 1690
15.7 Context menu 1692
15.8 Events 1694
15.9 Useful options 1706
15.10 Information and tips 1706
15.11 Command reference 1706
15.12 Tutorial 1765
16 Video interface 1802
16.1 imc STUDIO GoPro 1802
17 Programming interface 1808
17.1 Scripting 1808
17.2 Third Party Device Interface 1917
17.3 API 1946

18 Miscellaneous 1965
18.1 Last Changes 1965
Index 1981

1 General introduction

1.1 Before you start

Dear user.

1. The software you have obtained, as well as the associated manual are directed toward competent and instructed users. If you notice any discrepancies, we request that you contact our tech support 8.
2. Updates during software development can cause portions of the manual to become outdated. If you notice any discrepancies, we request that you contact our tech support.
3. Please contact our tech support if you find descriptions in the manual which you believe could be misunderstood and thereby lead to personal injury.
4. Read the enclosed license agreement 10. By using the software, you agree to the terms and conditions of the license agreement.

Note Notes on the descriptions and the screenshots

- The help may also contain parts **shared imc software components**. These parts may differ from the rest of the help in terms of style and structure. All help files are equipped with a full text search functionality and have an index.
- The screen shots appearing in this documentation were created with a **variety of Windows versions** and their appearance may thus differ from that of your installed program.
- imc STUDIO works with user groups and access rights 146. The **user group** has wide-reaching influence on the visibility and/or operability of menus, icons, etc. Note that in all imc STUDIO documents the administrator roll is assumed to be filled. All descriptions are therefore related to the full utilization of the user interface.
- The following descriptions and screenshots always pertain to the **view with the full scope of functions** (*Complete*). Many of these functions are also found in the minimized view at another location.

1.2 Tech support

If you have problems or questions, please contact our tech support:

Phone:	(Germany):	**+49 30 467090-26**
E-Mail:	hotline@imc-tm.de	
Internet:	https://www.imc-tm.com/service-training/	

Tip for ensuring quick processing of your questions:

If you contact us **you would help us**, if you know the **serial number of your devices** and the **version info of the software**. This documentation should also be on hand.

- The device's serial number appears on the nameplate.
- The program version designation is available in the About-Dialog (click on the symbol in the menu bar).

Product Improvement and change requests

Please help us to improve our documentation and products:

- Have you found any errors in the software, or would you suggest any changes?
- Would any change to the mechanical structure improve the operation of the device?
- Are there any terms or explanations in the manual or the technical data which are confusing?
- What amendments or enhancements would you suggest?

Our tech support 8 will be happy to receive your feedback.

1.3 Service and maintenance

Our service team is at your disposal for service and maintenance inquiries:

E-Mail: service@imc-tm.de

Internet: https://www.imc-tm.com/service

Service and maintenance activities include, for example calibration and adjustment, service check, repairs.

1.4 Legal notices

Quality Management

imc Test & Measurement GmbH holds DIN EN ISO 9001 certification since May 1995 and DIN EN ISO 14001 certification since November 2023. You can download the CE Certification, current certificates and information about the imc quality system on our website: https://www.imc-tm.com/quality-assurance/.

imc Warranty

Subject to the general terms and conditions of imc Test & Measurement GmbH.

Liability restrictions

All specifications and notes in this document are subject to applicable standards and regulations, and reflect the state of the art well as accumulated years of knowledge and experience. The contents of this document have been carefully checked for consistency with the hardware and the software systems described. Nevertheless, it is impossible to completely rule out inconsistencies, so that we decline to offer any guarantee of total conformity. We reserve the right to make technical modifications of the systems.

The manufacturer declines any liability for damage arising from:

- failure to comply with the provided documentation,
- inappropriate use of the equipment.

1.5 imc Software License Agreement

imc Test & Measurement GmbH
Voltastr. 5
13355 Berlin
Commercial register: Berlin-Charlottenburg HRB 28778
Managing director: Michael John Flaherty

imc Test & Measurement GmbH
Terms and Conditions
Governing the Use of imc Test & Measurement GmbH Software
As of: October 1, 2023

§ 1 Objects of the Agreement

(1) In addition to the "General Terms and Conditions Governing imc Test & Measurement GmbH Deliveries and Services to Customers", these terms and conditions apply to all contracts concluded with imc Test & Measurement GmbH (hereinafter referred to as "imc") which involve the transfer of rights of use to any software developed by imc (standard software, software created or adjusted specifically for the Customer, which is recorded on the machine-decodable data carriers such as data files, databases and database material, updates, upgrades, releases, etc., including corresponding documentation, information and materials, hereinafter referred to as "Software").

(2) The Software is provided to the Customer as an executable object program on machine-decodable data carriers specified in the "Objects of the Agreement". The Software's product documentation is also supplied to the Customer either in print or on a machine-decodable data carrier. Unless otherwise expressly agreed in writing, the Customer is not issued the source code of the Software.

§ 2 Rights of Use, Scope

With regard to any transfer of rights of use to Software created by imc, the following provisions apply:

(1) Basic provisions

a) The Customer is granted a non-exclusive and – subject to the terms and conditions governing the use of Software by third parties, resale and leasing – non-transferrable right of use to the Software for its own purposes. "Use" signifies running the programs and editing the data records.

b) Until each due fee is paid in full, the Customer is entitled to use the Software solely on a revocable basis. If the Customer is in default with regard to the payment of fees, imc is entitled to revoke the use of the respective services for the duration of the default. The Customer is granted the permanent right to use copyright protected services, in particular the Software, only upon full payment of the agreed fee.

c) The Customer agrees to undertake appropriate precautionary measures to prevent unauthorized access by third parties to the Software. The original data carriers and the data carries used to make copies as per the agreement, as well as the documentation, are to be stored in a secure location. Employees are to be notified that the production of copies beyond the scope of the agreement is not permitted.

d) If the right of use is revoked or expires due to another reason, the Customer is obligated to return to imc the Software, the copies made by the Customer and the documentation. Provided that a physical return of the Software and the copies is not possible due to technical reasons, the Customer is obligated to delete such and confirm deletion to imc in writing.

(2) Reproduction

a) The Customer is entitled to make copies of the Software only if copies are necessary to use the Software in accordance with the contract. The following are considered cases in which reproduction is necessary: installation of the Software from the original data carrier onto the hard disk drive of the hardware used, as well as loading the Software into the computer memory.

b) The Customer is entitled to create a backup copy if such is necessary to safeguard future use. Copies may only be made for other purposes after prior written consent has been issued by imc.

c) The Customer is not allowed to make any reproductions other than those expressly permitted under the provisions of this agreement.

(3) Use of the Software by Third Parties, Resale and Leasing

a) The Software may be used for the purposes stipulated in this contract, in particular for the Customer's business operations. Access to the Software may also be provided to parties which rely on using the Software as instructed by the Customer. In particular, the Customer is entitled to operate the Software or allow the Software to be operated on data processing devices, which are located on the premises of and are directly owned by a third party company (outsourcing). The prohibition against multiple use remains unaffected.

b) The Customer may permanently sell or give the Software to third parties provided that the Customer is granted permanent use of the Software. In the context of its period of use, the Customer may temporarily transfer the Software to third parties for a fee or free of charge. The prohibition against multiple use remains unaffected. The Customer is expressly notified that transfer to third parties is not permitted and use by third parties is technically not possible if an individual license must be acquired or an individual activation is required for third party usage, such as in the case of runtime licenses.

c) With regard to the valid use of Software by a third party, the Customer is obliged to ensure that the third party acknowledges the provisions of this agreement governing the rights of use as binding for such third party. The Customer may not transfer Software and documentation to third parties if there are grounds to suspect that the third party may infringe upon the provisions of this agreement governing the rights of use, in particular with regard to the unauthorized production of copies.

d) Subject to the provisions stipulated in § 4 Paragraphs 1 and 2 or a deviating express agreement in writing, the Customer may not use the Software while the Software is being used by a third party (prohibition against multiple use); in the event that the Software is transferred to the third party, the customer is obliged to surrender to imc all Software copies including, if applicable, all existing backup copies, or to destroy copies not surrendered.

(4) Decompilation

The reverse translation of the provided program code into other code forms (decompilation), disassembling and other forms of reverse engineering of the various production phases of the Software is not permitted. If interface information is required to achieve the interoperability of a separately created computer program, such may be requested from imc, or a third party to be named by imc, for a minor fee. Section 69 e of the German Copyright Act ("UrhG") remains unaffected by this provision.

(5) Changes by imc

If imc conducts adjustments, changes or enhances the Software on behalf and on account of the Customer, the Customer thus acquires the corresponding rights of use to the changes or enhancements of the Software to which he is entitled according to the stipulations of this agreement.

(6) Exceptional Usage Requests by the Customer

If the Customer requests to use the Software according to terms which deviate from the requirements stipulated in Paragraphs 2 through 5, this exceptional use of the Software must be agreed in writing by imc. In such an instance, the Customer agrees to provide imc with information about the desired scope of use, the pertinent field of application, etc. If imc subsequently grants a license covering the Customer's special intended use, the parties agree that a new license fee is owed by the Customer, which is independent of payments made by the Customer for the previously existing license.

§ 3 Copyright, Protection of the Software

(1) The intellectual property, in particular the copyright as well as all industrial property rights and trade secrets, are retained by imc and are not transferred to the Customer. The Customer's ownership of the machine-decodable data carries and data processing units remains unaffected.

(2) Copyright notices, serial numbers as well as designations and reservations of rights which serve as program identification or a protective right may not be removed or changed. The Customer is obliged to transfer the existing protective right notices to all copies. In particular, backup copies of the Software must be expressly designated as such.

§ 4 License Types, Multiple Use

(1) In the case of a Single-User License, the Software may be activated and run on only one data processing unit. "Activation" refers to the process of transferring the license to the data processing unit.

If the technical specifications for the Software permit a second activation, then the Customer may additionally activate the Software on a second data processing unit. However, the Software may only run on one data processing unit at any one time, not on both simultaneously.

(2) With a Network License, the Software may be run on as many data processing units as the amount of licenses obtained. In this case a central data processing unit acts as the license server for which the activation process is performed.

If the technical specifications for the Software permit a second activation, then the Customer may additionally activate and run the Software on as many data processing units as the amount of licenses obtained. However, these additional data processing units must be used by the same users who operate the Software via the license server.

(3) Subject to the provisions in Paragraphs 1 and 2 or a deviating express agreement in writing regarding network use, multiple use of the Software is not permitted.

(4) If the data processing unit is changed, the Customer is obliged to delete the Software from the hard disk drive of the previously used hardware.

§ 5 Software-Subscription

If the Software used is a software-subscription, the following additional restrictions apply:

(1) The right of use is valid for a limited time period. The start and end of the time period are specified. After the end of the time period, the right of use is expired.

(2) If the Customer wishes to continue using the Software after elapse of the specified time period, the subscription must be renewed.

§ 6 Trial Version

If the Software used is a free trial version, then the following additional limitations apply:

(1) The trial version only entitles the user to test the Software. In particular, commercially productive utilization is not permitted.

(2) The rights of use granted expire after the elapse of a period stated in the product description.

§ 7 License Key

(1) Upon delivery of the Software the Customer receives a License Key. Using this License Key, the Customer is able to activate the Software purchased. By means of this License Key the Customer can also view his license status and order updates and upgrades.

(2) The License Key is to be protected against access by third parties in order to prevent misuse. If, however, a third party gains unlawful access to the Key, the Customer is obliged to notify imc immediately via telephone, as well as in writing, so that the previous License Key may be suspended and a new one issued.

§ 8 Conclusion

(1) The law of the Federal Republic of Germany shall apply under exclusion of private international law. The provisions of the UN Convention on Contracts for the International Sale of Goods (CISG) do not apply.

(2) The place of performance for all obligations arising from this agreement is imc's registered seat. Insofar as the Customer is a merchant as defined by the German Commercial Code (HGB), a legal entity under public law, or a special asset under public law, the exclusive place of jurisdiction for all disputes directly or indirectly arising from the contractual relationship is agreed as imc's registered seat. The same applies to persons who have no general place of jurisdiction in Germany, as well as to persons who have moved their place of residence or usual whereabouts abroad since conclusion of the contract, or whose place of residence or usual whereabouts is unknown at the time the action is filed. In addition, imc is entitled to file suit at the statutory venue.

(3) Oral side-agreements are not valid. Deviating or supplementary conditions as well as modifications of this contract, including this written requirement clause, are only valid if agreed in writing and expressly marked as a modification or supplement.

(4) If certain provisions of this contract are inoperative or unfeasible, this does not prejudice other provisions of the contract. The contracting parties agree to contractually substitute an operable provision which approximates the commercial intention of the contract as closely as possible for any inoperable one.

© 2023 imc Test & Measurement GmbH

2 Chapter overview

In this manual, you will find a detailed description of how to operate the imc STUDIO software.

Section	Synopsis
General introduction	Quality Management 9, Tech support 8, License agreement 10
Setting Up 15	Software setup operations • System requirements 16, Installation 17, Start 31
Setting Up - Connect the device 45	Connecting via LAN 46 and Optimize the network 62
Tutorials 71	• **The first run 37 - What steps are necessary in order to obtain measured data?** • **Tutorials 71 - Begin with the tutorial on device configuration and measurement data display.**
imc STUDIO (general) 110	The imc STUDIO user interface and its operation • Project Menu 121, Creating an experiment 116 and "What is saved where?" 162
Setup - Device (general) 186	The device configuration • Performing a measurement 241 and Devices supported (Device Overview) 187 • Metadata, additional columns 247
Setup pages - Configuring device 288	The device configuration • Configuring device (Setup pages) 288 • Synchronization 292, Storage options 411
Setup - Advanced Device Functions (imc DEVICES) 769	Additional functions of the devices of firmware group A (imc DEVICES); including: • Fieldbusses 474
Data analysis and signal processing 843	Various possibilities for processing and analysis of measured data • during a running measurement on the device: imc Online FAMOS 851 (function reference 896) • during a running measurement on the PC: Inline FAMOS 851, Inline Analysis 1064, Bus Decoder 1101, Powertrain Monitoring 1106 • subsequent analysis (post-processing): imc FAMOS 1088
Channel types, variable types, data types and data storage formats 1130	In imc STUDIO, the various components work with variables, and some work with files as well. Variables can take a variety of formats and types. This chapter provides an overview of the various possibilities and applications. • Data types 1130 • Various variable types - Channel type 1134 such as User-defined variables 1149 • Data file format - Saving measured data and export/import formats 1152 such as Parameter set 1159
Panel 1205	Display of measured data and operation via the user interface • Curve Window 1265 and Widgets instruments) 1235 • Variable linkage 1550 ans Data Browser 1214
Automation 1557 **Tutorials** 1628	Editor for the Tasks 1584 (routines) for open- and closed-loop control operations Getting started 1559 and Process 1561

Section	Synopsis
Sequencer, events and commands 1681 **Tutorials** 1765	Creating automated routines • Sequence table 1684 and Creating a sequence of commands 1690 • Events 1694 (event handling) and User-defined events 1696 • Command reference 1706
Programming Interface	Scripting 1808: Interface for using Scripts in imc STUDIO (Tutorial 1908). Third Party Device Interface 1917: Interface for using third party devices in imc STUDIO (Tutorial) 1942. API 1946: Interface for developing custom application with access to imc STUDIO functions.

Tech support

If you have any open questions about our products, please contact our tech support.

Questions or problems? Contact our Tech support 8.

3 Setting Up - Software

This chapter describes the **first steps** in operating imc STUDIO and the **installation of additional imc products**.

imc STUDIO is the **common framework** forming a **product package** through the combination of modular components (plug-ins).

Which components are available depends on the product installation (order).

Chapter overview

Synopsis	Section
Startup operations for imc STUDIO - **installation** and **product configuration** prior to first use	• System requirements 16 • Installation / Uninstall 17 • **Product Configuration / Licensing** 29
The first start, important settings	• **The first start** 31 • Device connection / Network / Firewall 33
Guided first steps - What steps are necessary in order to obtain measured data?	• Training - Guided first steps 37
List of components used and their versions	• Version Information 39

3.1 System requirements

Supported operating systems
Windows 10*/11* (64 bit)

*released in conformance with the version of Windows 10/11 applicable at build date of imc software

Minimum requirements for the PC
4-core CPU 2 GHz [1]
8 GB RAM (recommended: 16 GB RAM) [1]
10 GB free hard disk space (recommended: SSD) [2]
Display resolution: 1366 x 768 (recommended: 1920x1080)

1 Requirements and recommended minimum configurations for the applied PC will rise with increasing number of connected devices and the resulting overall system data rate. Another crucial factor will be the extend of used live analysis and visualization functions on the PC (in particular Data Processing, imc Inline FAMOS and active curve windows).
A smooth operation will also heavily rely on sufficient RAM memory resources on the PC: It must ensure to satisfy all vital functions without any outsourcing to slow external swap file (HDD/SSD) in order to guarantee sustainable performance.

2 The hard drive memory requirement increases if measured data are saved to the PC.

Other operating system components

The following components are installed with the imc STUDIO setup if they are not already present:

Component	Version	Folder in "System" directory
Microsoft .NET Framework	4.8	DotNetFx4.8
Microsoft VC 2015-2019	14.28.29910	Microsoft Visual C++ Redistributable\2019
Microsoft VC 2010	10.0.402219.1	Microsoft Visual C++ Redistributable\2010
Microsoft VC 2005	6.0.3790.0	Microsoft Visual C++ Redistributable\2005
Microsoft Build Tools 2015	14.0.23107.10	MSBuildTools2015

Supported Measurement Devices

Which devices you can use in imc STUDIO is described in the documentation on the "*Setup*" > "*Device Overview* 187", as well as in the "*Technical Data Sheet*".

3.2 Installation - Preparation

Software requires a license

This means that the program may only be started **upon obtaining a license** (see Product Configuration / Licensing 29).

Administrator rights required

For the purposes of installation and uninstallation, a user account with **administrator privileges for the PC is required**.

If you are logged on to the PC **without administrator rights**, **log yourself out** and log back on with an administration-level user account. If you don't possess the appropriate account type, you will need the support of your system administrator or IT department.

See also Notes on Windows User Account Control 20.

Restarting the computer in the process of installation

During the installation process, the installation program will prompt you to restart the PC.

Note Restarting

After the restart, login to **the same user account** with which you had previously begun the installation.

Coexisting applications: imc STUDIO, imc STUDIO Monitor, imc WAVE, ...

Some imc programs are installed as an independent and specially adapted instance of imc STUDIO. They are based on imc STUDIO.

Unless otherwise stated, these programs are installed and used in parallel. As long as these instances are based on the same version of imc STUDIO (e.g. 2022 R1), they are all subordinated to the same program installation, which means they share resources.

For this reason, installation of the instances in this case must be performed in one single shared Setup procedure. Any attempt to perform installation in succession or subsequently will cause the already existing instances to be deleted.

This applies particularly to the joint and parallel installation of imc STUDIO Monitor, imc WAVE and imc STUDIO, which must be performed in a single step.

Do not install imc STUDIO, imc WAVE and imc STUDIO Monitor in succession, but always at the same time.

Update or parallel installation

The Setup utility checks whether any version of imc STUDIO is currently installed on the computer. If so, it can be uninstalled by means of the Setup. An associated confirmation prompt is posted in that case. All user data such as the database remain intact.

The **two versions** can be **installed in parallel** provided that they have differing version numbers (e.g. 5.2 and 2023). The new version can be installed in the same folder (default case: "C:\Program Files\imc"). In this folder, a new subfolder with the new version number is created for imc STUDIO.

In both cases, you are able to **adopt** a variety of **settings** from the old versions. These include the project settings and views. By contrast, other settings such as the product configuration and the database path need to be set up from the beginning.

Adopting the settings with the help of the existing database (recommended)

When an existing database continues in use, any settings which can be retained will be retained. For a parallel installation, two data bases are needed. Create a copy so that the old imc STUDIO version can continue to work with the existing database. A confirmation prompt about making a copy is posted.
See: Update with the help of the existing database 19.

Adopting the settings without using the existing database

Some settings, e.g the views, can also be retained without the database. To do this, save and import the appropriate settings.
See: Update without using the existing database 19.

The subsequent procedure - an overview

Follow the installation program instructions (see: "*Installation step by step* 21").

- The installation process first checks whether the **required system software** is installed. If ones is missing, it will be installed automatically.
- After restarting the system, the selected products are installed.
- After the installation is concluded, it is possible to start imc LICENSE Manager directly to activate your license.
- Once the license has been activated, imc STUDIO is ready for use.

Uninstalling

Un-installation is performed via the menu item "*Windows-Settings*" > "*Apps & features*".

Reference See also

- Unattended Installation - Silent Installation 43
- Installation - Installing projects 42
- Recommended Virus Scanner Settings 41
- Changing Languages and Installing Additional Languages 40

3.2.1 Update with the help of the existing database

When an existing database continues in use, any settings which can be retained will be retained. This includes Views, any Setup-columns created by the user, the user administration and all experiments.

If you wish to run multiple versions of imc STUDIO or maybe restore an older version at a later time, create a copy of the database.

Note The database

It is impossible to use the imc STUDIO database by both versions in parallel.

- If the same path is selected in the new imc STUDIO version, the database is used automatically. When loading old experiments, a note is posted in the logbook that the **experiments belong to an older version. Once saved, they can no longer be loaded** with the old version.
- If the database structure has been altered, you will be notified of this. A **dialog for performing conversion** appears. This dialog can be used to convert the database and, if desired, copy it first. After the conversion, the entire **database can no longer be used with the old version**.

Note Using new views

- Be aware that the new version provides new functions, such as new or enhanced Setup pages and new menu items.
- **Use of the new Views is recommended, in order for these new functions to be accessible! Please look under "What's new" whether there have been any changes in this regard.**
- User-made columns such as meta data columns are not automatically inserted into the pages. The configurations of these columns, however, is adopted from the old View. You can re-insert these columns at the desired position (by means of the Column Chooser).
- Multiple views are provided. Select a view and add whatever Setup pages are required or saved, and save the view under a new name.
- How to save Views is explained in the section "*imc STUDIO (general)*" > "*Views* 155".

3.2.2 Update without using the existing database

In imc STUDIO, various settings were saved with the respective **project** and apply to all the experiments it contains. Among other things, these include the view settings: user-defined views, column configurations (e.g. metacolumns).

Please first check which settings you will need.

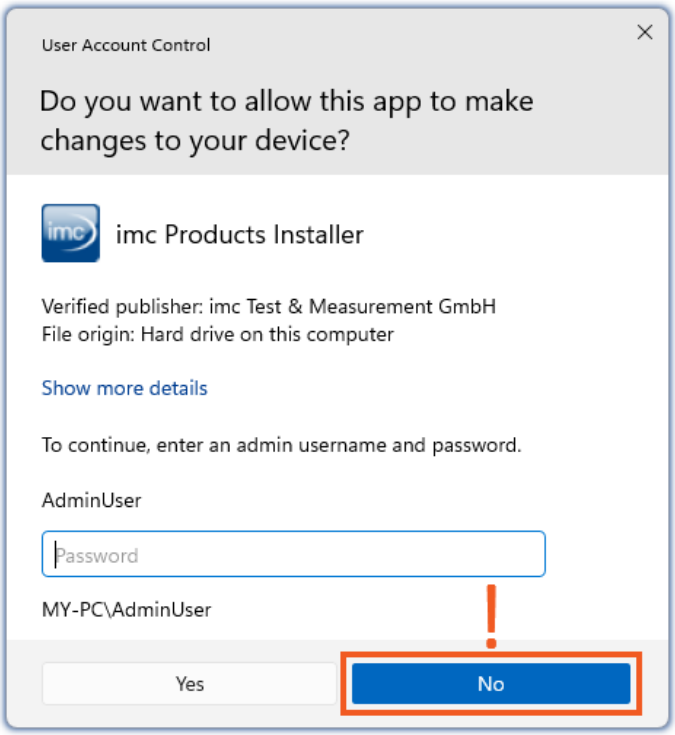
In order to **retain these settings** in the new version after having **performed the update** without using the existing database, export of the settings are necessary.

- If you wish to reuse **all of the project's settings**, export the project settings 117 (by means of the Projects-dialog: Menu ribbon: "*Project*" > "*Manage Project*").
- If you wish to reuse **only the viewing settings**, export the views 157.

3.2.3 Notes and troubleshooting

Notes on Windows User Account Control

Do not change the user account

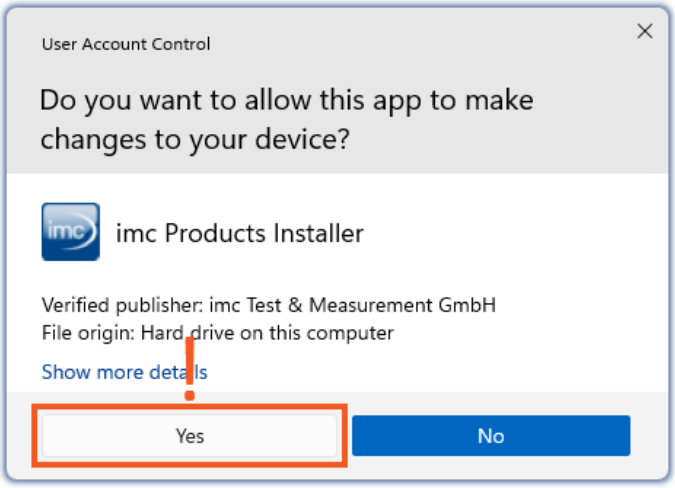


Example of a password request.
Select "No".

With Windows, installation can be started without administrator rights. In this case, the operating system requires selection of a user account and associated password, as shown in the example.

Do NOT continue with the installation in this way, otherwise it will be incorrect. **Select "No"**.

Confirm the prompt by the user account control



Select "Yes", to start the installation.

When you are logged in to a user account possessing administrator rights, you may receive a prompt by the Windows User Account Control (UAC) to allow changes. Select **"*Yes*"**.

Notes on the security software

Some virus scanners **prevent correct installation** of imc programs. We are currently aware of this affecting products from the companies McAfee and ESET. In principle, almost any virus scanner can have settings possibilities which prohibit necessary functions during installation.

For installation purposes, certain steps are necessary, such as:

- registration of programs for Autorun
- registration of programs as a service
- running of scripts from the TEMP-folder
- ...

The error profile may include various messages during installation. Or installed programs failing to launch.

In such cases, please contact your administrator to find out whether certain rules can be suspended for the duration of the installation. For any further questions, please contact our tech support 8.

Reference See also

Recommended Virus Scanner Settings 41

3.3 Installation - Step by step

Actual texts appearing in the user interface may differ from those shown in the screen shots, depending on the product configuration (path/version name).

Download and Installation

The most up-to-date version can be downloaded from the imc website under "*Downloads*" > "*imc STUDIO*". Both a complete and a reduced installation package are offered. The reduced installation package is sufficient if you have only purchased imc STUDIO. Which products are included is indicated next to the Download button.

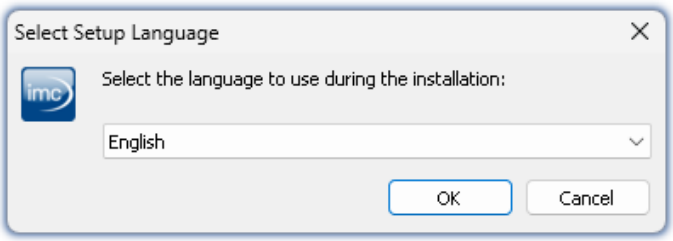
Start the download first, and next the file (e.g. "*Installer_imc_STUDIO_2023_R4.exe*").

The Installer unpacks the installation files. Subsequently, it independently starts the actual installation. Select an appropriate folder into which to unpack the files. The system's default suggestion is the Desktop.

Hinweis Installation files

The unpacked files can be deleted or saved once installation is completed. You can also use the unpacked files for installation at a later time. You do not necessarily need the Installations-exe file.

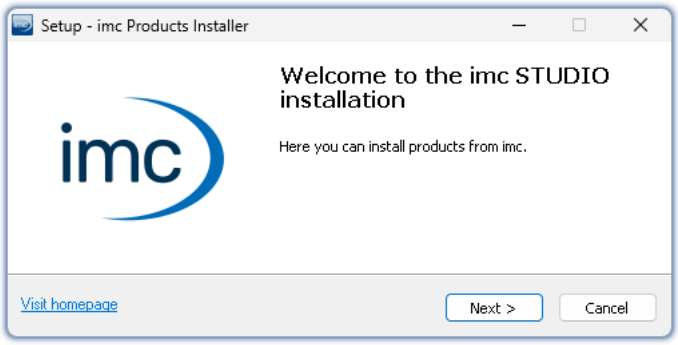
Installation language settings



Selecting the language during installation

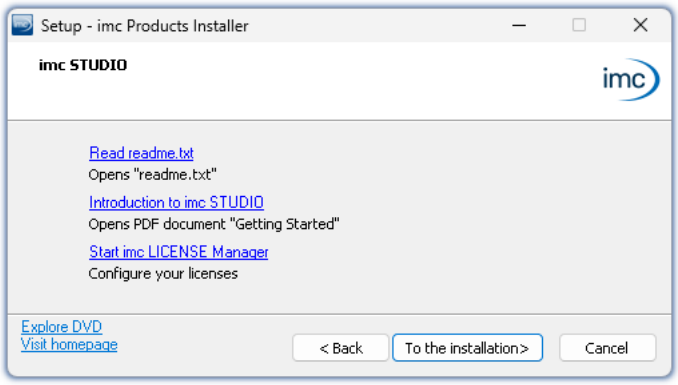
After starting installation, a window appears in which to select the installation language.

Performing installation



Installation setup welcome page

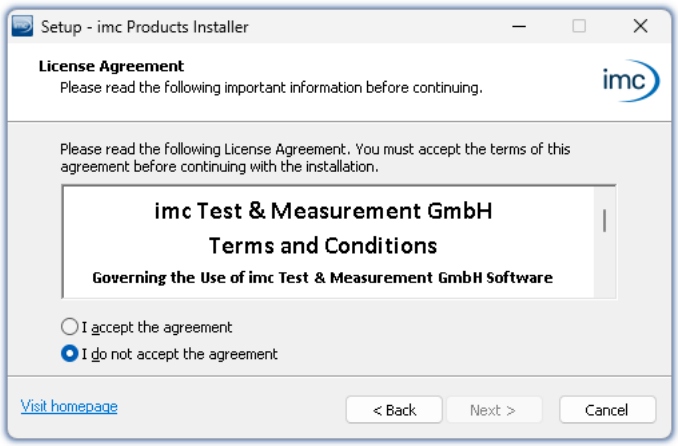
The installation setup then starts in the language selected.



Before starting installation

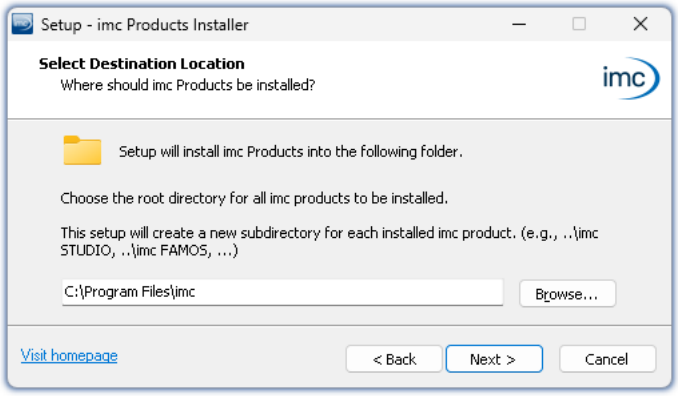
On the installation setup's next page, you are offered the following choices:

- open the "*Readme*"-file,
- open the "*Getting Started*"-document,
- separate installation of imc LICENSE Manager and
- display of the installation medium contents.



License agreement

You must accept the license agreement to continue the installation.



Entering the installation path

The next step is to select the installation folder. Under this folder, a separate subfolder will be created for each imc product (e.g. imc STUDIO 2023). It is recommended to end the associated pathname with "*\imc*".

3.3.1 Select the products / Installation type

Now you can select which components should be installed. There are three types available:

- "*Full functionality for 30-day demo*" : hereafter called "*Demo*"
- "*Typical functionality incl. imc STUDIO Professional*" : hereafter called "*Typical*"
- "*User-defined* 25"

"*User-defined*" gives you the ability to affect all installation settings.

For the "*Demo*" and "*Typical*" installation types the installation settings are already configured. If you need further settings choose the "*user-defined* 25" installation type.

Installation type	Beschreibung
Demo	Use the installation type "*Demo*" in order to test the full scope of **imc STUDIO** and/or **imc WAVE** for 30 days. No other products which require a password will be installed.
Typical	Use the installation type "*Typical*" in order to install the **Professional edition** of **imc STUDIO** including all necessary components. No other products which require a password will be installed.
User-defined	Use the installation type "*User-defined*" if you wish to **configure the individual products**. With this type, it is possible to install **password-protected components**. Further, you can install and configure additional imc products such as **imc FAMOS**. Note that some components can require a separate license.

Note imc STUDIO reconfigure afterwards

Regardless of which variant of the installation is selected, when imc STUDIO is installed, all components are included. This way, after successful installation it is possible to adapt the imc STUDIO configuration at any time by means of the "*product configuration* 29" and to select the appropriate edition.

Products	Demo	Typical	User-defined	Description
				● : included ○ : optional
imc LICENSE Manager	●	●	●	Managing the licensing (internal functions, does not appear in the list)
imc Shared Components	●	●	●	Proving common components of the imc products, such as the curve window (internal functions, does not appear in the list)
Required system components	●	●	●	System components require using of imc products.
imc SENSORS (1)			○	imc SENSORS is a database application for administering and editing sensor information. Applicable with devices of the firmware group A 187 (imc DEVICES).
imc STUDIO Developer (Demo) (2)	●		○	imc STUDIO is modular software platform handling all aspects of modern measurement technology from simple data acquisition to sophisticated test configurations.
imc STUDIO Professional (3)		●	○	
imc STUDIO any edition (3)			○	
imc STUDIO Monitor (3)			○	imc STUDIO Monitor makes it possible to connect with one or more measurement devices, particularly for the purpose of monitoring the instantaneous readings. Data can be viewed and edited live on multiple workstations. It is based on imc STUDIO and is installed as a separate instance of imc STUDIO.
imc WAVE (Demo) (2)	●		○	imc WAVE is a software package for NVH (Noise Vibration and Harshness) analysis. It is based on imc STUDIO and is installed as a separate instance of imc STUDIO. It can be equipped with multiple, separately licensed analyzers.
imc WAVE (3)			○	
Firmware and driver package imc DEVICES	●	●	○	Package of device drivers and firmware required by imc STUDIO and imc WAVE for devices of the firmware group A 187 (imc DEVICES - e.g. CRONOS family, C-SERIES, SPARTAN, BUSDAQ).
Firmware and driver package imc DEVICEcore	●	●	○	Package of device drivers and firmware required by imc STUDIO and imc WAVE for devices of the firmware group B 187 (imc DEVICEcore - e.g. ARGUS*fit*, EOS).
imc FAMOS Enterprise (Demo) (2)	●		○	imc FAMOS is a program for analyzing, evaluating and documenting measurement results.
imc FAMOS Reader (3)	●	●	○	
imc FAMOS any edition (3)			○	
imc Format Converter	●	●	○	Converts imc measurement data to other formats such as EXCEL and ASCII.
imc Documentation	●	●	○	Installs the documentation on the selected products. If the checkmark is removed, it is not possible to open the documentation via the software.

1: imc SENSORS requires a password for installation.

2: The associated Demo license must be activated after concluding installation by means of imc LICENSE Manager, otherwise it is not possible to start the product. Once the demo trial period has elapsed, use the product configuration 29 to covert the product to the license you have purchased.

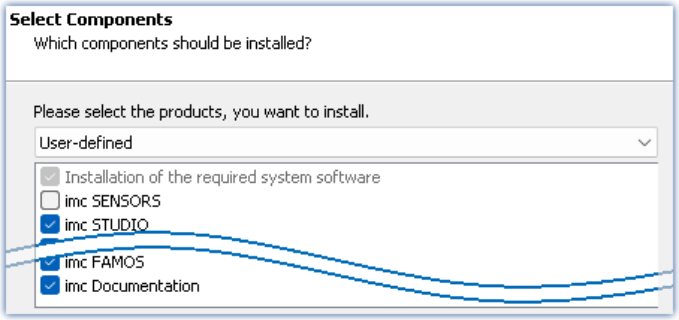
3: The associated license must be activated after concluding installation by means of imc LICENSE Manager, otherwise it is not possible to start the product.

When you select "*Demo*" or "*Typical*", the start of the installation follows immediately (see: "*Starting the installation* 28"). By contrast, if you select "*User-defined*", the installation is first configured 25.

3.3.2 User-defined

The installation variant can be configured in detail. Based on your selection of components desired, different installation steps are displayed.

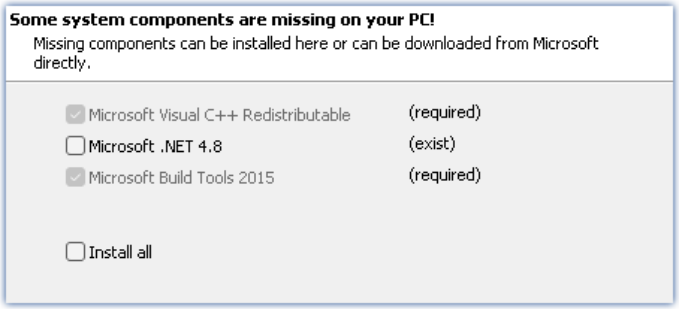
Selecting components



Selecting User-defined installation

Once you have selected the installation type "*User-defined*", it is possible to select/de-select the desired products in the box at the bottom.

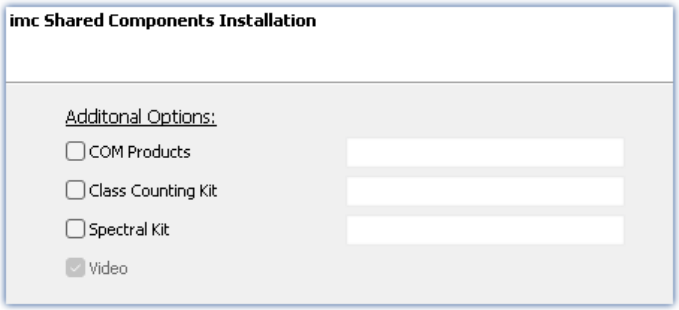
Required system components



Required system components (example)

In the next step, the system components which the products selected require are displayed, along with a note about which are already installed on your system. It is also possible to re-install any already installed components. Which components are missing/present depends on the system and it update status, so that the illustrations are intended as an example only.

Configuring imc Shared Components

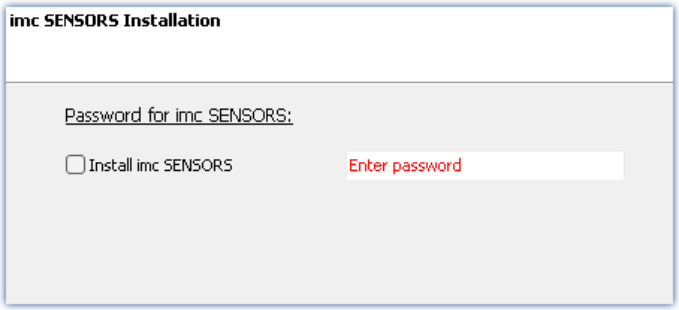


Configuring imc Shared Components

In this installation step, it is possible to install password-protected components of imc Shared Components.

Option/Component	Description
COM Products	The imc COM programming interface is a system integration tool
Class Counting Kit	Function library for imc COM products
Spectral Kit	Function library for imc COM products

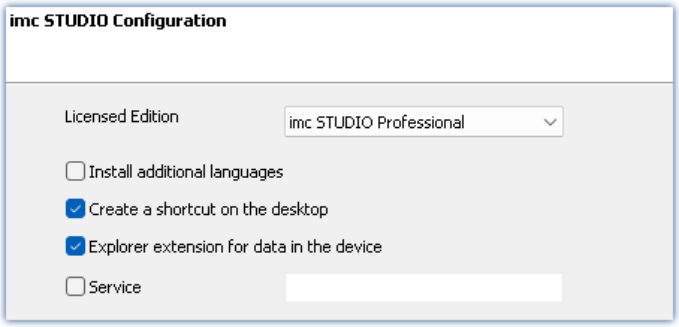
Password for imc SENSORS



Password prompt for imc SENSORS

If you have selected imc SENSORS, you are prompted to enter the password prior to installation. But if you do not wish to install imc SENSORS, de-select it here.

Configuring imc STUDIO

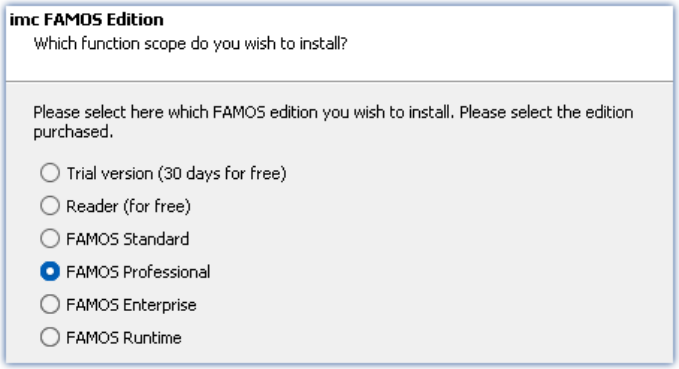


Configuring imc STUDIO

It is possible to make settings for your licensed product edition here. In consequence, later product configuration 29 can be omitted.

Option/Component	Description
Licensed edition	**Information on the editions** is provided in the "*Technical data sheet*" for imc STUDIO.
Install additional languages	With this option, imc STUDIO is installed in all available languages. (This will substantially prolong the installation procedure). Please be aware that some functions may require the other languages. For instance, foreign-language parameter sets can only be imported if the associated language is installed. If this option is deactivated, English language is installed plus the language of the operating system, if available. If needed, all other available languages can be installed at a later time (see the section "*Installing additional languages* 41").
Create a shortcut on the desktop	Program starting links can be created on the Desktop.
Explorer extension for data in the device	Access to the device hard drive via the Windows-Explorer. for devices of the firmware group A 187 (imc DEVICES)
Service	Service support. Install this component only if requested by our tech support. for devices of the firmware group A 187 (imc DEVICES) and group B 187 (imc DEVICEcore)

Configuring imc FAMOS



Configuration of the imc FAMOS installation

In this step, you configure the imc FAMOS installation. Select the editions you wish to install. Note that all editions except the *Reader*-edition require a license.

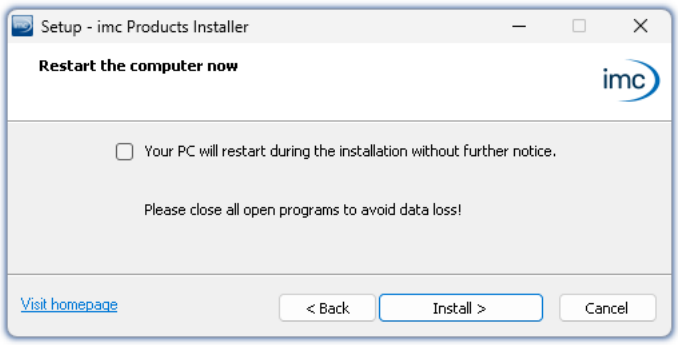
Follow the Assistant's instructions and select your preferred language for the help files and example files like projects, sequences and dialogs. Select any optional components which you wish to install along with imc FAMOS. Additionally, a folder for sample files is needed.

Reference imc FAMOS editions

More detailed information on the imc FAMOS editions is provided in the imc FAMOS manual.

Once all products have been configured, the installation can start.

3.3.3 Starting the installation



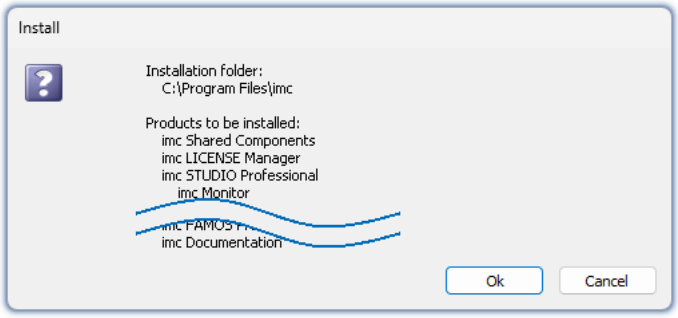
Concluding the installation setup and starting installation

Before the installation, a **system reboot** is performed. For this reason, first **close any running programs** before continuing!

Confirm the reboot by means of the checkbox and then continue (button: "*Install*"). Initially, the required system components are installed. Subsequently, the PC reboots automatically.

Warning Windows user account

After restarting, **be certain to log in with the same user account** with which you started the installation process. Using a different user account can cause failure of correct installation of the components.



List of components to install, following restart (example)

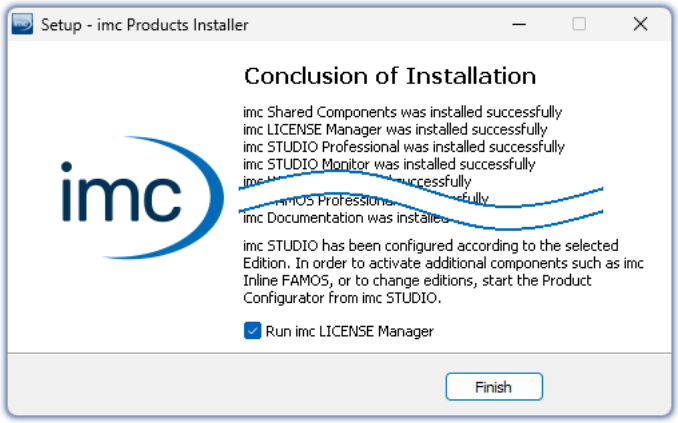
After logging in, the actual installation of the products begins. You are presented with a list of which components are to be installed, which you can confirm by clicking on "*OK*".

The figure shows an example; the actual dialog may appear differently depending on your respective previous selections.

Next, installation of the individual imc products begins.

Note Disabling the system start

While installation is running, the operating system is disabled from starting again, meaning that no other programs can be started.



Concluding installation

When installation is finished, the following message is posted.

Next, you can start imc LICENSE Manager in order to activate your license. It is also possible to start the imc LICENSE Manager later via the Start menu.

3.4 Product configuration / Licensing

imc STUDIO can be ordered and licensed/activated in various product configurations. Further details can be found in the order forms or on https://www.imc-tm.com. For a list of all possible editions and components, see the "*Technical data sheet*" or contact our tech support 8.

Editions and components

The following editions are available for imc STUDIO, and each include a certain basic package of components/functionalities.

Edition	Order code
imc STUDIO Runtime	imc STUDIO-RUN
imc STUDIO Standard	imc STUDIO-STD
imc STUDIO Professional	imc STUDIO-PRO
imc STUDIO Developer	imc STUDIO-DEV

With an appropriate "*Engine*" activated, any edition can run the configurations of higher editions.

Additional optional or individually licensable components can also be integrated.

Each edition is able to apply configurations created with a higher edition, but not to modify them.

For the Edition **Standard**, **Professional** or **Developer**, the corresponding license is required. These can be activated in the imc LICENSE Manager. The Edition **Runtime** is a restricted version, free of charge. You can register this edition in the imc LICENSE Manager. There, select the option "**Trial versions and freeware**".

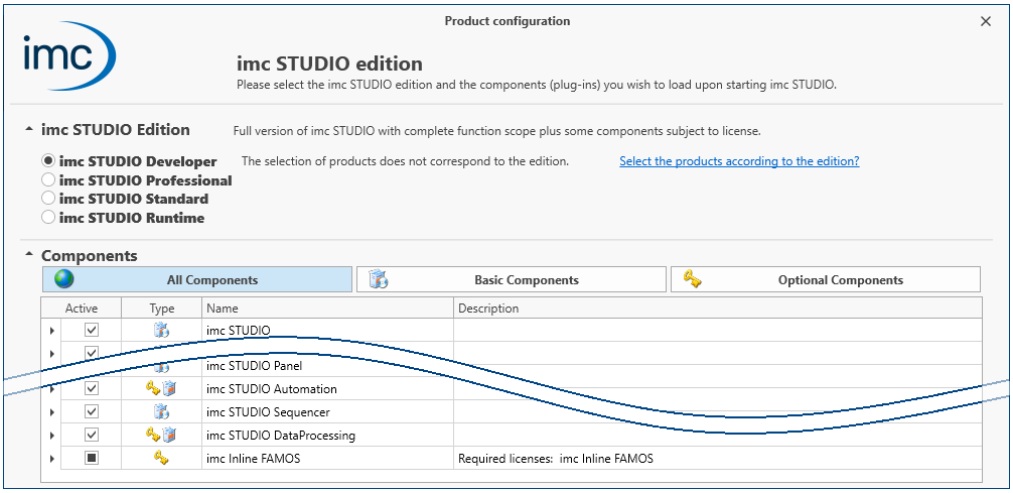
Licensing (imc LICENSE Manager)

The software licenses are managed with "*imc LICENSE Manager*". After concluding installation, you can start imc LICENSE Manager directly in order to configure your license (e.g. by means of the Windows Start menu: group "*imc*" > "*imc LICENSE Manager*"). If imc STUDIO is started without an appropriate license having been configured, the product configuration program opens, from where it is also possible to start imc LICENSE Manager.

Follow the instructions of the imc LICENSE Manager. imc LICENSE Manager offers a separate documentation. Start imc LICENSE Manager and click on "*Help*".

Changing the product configuration

You can change the product configuration in order to adapt it to the license purchased. For instance, if during installation you have selected a different edition or if you have purchased a product upgrade. To do this, start imc STUDIO and click in the menu bar (top right) on the symbol . The product configuration assistant now opens. You can change the configuration to correspond to the licenses you have purchased. Follow the instructions and close the assistant.



Assistant for altering the product configuration

After completing the configuration, you must close and restart imc STUDIO to apply the new settings.

Changing the edition

In the dialog's upper region, a number of editions are available for selection. The default selection is the edition currently being used. If you change the selection, you can decide whether the available components are to adapt to the selection. In most cases this is recommended, since this is the only way all of the edition's functions will be available after an update.

Optional components are not affected by the edition selection and remain in their previous state.

> Please only change the edition if you have activated an appropriate edition in the imc LICENSE Manager.

Changing the components used

Additional components for the edition selected are available to choose. Some of these components require a separate license.

In the bottom region of the dialog, the available components are shown. The set of components currently used is selected by default (except subsequently to an edition change).

Column		Description
Aktive	☑	The component(s) is/are activated.
	☐	The component(s) is/are deactivated
	▬	Some of the components are activated. After clicking here, the components underneath are deactivated.
Type		The component requires an extra license in addition to the license for the edition.
	/	The component can be activated for the selected edition without any additional license.
Name		Product name of the component
Description		Short descriptive text about the component. More detailed descriptions of the individual components are provided in the technical data sheet.

Activate/deactivate the individual components in the column "*Active*".

FAQ

Why are there sometimes two components for the same product? E.g. "Automation Editor" and "Automation Engine"

Answer: Many components are subdivided into their actual function, on one hand, and on the other their respective settings interface. This distinction exist for the following reason: each edition can run any function created with a higher-level edition, but not modify it.

In other words, using the component "Automation Engine", you can run Automation-Tasks in the Standard edition. However, the Editor ("Automation Editor") is only available in the Developer edition.

When should the "Engine" be deactivated

Answer: In most cases, it is not necessary to deactivate the engine if the component is not required. However, be aware that every component activated makes resource demands.

At a minimum, keeping unnecessary components active slows down the launch of the system.

Disadvantage: All functions belonging to the deactivated components no longer work. Example "Sequencer": Commands at Widgets, hotkeys/keyboard shortcuts, user-defined buttons, user-defined events.

3.5 Start

Start the software by means of the associated desktop icon or via the Start menu.

Further, as part of installation an **icon is set up on your desktop** - depending on your choice during installation 21.

If no icon was set up, then open the Windows Start menu. There you find the group "*imc*" and in it a link to start the product.

If product configuration / licensing 29 is missing, the Product configuration starts automatic.

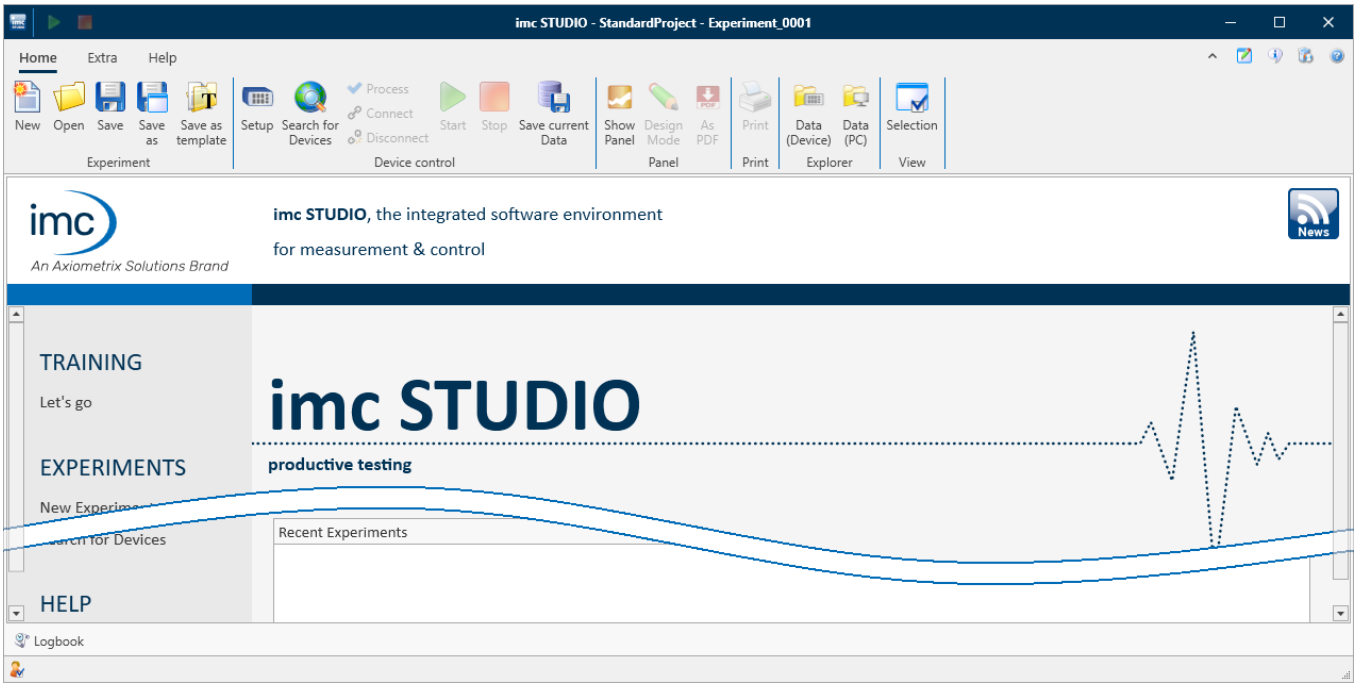
Splash screen

Next, a splash screen appears, indicating that software components are being loaded. Depending on the installation and the PC capacity, this could take some time.

Loading the software components after start

Homepage - Start page

By default the Homepage is shown when the software is started. The Homepage shows certain functions, depending on the particular product, as buttons. Click on the corresponding buttons, to select your desired functions. After the first installation, the "***Homepage***" may appear as shown in the example below:



Start page (example)

imc STUDIO starts with a simplified view

Along with installation of imc STUDIO, a number of **Views** are provided. Views in this sense are defined designs for the user interface's appearance.

imc STUDIO offers a wide variety of possibilities and functions, but the entire scope of these is not always needed for a single simple measurement.

Note — imc STUDIO starts with a simplified view

In order to make it easier to get familiar with the software, the program starts with a **simplified view** with limited choices: "***Standard***". All important functions for measurement and visualization of measured data are available. The menu ribbon is structured in such a way that proceeding from left to right accesses all of the Main windows 33.

Changing from the simplified view to View: Complete

You can switch to a different view at any time, in order to obtain access to all of the existing functions. To do this, go first to the menu item "*Extra*" and in the drop-down list of views, select: "***Complete***".

Note — Notes on the descriptions and the screenshots

The following descriptions and screenshots always pertain to the view with the full scope of functions. Many of these functions are also found in the minimized view at another location.

3.5.1 Device Connection / Network / Firewall

To use an imc measurement device, imc STUDIO must set up a **connection via a network** (mostly LAN). Details on the network settings are presented in the chapter "*Setting Up - Connect the device* 45".

Firewall

The firewall used might **prevent** establishment of a **connection** between the operating software and the measurement device. If the software is unfamiliar to the firewall, then a corresponding prompt usually appears upon **accessing the network**. This happens, for example, after starting the software and upon first accessing the device (e.g. in a device search). Some firewall programs can also block access to system and hardware-components.

In all cases, **correct operation** is only possible if the **connection is not blocked**. The following programs are usually affected:

Program	Default path
imc STUDIO: imc.Studio.exe	C:\Program Files\imc\imc STUDIO...
imc DEVICES: imcDevices.exe	C:\Program Files (x86)\imc\imc DEVICES...
imc DEVICES Service: imcDevicesService.exe	C:\Program Files (x86)\imc\Shared

For operation on LAN, it is sufficient to **enable "*Private networks*"**. Enable this access type. You may wish to read the operating instructions for your firewall, and / or ask your administrator / IT-department.

Windows Defender Firewall: Unblocking is performed automatically upon installation.

3.5.2 Main Windows / Components

imc STUDIO provides the framework for various ***components***. Components appear within imc STUDIO as ***main windows***. A component can have one or more main windows.

The main windows appear in the Navigation pane.

On the left you see an example of the Navigation pane with the main windows **Homepage**, **Setup**, **Panel**, **Automation** and **Sequencer** (**Panel** is selected).

To jump to one of the main windows, click on the corresponding button in the Navigation pane.

Navigation pane 139: Select this link to go to further information on operation of the Navigation pane.

Note Notes on the use of the simplified view

- When you use the simplified view 32, the Navigation pane is hidden. In this case, in order to go to the main windows Setup and Panel, use the menu ribbon.
- The other main windows can only be reached via the Navigation pane.
- If you need it, you can show the Navigation pane 141, or change the view 155.

Tool window

Most components have their own tool window (e.g. the Panel with the tool windows: Widgets and Data Browser). The tool windows are described in their respective sections.

Reference See also

Tool window 140: Select this link to go to further information on operating the tool windows and for a description of the tool windows belonging to the imc STUDIO frame.

3.5.3 Important settings

User and User groups

imc STUDIO has the ability to administer various **users** which are permitted to use the program in various ways. For details see the section "*User administration* 146".

Options

Before beginning to work with imc STUDIO, you can change essential settings, such as the **folder path where the experiments are saved**.

You can change the folder path in the options 128 (under "*Project Management*" > "*HDD settings*"):

Ribbon	View
Extra > Options ( )	all

Note Reading and writing access rights are required

Please note that this setting will apply for every user of imc STUDIO. Each user must possess reading and writing access rights for the specified folder path.

Checking and modifying the time zone

Applicable to devices belonging to the firmware group A 187 (imc DEVICES): In order to assign a definitive point in time for all measured data, a specific time zone is set for all imc devices used. The factory setting is "***UTC+01:00***". If applicable, modify the time zone setting. For this purpose, the "*Devices properties* 221" dialog contains an entry "*Device time and synchronization*" > "*Time zone*".

Ribbon	View
Setup-Configuration > Device Properties	Complete

3.5.4 Command line parameters

Opening an Experiment file

An experiment file 110 (e.g. MyExperiment.imcstudio) can be opened directly in the Windows Explorer by double-clicking. This starts a new instance of imc STUDIO and opens the experiment file. The same procedure also applies for linkages to experiment files, for example files on the Desktop.

Experiments can be started with command line parameters. To do this, create a link to an experiment file and enter the desired parameter.

["imc STUDIO-Installation"] "Experiment-File" [/fullscreen] [/do[StartMeasurement]] [/do[StartSequencer]]

["imc STUDIO-Installation"]	Open Experiment with a specific imc STUDIO installation (optional)

In order to open an experiment file with a particular installation of imc STUDIO, extend the link with the respective installation's name. For example:

```
"C:\Program Files\imc\imc STUDIO 2023\imc.Studio.exe"
"imcDB://DB\StandardProject\MyExperiment"
```

The experiment file `MyExperiment.imcstudio` is started in the imc STUDIO installation.

"Experiment file"	Experiment to be loaded

Mandatory entry. Opens the experiment entered with the imc STUDIO installation which is set up as the default program under MS Windows. Mostly this is the last imc STUDIO version installed, or imc STUDIO Monitor/imc WAVE, if this component was installed along with it.

One possibility is to specify the path in absolute terms:

```
"C:
\Users\Public\Documents\DB\StandardProject\MyExperiment\config\MyExperiment.imcStudio"
```

Another way is to specify it in reference to the name of the database and of the project (the appropriate database folder must be selected):

```
"imcDB://DB\StandardProject\MyExperiment"
```

/fullscreen	Fullscreen mode (optional)

In order to start an experiment in fullscreen mode, the experiment file is opened with the parameter `/fullscreen`.

```
"imcDB://DB\StandardProject\MyExperiment" /fullscreen
```

After opening the experiment file, the Panel is started in fullscreen mode, if the experiment contains at least one Panel page.

/do[StartMeasurement]	Starting the measurement (optional)

In order to start the measurement automatically upon opening an experiment file, The experiment file is opened with the parameter `/do[StartMeasurement]`.

```
"imcDB://DB\StandardProject\MyExperiment" /do[StartMeasurement]
```

or

```
"imcDB://DB\StandardProject\MyExperiment" /fullscreen /do[StartMeasurement]
```

/do[StartSequencer]	Starting the Sequencer (optional)

In order to automatically start Sequencer upon opening an experiment file, the experiment file is opened with the parameter `/do[StartSequencer]`.

```
"imcDB://DB\StandardProject\MyExperiment" /do[StartSequencer]
```

or

```
"imcDB://DB\StandardProject\MyExperiment" /fullscreen /do[StartSequencer]
```

3.5.5 Error causes upon starting

Corrective actions for when the logbook opens with error messages directly upon starting imc STUDIO.

Message	Corrective action
No matching DEVICEcore found.	In the product configuration, the **firmware imc DEVICEcore** is activated. However, no appropriate firmware has been found or installed. By default all firmware products are activated. You can take the following corrective actions: • install the appropriate imc DEVICEcore-firmware. This is only necessary if you wish to use a device belonging to this firmware group. • deactivate the component "*imc DEVICEcore device adapter*" in the product configurator 29.
Error while initializing the plug-in "imcDevices V2.x Adapter" ...	In the product configuration, the **firmware imc DEVICES** is activated. However, no appropriate firmware has been found or installed. By default all firmware products are activated. You can take the following corrective actions: • install the appropriate imc DEVICES-firmware. This is only necessary if you wish to use a device belonging to this firmware group. • deactivate the component "*imc DEVICES V2.x Adapter*" in the product configurator 29.
Could not load file or assembly 'imc.Studio.Plugins.DevSetup.Sensors.Stubs.dll' ...	In the product configuration, the component **imc STUDIO Sensors** is activated. However, no appropriate **imc DEVICES firmware** was found or installed. You can take the following corrective actions: • install the appropriate imc DEVICES-firmware. This is only necessary if you wish to use a device belonging to this firmware group. • deactivate the component "*imc STUDIO Sensors*" in the product configurator 29.
The loaded Project-settings are from an older imc STUDIO version. If you save these settings, an older imc STUDIO version would not be able to read them correctly.	With every Version/Revision, new enhancements and other changes are introduced. In consequence of these changes, some saved projects may become no longer compatible with older versions. Make sure that no older version of imc STUDIO needs to access the database.
More than one DEVICEcore installation found in STUDIO installation directory.	Solution: Uninstall the older imc DEVICEcore version.
Could not connect to the Guardian process. Some functions (e.g. delete measurements) will not work.	The Guardian process is not running. Please start either the Guardian- or WatchDog-service. For more info, see the section "*Guardian* 165"

3.6 Training - Guided first steps

What steps are needed for obtaining measured data?

An Assistant is provided to help you become familiar with the system. The first steps – selection of the device, channel configuration and display of the measured data – are usually identical.

Note

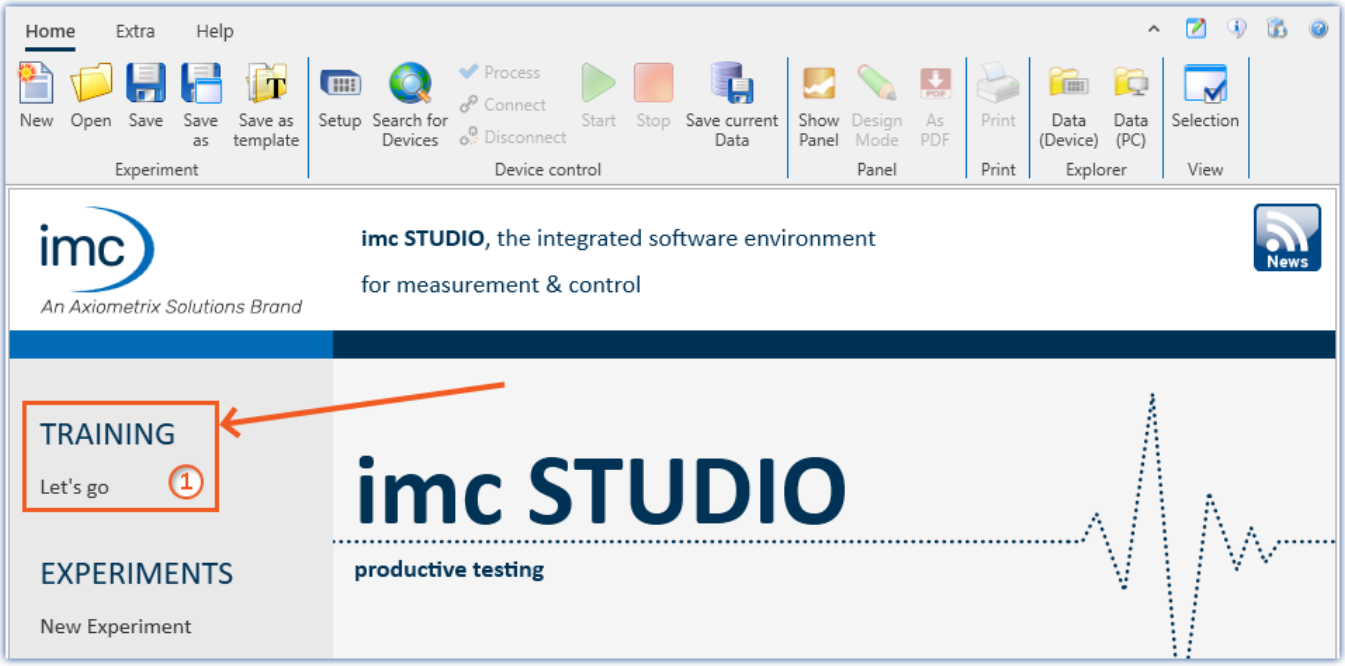
Restricted view

The Assistant is optimized for the Standard-view 32. It is also possible to start the Assistant in the "*Complete*" view. This omits the highlighting of some elements.

Product configuration

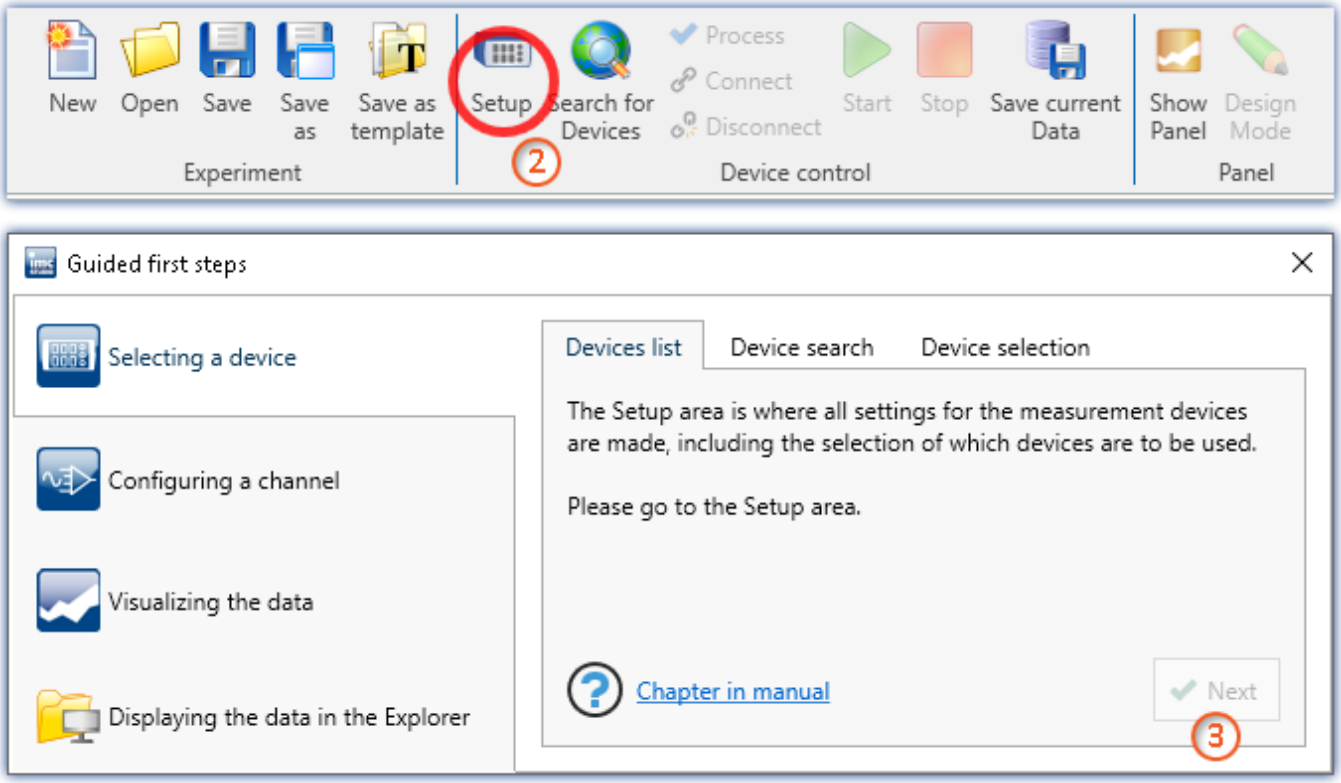
Before starting the Assistant, modify your product configuration 29. For instance, if you have activated imc Inline FAMOS, additional steps will appear.

Start the Assistant from the homepage directly after starting imc STUDIO. Click on "*Let's go*" (1). You are able to start and repeat the Assistant again at any time.



Starting the Assistant

A dialog appears, in which the next steps are explained to you. As an aid in orientation, a pulsating circle ② appears in the imc STUDIO user interface in order to highlight the interface element with is the topic of the descriptive text. As shown in the example below, the next step is to open the Devices list. To do this, it is enough to click on the button "*Setup*" in the menu ribbon. The Assistant recognizes when the assignment has been completed, and only then enables the "*Next*"-button ③.



The Assistant provides guidance using graphical highlighting and inforative texts

By means of the tabs, you can repeat or skip steps.

Each page contains a link to the pertinent section in the user's manual, where you can find advanced descriptions of the respective pages and steps.

Note — Firmware and network configuration

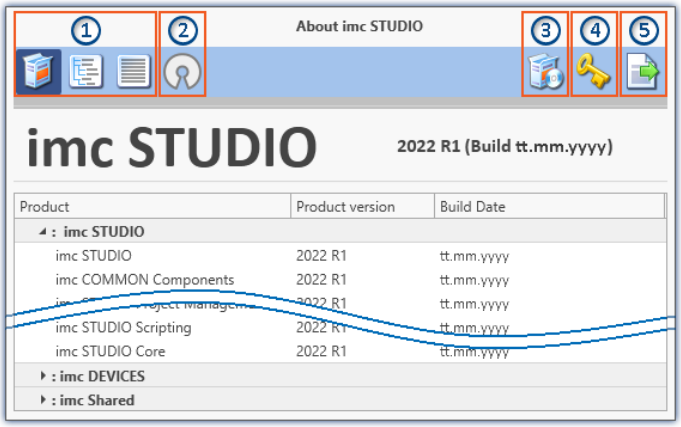
If the device's firmware is either wrong or the device cannot be found, then please see the section "*Setting Up - Connect the device* 45".

Reference — Tutorials

Once you have worked through the first steps, we recommend you follow the Tutorials 71. Here you will receive helpful tips and learn about additional operation possibilities.

3.7 Info / Version information

imc STUDIO consist of multiple components. In order to see which components are included in your package, click in the menu bar on the symbol . Then a pop-up window appears displaying the product names and details on the components:



Version information (example)

With the buttons on the left side ①, you can change the list's depth of detail and how it is sorted. Additionally, you will receive all copyright notes regarding "*Open source software*" ( ) ②.

With the buttons on the right side, you can start the assistant for the product configuration 29 ③, start the imc LICENSE Manager 29 ④ or export the product selection ⑤.

While performing the export, you can save the contents as a file in XML format. For inquiries to our tech support, it may be necessary to send it as an e-mail (see also chapter "*Tech support* 8").

3.8 Information and tips

3.8.1 Changing languages and installing additional languages

Following installation, imc STUDIO is available in **English** and, if applicable, in the **language of the operating system**. Additional languages can be installed by selecting the option "***Install additional languages***" in the installation variant "User-defined 25".

Note Observe operation system settings

- Please ensure that your operating system is set correctly to the display language. Some languages require modifications to be made accordingly. In particular, the "***System locale***" must be set correctly to correspond with the display language.
  Otherwise, with some languages there can be problems with the characters. Languages affected include Japanese, Chinese, Russian, ...
- **Mixed language settings** for the **format in Windows** cause problems with the display of units with special characters: see "*imc CANSAS Software* 211"

Changing the language

By default, the imc software will start in the same language as the Windows version installed. If this language is not supported, then the English program version appears.

The language can be changed, irrespective of the Windows version. Use the program "*imc Language Selector*". The program is found in the Start menu under the group "*imc*".

Note Constraints

Only the imc software's texts are converted. Components which depend on the operating system's language setting are not affected.

Please only use one of the following two languages:

- Default: the operating system's language
- English

Any other selection may cause malfunctioning if the operating system and the imc programs use different languages.

Installing additional languages

If desired, all other supported languages can be installed later (no installation medium is required).

Note — Which languages are be installed subsequently

In subsequent installation of languages, **all languages supported** are installed. There is **no selection** possibilities.

Step by step

- Administrator rights are required.
- Open the **installation folder** of imc STUDIO (e.g. "*C:\Program Files\imc\imc STUDIO...*"), e.g. with Windows Explorer or by means of the command line.
- There, open the folder "***Languages***"
- Run the file "***InstallLanguages.bat***".
- Wait until the script stops and posts the message "*Failures: 0*". This indicates that the installation has completed successfully.

Note — Instruction notes

If installation was unsuccessful, the necessary user rights may be missing:

- If "*InstallLanguages.bat*" is called via the **command line**, run the **command line as Administrator** (context menu: "Run as Administrator")
- If "*InstallLanguages.bat*" is called via the **Windows Explorer**, run the **call as Administrator** (context menu: "Run as Administrator")

3.8.2 Recommended virus scanner settings

The presence of many channels can **burden the PC** if a virus protection program scans the data captured. This may cause a **jiggling display** and high demands on the processor.

We strongly recommend **removing imc STUDIO from the virus scan**. Most virus protection programs are able to classify individual programs as low-risk. To do this, please refer to the instructions for your virus scanner.

Low-risk	Default path
Add the programs **imc.Studio.exe**, **imc.Monitor.exe** and **imc.WAVE.exe** to the list of low-risk programs	C:\Program Files\imc\imc STUDIO...
Add the program **imcDevices.exe** to the list of low-risk programs	C:\Program Files (x86)\imc\imc DEVICES...
Add the **database** path along with all subfolders to the list of low-risk folders	C:\Users\Public\Documents\DB

Ensure that its data are not searched for viruses during reading or writing.

Reference — Installation

See also the installation note: "*Notes on the security software* 21"

3.8.3 Enhanced installation

Installing projects

You can prepare projects which are automatically included with the installation. E.g., after the installation, special views and experiments will already be available.

- Create a project-export-file (".imcStudioExport") for the purpose. Make sure that the project is exported and not only the experiments (selection). Upon exporting, you can define what the file is to contain. E.g., only the project settings, or also the experiments.
- Save the file(s) on the installation medium in the following path: ..\Products\imc STUDIO\Projects

Any existing projects having the same name will be overwritten.

Produce an installation medium and run the installation as usual or install from the hard drive.

After the first launch of imc STUDIO, projects are imported into the database.

FAQ

Question	Answer
What happens if the project exists already?	All project settings will be overwritten.
What happens if the project has contents which do not exist in the project to be imported?	All content besides project settings remains intact. E.g. experiments, measured data, metadata, ...
What happens if the project has contents which already exist in the project to be reported?	These will be overwritten. As well, any experiments, measured data, metadata, ... are overwritten.
What happens if there are any other projects present?	These projects remain intact.
What all is imported from the project?	All project settings and any experiments, measured data, metadata, ... the projects contain

Adding your own personal programs

On the installation medium, in the folder "*Products*", create an extra folder for your own product. Save your installation file there. The Installer will now automatically detect the product and offer it for installation.

Optional: By means of an ini-file, you can cause your product to be the default selection in the Product-Selection list.

On the installation medium (on the same level as the folder "*Products*"), create an additional folder named "*Configuration*". There, create a file named "*Setup.ini*". Fill this file with the following text:

```
[SetupX]
SetupX1=<PRODUCT>
```

For <PRODUCT>, insert the name of the folder in which your installation file is located.

3.8.4 Unattended installation - Silent installation

For unattended installation ("*Silent installation*"), certain parameter files (.ini) are required. Create these files by means of a command line call.

```
Setup.exe /CREATEINIFILES="<TargetPatch>"
```

To do this, proceed as follows:

- Start the installer - e.g. "*Installer_imc_STUDIO_2023_R2_2023-04-20.exe*".
- The installer begins by unpacking the content of the installation medium. Select an appropriate target folder on your hard drive: e.g under "C:\DVD\".
- As soon as the unpacking process has concluded, configuration for the installation begins. Cancel this procedure!
- Start Windows' command prompt ("cmd", "command shell").
- Navigate to the appropriate file path and call the Setup with the following command line (in the example here, the folder is: "S:\DVD\"):

```
Setup.exe /CREATEINIFILES="C:\DVD\"
```

The frame setup is started and may be configured as desired. When the settings are finished being made, there is no restart, as there usually would be. Instead, at the specified target location, *.ini-files are created which have the following structure:

Main file

DVD/Configuration/Setup.ini

Product files

DVD/Products/$Productname$/Configuration/$Filename$.ini

z.B.

DVD/Products/imc STUDIO/Configuration/imc STUDIO.setup.ini

DVD/Products/imc FAMOS/Configuration/Setup_imcFamos.ini

The ini-files contain the settings which have been made.

Subsequently, the DVD folder can be copied to a data carrier and used for the unattended installation.

Calling the unattended installation

The installation process is called with command line parameters, where "`/SILENT`" causes the silent setup installation to start. The following is an example of a 64-bit system:

```
DVD/Setup.exe /Lang=en /DIR="C:\Program Files\imc" /SILENT
```

With the additional parameter "`/UNINSTALL=ALL`" the de-installation of all previous versions is inhibited.

If the switch is set to "NECESSARY" only the necessary version are de-installed. When setting the switch to "NONE", no version is removed.

In special cases a de-installation is needed, e.g. if you install the 5.2 R23 to a 5.2 R22. The 5.2 R22 needs to be de-installed.

Note

Creating ini-files in a different folder

You can also create the ini-files in a different folder. In this case, merge the two folders afterward (not recommended because prone to errors).

Please always create new ini-files each time

With any new version, it may occur that new keys are entered in the ini-files. For this reason, create new ini-files for any new version.

4 Setting Up - Connect the device

There are multiple ways to **connect the imc measurement devices with the PC**. In most cases, the **connection via LAN** (local area network, Ethernet) is implemented. See section "*Connecting via LAN in three steps* 46" for the **quickest way to connect** PC and measurement device.

But there are also other connection types:

- WLAN 59
- LTE, 4G, etc. (via appropriate routers)

These are described in a separate section: "*Special options for connecting to the device* 48".

The devices use the **TCP/IP protocol** exclusively. With this protocol, some settings and adaptations for your local network may be necessary. For this purpose, the support of your network administrator may be necessary.

Note

When **using multiple TCP/IP connections**, e.g. LAN, WLAN and modem, observe the remarks in the section "*Computers with multiple TCP/IP connections* 51".

Recommended network configuration

The latest and high-performance network technologies should be used to achieve the maximum transfer bandwidth. This means especially 1000BASE-T (GBit Ethernet). GBit Ethernet network devices (switches) are downward compatible, so that imc devices that only support 100 MBit Fast Ethernet can also be operated on them.

The cable length between the switch and a PC or a device should be less 100 m. Use a shielded cable. If the length of 100 m is exceeded, then you have to insert another switch.

If the system is being integrated into an existing network, you must ensure that the minimum data rate can be guaranteed. Under some circumstances, this may require using switches to subdivide the network into separate segments in order to govern the data traffic in a targeted way and thus optimize the data rate.

In very demanding applications, you might consider grouping multiple GBit Ethernet devices via even higher-performance sections lines of the network (e.g. via 5 GBit Ethernet) and to connect these groups to NAS-components, for instance, via these lines.

When such imc devices are included which use network-based PTP-synchronization (e.g. CRXT or CRFX-2000GP), then it is necessary to use network switches which fully support this protocol on the hardware side. Appropriate network components are also available as imc accessories (e.g. CRFX/NET-SWITCH-5) and are then electrically and mechanically fully compatible with the imc systems.

© 2023 imc Test & Measurement GmbH

4.1 Connecting via LAN in three steps

The most common case is described below: the PC and the device are connected via cable or network switch. The device's IP address must be set in the PC's address range. Subsequently, the device can be connected with the PC. If a connection has ever been established previously, the software recognizes the device's hardware configuration. In that case, experiment configurations can be prepared without any connection to the device.

Step 1: Connecting the measurement device

To connect via LAN there are two options:

1. The measurement device is connected to an **existing network**, e.g. via network switch. Only with a switch is it possible to run multiple devices.
2. The measurement device is connected directly to a network adapter on the PC (**point-to-point**).

In a LAN, the first case is typically implemented. Modern PCs and network switches are usually equipped with Auto-MDI(X) automatic crossover recognition, so that it is not necessary to distinguish between crossed and uncrossed connection cables. Thus both cable types can be used.

Step 2: IP-configuration

Start imc STUDIO. Click the "*Device interfaces*" button ( ) to open the dialog for configuring the IP address of the device.

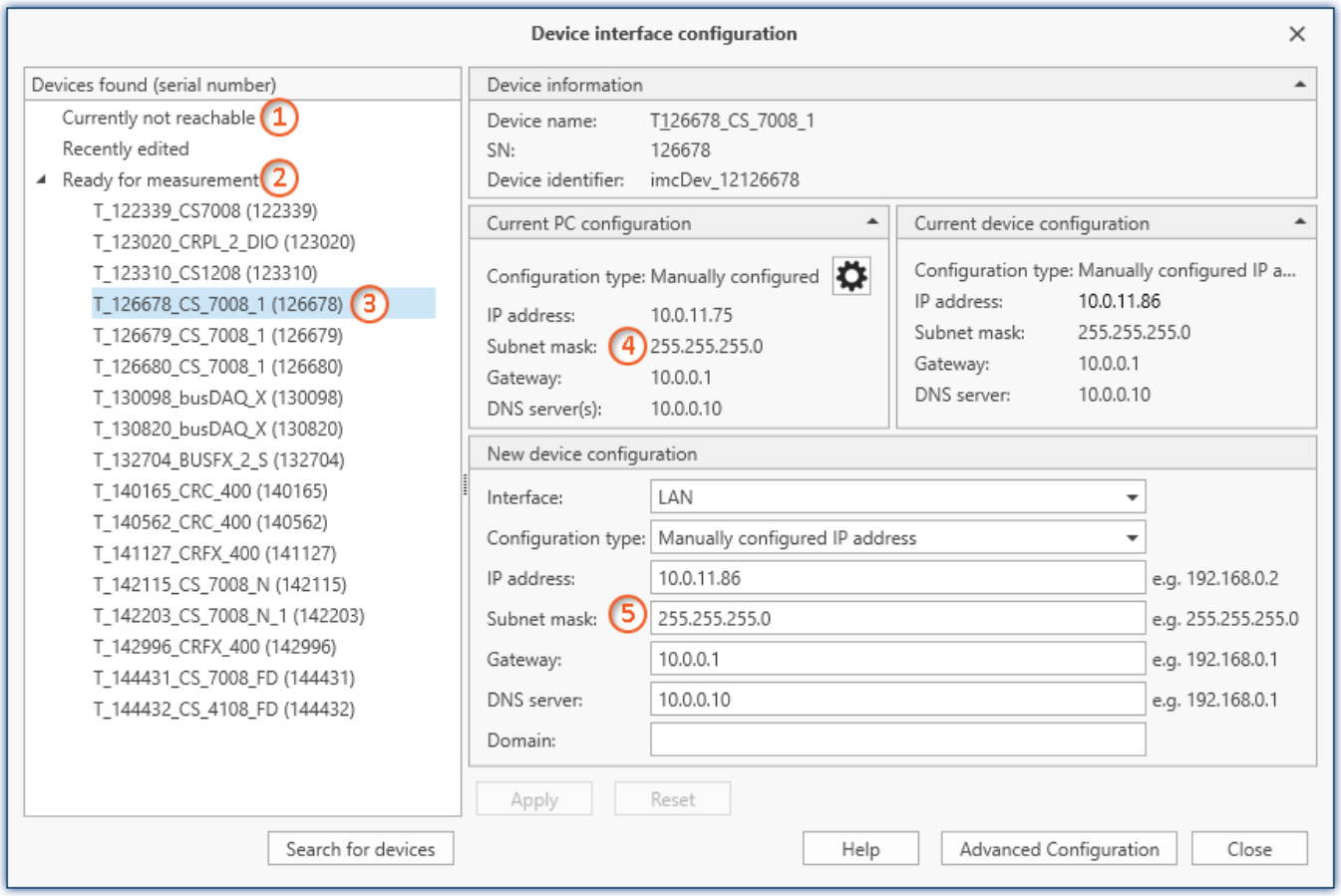
Ribbon	View
Setup-Configuration > Device interfaces ( )	Complete

If this **button** is **not present** in the view, it is also possible to open the dialog after a device search if it failed to find any new devices. Subsequently, a prompt appears asking whether to search for devices with an inappropriately configured network interface. Close this message box by clicking "*Yes*".

Once the dialog starts, the system automatically searches for all devices in the network. In the tree diagram, all available devices are indicated. If the device appears among the group "*Currently not reachable*" ①, it is necessary to modify the device's LAN-settings. If the device appears among the group "*Ready for measurement*" ②, you can leave the settings as they are or review them.

If there is any IP-conflict, devices affected will not be listed.

Select the device for making modifications ③.



Display of measurement devices found and of the IP address

Set the **IP address manually** if you are not using DHCP. The device's IP address ⑤ must match with the PC's address ④. To conform to the network mask, only the device portion may be different (see example).

Example

In the example shown, the fixed IP 10.0.11.75 with subnet mask 255.255.255.0 is selected for the PC. For measurement devices, any numbers would be suitable which begin with 10.0.11. and then do not contain 0, 75, or 255. The 0 and the 255 should not be used, if possible, due to their special significance. The 75 is the computer's number.

Example for IP settings	PC	Device
IP address	10 . 0 . 11 . 75	10 . 0 . 11 . 86
Network mask	255 . 255 . 255 . 0	255 . 255 . 255 . 0

If the configuration type: "*DHCP*" is used, **the IP address is obtained automatically** from the DHCP-server. If it is **impossible to obtain any setting values** via DHCP, the **alternative values are used**. These could lead to errors in the connection (different networks, same IP addresses, etc.).

If there is a **direct connection** between the device and the PC by a cable, then **DHCP should not be used**.

In order to apply the changes, click on the button "*Apply*". Wait for the device to restart and then close the dialog.

Note — Connection via modem or WLAN

For devices of the firmware group A (imc DEVICES):
If the connection to the device is established via a modem or WLAN, start the program "*imc DEVICES Interface Configuration*" by clicking on the button: "*Advanced Configuration*" (see previous figure). An exact description is found in the software manual chapter: "*Setting Up - Connect the device*" > "*Special options for connecting to the device* 48".

Step 3: Integrating a device into an experiment

Now you are ready to add the device to the imc STUDIO experiment. If your device is unknown to the system, first perform the "*device search* 199".

Ribbon	View
Home > Search for devices ( )	all
Setup-Control > Search for devices ( )	Complete

Select the desired device: Once you click in the checkbox "*Selected*" for the desired device, it is ready to use in the experiment (see *Device Search - Known and Selected* 241).

Selected	Device name	SN	Device specification
☑	T_124835_C1_1_LEMO_ET	124835	imc C1-1 LEMO
☐	T_130039_busDAQ_X	130039	busDAQ-X
☐	T_130311_SPARTAN_U32_CAN	130311	imc SPARTAN

You can also select multiple devices for your experiment.

Now the device is "*known*". After the next program start it is available for selection. For further information, see the documentation on the component "*Setup* 186".

Reference — Time zone

Now check whether the correct time zone is set for the device: see "*Important settings* 34".

4.2 Special options for connecting to the device

Additional ways to connect with the device are presented below. For other special cases, please see the section on LAN connection.

4.2.1 Devices and TCP/IP

General preparations

A imc device with TCP/IP supports a classic network interface (LAN).

For a device having a LAN-interface, the following parameters must be set appropriately:

- IP address
- Subnet mask
- DNS-server-address (optional)
- Domain (optional)
- Router-address (optional)

If your network is equipped with a DHCP-Server, these data can be accessed automatically by the device upon switch-on. Have your network administrator make the necessary configuration of the DHCP-server and the DNS-server. You can skip the rest of this section, in that case, and proceed with Configuration of the PC 50.

Note Notes for the Administrator

- We recommend using a fixed assignment of IP addresses to the devices and to set an unlimited lease-time. A device's host name cannot be changed. It is comprised of a prefix and the device serial number, e.g. " imcDev__99030143".
  For a device with a PPP-interface, it makes sense to set the following parameters:
- local and remote IP address,
- Baud rate and protocol of the serial interface to modem or cable,
- DNS-server address (optional),
- user name and password (optional),
- scripts for connecting and disconnecting (optional).

If there is no DHCP-server, the necessary device settings must be made as described further below. Have your administrator issue IP addresses for your device units and make note of the address assignments. Have the administrator enter the device units in the DNS-server if there is one. In this case, also make note of the DNS-server's IP address and the name of your network domain.

Find out the subnets in which the devices are to be operated and make note of the corresponding subnet masks. You will also need the IP addresses of any routers present which your devices units are to use.

Once you have gathered all the above information, you can proceed with the next section.

4.2.1.1 Configuration of the PC

To operate TCP/IP devices you need a PC on which the TCP/IP-protocol is installed and ready. We assume here that your PC is already set up appropriately. If this isn't the case, follow the instructions in your network card's documentation and in the WINDOWS help texts in order to install TCP/IP on your PC.

Avoid changing your PC's IP address if the PC is on an existing network! Simply adjust the configuration of the measurement devices!

Firewall and Ports

The firewall used might **prevent** establishment of a **connection** between the operating software and the measurement device. If the software is unfamiliar to the firewall, then a corresponding prompt usually appears upon **accessing the network**. This happens, for example, after starting the software and upon first accessing the device (e.g. in a device search). Some firewall programs can also block access to system and hardware-components.

In all cases, **correct operation** is only possible if the **connection is not blocked**. The following programs are usually affected:

Program	Default path
imc STUDIO: imc.Studio.exe	C:\Program Files\imc\imc STUDIO...
imc DEVICES: imcDevices.exe	C:\Program Files (x86)\imc\imc DEVICES...
imc DEVICES Service: imcDevicesService.exe	C:\Program Files (x86)\imc\Shared

For operation on LAN, it is sufficient to **enable "*Private networks*"**. Enable this access type. You may wish to read the operating instructions for your firewall, and / or ask your administrator / IT-department.

Windows Defender Firewall: Unblocking is performed automatically upon installation.

Note

Alternatively, you can enable the ports. To do this, each port must be entered for UDP and TCP. The following ports are used:

- UDP 1200
- TCP 1200 through 1202

4.2.1.2 Use of TCP/IP devices in various network situations

In some cases it's necessary to be able to run the PC together with the measurement device both in a network (e.g. company network) and, for mobile applications, with a point-to-point connection. In order to enable convenient and reliable operation in both settings, the following manner of proceeding is recommended.

In larger networks, the IP addresses for the network's PCs are usually assigned automatically ("*Network settings*" > "*Internet protocol (TCP/IP)*" > "*Properties*" > "*Obtain IP address automatically*"). For that purpose, the network comes with a DHCP-Server (DHCP: **D**ynamic **H**ost **C**onfiguration **P**rotocol).

If the Windows operating system is started on the PC in such a network (a physical connection between PC and the network (ethernet cable) exists!), then the PC is automatically assigned an appropriate IP address and the value for the subnet mask is set. The measurement devices can obtain appropriate IP addresses and subnet masks in the same way if the use of DHCP for the device is enabled. Thus, within a network with a DHCP-server, there aren't any address conflicts and no problems finding the devices in the network.

If the same PC is now started outside of the network environment (where there isn't any DHCP-server; e.g. on board a vehicle), then assignment of the IP address proceeds according to the APIPA-protocol (**A**utomatic **P**rivate **IP A**ddressing). In such cases, an IP address is assigned by the Windows operating system from a reserved IP address range (169.254.0.1 ... 169.254.255.254). The subnet mask is set to the value 255.255.0.0. See also AutoIP 52.

In this case the measurement devices use the IP address set in the device.

If an address from the range reserved for APIPA is selected in the device as the IP address, and the subnet mask is set to the value 255.255.0.0 (the use of DHCP for the device remains enabled!), then it's also possible to set up a connection between the PC and the device within the company network without any problems.

Note

- The IP address actually used for the PC and the value for the subnet mask can be determined by entering the following command line in the "(MSDOS-) input prompt": "`ipconfig /all`"
- If an IP address for a device is to be set from the range reserved for the APIPA, it must first be checked which IP address has been assigned for the PC, if no DHCP server is available. To do this, the PC is deactivated and disconnected form the network. After activation of the PC, the IP address currently in use is determined as described above. Make a note of this IP address!
- Ensure that each device is assigned its own IP address. This also goes for assignment of IP addresses from the range reserved for the APIPA! This prevents address conflicts even in networks without a DHCP-server.
- Fundamentally, the physical connection (ethernet cable) between the device and the network should be set up before the device is activated, so that automatic assignment of the IP address can work. If the automatic IP address reference fails, the IP address already set remains valid from this moment until the device is deactivated. However, this address isn't generally suitable for the IP addresses used in the network and address conflicts can result, or locating the device in the network will fail!

4.2.1.2.1 Computers with multiple TCP/IP connections

If you have set up multiple TCP/IP interfaces in your PC system, be sure to enter different subnets for each one! Otherwise random conflicts will occur. Reasons for having multiple TCP/IP interfaces include:

- You separate your connection to the company network and your connection to the measurement device by means of two network boards.
- You occasionally connect your measurement device via modem (PPP)
- You have added a Bluetooth connection.

The settings for the addresses could look like this, for instance:

	Measurement device	Router-company	Network Board1	Network Board 2
IP address	192.168.1.3	192.168.0.1	192.168.1.26	192.168.0.13
Subnet mask	255.255.255.0	255.255.255.0	255.255.255.0	255.255.255.0

In this example, your PC would communicate via Network Board 1 with the device and via Board 2 with the company network.

4.2.1.2.2 DHCP and APIPA

The settings presented here apply to the imc device, not the PC:

Setting	Description
Use DHCP	(Dynamic Host Configuration Protocol) The device tries to get its settings from a DHCP server. If the server is not available, the fixed settings are used! This corresponds to the alternative configuration in Windows PC.
DHCP ClientID	The *DHCP ClientID* can be entered if the DHCP server requires a special DHCP ClientID for IP assignment. If there is no entry, the MAC-address 61 is used. Supposing measurement devices are to be assigned IP by the DHCP server from a particular range. In that case, 'MeasDev 1' etc. could be entered, for example.
DHCP Hostname	If your administrator makes the device accessible via a DHCP Hostname, enter it here. Otherwise, leave this box empty.
Auto-IP (DHCP + APIPA)	(*Automatic Private IP Addressing*) *Use DHCP is* automatically activated along with this option. If the DHCP server cannot be reached, the fixed configuration for the address is not used. Instead, the device uses a random number generator to get an address in the range from 169.254.1.0 to 169.254.254.255. Afterwards it tests whether the address generated is available. If it is already used for another device, the search is repeated until a free address is found.

Note

If the PC's network interface is configured to "Obtain IP address automatically" (e.g. DHCP), then there is a problem with point-to-point connections when the network connection is cut and then restored (e.g. restart of the device interface in the process of a firmware update).

In such a case, the restoration of the device connection is delayed until the PC's network interface is completely reactivated (with a permanently valid IP address).

4.2.2 Start the network configuration - Interface configuration

By means of the dialogs "*Device Interfaces*" and "*Interface-Configuration*", it is possible to find out whether connection to the device can even be established at all. You are able to review the device's network system and to adapt the settings to the local network.

All entries are checked for plausibility, thus preventing incorrect entries. Even so, making the configuration requires system expertise and you may need support from your network administrator.

With imc STUDIO various options for interface configuration are supplied:

Programs	Description
Device Interfaces 53	Offers a simple guided interface for modifying the devices' LAN-configuration. If the simple configuration is inadequate, it is possible to start the advanced Assistant via the dialog: Interface-Configuration 53. The following connection types can/can't be configured via the dialog: **yes** / **no** Device connection via LAN / Device connection via Modem (TCP/IP with PPP) Device connection via WLAN (IP-configuration only) / Device connection via WLAN (advanced configurations, such as speed, encryption, ... but must be pre-configured via Interface-Configuration 53.)
Interface-Configuration 53	The program is used for the extended configuration of imc devices.

Start the configuration with "Device interfaces"

Start imc STUDIO. Click the "*Device interfaces*" button ( ) to open the dialog for configuring the IP address of the device.

Ribbon	View
Setup-Configuration > Device interfaces ( )	Complete

If this button is not present in the view you are in, it is also possible to open the dialog after a device search if it failed to find any new devices. Subsequently, a prompt appears asking whether to search for devices with an inappropriately configured network interface. Close this message box by clicking "*Yes*".

Reference Make settings

See further: "*Device connection via LAN* 55"

Start the configuration with "Interface-Configuration"

Start the program "*imc DEVICES Interface-Configuration*".

Start via:	Description
imc STUDIO	Click the "*Device interfaces*" button ( ) to open the dialog for configuring the IP address of the device. (See "*Start the configuration with "Device interfaces"* 53") There, click on the button: "*Advanced Configuration*".
the Start menu	Start the program: imc DEVICES Interface Configuration, from the program group: "imc" .

At the left there is a tree labeled "*PC*" and below it one entry for each network adapter which uses TCP/IP. The name or IP address of your PC is usually displayed here. The respective entry can consist of either a domain name or the associated adapter's IP address. This includes the dial-up network's adapters.

Note Inactive entries

If all the entries have been set as inactive or there is no entry, your PC is not configured correctly. Exit the program and check all network settings. Subsequently, repeat the configuration.

Reference Make settings

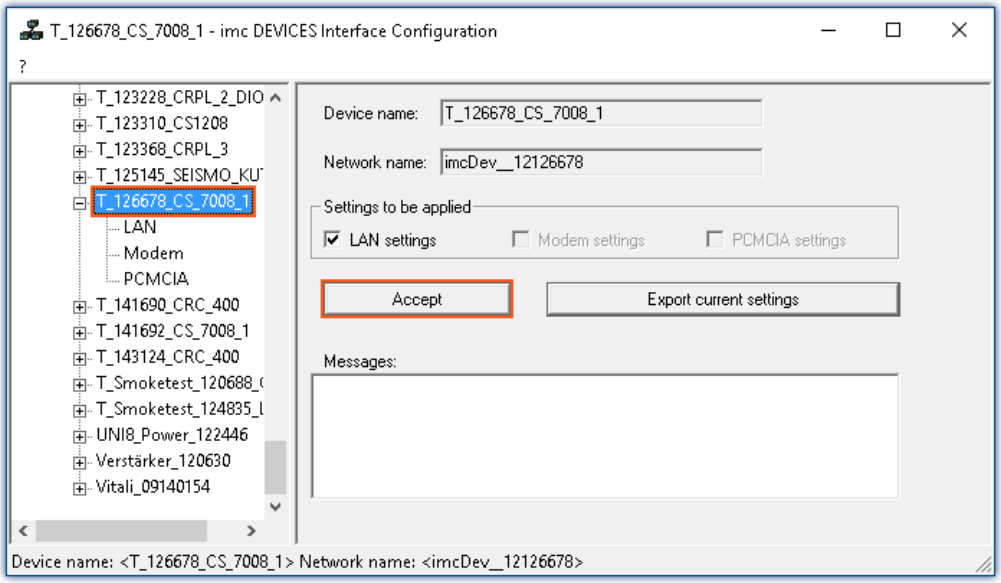
See further:

- Device connection via LAN 57
- Device connection via WLAN 59

Accept changes

The configuration is only written to the respective measurement device once you click on the button "*Accept*".

- After the configuration, select the device again in order to return to the overview.



- In the group "*Settings to be applied*", select the device adapters for which the changes are to be applied.
- In order to write the changes to the measurement device, click on the button: "*Accept*"

Once the configuration has been written to the device, a warm start is carried out so that the new settings take effect. This process requires some time.

Reject changes

To reject all changes, simply exit Interface-Configuration and respond to the prompt "Reject all changes?" with "Yes". If you select "No" for this prompt, the entries you made on the right side of the dialog remain intact. But they will only be applied once you click on the button *Accept*.

Export current settings

For documentation purposes, it is possible to export the interface settings currently applicable in the device to a CFG-file. Import is not available. The device folder is automatically the destination and the suggested file name is the device name with a suffix.

Note

After confirmation "Interface-Configuration" allows, that the subnet mask has a format, that is not common for that network class.
Example: Networks starting with 192 are so called class C networks. They always use the subnet of the format 255.255.255.X. In special situation it is necessary to change the format to 255.255.X.X.

4.2.3 Device connection via LAN

Preparation

Connect the devices to the LAN 46 and switch the devices on.

Note

No devices found

If the devices described below fail to find any devices, check whether your PC and the devices are properly connected to the network and activated.

If finding devices still does not succeed, set up a direct connection to the PC. You should also check whether the Firewall and port 50 setting is preventing a network connection to the device.

DHCP

DHCP (Dynamic Host Configuration Protocol) in conjunction with an appropriate server enables dynamic assignment of an IP address to the devices (see "*DHCP and APIPA* 52").

The devices ship set for DHCP. If an DHCP server is activated in your network, no further manual settings are needed. See your network administrator if you have any questions.

Warning Valid IP address

To set a valid and available IP address for the device in an existing network, be sure to contact your network administrator. If you set an IP address for the device which is already used by another device in the network, the result will be that one of the devices is no longer available in the network.

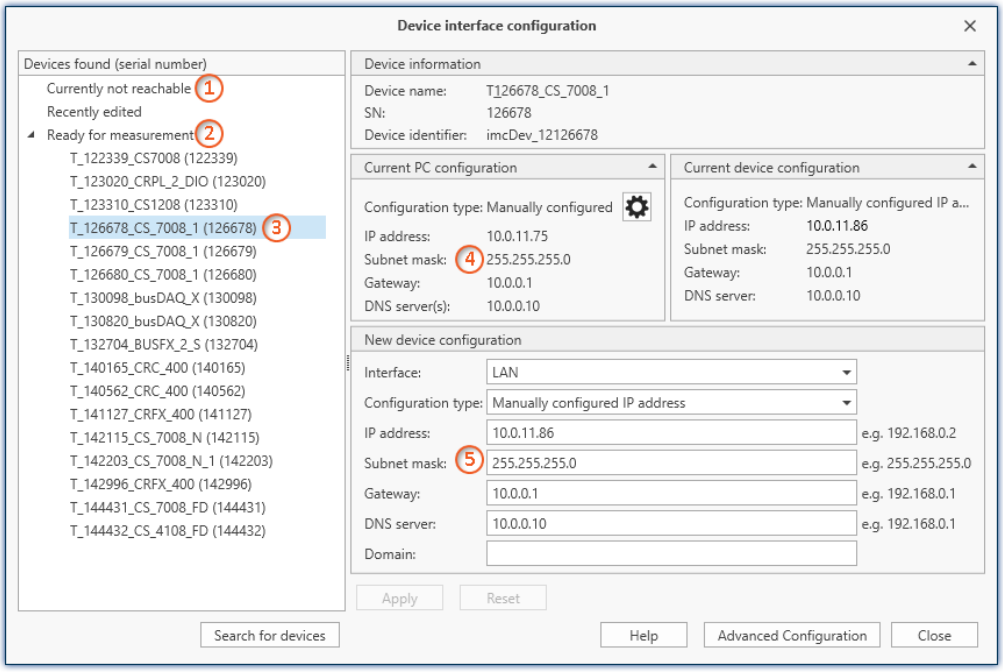
LAN-Configuration with Device Interfaces

Click the "*Device interfaces*" button ( ) to open the dialog for configuring the IP address of the device. See "*Start the network configuration - Interface configuration* 53".

Ribbon	View
Setup-Configuration > Device interfaces ( )	Complete

Once the dialog starts, the system automatically searches for all devices in the network. In the tree diagram, all available devices are indicated. If the device appears among the group "*Currently not reachable*" ①, it is necessary to modify the device's LAN-settings. If the device appears among the group "*Ready for measurement*" ②, you can leave the settings as they are or review them.

Select the device for making modifications ③.



Display of measurement devices found and of the IP address

The right-side portion is subdivided into four regions:

Region	Description
Device information	Lists information for identifying the device.
Current PC configuration	Lists the current configuration of the PC-adapter via which the device was found. By means of the cogwheel-icon (⚙) you can directly open the PC's adapter settings.
Current device configuration	Lists the current configuration of the device adapter.
New device configuration	Here, you can modify the device interface configuration.

Support by a configuration suggestion

This dialog offers support in configuring the interface. If the device's configuration is not appropriate for the PC, an appropriate configuration is found and suggested. Suggestions appear in "green" shade.

Note — Check the suggestion

Please check the suggestion exactly! Do not accept the suggestion without first verifying the IP address: Is it a valid and available IP address?

Subject to the PC-configuration, the next free IP address is offered as the suggestion. Free IP addresses might be occupied by devices which are currently switched off. In most cases, the suggested configuration can be used for a "point-to-point connection", since there are no other devices in the network in this case.

No suitable suggestion:

In some cases it may occur that no IP address is suggested. For IT reasons, the software is not able to conduct a search across all IP addresses and only a small number of IP addresses are checked. If all addresses checked are occupied, no result is returned. If the device is selected once again, a new search is started among other addresses.

If no address is suggested, it is best not to conduct any more searches but rather to manually enter a suitable address.

Procedure

- Select the device
- Check the settings and make the appropriate changes
- To write the changes to the measurement device, click on the button: "*Apply*"

Once the configuration has been written to the device, a warm start is carried out so that the new settings take effect. This process requires some time.

Advanced configuration

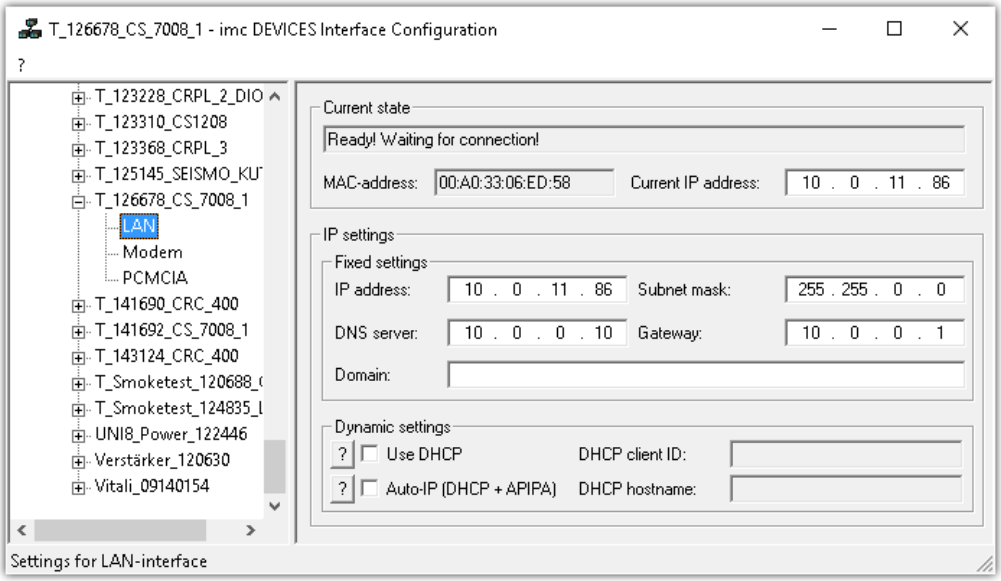
In some networks, configuration via the dialog is not adequate. For instance, it may be possible to activate the device connection via WLAN, but not to configure it. A device connection can not be established via modem (TCP/IP with PPP). In these cases, click on the button: "*Advanced Configuration*". The dialog: "*Interface-Configuration* 57" opens. Follow the instructions for configuring the device interface.

LAN-Configuration with Interface-Configuration

You can adjust the LAN configuration via the "*Interface-Configuration* 53" dialog.

Double-click on the network adapter's entry in order to start an automatic search. Depending on the network and the number of devices connected to it, this can take a while.Then the tree diagram will display all available devices under the adapter.

There, select the device desired and click on the sub-item "*LAN*". You can now see the measurement device's current configuration, as shown below:



TCP/IP configuration

Enter the IP settings in the right-side region.

Region	Description
Current state	Lists the device-adapter's current configuration.
IP settings	Here, you can modify the device interface configuration.

To write the changes to the measurement device, click on the button: "*Accept* 54"

Once the configuration has been written to the device, a warm start is carried out so that the new settings take effect. This process requires some time.

4.2.3.1 Connection via a direct address - User-defined device

In a structured network (network with routers, Internet, ...), imc devices could not be integrated by means of a network search. With the knowledge of the IP address or of the domain name (DNS name), it is now possible to integrate a device into the "*Device Table* 290" and to establish a connection.

To do this, run a devices search by IP address or DNS name.

Ribbon	View
Setup-Control > Search for devices by IP/DNS ( )	Complete

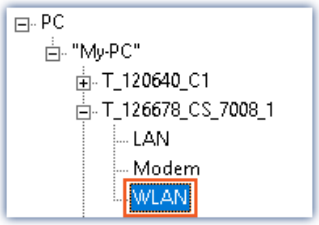
Proceed as follows:

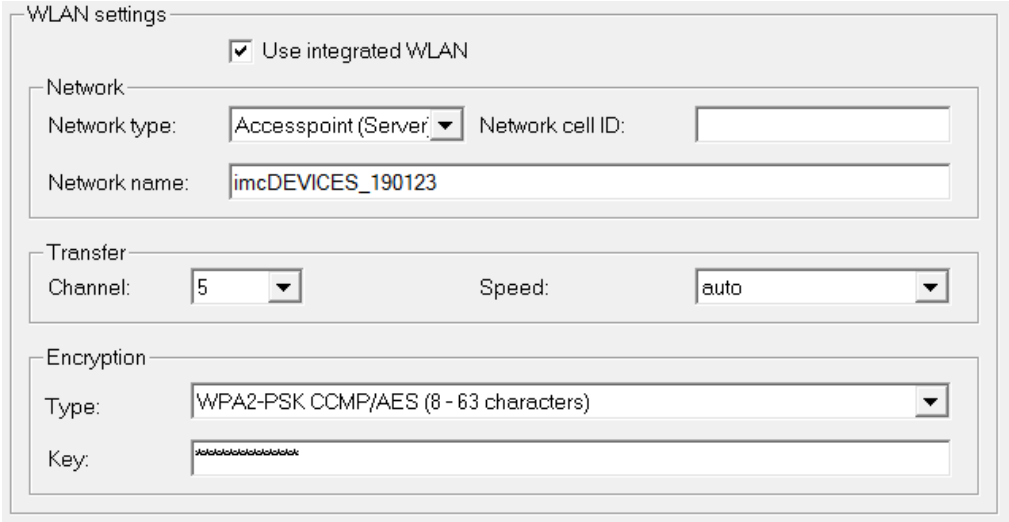
1. Open the dialog "Search for devices by IP/DNS".

2. Enter the IP address or the DNS name.
3. Clicking on the button "*Read device information*" initiates an attempt to establish a connection. If it is successful, then the device's name, its serial number and its product type are displayed and the device can be integrated into the experiment.

4.2.4 Device connection via WLAN

You can adjust the WLAN configuration via the "*Interface-Configuration* 53" dialog. This can be reached via the dialog for "*Device interface-configuration*"-> "*Advanced Configuration*" 57. There, select the device desired and click on the sub-item WLAN.





Parameter	Description
Network type	• **Managed (Client):** In most cases, the measurement device is connected at an Accesspoint. For this purpose, the device must be operated in the mode "*Managed (Client)*" with the encryption specified (usually "*WPA2-PSK*"). It is crucial for the WLAN IP address range to be different from the LAN-IP address range. Example: LAN: IP 192.168.1.20 -> WLAN **may not have** an IP from the range 192.168.1.x! • **Accesspoint (Server):** If no Accesspoint is available, it is possible to operated some imc devices as an "*Accesspoint (Server)*". Here, too, the use of "*WPA2-PSK*" encryption with a secure password is recommended. Both the measurement device and the PC must have a permanently set IP if there is no DHCP Server available. • **Ad-Hoc mode** is no longer supported under Windows 10.
Network cell ID	This may be required in order to select an Accesspoint (4 hexadecimal characters); otherwise, leave this empty.
Network name	Name of the network to which you intend to connect the imc device: • For the network type *Accesspoint (Server)*, the imc-device automatically provides the WLAN name via which the PC connects with it. • For the network type *Managed(Client)*, you must enter here the WLAN-name via which the imc device is to be accessed.

Parameter	Description
Channel	Channel 1-13 for 2.4 GHz. If the WLAN-module supports 5GHz, there are >36 channels. Observe all national regulations. For instance, in the USA, channels 12 and 13 may not be used.
Speed	The meaning of *auto* depends on the integrated WLAN-card! *11 Mbit/s* ***max*** means that the system attempts to transfer data at 11 Mbit. If the connection is too bad, however, the transfer rate is reduced automatically. With the setting *11 Mbit/s* ***fixed***, the transfer rate is not adapted automatically.
Encryption	The key must be the same for all participants. WPA-PSK or WPA2-PSK; the latter is more secure. Radius-authentication is not supported. WEP has become obsolete and is not recommended.

Note

- Do not enable "***Use DHCP***" in the "*Dynamic settings*" section if the device operates as an **Access point (Server)**. However, the device does not assign IP addresses! These must be assigned in the same IP address range in the PC-WLAN settings. Make sure that the subnet mask on the PC corresponds to the device.
- If a maximum value has been set for the speed and the connection quality is bad, the speed is adjusted automatically. In consequence of frequent speed adjustments, there may be data overflows. In such a case, set a lower maximum speed.
- The device does not support data exchange between the WLAN connection and the LAN connection. Even if unauthorized access to the WLAN takes place, there is no danger of access to the device via the LAN.
- For devices as of group A4 and higher (SN>13xxxx), up to **54 Mbit/s** can be transmitted.
- For devices as of group A7 and higher (SN>19xxxx) up to **300 Mbit/s** can be transmitted. Therefore *Speed* at *Transfer* must be set to "*auto*". Furthermore the *Network type "Managed*" may be activated and please select as Encryption "*WPA2-PSK CCMP/AES (8-63 characters)*". If you select another encryption, then the speed will be reduced to 54 Mbit/s conforming to the 802.11n Standard.

4.2.4.1 WLAN connection setup possibilities

PC via Access Point - Device via integrated WLAN

PC via Access Point - Device via integrated WLAN

The PC is connected with an Access Point. This connection is made either with a cable or a stabile WLAN connection. The Access Point can be chosen in such a way to ensure adequate power supply for a strong antenna.

PC via Access Point - Device via Access Point

PC and device via Access Points

This option is well suited for wireless transmission across especially long and problematic expanses.

At distances of over 30m, we recommend the use of Access Points at both ends. This means that strong, directed antennas can be used whose power supply can be ensured by the choice of appropriate Access Points.

The connection between the Access Point and the PC or the device is made by an cable or stabile WLAN hookup.

In this case, the interface from the PC and the measurement device is configured like a normal LAN connection 55. The cable is replaced by the wireless connection and administered by the Access Points. These must be configured in accordance with the instructions provided for them.

4.3 Information and tips

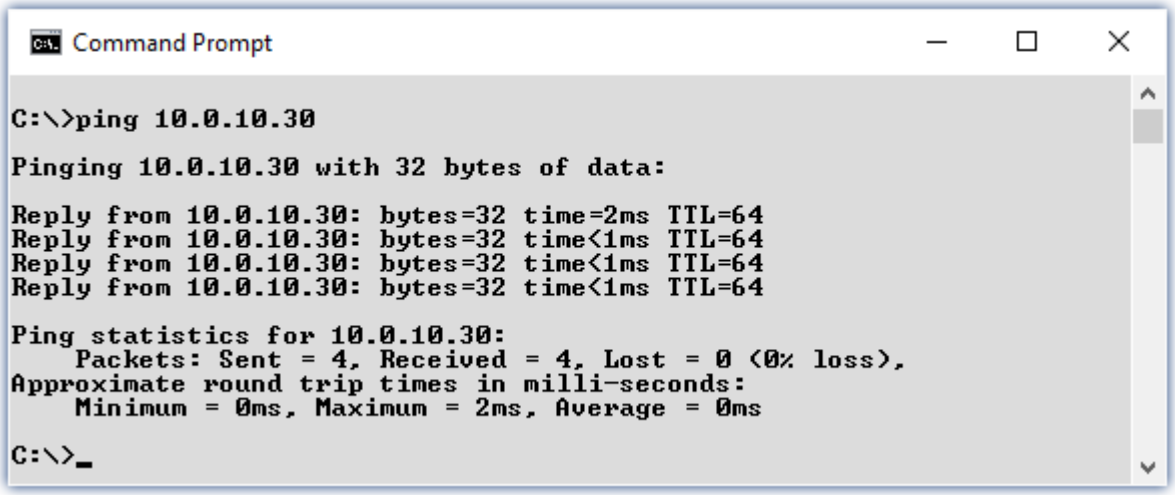
4.3.1 Determining the MAC-address of the measurement device

In order to find your imc measurement device's MAC-address, start the "input prompt" (Keyboard combination [<Win> + R]. Type cmd and confirm by pressing Return).

There, begin by accessing the device with a ping and the IP address. The IP address can be found with the help of the program Interface-Configuration 53.

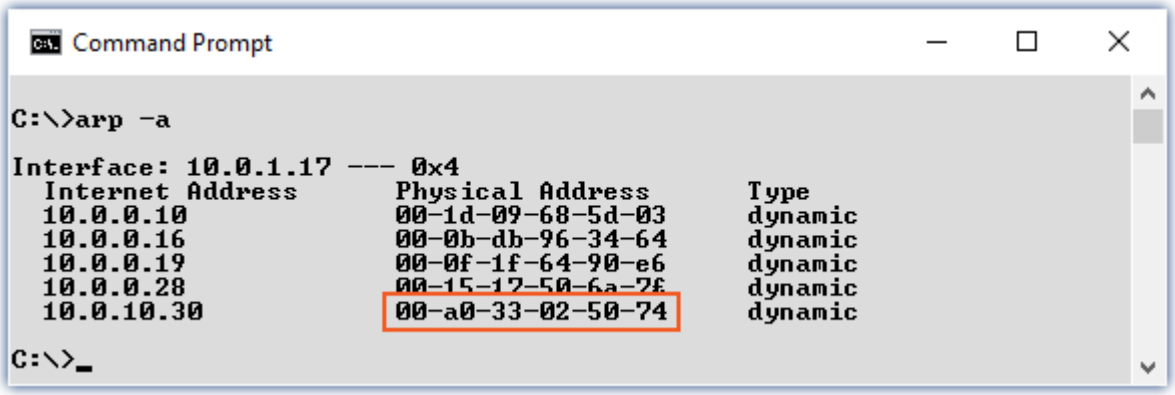
Example for the ping command:

```
ping 10.0.9.30
```



Next, enter the following command:

```
arp -a
```



In the second column ("Physical Address") you will find the MAC-address.

4.4 The network

Optimize the network

You should observe the following notes for networks with high data rate, to reduce the risk of a data overflow.

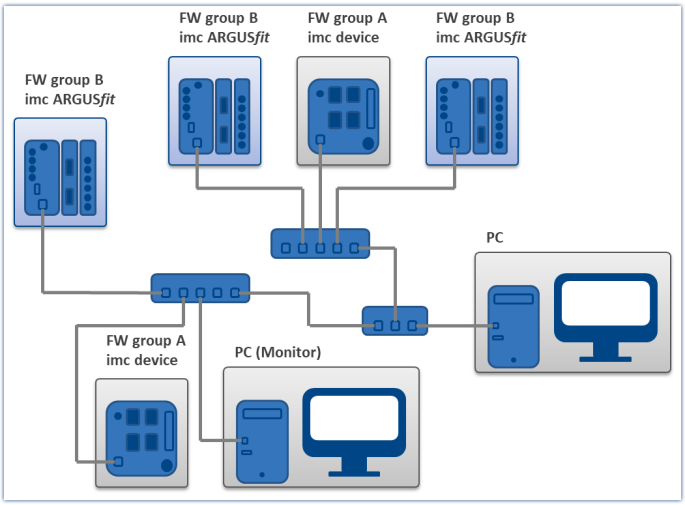
1. The network between switches and PC should be 1GBit/s
2. The actual data rate is decreasing with the number of channels. Avoid unnecessary channels.
3. The RAM buffer should be as big as possible, not "*Auto*". See also RAM buffer time 427.
4. Save only channels you need to the internal disk.
5. Avoid curve window setting that need a lot of performance. Choose scroll mode. Avoid the rezoom function after a long measurement duration. Don't set the sample points to symbols. Try to find a curve window configuration, that shows no bucking.
6. Observe the recommended virus scanner settings 41! It would check every sample written to the hard drive. That takes most of the performance of a PC and the hard drive.
7. Check the processor and network load with the PC task manager

Using a second network

To avoid disturbing data transfer between devices units and PC's, a second network using a second network card per PC can be set up. This second network can be used for transferring data between the PC's.

Example of a Complex Network

Until now, we have only considered networks consisting of a PC and n imc devices units. However, any arbitrary combinations are possible.



Monitoring

The PC on which imc STUDIO is installed is declared as the Master-PC, which configures the measurement device.

For devices belonging to the firmware group A 187 - e.g. CRONOS family, C-SERIES, SPARTAN, BUSDAQ:

Various other clients can access the measuring device via the network (e.g. via imc STUDIO Monitor or imc LINK or, it is set up accordingly, using a browser via imc REMOTE WebServer).

The well-known network restrictions are also valid here. In particular, it is impossible for more than one person to write to the same file. Furthermore, it is impossible for one person to open a file while another writes to it. In general, all network restrictions that apply to files also apply to the individual Devices units.

In consequence, the software con prevent multiple users from configuring the same device.

4.5 Firmware version

imc DEVICE*core* and imc DEVICES as a driver package for imc STUDIO

The drivers and firmware components for imc devices are provided in "imc DEVICE*core*" and "imc DEVICES". Thus, at least one of the two driver packages is a necessary requirement for operating imc devices under imc STUDIO.

- imc DEVICE*core*: For devices belonging to the firmware group B 187 - e.g ARGUS*fit*, EOS
- imc DEVICES: For devices belonging to the firmware group A 187 - e.g. CRONOS family, C-SERIES, SPARTAN, BUSDAQ

Both packages can be installed and operated in parallel. The software and the devices automatically apply the appropriate package.

The software only works with devices having the right firmware. Upon the connection/preparation, the versions are compared. If they don't match, the firmware update 64 is carried out.

Note Parallel firmware versions

imc DEVICE*core*: The associated firmware- and driver package ships with the software used, and is installed in the software's installation folder. Any other installations are not recognized and thus cannot be used.

imc DEVICES: It is possible to work with multiple groups of imc DEVICES simultaneously and in parallel. In special cases, this may be relevant or even recommendable, particularly in conjunction with the use of multiple devices. Additional notes on this subject are provided in the chapter: "*Associated Firmware (imc DEVICES)* 68"

4.5.1 Firmware update

Every software version comes with matching firmware for the hardware. The software only works with devices having the right firmware.

Once the program connects up with the unit, the device's firmware is checked. If the software version doesn't match the device's firmware version, you are asked if you want to perform a firmware-update.

Note

The firmware update is only required if the software was obtained as an update. If you obtained your hardware equipment together with the software, no firmware update is necessary.

Warning Do not interrupt the firmware update!

Be absolutely certain to observe the following:

1. Under no circumstances should the device or its power supply be deactivated during the firmware update!
2. The network connection may not be interrupted. Use a cable connection, not WLAN!

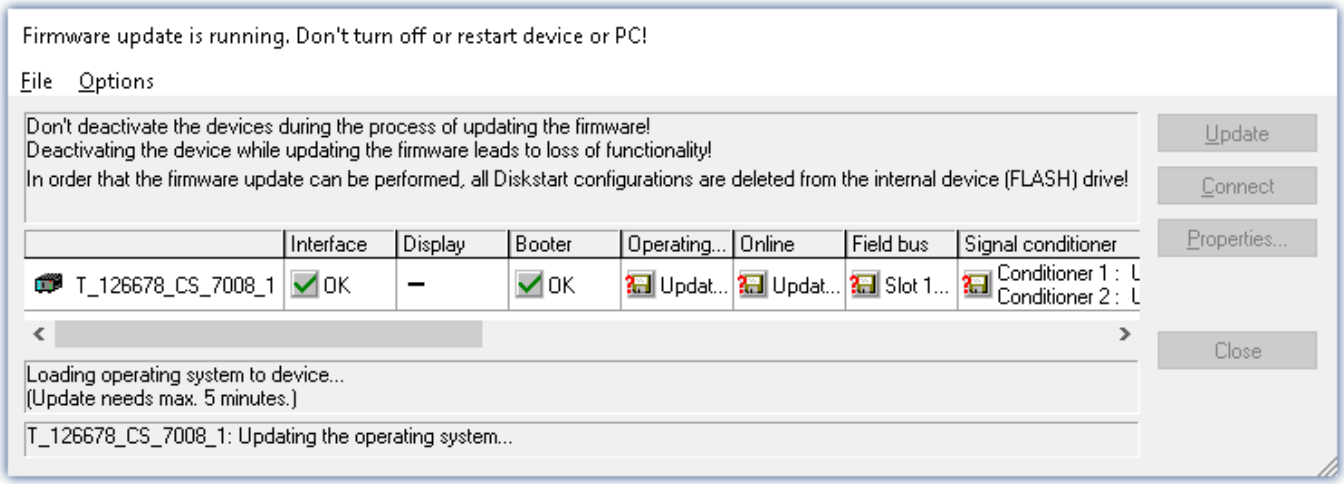
Firmware update - imc DEVICES

Note Firmware and driver package

The following description applies to devices of firmware group A (imc DEVICES) - e.g. CRONOS family, C-SERIES, SPARTAN, BUSDAQ.

Depending on the device type, the following components are loaded automatically: Interface-firmware (Ethernet, modem, ...), booting program, amplifier firmware, firmware for the signal processors.

The dialog for the firmware-update looks like this:



Start of the firmware update (example of a single device)
The state of the components of the firmware is diaplayed in the list.

Component	Description
Interface	Interface-Firmware (Ethernet)
Booter	Start-up program for the device upon switching-on
Operating system	Device operating system
Online	Online-functions and hard drive controller
Display	Operating system of the connected displays 786
Fieldbus	Fieldbus interfaces (e.g. CAN etc.)
Signal conditioners	Amplifiers

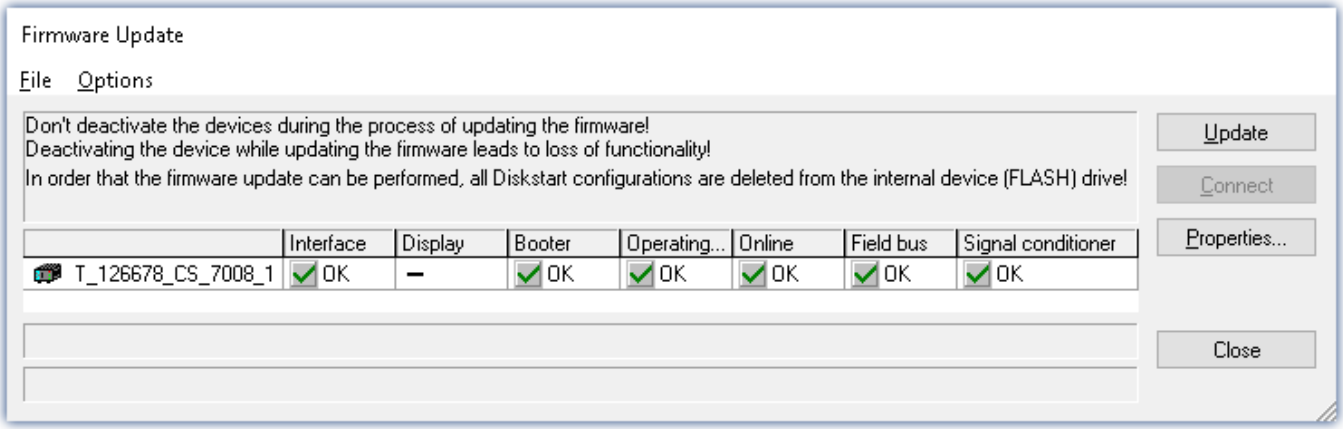
The following symbols for the individual firmware components appear in the list:

Symbols	
	not current
✓	firmware conforms to current standards
	error occurred during update procedure
—	this option is not available on the device

If no status indicators are displayed, no connection could be made to the corresponding device.

The duration of the update depends on the number of amplifiers (can last up to several minutes). You will be informed on the progress.

You are notified when the firmware setup concludes successfully, as shown below:



Conclusion of the firmware update (example of a single device)

Choose "*Close*". The device can now be used with the product software.

Warning Be sure to observe in case of error

- For a variety of reasons, the firmware update sometimes does not conclude properly, for example due to interruption of the power supply. For instance, the "handshake signal" at the end of the procedure may be missing. In this case, no measurement channels would be displayed initially. However, restarting the device and its software and performing the firmware update again usually restores everything to normal. It may be necessary to call the menu function "Update all components" in the Firmware update dialog's Options menu. This scenario only results in permanent damage in the most rare cases, and it is very worthwhile to repeat the procedure before sending a device in for repair.
- Behaviour under error condition, Windows cuts off the network connection without the user's knowledge; but this can be prevented using the PC's Control Panel.
  Background: During the firmware updates there is no data transfer for a few minutes and thus no network activity; Windows detects inactivity of the connection and the following mechanisms are set in motion:
  a) Windows' energy saving mode switches the LAN adapter off, consequently interrupting the network connection!
  b) Windows switches to the next LAN adapter if there is one (some PCs have multiple adapters in order to, for instance, to access services in parallel that are accessible via separate networks.)
  c) Other scenarios are feasible, e.g. if switches are activated, which can also respond to missing data traffic.

If an error message is posted during the firmware update, leave the device on and contact our tech support 8. The firmware update may be continued with guidance from the tech support.

Note

Firmware logbook

The "*File*" menu offers a function for working with the firmware log file. Every action taken during a firmware update plus any errors which may occur are recorded in a log file. This log file can be displayed with menu "*File*" > "*Show log file*".

Update all components

The "*Options*" menu offers the option to "*Update all components*". This makes it possible to earmark all the components of the selected device for an update. The function is only to be used in compliance with instructions from the tech support.

Firmware update - Password locking

It is possible to prevent an unintentional firmware update. For this purpose, the "*Access protection*" > "*Password*" and "*Use for firmware update*" control is provided in the "*Device properties dialog* 221".

Ribbon	View
Setup-Configuration > Device Properties	Complete

No default password is supplied in the manufacturing. When a firmware update is started, the system checks whether firmware updating is enabled in the device configuration; otherwise the firmware update is canceled with an error message.

⊞ **Device**	
⊟ **Access protection**	
Password	not active=0
Use for firmware update	don't use=0
Use for establishing connection	don't use=0
FTP-login name	imc
FTP-password	not active=0
Anonymous FTP access	enabled=1
⊞ **UPS**	

Blocking / Unblocking of the firmware update

Blocking and unblocking

The firmware update is blocked with a freely defined password. To unblock, you must enter the password which was used to install the blockage. It is only removed if the password is identical with the password saved in the device.

Note A general password is available

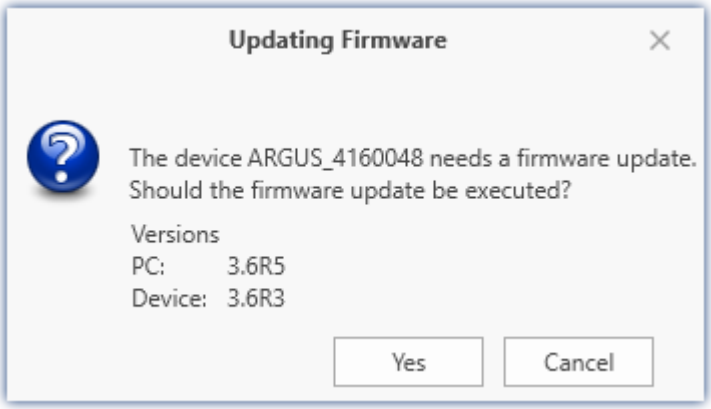
Resetting of the password can also be accomplished by a **General password**. This general password is derived from the device's serial number and can be provided by imc if necessary.

Firmware update - imc DEVICEcore

Note Firmware and driver package

The following description applies to devices of firmware group B (imc DEVICEcore) - e.g. ARGUS*fit*, EOS.

Upon the connection/preparation, the versions are compared. If they do not match, the next dialog notifies you about this, and indicates both which version is being used and which version is required.



When you click "*Yes*", the firmware update starts. The duration of the update depends on the number of amplifiers (can last up to several minutes). You will be informed on the progress.

The system notifies you when the firmware update has been completed successfully. Close the dialog. Subsequently the device can be used with the device software.

FAQ

What to do in case of an error occurring in a firmware update?

Answer: When a firmware update fails to run correctly, the last active firmware is restored. The device remains operational by means of the previous installed firmware. Repeat the procedure, and if the repeat procedure also fails, please contact our tech support.

What is an optional update?

Answer: In some cases, a firmware update is not mandatory. In such cases, the update is available to you as "*optional*". However, we still recommend performing the update, since every update not only implements improvements, but also fixes any bugs. Nevertheless, with the new version of imc STUDIO, you are able to work with the device running the old firmware edition, if you choose.

4.5.2 Associated firmware (imc DEVICES)

If multiple firmware versions are installed on the PC, then a defined version must be selected for each device. The selection strategy is set in the Options: "*Setup*" > "*Device options*" > "*Selection of the imc DEVICES version*".

Ribbon	View
Extra > Options ( )	all

Option	Description
Selection of the imc DEVICES version	If multiple imc DEVICES versions are installed on the PC, a specific version must be selected for the operation of each device. This option controls the selection strategy. If only one imc DEVICES version is installed, this setting has no effect. • **Manual:** With "**Manual**", the system always asks which firmware version to use when device are selected or an experiment is loaded. • **Automatic:** "**Automatic**" avoids firmware updates. The version currently running on the device is used preferentially. • **Always use latest:** With this selection, the most current firmware version compatible with this imc STUDIO version is always used.

Note Use of older firmware

Please note that a new firmware version not only contains new functions, but also reflects resolved issues. These changes only take effect if the device is also using the new firmware.

FAQ

Question: Which firmware versions support my imc STUDIO version?

Answer: In regard to the use of imc DEVICES as the device driver, there is a clear distinction determining the compatibility of versions/version groups:

imc STUDIO version	assigned firmware / version group	additionally compatible
5.2	2.10, 2.11, 2.13	
2022	2.15	2.14
2023	2.16	2.14, 2.15
...	...	

For version 5.2, the applicable rule is: the assigned version group and all **predecessor groups going back to 2.10** are compatible with the imc STUDIO-version used.

As of version 2022, the applicable rule is: the assigned version group and all **predecessor groups going back to 2.14** are compatible with the imc STUDIO-version used.

Question: Can devices in the same experiment use different firmware versions?

Answer: Yes. If multiple devices are used, each one may use a different firmware version. Prerequisite: The imc STUDIO-version used supports the firmware versions.

Question: Can I install a new firmware version without changing the imc STUDIO version?

Answer: Yes. Prerequisite: The imc STUDIO version used supports the firmware version.

Question: Is there a way to determine what firmware the device is using?

Answer: Yes.

- On the Setup page: "*Devices* 289", a column can be added: Device firmware
- In the Device properties 221.

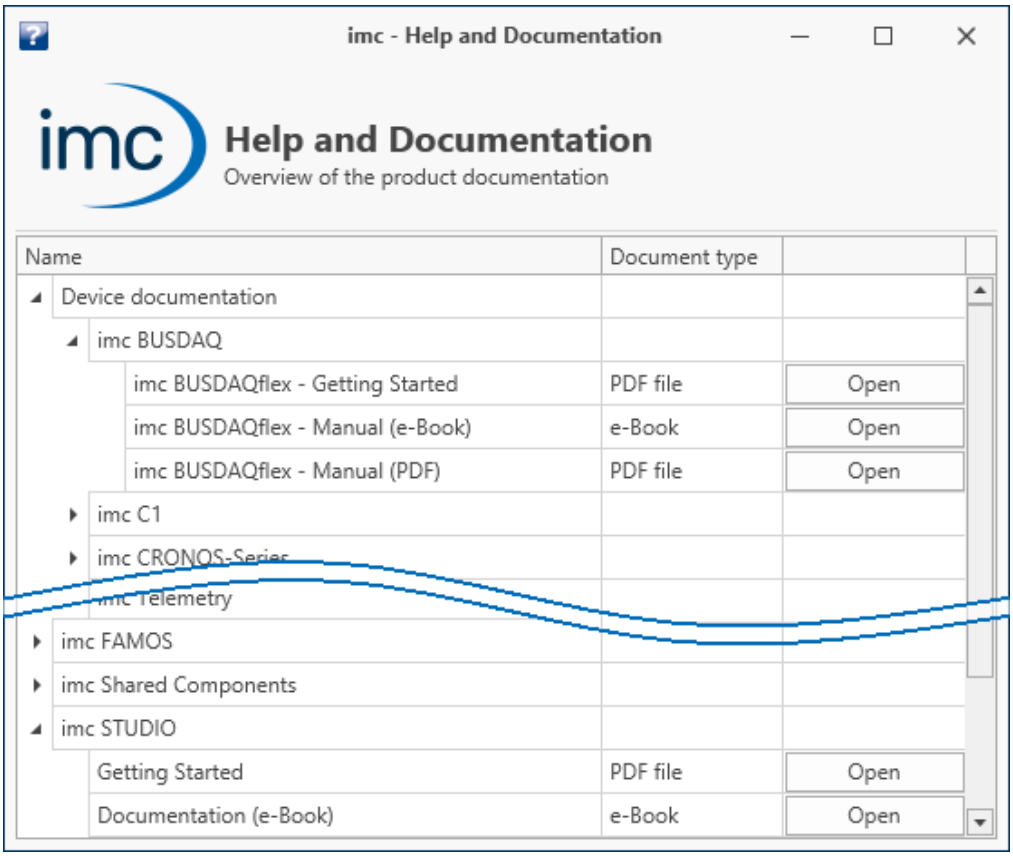
5 Documentation of devices - Help and Documentation

This manual is mainly about the device software. Separate documentation is available for each device model. Read this as well in order to obtain optimum utility from your device.

How to find the device manuals?

To do this, use the program "*imc Help and Documentation*" which helps to locate the documents. A few clicks of the mouse will open the desired document. This program is available

- via the Ribbon: "*Help*" > "*Additional Documents*",
- via the Startmenu under the group "*imc*" or
- on the installation medium ("*imc Help and Documentation.exe*")



"imc Help and Documentation" lists here the imc C-SERIES documents "First Steps" and "Manual" in both PDF and e-Book formats.

Note — Uninstalling the documentation

The documentation for older, already uninstalled software versions is not deleted automatically. Thus it may happen that multiple manual versions are listed. It is advisable to delete the old versions.

To delete documents manually, start the process of uninstalling the program: "*imc Help and Documentation*". A prompt appears to ask which documents to uninstall.

As long as not all of them are selected, only the selected documents are uninstalled. The program "*imc Help and Documentation*" is only included for uninstallation once the last documentation is uninstalled.

Thus, you will be able to uninstall old document versions without any difficulties.

6 Tutorials

The following are **tutorials** on **device configuration** and **measurement data display**. The subsequent chapters also include tutorials on other topics:

Reference

Guided first steps

If you are working for the first time with imc STUDIO, we recommend that you work through the guided first steps 37 before beginning any tutorials.

There is an assistant which helps introduce you to the program. The first steps – device selection, channel configuration and displaying of measured data – are usually identical.

Other tutorials

- Sequencer 1765
- Automation 1628
- Scripting 1908
- Third Party Device Interface 1942

Tutorials - Device configuration and display of measured data

This chapter presents some examples involving the components "*Setup*" and "*Panel*".

- Simple measurement - First steps 72
- Circular buffer memory for the display 79
- Subsequent saving of measured data 83
- Simple triggering 87
- Simple triggering - Enhanced: Trigger status 93
- Triggered measurement 96
- Thermocouple measurement 102
- Artificial sine - signal generation using imc Online FAMOS 106

6.1 Simple measurement - First steps

Assignment:

Set up an experiment. A measurement on an analog channel is to be started and the measurement results are to be displayed on a Panel page. The signal applied in this example is a ±2 V sawtooth voltage with a 1 Hz base frequency. If you use a different signal, modify the channel settings accordingly.

Learning goals:

- Creating an experiment
- Simple device operation

Elements used:

- Search for devices, device selection , connection with a device, channel settings

Prerequisite:

- imc measurement device with an analog channel

Procedure:

Connect a **signal generator** with "*Channel_001*" on your device. Refer to the device manual to see how to connect a signal with your device (plugs, etc.).

Start imc STUDIO as described in the section "*Start* 31".

6.1.1 Setup - Settings

To begin, the measurement device is selected and configured.

- Open the **device configuration page**: "*Setup*".
- Go to the **tab**: "*Devices*" .

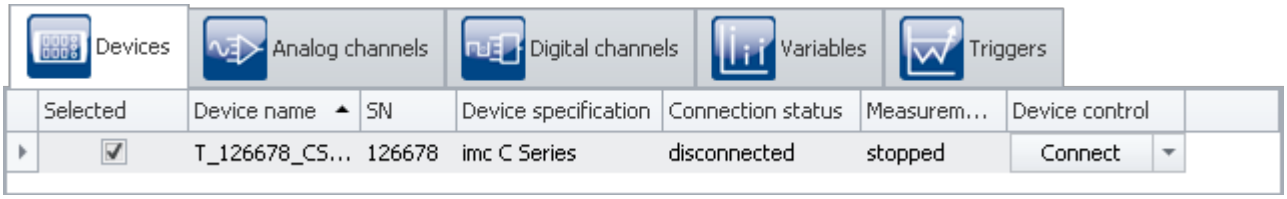
Here you will find any devices which are known to the system due to having been used before. Following the first installation of imc STUDIO, the list is empty.

Select your device:

- If the device desired **does not appear in the list**, perform a device search 241 ( ). Subsequently, the list will display all devices found.

Ribbon	View
Home > Search for devices ( )	all
Setup-Control > Search for devices ( )	Complete

- Click on the Checkbox-symbol ( ) in order **to select the device**.



Selecting a device

The device is now known and selected, and is available each time imc STUDIO is started.

Configuring a channel

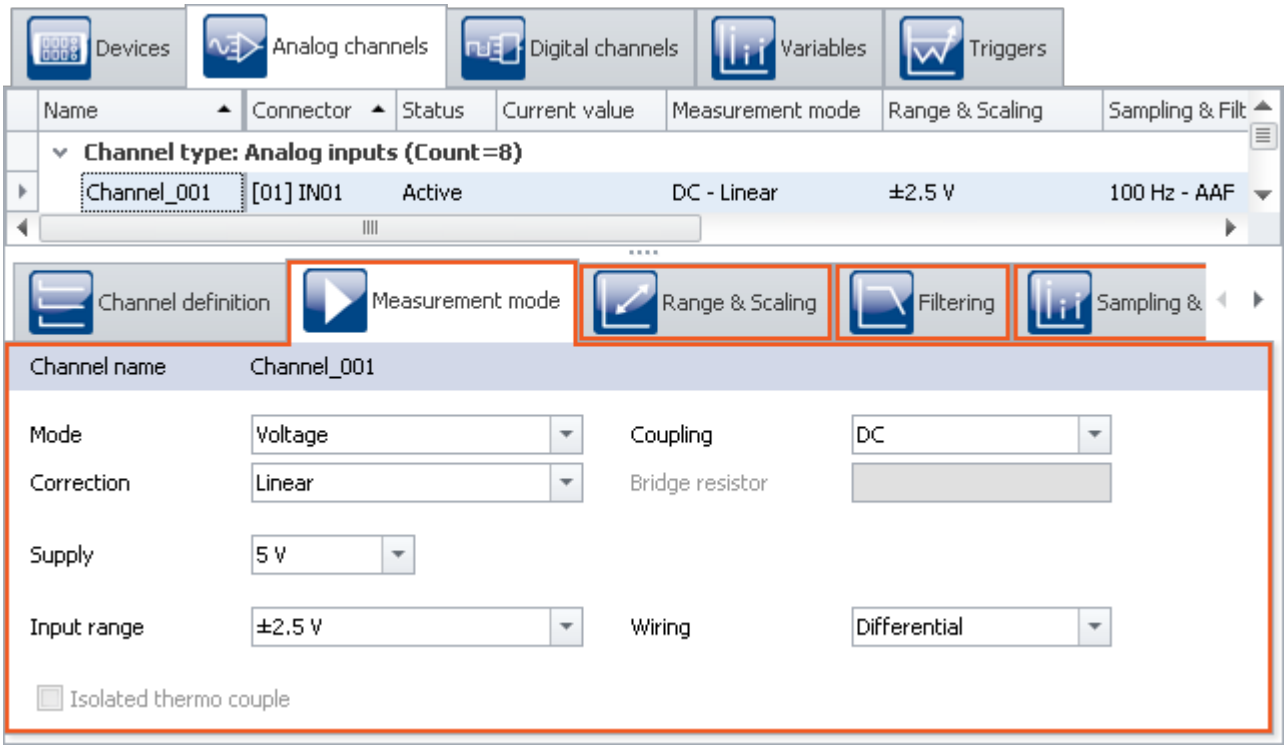
Now the channel used is configured:

- Go to the **tab**: "*Analog channels*" .

Under **Channel type**: "*Analog inputs*", select "*Channel_001*". By default, this is set to "*active*". In order to make settings, click in the respective box and change the entry.

Set "*Channel_001*" in accordance with the input signal from your generator. In this example:

Parameter	Value	Dialog (View: *Standard*)	Dialog (View: *Complete*)
Mode	Voltage	Measurement mode	Measurement mode
Coupling	DC	Measurement mode	Measurement mode
Input range	±2.5 V	Measurement mode	Range & Scaling
Type	AAF	Sampling & Filtering	Filtering
Sampling rate	100 Hz	Sampling & Filtering	Sampling & Preprocessing
Duration	10 s	Sampling & Filtering	Sampling & Preprocessing



Configuring measurement channel by the dialogs

- To **apply the changes**, click on the button "*Process configuration* 241" ( ).

Ribbon	View
Home > Process ( )	all
Setup-Control > Process ( )	Complete

By this means, the device settings are available to other components.

6.1.2 Panel - Settings / Start measurement

To display measured values, create a Panel page.

- Open the page: "*Panel*".

Reference Panel description

How the Panel is structured is presented in the chapter: "*Panel - Data visualization and controls* 1205".

Editing can only be performed in **Design mode** (e.g. add new **Widgets**, change **properties** etc.).

- Activate the Design mode ( )

Ribbon	View
Panel-Navigation > Design Mode ( )	Complete
Panel-Design > Design Mode ( )	Complete
Home > Design Mode ( )	Standard

Note Notes on Data Browser

The "*Panel*" has various **tool windows**. In "*Data Browser*" you can see the **device's channels** as shown in the image below. **Following changes** to the "*Setup*", this list must be updated using the command "*Process configuration* 241".

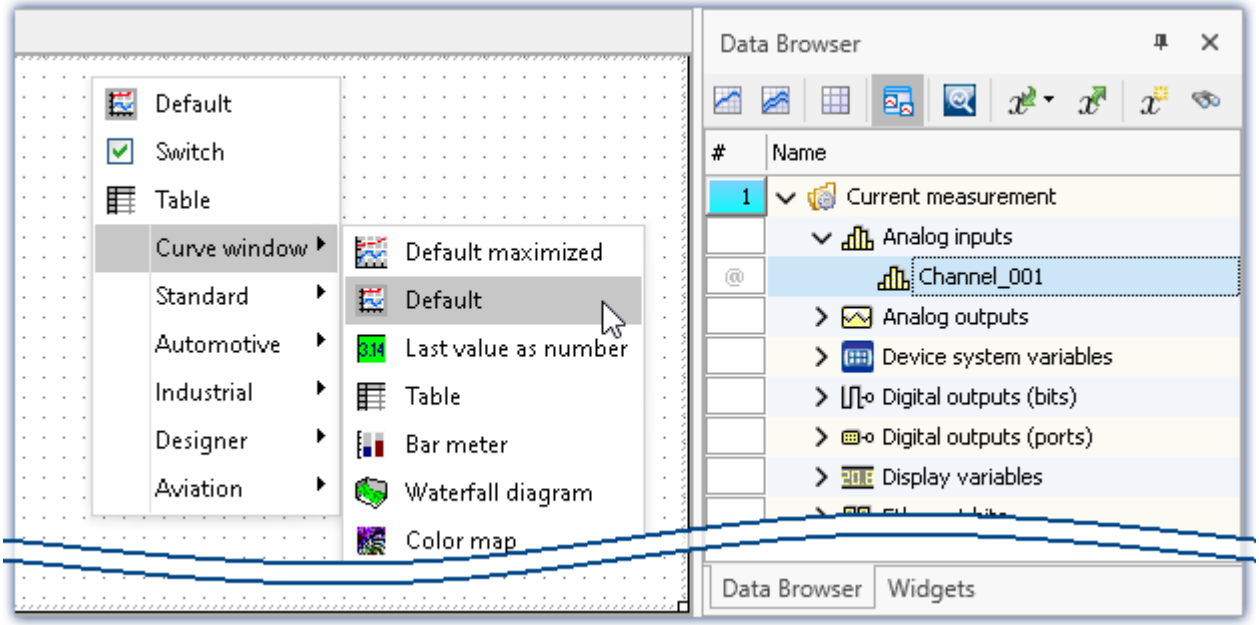
- Open the tool window "*Data Browser*" and select the **analog input**: "*Channel_001*".
- Use Drag&Drop to drag this channel to the page, either by grabbing its icon or its name (but do not grab the @-symbol; this will be explained in the next section).

A menu appears offering Widgets appropriate for the analog channel's data type.

- Select one of these Widgets. In the example, a "*standard curve window*" is selected.

Reference Curve window documentation

This Widget is the curve window, which is also used in other imc software packages. More information is provided in the section. Chapter:
"*Panel*" > "*Curve window* 1265".



Dragging an analog channel to the Panel page

There are a variety of Widgets available. The example (in the picture above) may differ from your configuration.

Following selection, the selected Widget appears on the Panel page and is linked with the analog input: "*Channel_001*".

As long as the Widget is selected, a cross appears in the center (⊕). This can be used to move the Widget or open a context menu there (by right-clicking).

- Deactivate the Design mode ( ).

Save experiment

Save the experiment:

- Click on the button: "*Save as*" ( ).
- Save the experiment under the name "*First Steps*".

Ribbon	View
Home > Save as ( )	all
Project > Save as ( )	Complete

Starting measurement

- To **transfer** the current **settings to the device**, click on the button "*Prepare* 241" ( ).

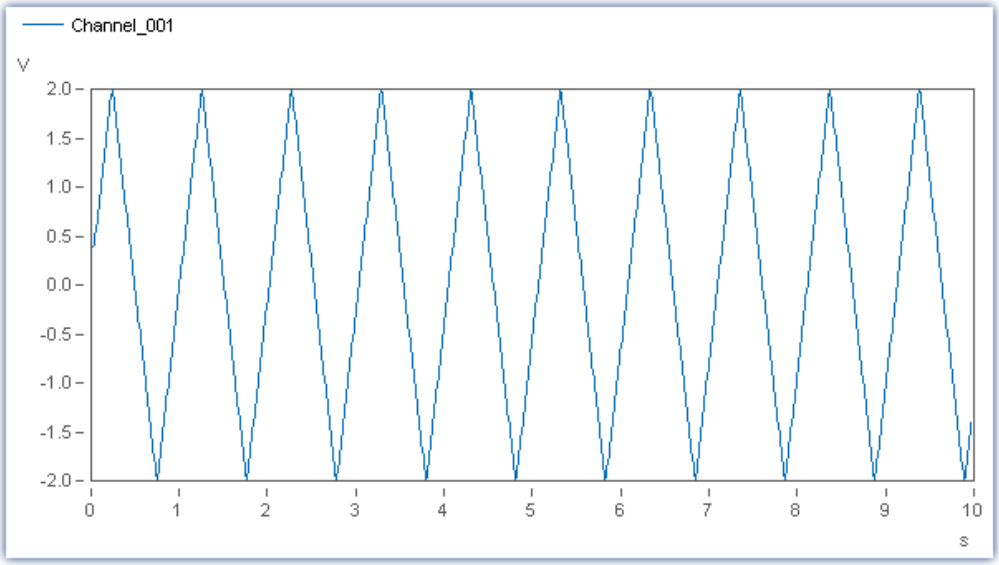
Ribbon	View
Home > Prepare ( )	Complete
Setup-Control > Prepare ( )	Complete

Note Preparation of the device

The menu item "*Prepare*" is not present in the "*Standard*" view. That is because it does not need to be performed manually. Instead, preparation of the devices is performed automatically unless it had already been initiated manually.

- Start the measurement 241 ( ).

Ribbon	View
Home > Start ( )	all
Setup-Control > Start ( )	Complete



Curve window

After ten seconds, the measurement ends automatically. You can also start measurement multiple times and stop it prematurely using the associated button ( ).

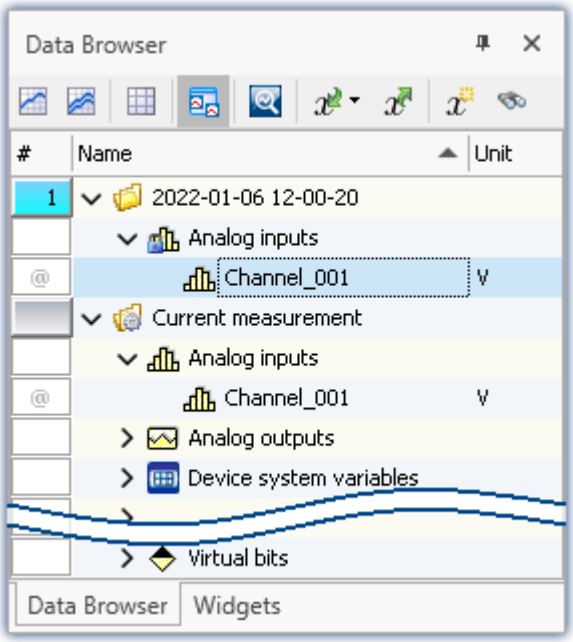
Ribbon	View
Home > Stop ( )	all
Setup-Control > Stop ( )	Complete

Using the curve window, you can analyze the data in a wide variety of ways.

Saved measurement data

At the end of measurement, the **measurements** are each saved in a separate subfolder in the **Experiment folder**. The folder is named according to the respective measurement's start-time stamp.

The saved measurement appears in the Data Browser:



Saved measurement data (example)

- Perform at least **two more measurements** (so that there are at least three measurements).

Note — Permanent data storage

By default, **data saving is activated for all channels**. Activation of data storage ensures that no measured data are lost. Subsequently, it is always possible to delete the measured data once they are no longer required.

It is possible to deactivate saving of the data. The next examples illustrate additional ways to save measured data.

Reference

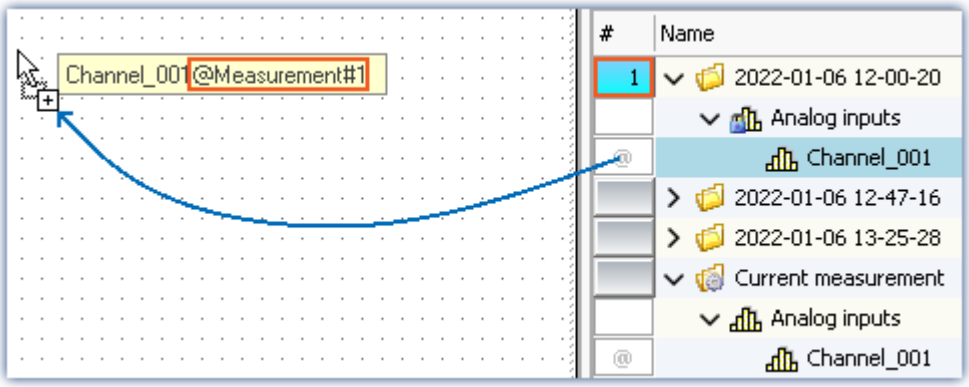
More information on **activating the data storage** is provided in the section:
"*Setup pages - Configuring device*" > "*Data transfer* 370".

6.1.3 Panel - Messungen nacheinander betrachten

The saved measurements also can be displayed on the Panel page. You can link a variable either by its **fixed name** or its **symbolic name** with a Widget.

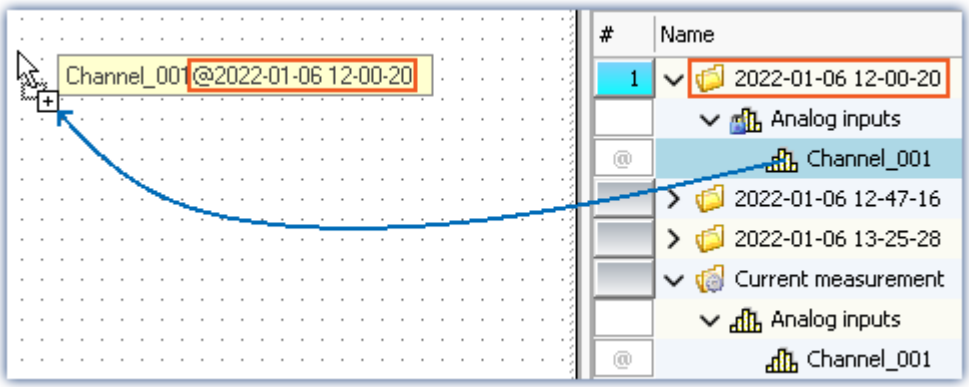
Fixed name	**<VariableName>@<Measurement Name>** e.g. **Channel_001** In this way, the Widget always indicates the current measurement. or **Channel_001@2022-01-06 12-00-20** in this way, the Widget is always linked with this concrete measurement
Symbolic name	**<VariableName>@Measurement#<Measurement number>** e.g. **Channel_001@Measurement#1** Contains the name of the variable followed by a measurement number. The measurement number can be assigned variably with the Data Browser. This enables you to **compare measurements**. After measurement, you can select the saved measurement (it is then assigned the the number "1"). Subsequently, **all Widgets** show the variables belonging to the saved measurement.

If you have opened a measurement in the Data Browser, the **"@" Symbol** (@) appears in the **Numbering column**. To link a variable with a Widget **via a measurement number** (symbolic name), **use Drag&Drop** to move the variable **from the "@"-symbol** to the page.



Variable linkage with symbolic name.
Example: "Channel_001@Measurement#1"

To link a variable with a Widget by its **fixed name**, move the variable to the page **from the name column using Drag&Drop**.



Variable linkage with fixed name.
Example: "Channel_001@2022-01-06 12-00-20"

Reference Linkage of variables

More information on linkage of variables is provided in the section:
"*Panel*" > "*Variable linkage* 1550".

- Create one new curve window for each of the **two ways described**

One of the curve windows is linked with a defined measurement. The other displays the channel belonging to the measurement selected.

- Select a **different measurement**: the measurement is automatically loaded and displayed.

Note Always display all data

If you intend for the complete measurement to always be displayed whenever changing measurements, the following settings for the curve window are optimal:

- **Scroll mode**: "*No*"
- **Timeline(x-axis):** "*automatic*"

This causes a similar situation as the "*Stretch*" mode. The difference is that then the complete data set is always shown (the left side also adjusts accordingly.) This mode is suitable for reports when measured data are to be loaded in succession and displayed completely in a curve window.

Attention: In cases of very large data sets, this mode can cause significant performance problems if all data points are displayed.

Save the experiment

You have thus concluded your first experiment. Next, save the experiment again:

- Click on the button: "*Save*" ( ).

6.2 Circular buffer memory for the display

Assignment:

Compare the data which is available during the measurement with the saved measured data.

Learning goals:

- Data flow: from the device to the PC, for display and storage purposes
- the effects of circular buffer memory on the measured data

Prerequisite:

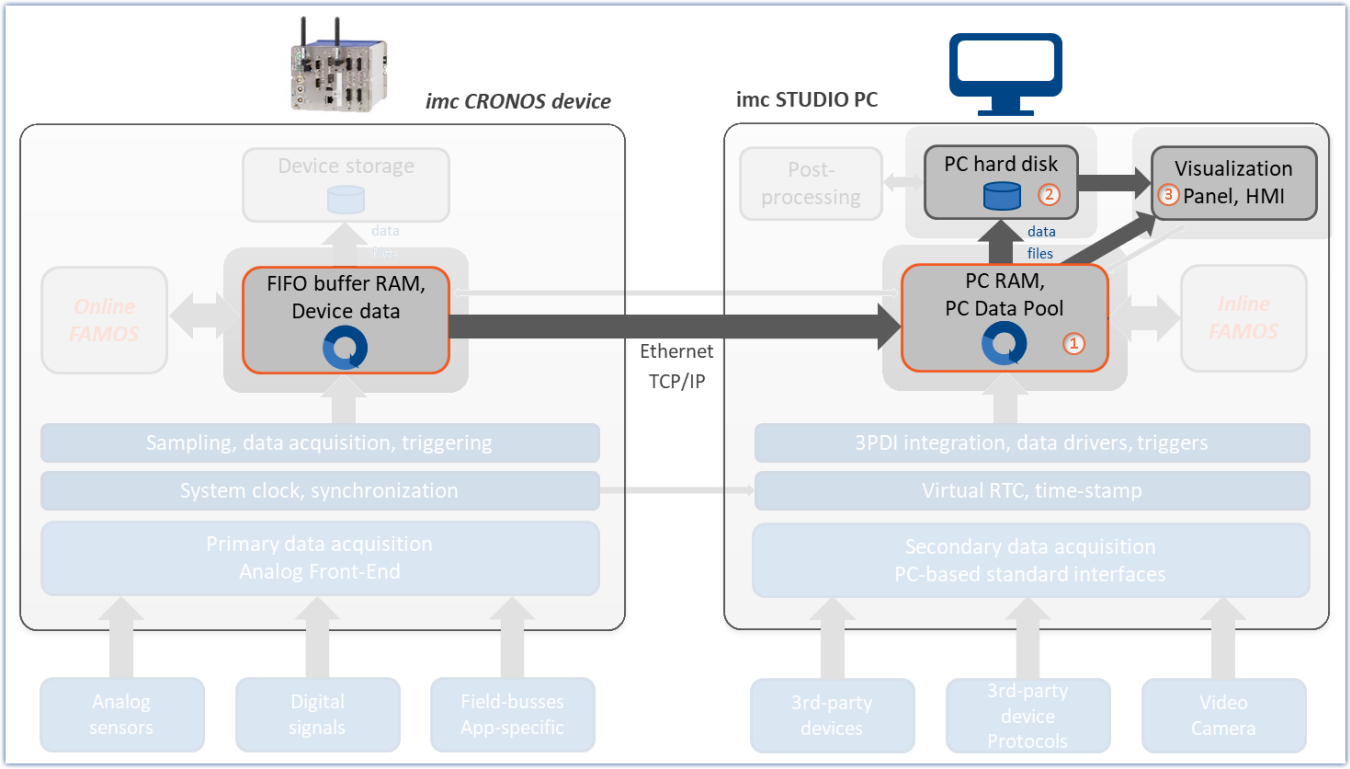
- imc measurement device with an analog channel

Procedure:

Begin by start imc STUDIO.

6.2.1 Background-info: The data stream

Once the measurement starts, the measurement device generates a data stream. The goal in every measurement is to process this data stream and to observe and or save it.



Simplified schematic of the data stream illustrated by the imc CRONOS device

The data stream is first written to FIFO RAM buffer memory 427 in the device.

Once the data have been processed in the device, they can be **transferred to the control PC** ①. The data can be **displayed** on Panel pages ③. Additionally, the measured data which have been transferred can be **saved** on the PC ②.

- By default, the **data for the display** are saved on the PC in a **circular buffer** ①. By using the circular buffer memory, it is possible to temporarily store the data in the higher speed ("rapid access") working memory (RAM). This improves performance and enables continuous data transfer no matter how long the measurement runs.
- For **data storage** purposes, circular buffer memory is not used.

Whether or not data are transferred to the PC and whether they can be saved can be set for each channel separately. By default, data saving on the PC is activated.

6.2.2 Setup - Settings

First, the measurement device is configured.

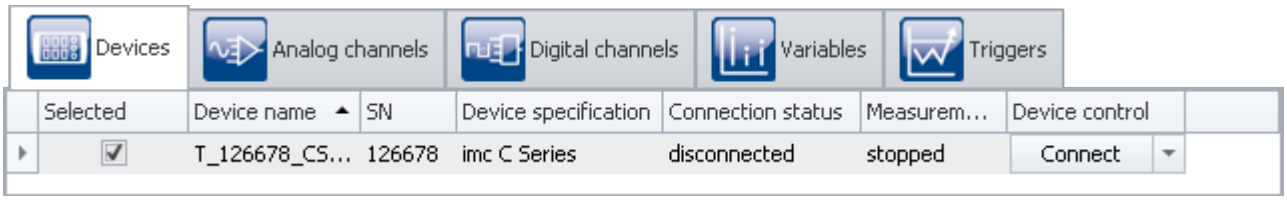
- Open the **device configuration page**: "*Setup*".
- Go to the **tab**: "*Devices*" .

Select your device:

- If the device desired **does not appear in the list**, perform a device search 241 ( ). Subsequently, the list will display all devices found.

Ribbon	View
Home > Search for devices ( )	all
Setup-Control > Search for devices ( )	Complete

- Click on the Checkbox-symbol ( ) in order **to select the device**.



Selecting a device

The device is now known and selected, and is available each time imc STUDIO is started.

Configuring the channels

- Go to the **tab**: "*Analog channels*" .

Under **Channel type**: "*Analog inputs*", select "*Channel_001*". By default, this is set to "*active*". Make settings for "*Channel_001*" in accordance with its input signal. Also ensure that the following settings are applied:

Parameter	Value	Dialog (View: *Standard*)	Dialog (View: *Complete*)
Duration	30 s	Sampling & Filtering	Sampling & Preprocessing
Circular buffer time	10 s	Data transfer	Data transfer
Save data (PC)	Checkbox activated (true)	Data transfer (below "Transfer to PC")	Data transfer (below "Transfer to PC")

- To **apply the changes**, click on the button "*Process configuration* 241" ( ).

By this means, the device settings are available to other components.

6.2.3 Panel - Settings / Start measurement

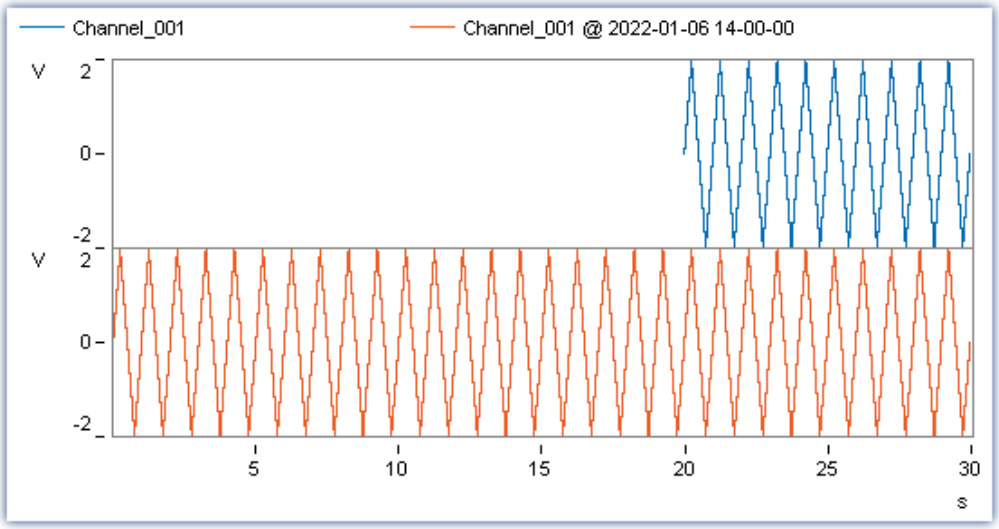
Now you can create the Panel page.

- Open the page: "*Panel*".
- Activate the Design mode ( )

Create a curve window showing the channel: "*Channel_001*" and start the measurement.

Observe what happens after the first 10 s → The first **data are deleted**. There are always only a maximum of 10 s worth of measured data.

After 30 s, measurement stops automatically and a completed measurement appears in the Data Browser. Also observe "*Channel_001*", which has been saved. It contains all 30 s worth of data.



The channel under "Current measurement" contains only the last 10 s of the entire measurement. The saved channel contains all 30 s.

Conclusions

For the purpose of displaying the channel data from indefinitely long measurement runs, use the circular buffer memory. This circular buffer displays the data for a specified duration. Note that many functions directly access the circular buffer; for instance, the imc FAMOS-command (during a measurement), or subsequent data storage in response to the button "*Save current Data* 83".

Give careful prior thought to how much circular buffer memory your application will need. If you select a volume which is too large, the performance can be adversely affected, especially when trying to display too much data in a curve window.

If the demands on the PC ever become too great for the data to be processed punctually, data overflow can occur.

Reference Data overflow

Additional useful notes on the topic of data overflow are provided in the following sections:

- "*Avoidance of data overflow* 755"
- "*Behavior upon data overflow* 755"

6.3 Subsequent saving of measured data

Assignment:

Perform a measurement without simultaneous data saving, and save the measured data subsequently on the hard drive.

Learning goals:

- Use of subsequent data saving

Elements used:

- Menu action: "*Save current Data*"

Prerequisite:

- imc measurement device with an analog channel

Procedure:

Load the experiment "*First Steps*" from the tutorial "*Simple measurement - First Steps* 72" or stop the current measurement ( ) if it is still running.

To load the experiment:

- click on the button: "*Open*" ( )
- select the experiment "*First Steps*" and confirm by clicking in the dialog on "*Open*"

Save the experiment under a new name: "*First Steps without data saving*"

- To do this, click on the button: "*Save as*" ( ).

6.3.1 Setup - Settings

Deactivate storage of the measured data for the active measurement channel.

- Open the **device configuration page**: "*Setup*".
- Go to the **tab**: "*Analog channels*" .

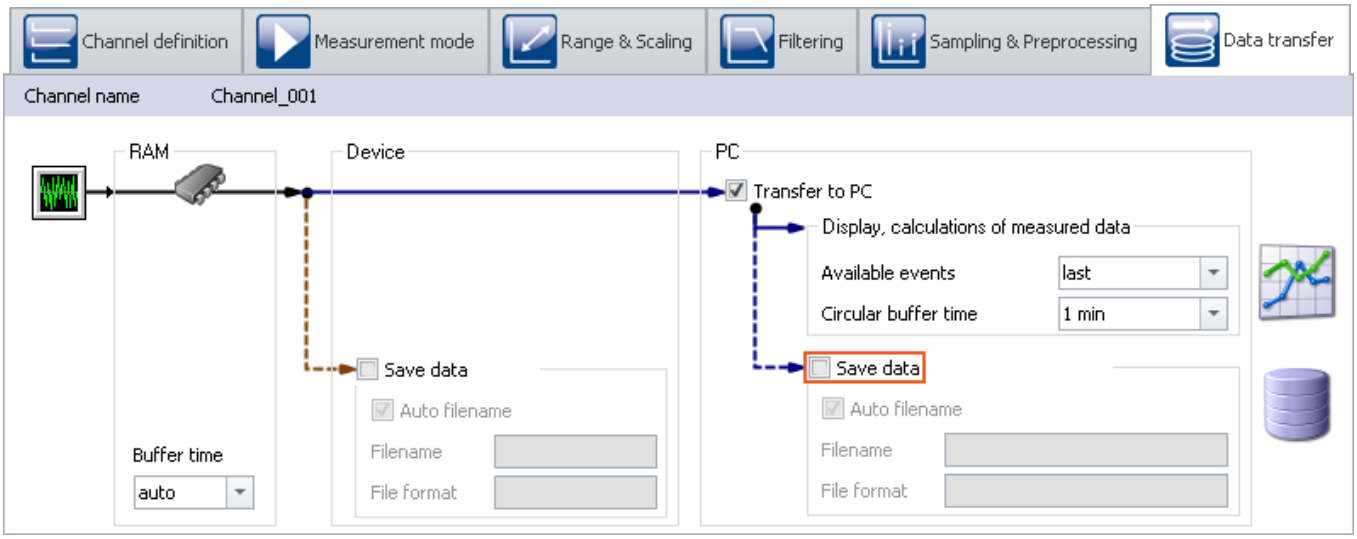
Under **Channel type**: "*Analog inputs*", select "*Channel_001*".

Deactivate storage of the measured data on the PC for the analog input: "*Channel_001*".

Parameter	Value	Dialog (View: *Standard*)	Dialog (View: *Complete*)
Save data (PC)	Checkbox deactivated (false)	Data transfer	Data transfer

Note

The option "*Save data*" appears on the "*Data transfer*" page in **two places**. Once where it activates **storage** of the **measured data on the device**, and once for **storage on the PC**.



Deactivating the storage of measured data using the dialogs

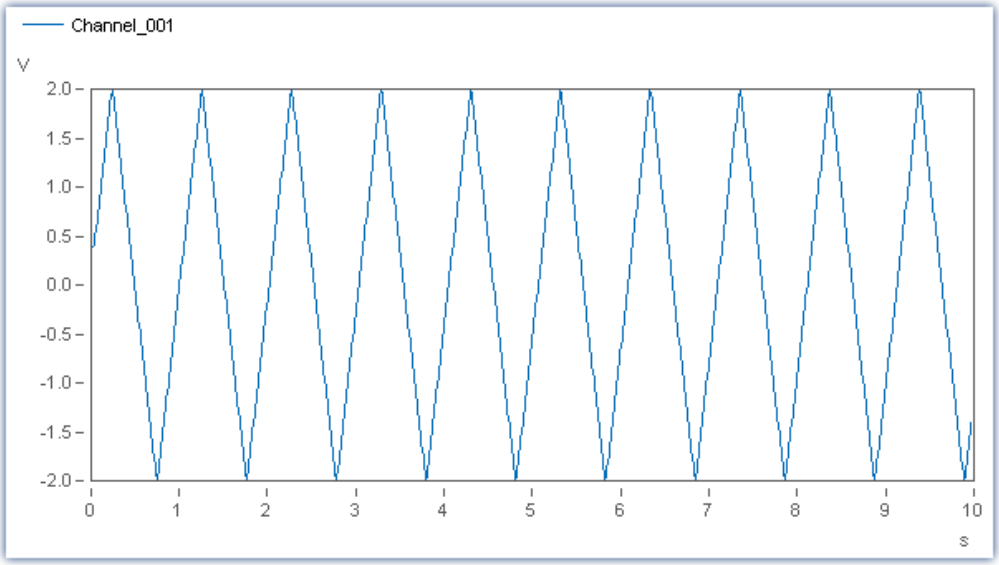
Note

By default, **data saving is activated for all channels**.

6.3.2 Panel - Starting measurement

Open the Panel page and perform a measurement.

- To **transfer** the current **settings to the device**, click on the button "*Prepare* 241" ( ).
- Start the measurement 241 ( ).



Curve window

After ten seconds, the measurement ends automatically.

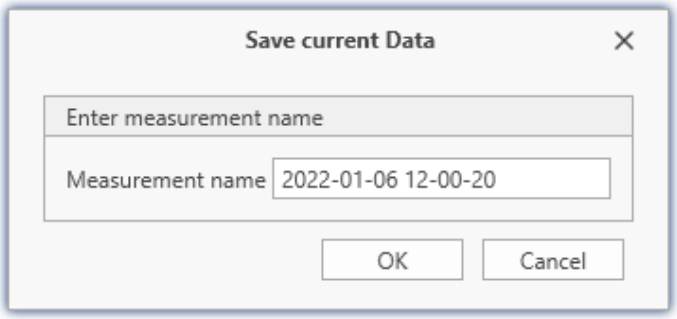
6.3.3 Subsequent saving of measured data

In this example, the data were not saved permanently. No data are on the hard drive, but these can be saved retroactively.

- Click on the button: "*Save current Data*".

Ribbon	View
Project > Save current Data ( )	Complete
Home > Save current Data ( )	Standard

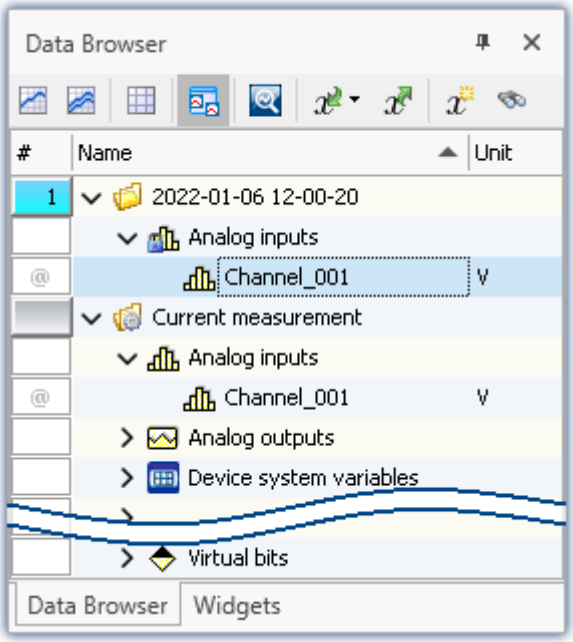
- If the action has not been configured otherwise, a dialog appears which prompts you to enter a folder name. The suggested default value is the current time stamp. Close the dialog by selecting "OK".



Dialog: Save current Data

Subsequently, all channels (including analog channels, virtual channels, ...) belonging to the experiment are saved to the hard drive. Within the experiment folder, a folder having the specified name is created, in which the data are then stored.

In the Data Browser, a folder corresponding to the folder name appears, which contains the saved measurement data. You can load and view it using the Panel.



Saved measurement data (example)

Note

- The measurement folder created represents a complete measurement. It is treated in the same way as any folder created by means of normal measurement data saving.
- **No single values can be saved in this way.** These include, for instance, Display variables.
- The measured data saved by means of "*Save current Data*" are always stored in the experiment folder. If you wish to save them to a freely selectable folder, use the button: "*Export current Data*". Be aware that measured data saved in this way are not recorded in the Data Browser and can thus not appear there.

Warning Constraints

Be aware that with this function, only such variables are saved, which are found in the Data Browser under "*Current measurement*". For these variables, a **circular buffer memory** is usually activated, so that not all results since the measurement start are available.
For device-variables, therefore, the Data Transfer Settings 370 are applicable. The storage type: "*Display, calculation of measured data*" is used.
See the tutorial: "*Circular buffer memory for the display* 78"

Reference

More information on the function: "*Save current Data*" is presented in the section:
"*Storage options and directory structure*" > "*Targeted Data Saving, or Saving Subsequent to Measurement* 439".

6.4 Simple triggering

Assignment:

Display a measurement on an analog channel. Use a switch on the Panel page in order to interactively start and stop measurement on this channel.

Learning goals:

- Creating an experiment
- Simple device operation
- Use of a trigger

Elements used:

- Trigger

Prerequisite:

- imc measurement device with an analog channel

Procedure:

Begin by starting imc STUDIO.

6.4.1 Setup - Settings

First, the measurement device is configured and an appropriate name is assigned to the variables.

- Open the **device configuration page**: "*Setup*".
- Go to the **tab**: "*Devices*" .

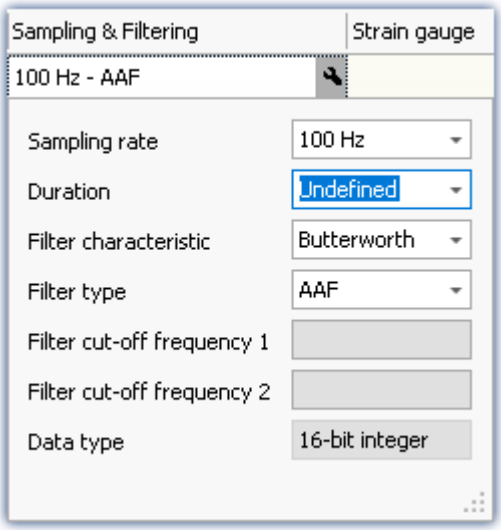
Select your device

Configuring the channels

- Go to the **tab**: "*Analog channels*" .

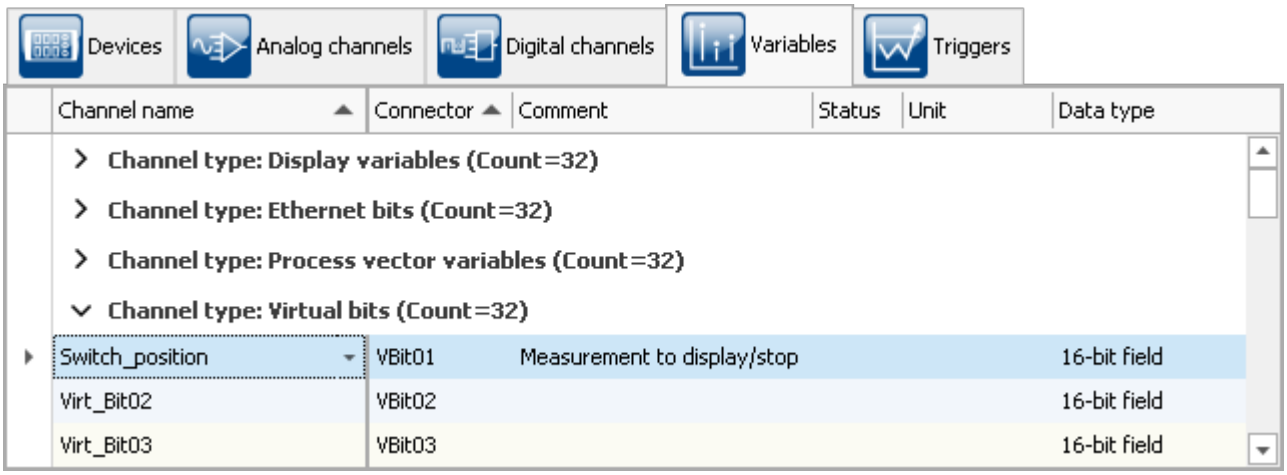
Under **Channel type**: "*Analog inputs*", select "*Channel_001*". By default, this is set to "*active*".

- In order to set a measurement of unlimited duration, in the **column** "*Sampling & Filtering*", on the **row** "*Duration*" set the **value** "*Undefined*" by means of the pop-down symbol ( ).



Unlimited measurement duration

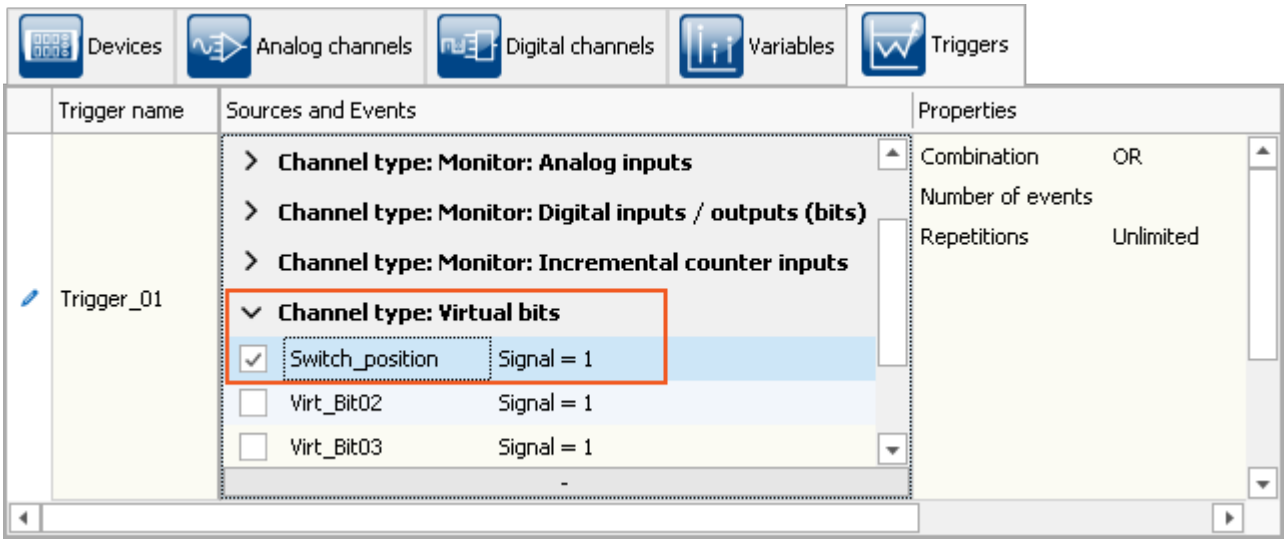
- Go to the **tab**: "*Variables*" .
- Under **Channel type**: "*Virtual bits*", click on "*Virt_Bit01*" and **rename** it to "*Switch_position*". If desired, you can enter a comment ("*Measurement to display/stop*").



Channel type "Virtual bits"

Defining triggers

- Go to the **tab**: "*Triggers*" .
- To define the **virtual bit** "*Switch_position*" as a source, click in the **column** "*Sources and Events*" on the plus-sign symbol (+).
- Select "*Switch_position*".



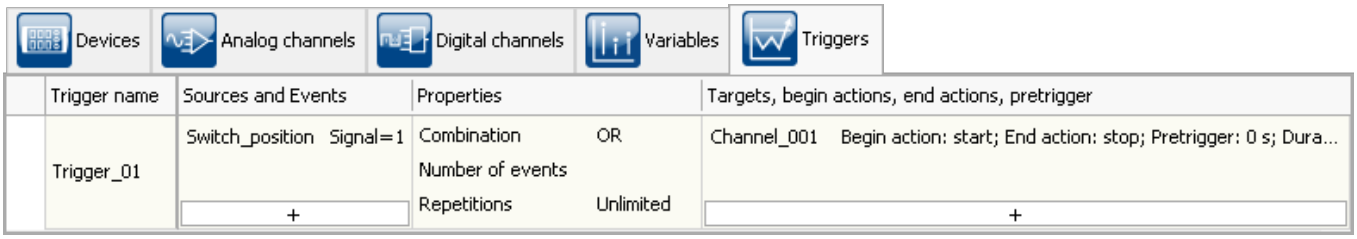
Trigger source

In order to be able to **release the trigger multiple times**, in the column "*Properties*" the "*Repetitions*" is set to "*Unlimited*".

- As the **target**, specify "*Channel_001*"

The measurement is to start when `Switch_position = 1` and stop when `Switch_position = 0`. To accomplish this, set the "*Begin action*" and "*End action*" as shown in the picture below.

Parameter	Value
Trigger begin action	start
Trigger end action	stop



Settings for the trigger

- To **transfer** the current **settings to the device**, click on the button "*Prepare* 241" ( ).

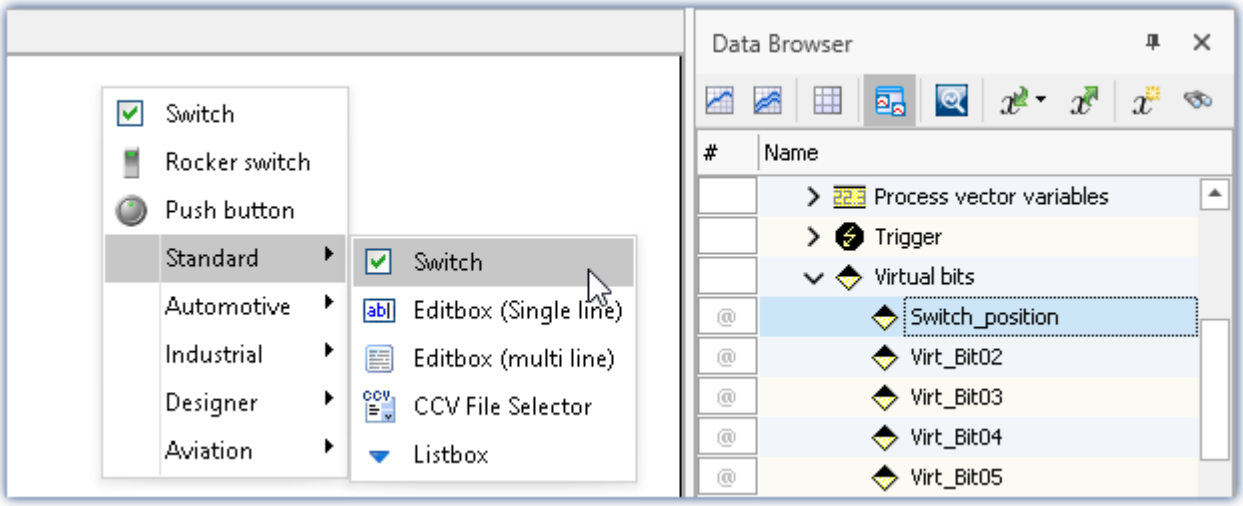
6.4.2 Panel - Settings / Start measurement

Now you can create the Panel page.

- Open the page: "*Panel*".
- Activate the Design mode ( )

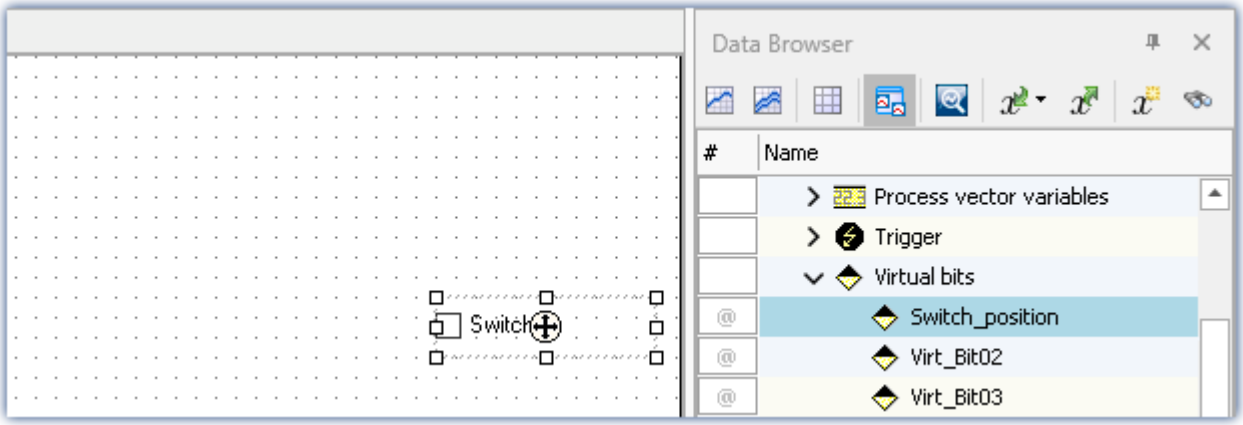
Switch

- From the Data Browser, you can now drag the "*Switch_position*" (found under **Channel type**: "*Virtual bits*") to the Panel page.
- After releasing the mouse button (Drag&Drop technique), a context menu appears. From this context menu, select "*Switch*".



Dragging a virtual bit to the Panel page

A wide variety of Widgets are now available. Please note that the image below is only an example and may not correspond to the Widgets available to you.



Now the switch on the Panel-page is **linked with** the **virtual bit** in the device named "*Switch_position*". Using the switch, you can interactively switch the virtual bit on and off.

By means of the Widget-property: "*Text*", you can label the switch; e.g. "*Trigger start*". To do this, open the context menu over the crosshairs ( ) within the selected Widget and select the entry: "*Properties*".

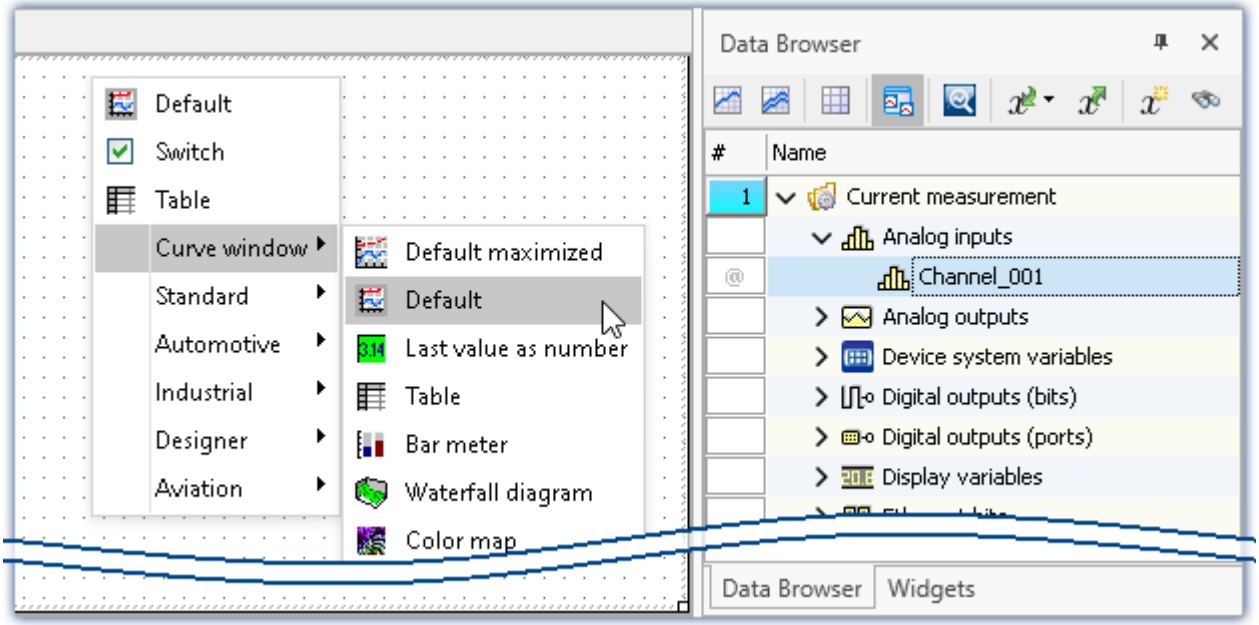
Note Additional Widgets

Depending on the product configuration, additional Widgets are available. Instead of a simple switch, you could select a "*Rocker switch*", for example, under the heading "*Automotive*".

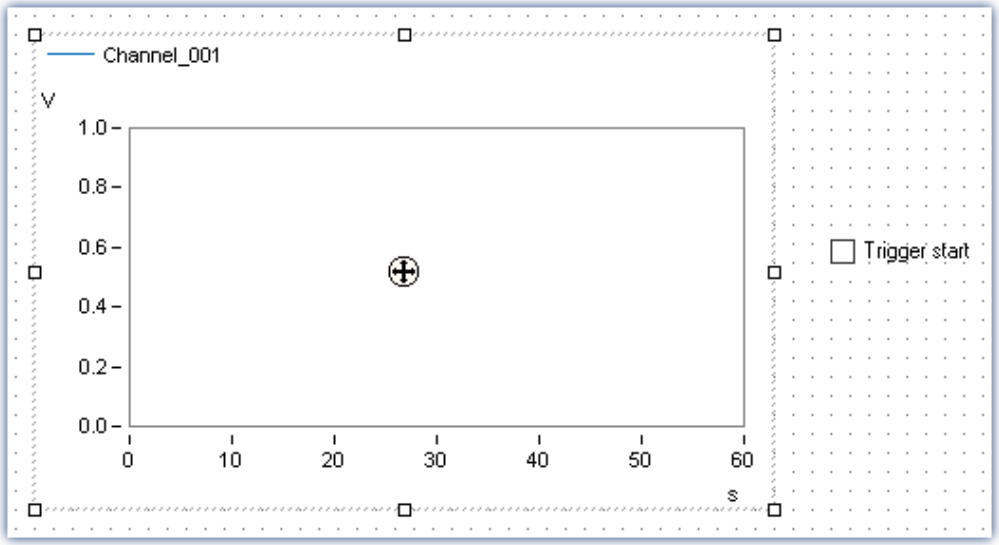
Curve window

After positioning the switch on the Panel page, next create the curve window:

- Drag the **analog channel** "*Channel_001*" from the Data Browser to the Panel page.
- Upon releasing the mouse button, you see a context menu where you will select the entry "*Standard*". This creates a standard curve window.



Dragging an analog channel to the Panel page



Example: Completed Panel page with two Widgets

As long as the Widget is selected, a cross appears in the center (⊕). This can be used to move the Widget or open a context menu there (by right-clicking).

- Deactivate the Design mode ( ).

Starting measurement, clicking buttons, stopping

- Start the measurement 241 ( ).

"*Channel_001*" is triggered by the virtual bit "*Switch_position*". This means that when you click on the switch, data capture on "*Channel_001*" starts/stops.

- Click on the switch multiple times.

Save experiment

With this, you have concluded the experiment. Save the experiment:

- Stop the measurement ( ).
- Click on the button: "*Save as*" ( ).
- Save the experiment under the name "*Simple Triggering*".

Reference

For an additional display of text on the Panel page to indicate the status of the measurement, please read about the enhancement: "*Einfache Triggerung - Erweiterung* 93".

6.5 Simple triggering - Enhanced: Trigger status

Assignment:

A text box on the Panel page displays the **trigger's state**. The states are ("*running*", "*stopped*").

Learning goals:

- Using trigger-variables

Elements used:

- Listboxes, Zones, Trigger variables

Prerequisite:

- This is an enhancement of the experiment: "*Simple Triggering* 87", which must already exist.
- The same conditions apply as for the experiment: "*Simple Triggering*"

Procedure:

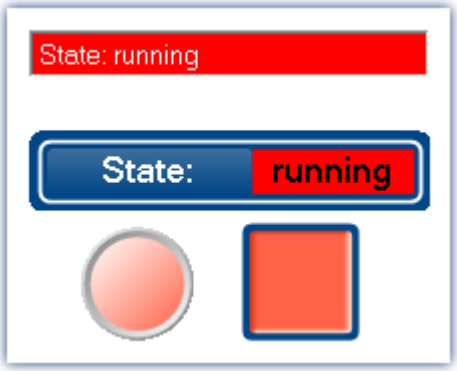
Begin by starting imc STUDIO.

- Load the experiment "*Simple Triggering*" or stop the current measurement ( ) if it is still running.
- Save the experiment under a new name: "*Simple Triggering - Enhanced*"
  - To do this, click on the button: "*Save as*" ( ).

6.5.1 Panel - Settings / Start measurement

Note Widgets

The status of the trigger can be displayed using a variety of Widgets. In this example, a simple list is used which is able to display texts. Other Widgets from the advanced groups offer more display options, but these are not available in all Editions.



Top Widget from the group: "*Standard*".

Other Widgets, belonging to the groups "*Automotive*", "*Industrial*" and "*Designer*"

- Open the page: "*Panel*".
- Activate the Design mode ( )
- Drag the Widget: "*Listbox*" to the Panel page:
  - Tool Window: "*Widgets*" - Group: "*Standard*" > "*Listbox*".

The "*Listbox*" is configured in the **Widget-properties**. To do this, open the context menu over the crosshairs ( ) within the selected Widget and select the item: "*Properties*".

- Under the **property**: "*Variable*", select the trigger's status-variable: "*Trigger*" > "*Trigger_01*" > "*State*".

Note Trigger status indication

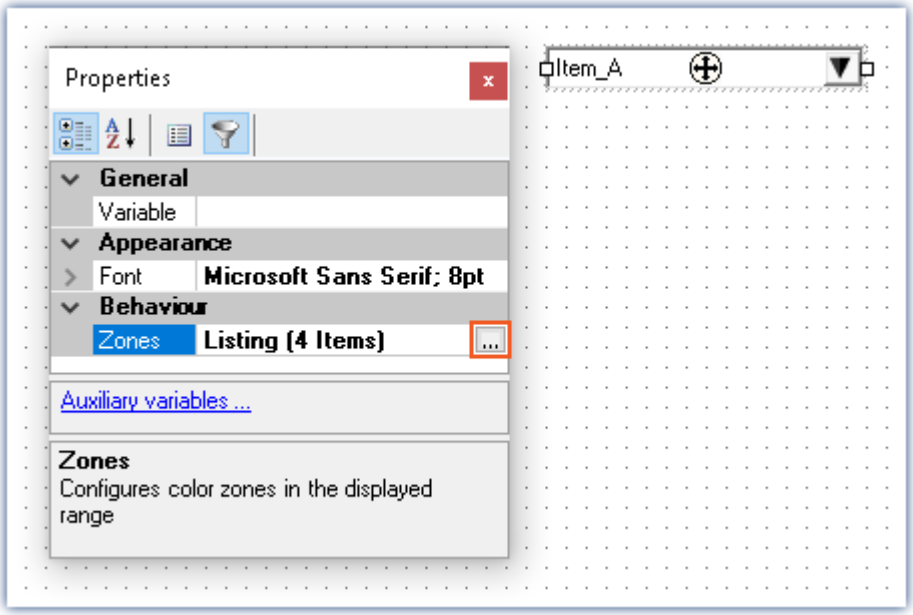
This provides an indirect linkage between the switch and the selection box. When the switch is actuated, the trigger releases. Once the trigger releases, the selection box indicates the operative status.

Conversely, the Widget also provides an indication when the trigger release failed. As a simple example, this can happen if the measurement had not previously been started.

Setting up and adapting states (zones)

The states displayed are to be adapted. For this purpose, some of the Widgets have the property: "*Zones*".

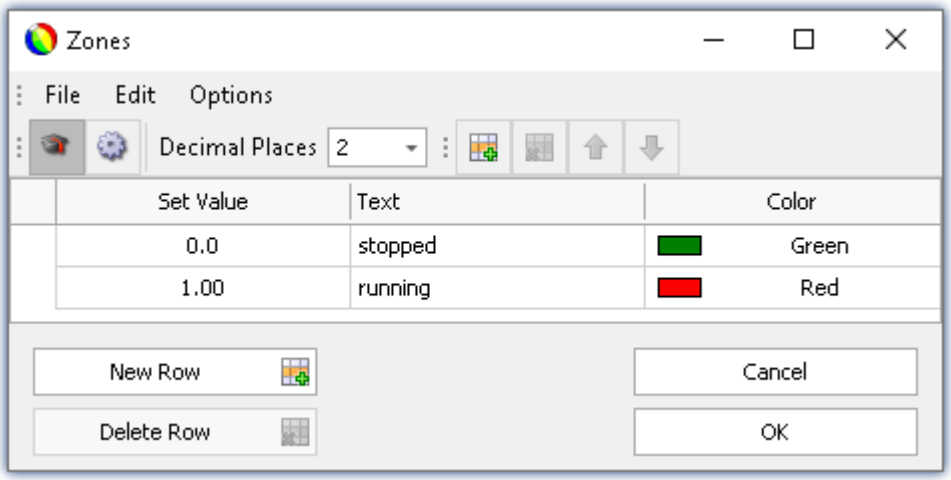
- Click on the symbol ... as shown in the following picture:



States (zones) defined for the selection box

In the dialog window, it is now possible to define the zones. The measurement has two possible states: "*stopped*" and "*running*".

- Only two zones are required; begin by deleting all but two zones
- Enter the zones as shown in the following picture:



Zones

- Deactivate the Design mode ( ).
- Start the measurement ( ).
- then click on the switch on the Panel page

The listbox changes (running, stopped) in response to the switch position:

Optional settings:
Button: "*Hide*"
Foreground color: "*White*"

Saving

With that, you have completed this example and can save it. Since the experiment has already been saved under a new name, it is sufficient to save the changes.

- Stop the measurement ( ).
- Click on the button: "*Save*" ( ).

Note Possible enhancements and additional exercises

- Indicate the count of previous trigger-releases. For this purpose, use the trigger-variable: "*Trigger*" > "*Trigger_01*" > "*EventNumber*".
- Indicate whether the measurement is running. For this purpose, use the device system variable: "*System*" > "*Experiment*" > "*Measurement status*".

6.6 Triggered measurement

Assignment:

The analog channel is to measure for 5 ms if the input signal exceeds the threshold 3 V. Apply a signal at the measurement device's analog input, with adjustable amplitude, e.g. by means of a function generator.

Learning goals:

- Use of a trigger
- Setting a channel according to the expected input signal

Elements used:

- Trigger event
- Monitor channel (this channel type cannot be used with imc EOS devices).

Prerequisite:

- imc measurement device with an analog channel
- A function generator or other signal source

Procedure:

Setting a function generator: Set up a sine function with a frequency of 1 kHz and initial amplitude of 2 V, for example.

Next, start imc STUDIO.

6.6.1 Setup - Settings

To begin, the measurement device is configured.

- Open the **device configuration page**: "*Setup*".
- Go to the **tab**: "*Devices*" .

Select your device

Configuring the channels

- Go to the **tab**: "*Analog channels*" .

Under **Channel type**: "*Analog inputs*", select "*Channel_001*". Make the following settings for the channel's parameter configuration:

Parameter	Value
Status	Active
Mode	Voltage
Coupling	DC
Input range	± 5 V
Sampling rate	100 kHz
Duration	5 ms

Under **Channel type**: "*Monitor: Analog inputs*", select "*Mon_Chan_001*". Make the following settings for the channel's parameter configuration:

For devices belonging to the firmware group A 187 (imc DEVICES) - e.g. CRONOS Familie, C-SERIE, SPARTAN, BUSDAQ

Parameter	Value
Status	Active
Sampling rate	10 kHz
Duration	Undefined

Note

The monitor channel's sampling rate can only be set to a value less than the analog input's sampling rate.

For devices belonging to the firmware group B 187 (imc DEVICEcore) - e.g. ARGUS*fit*, EOS

Parameter	Value
Status	Active
Function (Preprocessing)	Arithmetic mean
Points	10
Duration	Undefined

Defining triggers

- Go to the **tab**: "*Triggers*" .

In the column "*Sources and Events*" set "*Channel_001*" as the source, with the following event:

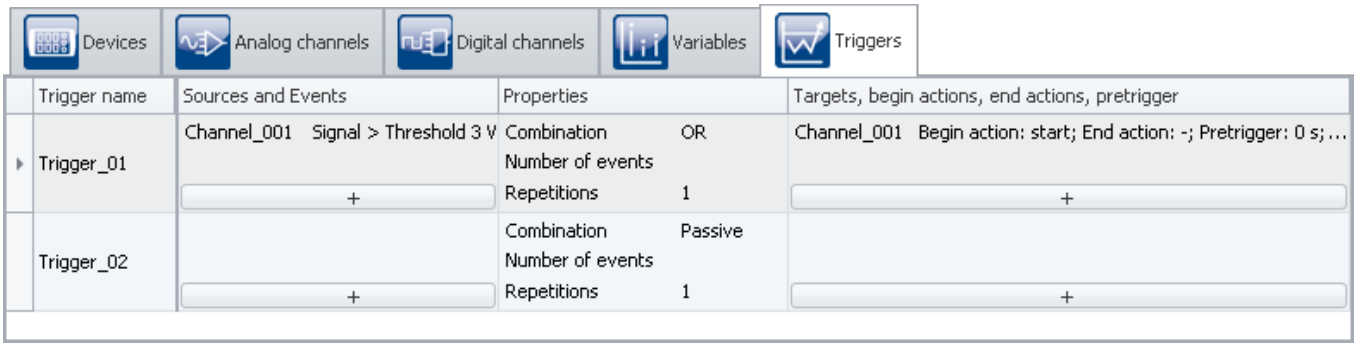
Parameter	Value
Event type	Threshold
Event	Signal > Threshold
Event threshold	3 V

In the column "*Properties*" set the following:

Parameter	Value
Combination	Or
Repetitions	1

In the column "*Targets, begin actions, end actions, pretrigger*" set "*Channel_001*" as well as a target, since it is to be used as a triggered event and measurement channel.

Parameter	Value
Begin action	start
End action	-
Pretrigger	0 s



Settings for the trigger

Prepare

- To **transfer** the current **settings to the device**, click on the button "*Prepare* 241" ( ).

6.6.2 Panel - Settings / Start measurement

Now you can create the Panel page.

- Open the page: "*Panel*".
- Activate the Design mode ( )

Create a curve window containing "*Channel_001*":

- From the Data Browser drag the variable: "*Analog inputs*" > "*Channel_001*" to the Panel page.
- In the Widget selection list, select: "*Curve window*" > "*Standard*".

Curve window properties

More Channels

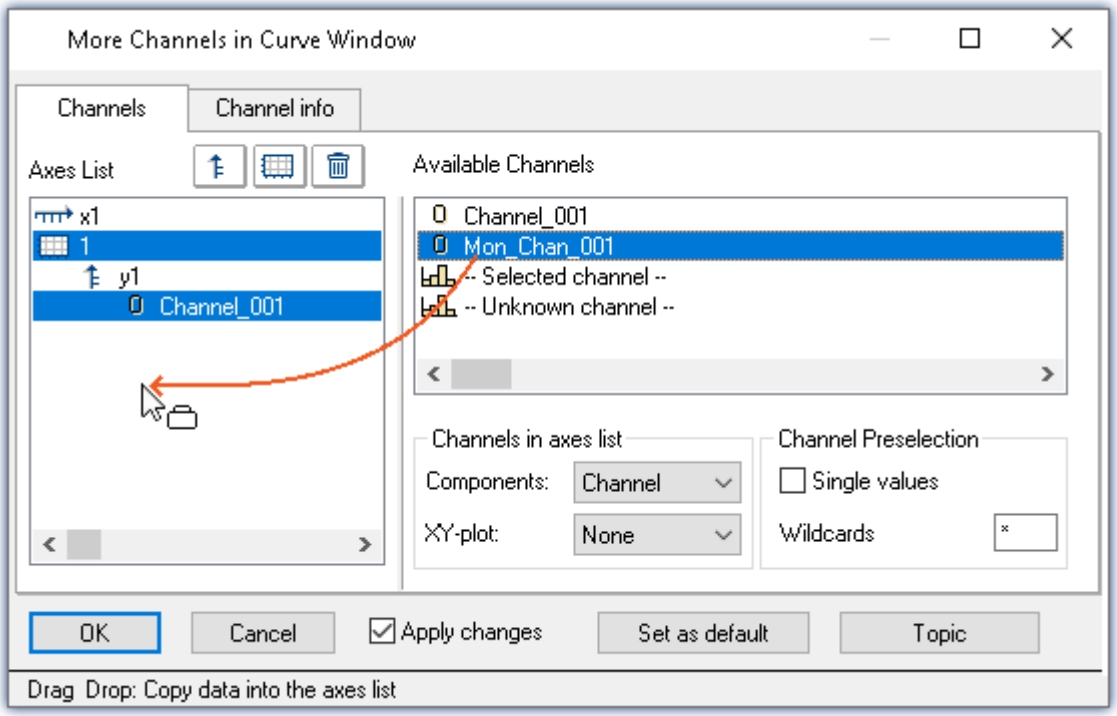
In a curve window it is possible to display more than one channel.

- Right-click the mouse over the curve window (not over the cross (⊕) in the center).
- In the context menu which then appears, select "*Configuration*" and subsequently the command "*More Channels*"

A dialog appears, in which you can **select additional channels** which are then displayed in the same curve window.

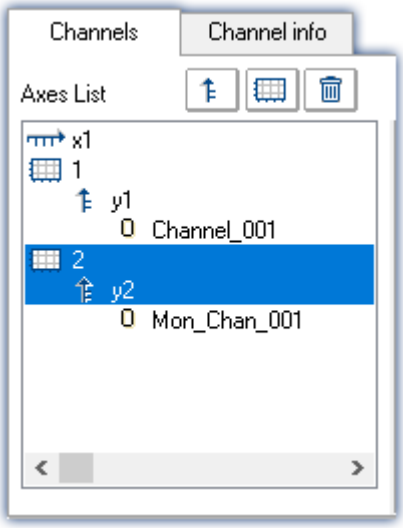
All data sets available for selection are displayed in the list at right: "*Available Channels*". The **data sets already displayed** in the curve window are indicated in the list at left: "*Axes List*".

- Drag "*Mon_Chan_001*" from "*Available Channels*" to the box "*Axes List*".



Channels for the curve window

- In order to display the two channels on different Y-axes, click on the coordinate-system symbol ( ).
- Next, select the channel "*Mon_Chan_001*". This now contains a single coordinate system.



To apply the settings, click on the button "*OK*".

Saving

With that, you have completed this example and can save it:

- Save ( ) the experiment under the name "*Triggered measurement*".

Starting measurement, observing results

- Deactivate the Design mode ( ).
- At the input: "*Channel_001*", for example, apply a sinusoidal signal with an amplitude of < 3 V an
- Start the measurement 241 ( ).
- after a while, change the amplitude to a value > 3 V.

As soon as the amplitude exceeds 3 V, "*Channel_001*" is recorded for 5 ms duration. "*Mon_Chan_001*" records the data already upon the measurement's start.

- Stop the measurement ( ).

Evaluating measured data

Observe the **complete measurement**. To do this, perform a rezoom in the curve window (context menu item: "*Edit*" > "*Rezoom*").

Notice that both channels start at the time point "0 s".

Adjusting the axes

The two channels have differing Trigger-times, but by default they are both displayed in the curve window stacked one above the other at x = 0. Thus in this example, each channel begins at x = 0, even if the triggered channel: "*Channel_001*" is started later.

In order that the two channels' **time coordinates to be synchronized**, the scale of the x-axis must be adjusted.

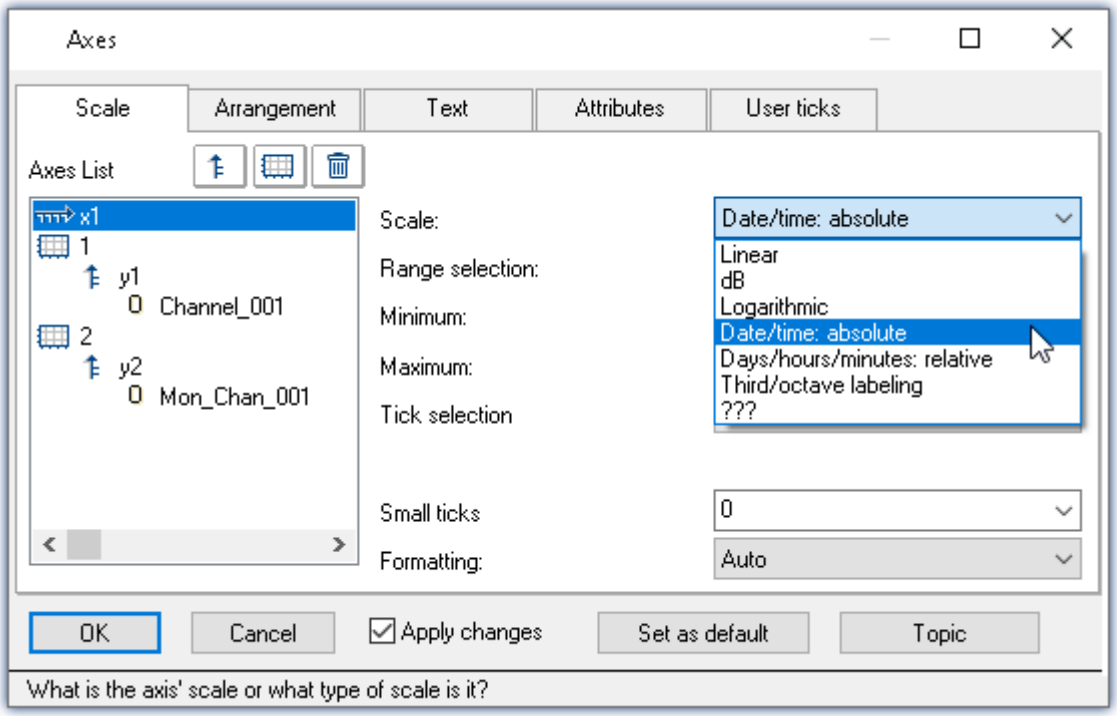
- Right-click he mouse over the curve window (not over the cross ( ) in the center).
- In the context menu which then appears, select "*Configuration*" and subsequently the command "*Axes*"

A dialog appears in which you can adapt the curve window's axes.

- In order to see the x-axis' settings, select the axis "*x1*" from the list at left: "Axes *List*".

Then make the following setting:

Parameter	Value
Scale	Date/time: absolute

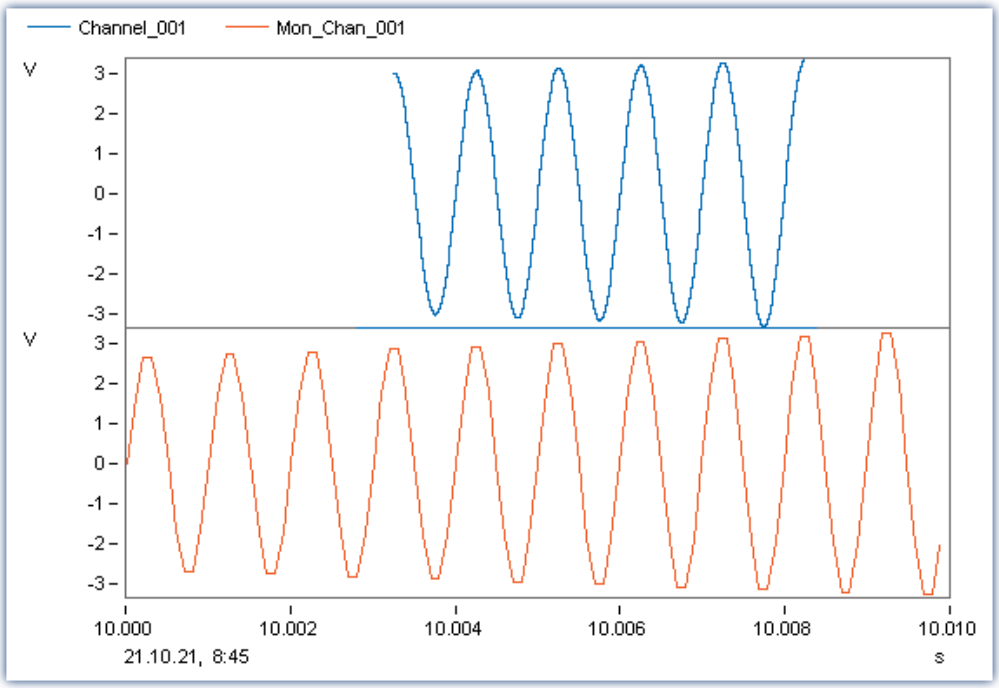


Setting the axis for the curve window

To apply the settings, click on the button "*OK*".

Now the two **channels are displayed in parallel with correct reference to time**. To investigate the data, you need to zoom in on the correct location in the curve window.

- Use the context menu item "*Edit*" > "*Zoom*"



Measurement of the channel's signal only starts once the trigger event occurs

Save the experiment

Save the experiment again:

- Click on the button: "*Save*" ( ).

6.7 Thermocouple measurement

Assignment:

A temperature is to be measured by thermocouple.

Learning goals:

- Advanced channel settings

Element used:

- Sensor & scaling
- Pre-processing of data

Prerequisite:

- imc measurement device with an amplifier for temperature measurement
- Type K Thermocouple
- An appropriate connector with which to connect the thermocouple with the imc measurement device

Procedure:

Connect the thermocouple to "*Channel_001*" of the imc measurement device.

Next, start imc STUDIO.

6.7.1 Setup - Settings

To begin, the measurement device is configured.

- Open the **device configuration page**: "*Setup*".
- Go to the **tab**: "*Devices*" .

Select your device

Configuring the channels

- Go to the **tab**: "*Analog channels*" .

Under **Channel type**: "*Analog inputs*", select "*Channel_001*". Make the following settings for the channel's parameter configuration:

Parameter	Value	Dialog (View: *Standard*)	Dialog (View: *Complete*)
Name	Thermocouple	Channel definition	Channel definition
Status	Aktive	Channel definition	Channel definition
Mode	Voltage	Measurement mode	Measurement mode
Coupling	DC	Measurement mode	Measurement mode
Correction	Type K	Measurement mode	Measurement mode
Input range	-270..500 °C	Measurement mode Range & Scaling	Measurement mode Range & Scaling
Filter type	AAF	Sampling & Filtering	Filtering
Sampling rate	1 kHz	Sampling & Filtering	Sampling & Preprocessing
Duration	Undefined	Sampling & Filtering	Sampling & Preprocessing
Function	Arith. Mean	-	Sampling & Preprocessing
Points	50	-	Sampling & Preprocessing

By means of the **correction** , the **input range** and the **unit** are automatically adapted.

The **arithmetic mean** of 50 sampled points produces a **resulting sampling rate** of 20 Hz, since groups of 50 points each are summarized by one point.

Prepare

- To **transfer** the current **settings to the device**, click on the button "*Prepare* 241" ( ).

6.7.2 Panel - Settings / Start measurement

Now you can create the Panel page.

- Open the page: "*Panel*".
- Activate the Design mode ( )

Create a curve window containing *Thermocouple*:

- From the tool window: **Data Browser**, drag the variable: *Analog inputs > Thermocouple* to the Panel page.
- In the Widget selection list: **Standard** (curve window).

Curve window properties

Adjusting axes

Adapt the axis settings:

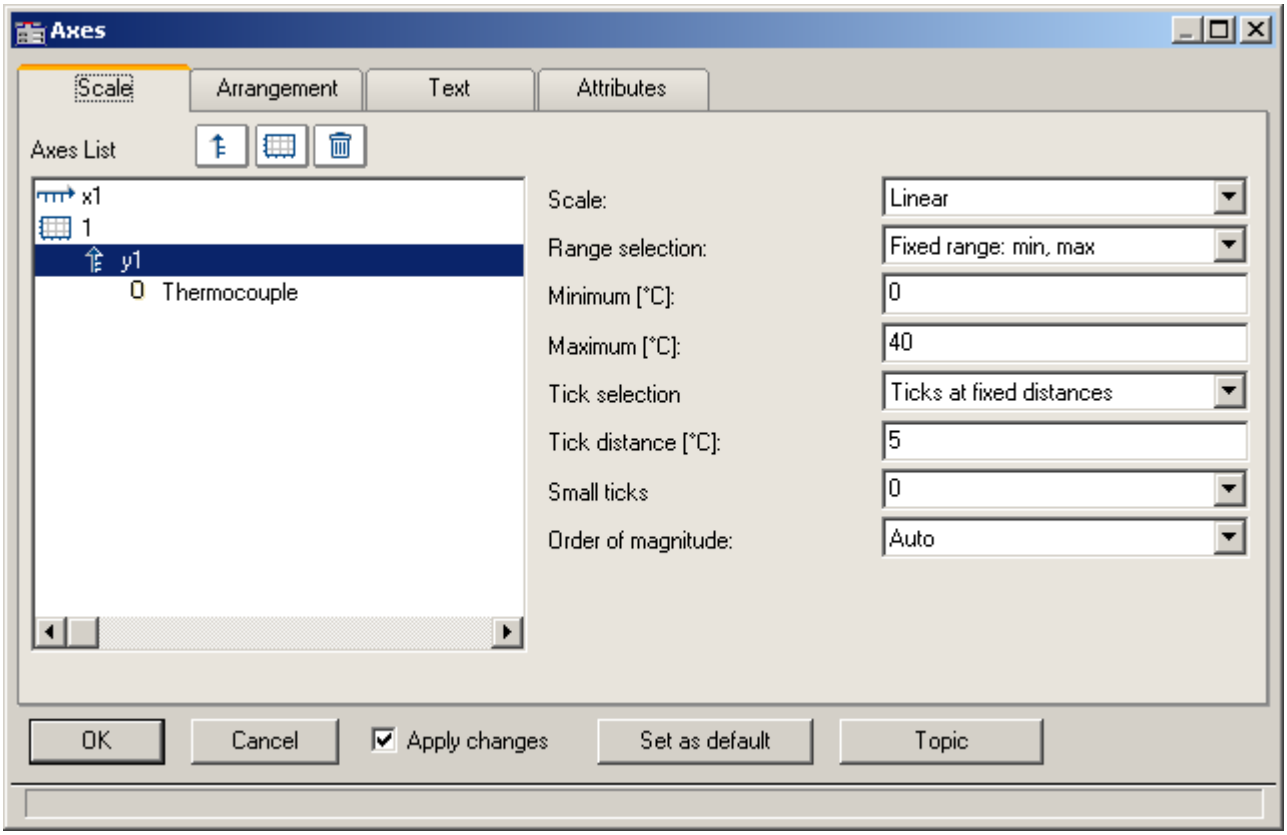
- Right-click the mouse over the curve window (not over the cross ( ) in the center).
- In the context menu then appears, select **Configuration** and subsequently the command **Axes...**

A dialog appears in which you can adapt the curve window's axes.

- In order to see the y-axis settings, select the axis *y1* from the list at left (**Axes List**).

Now make these settings:

Parameter	Value
Scale	Linear
Range selection	Fixed range: min, max
Minimum [°C]	0
Maximum [°C]	40
Tick selection	Ticks at fixed distances
Tick distance [°C]	5



Settings for the y-axes

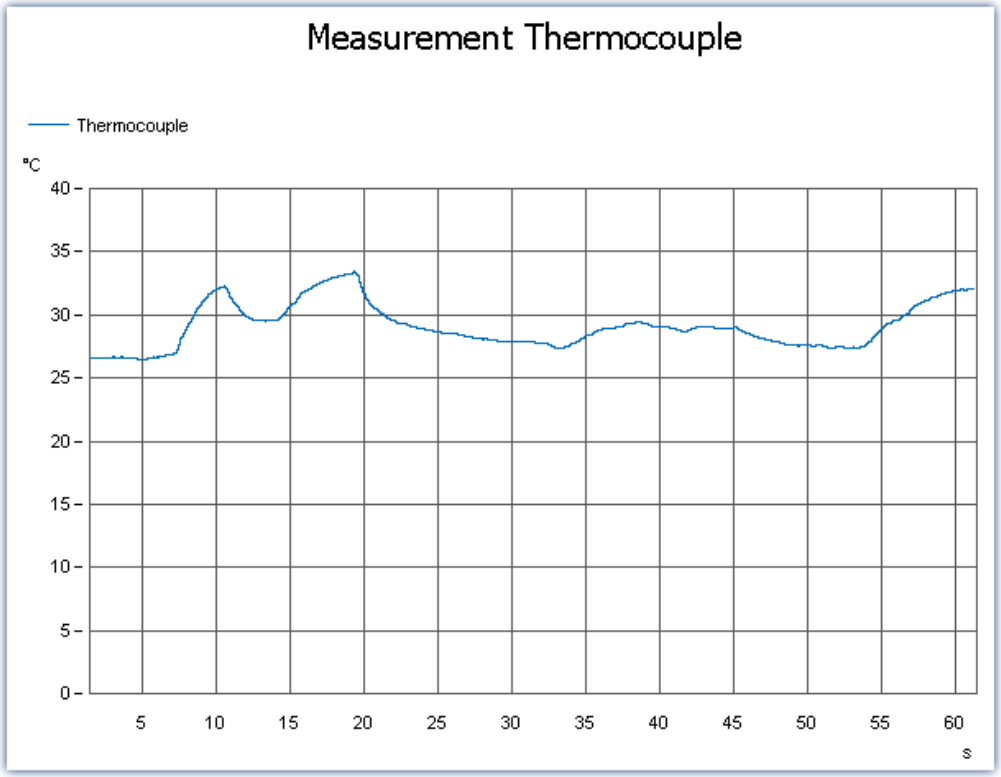
In order to apply the settings, click on the button "*OK*".

- Display a grid by means of the curve window's context menu item **Configuration > Grid**.
- Insert a **text box** with the text: *Measurement Thermocouple* and adjust the font size and the Widget-size.
- Deactivate the Design mode ( ).

Starting measurement, observing results

- Start the measurement 241 ( ).

The following picture illustrates one possible display of the course of the temperature signal.



Curve window with plot of temperature

Saving

With that, you have completed this example and can save it:

- Stop the measurement ( ).
- Save ( ) the experiment under the name "*Thermocouple measurement*".

6.8 Artificial sine - signal generation using imc Online FAMOS

Assignment:

Generate an artificial sinewave imc Online FAMOS and display it on the Panel page. The frequency and amplitude are to be adjustable by Widgets.

Learning goals:

- Use of imc Online FAMOS
- Exercising control via the Panel

Element used:

- Connection with a device
- Channel settings
- imc Online FAMOS

Prerequisite:

- imc measurement device with an analog channel

Procedure:

Begin by start imc STUDIO.

6.8.1 Setup - Settings

First, the measurement device is configured and the variables used are assigned an appropriate name.

- Open the **device configuration page**: "*Setup*".
- Go to the **tab**: "*Devices*" .

Select your device

Configuring the channels

- Go to the **tab**: "*Analog channels*".

Under **Channel type**: "*Analog inputs*", select "*Channel_001*". Make the following settings for the channel's parameter configuration:

Parameter	Value
Name	Clock
Status	Active
Mode	Voltage
Coupling	DC
Input range	± 5 V
Sampling rate	1 kHz
Duration	Undefined

- Go to the **tab**: "*Variables*".
- Under the **Channel type**: "*Display variables*", rename the following variables:
  - *DisplayVar_01* > *Amplitude*
  - *DisplayVar_02* > *Frequency*

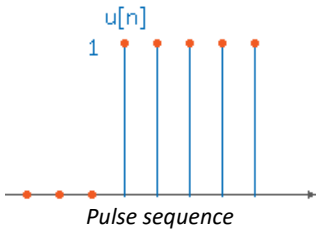
imc Online FAMOS

- Open imc Online FAMOS ()

Ribbon	View
Start > Online FAMOS ()	all
Setup-Configuration > Online FAMOS ()	Complete

- Enter the following source code in the imc Online FAMOS Editor:

```
t = integral(Clock * 0 + 1)
Oscillation = Amplitude * Sin(6.283185 * Frequency * t)
```

Note	Explanation
 *Pulse sequence*	The expression `Clock * 0 + 1` generates the sequence of unit pulses `u[n]` after the measurement starts. Since the channel's sampling frequency is 1 kHz, the pulse sequence consists of 1000 values per second. By integrating the unity pulses, one obtains a straight line with slope 1. The time "t" thus increases linearly.

The general shape of the sine: `x(t)=A*sin(u*t)` with `u=2*Pi*f=6.283185*f`.

Here it is to be noted that the signal is also discrete and has 1000 values per second. In order to change the resolution, you simply need to change the channel's sampling frequency.

The virtual channels you defined, "*t*" and "*Oscillation*" are generated automatically by this assignment.

- Select the Syntax Check () and exit imc Online FAMOS.

Prepare

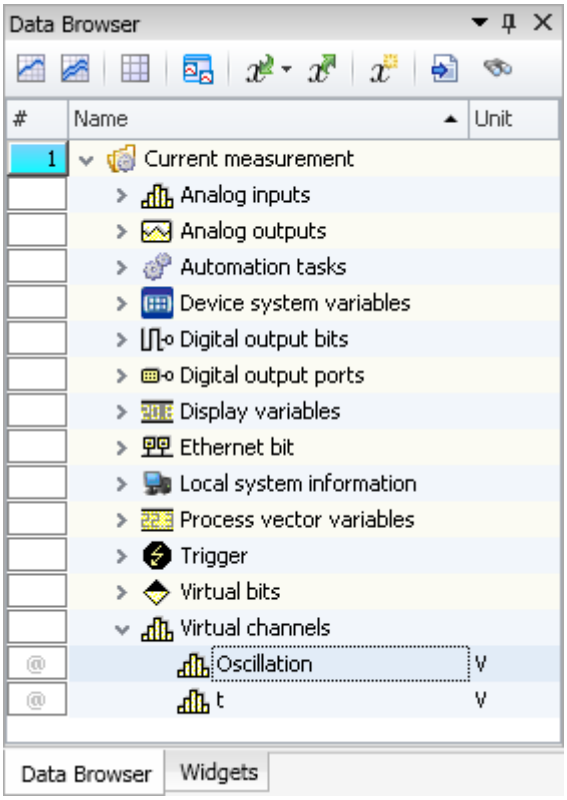
- To **transfer** the current **settings to the device**, click on the button "*Prepare* 241" ( ).

6.8.2 Panel - Settings / Start measurement

Now you can create the Panel page.

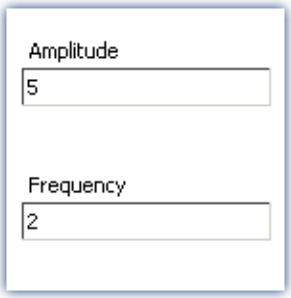
- Open the page: "*Panel*".
- Activate the Design mode ( )

In the tool window: Data Browser you should find the **virtual channel** "*Oscillation*" created in imc Online FAMOS.



- From the tool window: **Data Browser** drag the variable: *Virtual channels* > *Oscillation* to the Panel page.
- In the Widget-selection list, select: **Standard** (curve window).

In order to change the values of variables *Frequency* and *Amplitude*, these are displayed on the Panel page with suitable Widgets.



- Drag the variable: *Display variables* > *Amplitude* to the Panel page.
- In the Widget selection list, select: group **Standard** > **Editor box (Single line)**.
- Drag the Widget: "**Label**" to the Panel page:
  - Tool Window: *Widgets* - Group: *Standard* > *Report* > *Label*.

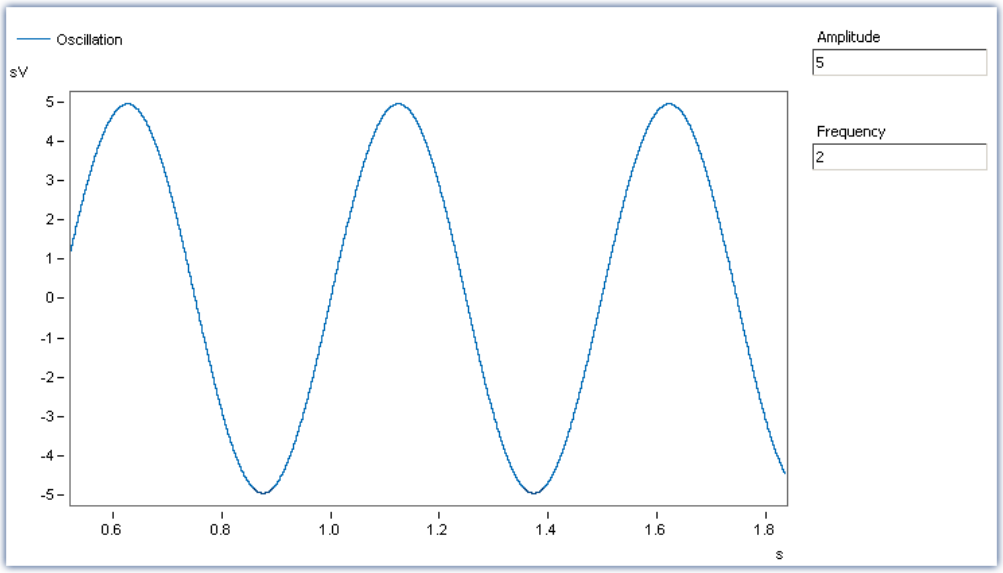
To adapt the **Text**, open the label's Properties.

- In the box **Text**, enter "*Amplitude*".

Display the variable *Frequency* in the same way.

Start measurement, observe results

- Deactivate the Design mode ( ).
- Start the measurement 241 ( ).
- change the *Amplitude* and the *Frequency* and observe how the curve window changes:



Sinus

Saving

With that, you have completed this example and can save it:

- Stop the measurement ( ).
- Save ( ) the experiment under the name "*Artificial Sine*".

7 imc STUDIO (general)

imc STUDIO is the **common framework** forming a **product package** through the combination of modular components (plug-ins).

Which plug-ins are available depends on the product installation (order).

Chapter overview

Synopsis	Section
Data storage: How are **data stored** and loaded	• Experiments, Projects and the Database 110 • Ribbon "Project": Open / Save experiments 121 • What is saved where? 162
Navigation through the entire **software**	• Navigation Pane 139
Feedback from imc STUDIO - information, warnings and error reports	• Logbook 142
Limited operability due to disabled user privileges	• User administration and Access rights 146
The software user's interface is flexible. Saving and restoring of views	• Saving / Loading views 155
Support of process automation by means of replaceable text modules	• Placeholders 166

7.1 Experiments, projects and the database

This chapter describes the interdependencies among the "*database*", "*projects*", "*experiment templates*" and "*experiments*".

Experiment

The experiment includes a record of all settings needed for generating the measurement data as well as viewing and evaluating them. The actual measured data are saved with the "experiment" to which they belong.

You can create a variety of experiments to accomplish a corresponding variety of measurement tasks.
imc STUDIO always works with exactly one experiment and all changes are saved to it.

Among others, the following settings are saved:

- the experiment file (filename extension: "*.imcStudio"),
- measurement files and metadata,
- various backup files and administration files

Among other things, the experiment file contains all the settings configured in the main windows and in the Setup-Assistants.

Each experiment has a unique name which is set when it is saved 116.

Measurement data

By default, the measured data are saved in the Experiment folder. They belong to the experiment. You specify the data structure in the Device-settings under "*Storage* 292" (Setup page: "*Devices*" > "*Storage*").

Project

A project is primarily a collection of various experiments. The factory settings of imc STUDIO are configured in such a way that only one project is present and that you are aware of it to the minimum extent possible.

All of your experiments are saved in this project ("*StandardProject*").

Some options and configurations are not saved along with the experiment. For instance, they might be saved along with the project and apply to all experiments belonging to the project. In some cases, you can define what has to be saved where. E.g. when generating variables, you can change the validity scope and, for instance, save it not just for that experiment but for all experiments in the project.

Whatever is saved at the Project level (and thus not with the specific experiment) is correspondingly denoted at the locations affected. A project can include:

- one or more experiments
- one or more experiment templates 119
- project settings (e.g. views, user administration and project-events)

After the first start of imc STUDIO, a standard project is set up in which your experiments are saved.

Project view - working with multiple projects

When you activate "*Project view*", there is a project-based tree diagram in the Open- and Save-dialogs 115. Here, you can create additional projects and load experiments from other projects.

Note — Creating a project resets variables

When you create a new project, the status of the experiment currently open is saved temporarily and reloaded later after the new project has been set up. In consequence, the values of the variables are reset to the condition they would have if the experiment were re-loaded.

In the process, the event "*Experiment_Loaded*" is not triggered. If you need this to happen for your experiment, please re-load it manually. If you save your experiment with the new project ("*Save As*"), something similar happens. Here, too, the variables are reset, but additionally the event "*Experiment_Loaded*" is triggered.

You can activate the "*Project view*" in the options (under "*Project Management*" > "*General options*"):

Ribbon	View
Extra > Options ( )	All

Option	Description
Project view activated 132	To create more projects, you must activate the **Project view**. This option is deactivated by default. (Information on Projects 110). There is an extended view available of some dialogs 114. Multiple projects can be created and used. Only activate the **Project view** if you wish to use and manage multiple experiments in separate projects. If this function is deactivated, some dialogs are displayed in **simplified form** (e.g. "*Save experiment*" and "*Open experiment*" are **missing Project selection**, for instance).

Database

The database is the data repository for imc STUDIO. Here, the projects and their associated experiments are saved. The database does not have its own settings and configurations.

The folder path for the **database can be freely selected** (in the Options under: "*Project Management*" > "*HDD Settings*").

Ribbon	View
Extra > Options ( )	All

Option	Description
Database folder 132	Here you set where the "**Database**" is to be saved. (Information on "*Data management* 161 ") The database is not user-dependent. Note that every user who is set up in the system has writing and reading rights for this path.

Database conversion

If the database structure has been changed, you are subsequently notified. One way this can happen is as a result of updating to a new version.

A **conversion-dialog** appears. You can convert the database or have it copied beforehand. After the conversion, the entire **database can no longer be used with the older version**.

Databases are not downward compatible.

In the upper region, the reason why the database does not fit with the current version is noted. E.g., the database may be too new or too old. In the list below, all databases found in the folder selected are listed. The "Status"-bar contains information on the database.

The following approaches are available:

Approach	Description
Select an existing database	Select the appropriate database and click on the button "*Apply*". If it is necessary to **convert the database**, an additional dialog appears. Here, the user is prompted to decide whether to **use the database under a new name**. If you choose: • **no:** the existing database is converted. It can no longer be used with the older version. • **yes** (*recommended*)**:** the database is copied. Enter a name for the new database. Now you have two databases. You will have a backup copy and can continue to use the old database with the old version.
Create a new database	Click on "*Create new*". Enter an appropriate name for the database. It will be created in the folder selected (parallel to any already existing databases).
Change the database's folder	Click on the button "..." next to where the folder is indicated. Select an appropriate folder. Here, select a folder where the database folder is to be created later. Not the database itself. E.g. the default folder: "*C:\Users\Public\Documents*". The database will be created in this folder, e.g. "*DB*".

Experiment Template

See: "*Experiment Templates* 119"

Creating one's own files on an experiment - in the folder "Meta"

You can **create your own files on the experiment**, such as metadata, imc FAMOS-sequences, info-files. To do this, use the **folder: "*Meta*"** in the experiment path. If you save or export the experiment under a different name, all files from the folder "*Meta*" come along with it.

The folder "*Meta*" is now ignored in all components.

- The content is not recognized as a measurement although containing measured data. This means that there is no entry in the Data Browser.
- Interval saving does not delete the folder. When the number of intervals is limited, the folders of measured data are deleted in succession once the specified count has been reached. The folder "*Meta*" is ignored, even when the folder contains measured data.

Reference See also

- Structure of and files in the database: "*Data Management* 161"
- What is saved where? 162

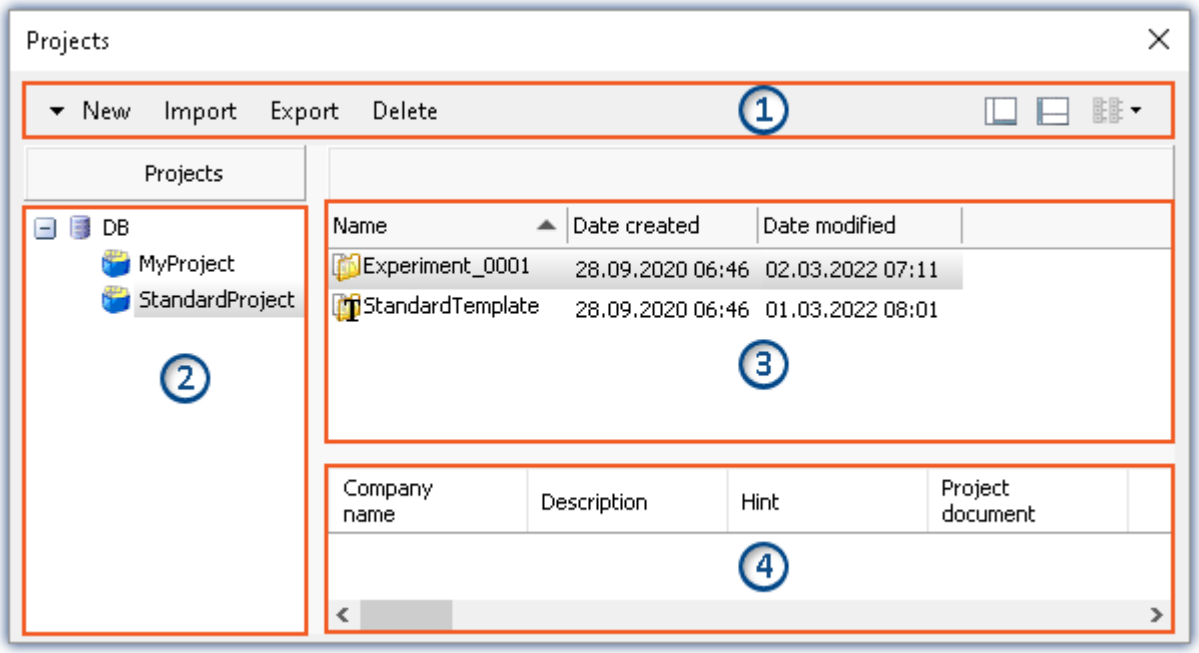
7.1.1 Dialogs: Project and Experiment

The Project Management dialogs are described below:

- with "**Project view**" and
- with "**Experiment templates**".

By default, these options 132 are deactivated, in consequence of which not all functions are available. Options-dependent functions are mentioned explicitly below.

The dialogs following the functions "**Manage Project**", "**New Experiment**", "**Save As**" and "**Open**" are structured similarly. The dialog is described below on the basis of "*Project Management*" as an example. Not all dialogs feature the complete scope of functionality.



Example: Manage project

The dialog is subdivided into four regions (from top to bottom):

1. Menu bar
2. List of all projects
3. List of all experiments and experiment templates in the selected project
4. Linked metadata information belonging to the selected experiment

By default, the regions 1 and 3 are displayed; the others can be activated if desired.

Region 1: Menu bar

Function	Description
New	**Create new project:** Creates a new project in the selected database (a database must be selected). **Create new experiment template:** Creates a new experiment template 119 (a project must be selected).

Function	Description
Import 118	Imports elements (projects, experiments and or experiment templates) from a file in the selected entry. The file can contain multiple elements. You can select either all or individual elements for import. Additionally, the target name can be changed. Be sure to first select the target correctly. You can only import a project if the database is selected. You can only import experiments and experiment templates if a project is selected. You can only import projects if the Project view is activated.
Export 117	Exports the selected entries to a file. Whole projects and individual/multiple experiments and experiment templates can be exported to a file. Manually created folders are not included with the export. The exception is a folder in the experiment path named: "*Meta* 113".
Delete	Deletes the selected entries. Complete projects and individual experiments/experiment templates can be deleted. If you select delete an experiment with saved measurement data, a prompt appears for confirmation that you wish to delete the data.
/	Show/Hide Region 2 (list of all projects)
/	Show/Hide Region 4 (metadata information)

Regions 2 and 3: List of all projects / experiments / experiment templates

In these two regions, the projects/the experiment and experiment templates are listed. When you select a project, all elements of the project selected are displayed in the region at right.

Region 2 is only displayed if the project view is activated and the region is shown (via the menu button:).

Region 2 only shows the experiment templates if these are activated.

Region 4: Linked metadata information

The metadata saved with the selected experiment are displayed in this region. When an experiment is saved, the metadata can be saved automatically along with it. In the Options 130 "*Metadata*" > "*Experiment - Metadata*" > "*Setup page*", you can select which source to use for the metadata.

Only shown if the region is shown (via the menu button:).

7.1.2 Creating/Saving/Copying an experiment

Each experiment has an identifying name which is specified when it is created/saved. Each name may be used only once per project.

When you create an experiment (ribbon menu item: "*Home*" or "*Project*" > "*New*"), or when you save an experiment under a new name, you are asked for a name. In this case, enter a unique name.

If the "Project view 132" is activated, you can additionally select the **target project**.

Action	Description
New Experiment	When you create a **new experiment**, it is generated from an experiment template (see "*Experiment template* 119"). • If experiment templates are displayed 119, select an experiment template. • If they are not displayed, the **preferred experiment template** 120 is used automatically (at default: "***StandardTemplate***"). All changes since the last save are discarded if a new experiment is created.
Save experiment as	When you save an experiment under a new name, a new experiment having the current settings is created. • Measurement data from the existing experiment are not transferred along to the new experiment (exception: if the experiment was previously never saved; in that case a prompt appears to ask whether to include these measured data). • Files from the folder "Meta" 113 are copied and are available in the new experiment.
Copying an experiment to a different project	You can copy experiments to a different project by means either of Drag&Drop or of the context menu. This means that all files located in the Experiment-folder are included: Measured data, saved measurement settings, parameter set files or user-made folders of metadata.

7.1.3 Exporting and importing experiments and projects

Using the project- and experiment-dialogs, you can export/import complete projects along with associated experiments and measurement data. On the other side of scale, you can also export/import only the project settings, or individual experiments.

Exporting

The elements selected are exported to a file.

Ribbon	View
Start > Open ( )	All
Start > Save as ( )	All
Project > Manage Project ( )	Complete
Project > Open ( )	Complete
Project > Save as ( )	Complete

Procedure: Select the elements you wish to export, e.g. two experiments. Click on "*Export*".

A dialog appears in which you can select what to package in the "*imcStudioExport*-file". Confirm your selection and select an appropriate destination.

Selection	Description
Project-settings	(Only when exporting projects) All projects selected are exported
Experiment-settings	Project-export: All experiments saved with the project are exported Experiment-export: All experiments selected are exported
Data	Among other items, the experiment's measurement data are included with the export (caution: the file can become very large)

Note Metadata

Manually created folders are not included with the export. The exception is a folder in the experiment path named: "*Meta* 113".

Importing

You can import previously exported projects or experiments by means of the project-/experiment-dialog.

Ribbon	View
Start > Open ( )	All
Project > Manage Project ( )	Complete
Project > Open ( )	Complete

Procedure: First select the location to which to import the content. Next, click on the "*Import*"-button and select the file desired.

The "*imcStudioExport*-file" may contain multiple elements. You can select either all or only individual elements for import. Additionally, the target name can be edited.

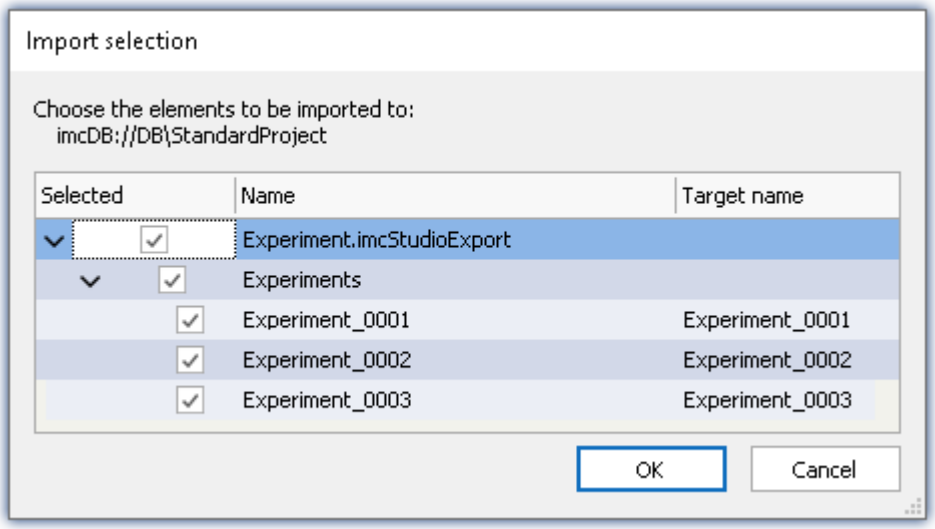
In the first dialog, select the main elements to import (which projects or which experiments). When importing projects, the individual projects are shown; when importing experiments, the various experiments are shown. In the column "*Target name*" you can edit the name. (Note: the dialog only appears when it contains more than one element, or when an element is renamed)

Be sure that you previously select the target correctly.

You can import a project if the database is selected.

Experiments, if a project is selected.

Notes: You can only import projects if the Project-view is activated.



Import multiple experiments from a single file

Confirm your selection.

Subsequently, another dialog appears (as for the Export procedure), where you can set which additional elements to import from the file.

Selection	Description
Project-settings	(Only when importing projects) All projects selected are imported
Experiment-settings	Project-import: All experiments saved with the project are imported Experiment-import: All experiments are imported from the file
Data	Among other items, the experiment's measurement data are included with the import

FAQ What happens when importing existing elements?

You can import project-settings and experiment-settings without deleting the underlying elements. E.g., you can swap out the project. The underlying experiments remain intact.
Thus, the project can be modified at a developing PC and then imported to the test station.
Only elements having the same name are overwritten.

7.1.4 Experiment templates

When you create a new experiment, it is generated from an experiment template. The new experiment contains all properties of the selected template. In templates, all settings which are saved will be saved also in an experiment.

After the first installation or after creating a new project, the project contains an "empty" experiment template.

Making experiment templates visible

You can activate the "*Experiment templates*" in the options (under "*Project Management*" > "*General options*"):

Ribbon	View
Extra > Options ( )	All

Option	Description
Show experiment template 132	In order to be able to use experiment templates, you must previously make the **experiment templates visible**. This option is deactivated by default. (Information on Experiment templates 110) An extended view of certain dialogs 114 is available. Experiment templates can be created and used. Only activate this function if you wish to create multiple similar experiments. If this function is deactivated, some dialogs are displayed in **simplified form** (e.g. "*New experiment*" and "*Edit project*" appear **without experiment template selection**).

Creating experiment templates

Ensure that experiment templates are visible. You can generate experiment templates from a variety of sources:

Source	Description
From present settings	The configuration currently set is used for the template. (see also "*Save as template* 121")
From existing experiment	Select an experiment from which to generate the template.
From default settings	An empty template is created.

- Click on the button: "*Manage Projects*"

Menu ribbon	View
Project > Manage Project ( )	Complete

- Select the project (if the projects are not displayed, click on the white background, in order to avoid selecting an experiment)

- In the dialog, click on the menu item: "*New*" > "*Add experiment template*"
- Select a source

The experiment template is created from the source and and is available when creating an experiment 116.

Preferred experiment template

You can **mark** an experiment template **as preferred** (template context menu > "*Set as preferred experiment template*"). If the experiment templates are no longer displayed, the new preferred experiment template is automatically used when a new experiment is created.

7.2 Ribbon

7.2.1 Project menu

The "*Project*" menu is available in every plug-in.

Some of the dialogs offer an extended view 132. By default, the dialogs are displayed without "**project view**" and "**experiment templates**".

Project

Menu item	Description
Manage Projects 114	Managing projects and experiment templates
Save Project	Saves the current project

Experiment

Menu item	Description
New 116	Creates a new experiment
Open	Opens an existing experiment
Save	Saves the current experiment
Save as 116	Saves the current experiment under a new name. Additionally, the experiment is reset. The variables are assigned the respective initial value (e.g. device variables = "0", or user-defined variables are assigned the respective initial value specified). The event "*Experiment_Loaded*" is triggered.
Save as template	When you click on the button, a new template 119 is generated which the system then uses automatically (keyword "*Preferred experiment template* 120"). (from the edition "imc STUDIO Professional")

Im-/Export

Menu item	Description
Import / Export 122	In this dialog you can import and export various components.

Current Measurement

Menu item	Description
Save current Data Export current Data	In addition to the optional automatic data saving, you can also perform targeted saving of the data currently in the PC (Current measurement) either during or subsequent to the measurement. See also Setup pages - Configuring device: "*Storage options and directory structure*" > "*Controlling Data Storage*" > "*Targeted Data Saving, or Saving Subsequent to Measurement* 439"

Note Without imc STUDIO Project Management

In the product configuration, the component "**imc STUDIO Project Management**" can be deactivated. If "**imc STUDIO Project Management**" is deactivated, a number of limitations apply.

Among others, there is no longer any database. The dialogs for saving and opening experiments match the standardized "Save As"- or "Open"-dialogs. Experiments can be saved at any desired location. The saved measurement data are stored in the designated experiment folder.

The functions of the Project Management are listed in the technical data sheet.

The description of the Project- and Experiment-dialogs n the following pages pertain to imc STUDIO with activated Project Management.

Changes in the menu without imc STUDIO Project Management:

Project - without Project Management

Menu item	Description
Save Project	Saves the current project. In this case, all "Project"-settings are saved as "Application"-settings and apply for all experiments.

Experiment - without Project Management

Menu item	Description
New experiment	Creates a new experiment. The storage location is first determined by means of "*Save experiment/Save experiment as*". The device currently selected remains selected. However, the device's configuration will be reset.
Open experiment	Opens an existing experiment. The experiment can be be saved anywhere in the file system.
Save experiment	Saves the current experiment at the storage location previously set using "*Save experiment as*"; if no storage location had been determined yet, "*Save experiment as*" is run automatically.
Save experiment as	Saves the current experiment under an arbitrary name at any location in the file system

7.2.1.1 Import / Export

In this dialog you can import and export various components.

Select the ribbon menu item *Project > Import / Export*.

Option	Description
Parameter set	Enables loading and exporting of values (data pool) and settings (device configuration). Select the desired settings/values. Details are provided in the documentation on the command: "*Import parameters* 1172" and "*Export parameters* 1167".
Views, metadata columns, (sensors,) ... 157	Exports views, metacolumns, ... and sensors set up in imc STUDIO to a file. Or conversely, import views and metadata columns, ... from a file. You can adapt the imc STUDIO user interface, for instance by altering the Setup pages or the tool window. These settings can be saved by means of exporting. in this way, views can be transferred to other projects/applications/PCs.

Option	Description
imc DEVICES experiment 283	Imports the devices configurations of an old imc DEVICES experiment or an imc DEVICES device configuration for a device.
Supplemental files	Supplemental files can be import and exported: • Characteristic curves and filter data for imc Online FAMOS / imc Inline FAMOS (*.dat) • imc Online FAMOS source code (*.ofa) • Synthesizer directory structures (*.dat) • Messaging configurations (e-mail, SMS, UDP, ...) (*.msg) Export: Shows the dialog to manage supplemental files 212, in which you can export in the experiment existing supplemental files. See the description: "*Setup*" > "*Ribbon*" > "*Configuration*" > "*Supplemental Files* 212".
Synthesizer-/controller configuration	Imports synthesizer-/controller configuration.
imc DEVICES adjustment values	Imports adjustment values created with imc DEVICES.
Device descriptions ('<Device name>')	Import: Imports device descriptions (umi files) as new devices in the Setup, a devices had been selected following a device search. Afterward, the device is available for configuration. Export: Exports the device description (umi file) for the device '<Device name>' into a selectable folder.
MFB configuration	Imports ARINC (*.idb), AFDX (*.xml) or CAN (*.cba) configurations.
Panel pages of an imc STUDIO experiment	Imports just the Panel pages of an imc STUDIO experiment without any device configuration data or measured data. Be aware that all pages present will be deleted.
Transfer experiment to other devices 281	Opens an experiment and offers a choice of which devices to use for the experiment.
Setup table description	Imports an exported Setup table description.
Setup column descriptions	Imports an exported Setup column description to an existing table description.
Import sensors	Imports sensors created in imc STUDIO from a view settings file 157.

7.2.2 Edit menu

The effects of some functions depend on the current main window.

Clipboard

Menu item	Description
Cut	Removes the selection (the selected elements / the highlighted text) to the Clipboard.
Copy	Copies the selection (the selected elements / the highlighted text) to the Clipboard.
Paste	Inserts the Clipboard contents at the position highlighted.

Edit

Menu item	Description
Undo	Reverses the effect of the last change to the current window.
Redo	Restores the previously reversed change to the current window.
Delete	Deletes the selection (e.g. the source text, the Panel-page, the Widgets selected or the selected Automation-element).
Select all	**Panel:** Selects all of the page's Widgets.
Aktualisieren	**Setup:** Metadata columns are updated. This affects file which, for instance, are accessed via a link (such as PDF-files). These are re-loaded if they have changed in the background.

Search

Menu item	Description
Find	Find specific text in the current window. **Automation:** Searches through the source text of all elements and lists all results found in the Search-dialog. It is possible to browse through the individual elements by clicking the mouse (see "*Source Code - find and replace* 1622").
Find and Replace	Find and replace specific text in the current window. **Automation:** Searches through the source text of all elements and lists all results found in the Search-dialog. All results located can be replaced here (see "*Source Code - find and replace* 1622").

Print

Menu item	Description
Print	**Panel:** Opens the dialog for printing the Panel pages. **Setup:** Creates a printable version of a report of the current configuration. Appearing in the list are multiple device- and channel-configurations, as well as the imc Online FAMOS-source text (see "*Report of the Setup-configuration* 285"). **Automation:** Creates a printable version of a report of the current configuration. There are three available varieties, extending from a broadly outlined structure to the contents with the source text of all elements (see "*Report of Automation-configuration* 1625").
Print Preview	**Panel:** Opens a print preview. Here you can see the possible appearance of printing results. This can be helpful when Widgets such as the curve window are used, which have their own color settings for the printout. **Setup:** Creates a report. See "*Print*". **Automation:** Creates a report. See "*Print*".

7.2.3 View menu

Views

Menu item	Description
Save View 156	Saves current view Overwrites the used view in the project settings.
Save View as 156	Saves current view in the project settings under a new name freely specified by the user, e.g. "Test Drive 1".
View selection 155	Loads view from the project settings.
Delete View 156	Deletes a saved view from the project settings.
Restore 156	Restores views from the factory settings.

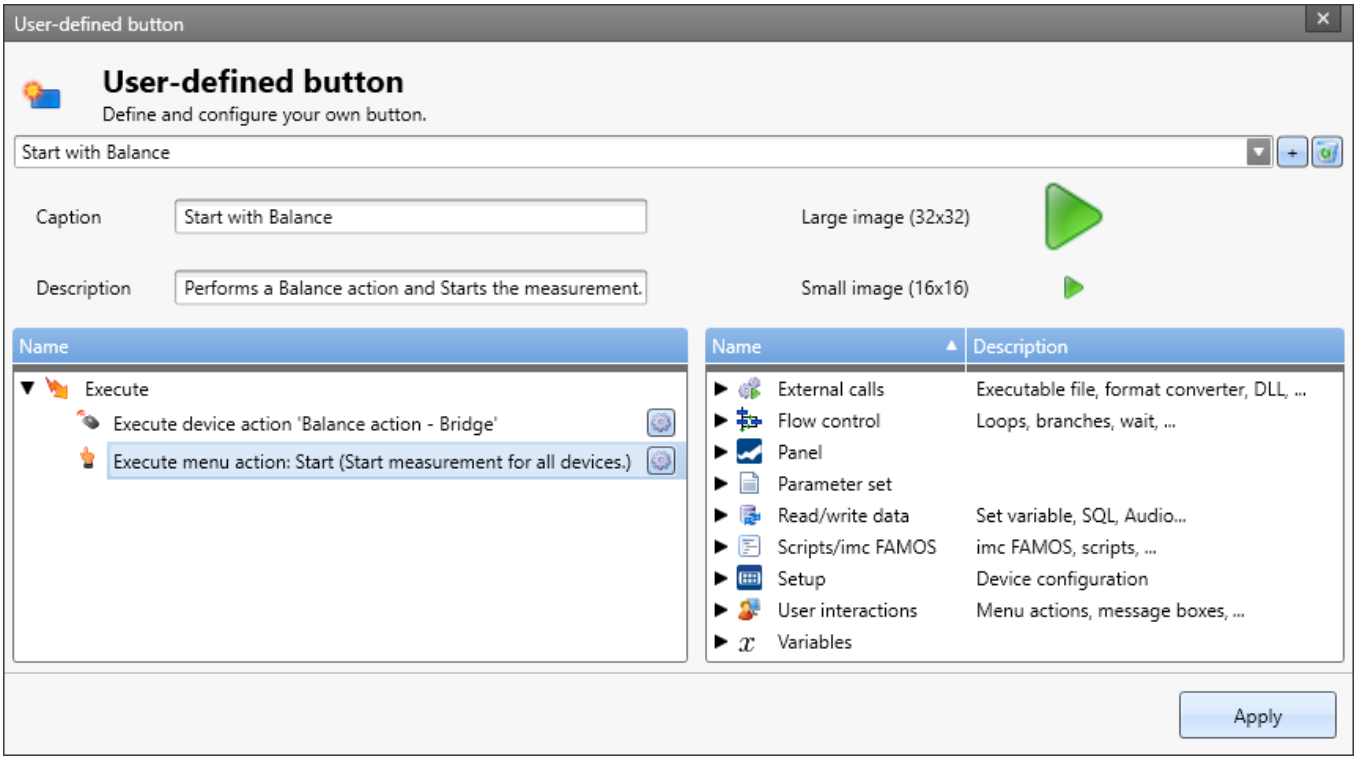
Layout

Menu item	Description
Reset Window Arrangement	This function sets the currently open main window's GUI back to its last saved condition. This affects the window arrangements: • the position and size of the imc STUDIO window • the position and size of the tool window
Customize Ribbon Menu 159	By means of this function the menu ribbon can be customized.
User-defined buttons 126	Creates user-defined buttons which cause commands to be carried out.
Choose Tool Windows 141	Opens the dialog: "*Tool Windows Chooser*" in order to show/hide the tool windows.

7.2.3.1 User-defined buttons

You can create your own personal buttons for the menu ribbon. Certain commands are associated with each button, which are run when the respective button is pressed.

The pop-down list contains all user-defined buttons available. The "Caption" of the respective button is used for identifying the button in this list. The list can be used to open existing buttons and to reconfigure/delete them.



Example of a user-defined button:
this one performs a channel balance and then starts measurement

Adding a new button

Click on the "+"-symbol to create a new button.

Configure the display in the button's menu ribbon:

- Caption: the displayed caption in the menu ribbon
- Description: the information in the button's tool-tip
- Image: the icon in the menu ribbon for the large and the small variants

Command configuration

Below, there are two areas:

- At left, the commands to be run
- At right, the available commands

Use the Drag&Drop technique to add commands to the list at left and configure the commands accordingly.

Deleting a command

If you wish to delete a command from the list, select it and click on the <Del>-key.

Adding a button to the menu ribbon

Click on the button "Apply". In consequence, the dialog for modifying the menu ribbon appears, but its scope is limited to the user-defined buttons. Add the button at the desired location in the menu ribbon. Information on this topic is presented in the chapter: "*Customize ribbon* 159".

Note Saving the button's configuration

The **configuration of the buttons** is saved 162 with the respective **project**. Thus, the button is available in all of the project's experiments.

The configuration **is only saved** whenever the **project is saved**.

The display in the button's menu ribbon is saved with the view.

7.2.4 Extra menu

The content of the menu "**Extra**" depends on what plug-ins are included.

Options

Menu item	Description
Options 128	General options

User Administration

Menu item	Description
Login 151	Logs in a user
Logout 151	Logs a user out
Administration 147	Sets up a user and assigns them to user groups
User access rights 154	Assigns rights to user groups
Startup behavior 150	For setting imc STUDIO's startup behavior

Sequencer

Menu item	Description
Start (Sequencer)	Starts the Sequencer. For more info see the description of the plug-in Sequencer 1681
Stop (Sequencer)	Stops the Sequencer. For more info see the description of the plug-in Sequencer 1681

7.2.4.1 Options

In this dialog you can make fundamental settings for the individual components.

Where are the options saved?

The individual options each have different storage locations. Some are saved with the project, others belong to the entire application (the imc STUDIO installation). A diskette-symbol indicates the storage location:

Scope	Description
Application option	Options which apply for the imc STUDIO installation. The apply for all projects and experiments belonging to the selected database.
Project option	Options applying to all experiments belonging to the current project.

Resetting options

By means of the selection list (at the bottom left), it is possible to reset either all options or individual pages.

Selection	Description
Reset	Resets the **currently opened page** to its **last state**. I.e., back to the settings which applied at the time the dialog was opened.
Reset All	Resets the options for **all pages** to their **last state**. I.e., back to the settings which applied at the time the dialog was opened.
Default	Resets the **currently opened page** to the **factory settings**. I.e., back to the settings which applied upon installation of the system.
Default All	Resets the options for **all pages** to **factory settings**. I.e., back to the settings which applied upon installation of the system.

Commands

E-mail options

Default values for the Mail-command. As long as no other settings are made in the corresponding command, these default values are used.

Please obtain the necessary settings values from your E-Mail provider.

Options - Mail options	Description
Mail server (SMTP)	Outgoing mail server of the E-Mail provider used.
Password	Password for authentication.
Port	SMTP server port used.
Postfix	This postfix will be appended to every e-mail.
Sender name	The name of the sender of the e-mail (your name).
Sender address	Sender address (your e-mail address).
User	User name for logging in.

Data Browser

General options

Options	Description
Grouping by categories	The variables are grouped in the Data Browser according to their respective categories (e.g. "Analog inputs", "User-defined variables"). This can be deactivated in order to sort the variables by their names. You can create your own groups by naming the variables according to the following syntax: "Group.Name". For instance, if there are two channels: MeasPoint1.Temperature and MeasPoint1.Voltage, these variables are automatically grouped in the group: "MeasPoint1".
Save the assignment of the measurement number	If a measurement number is assigned to a measurement, the assignment is saved with the experiment.

General options

Default dialog options

Here it is possible to specify default answers for dialog queries. If a value other than "Show dialog" is selected, the dialog is not shown and instead the selected answer is used.

Storage location measurement data

Options	Description
Custom base path for measurement data storage	The base directory path for storage of measured data represents the default filepath for experiments. In this folder, all measurements are saved (according to the structure specified in "*Folder naming*"). See also "Setup pages - Configuring device": "*Storage options and directory structure*" > "*Controlling Data Storage*" > "*Customized storage location for measurement data* 435".
Custom folder naming for measurement data storage	This specifies the structure below the base path. For instance, folders are set up here according to the settings for the directory structure 433. See also "Setup pages - Configuring device": "*Storage options and directory structure*" > "*Controlling Data Storage*" > "*Customized storage location for measurement data* 435".

Logbook

Logbook

Options - Logbook	Description
Delete Logbook-files [days]	After the specified number of days elapse, the logbook files are deleted automatically. The deletion verification test is performed upon starting the program and at 0:00 (12 AM).
Default settings for the filters in the logbook tool window	Category of messages for which the filter presets are to be made.
Open logbook in response to a report of category:	For this category, the logbook is opened and attached.

Metadata

General Options

Options - Experiment - Metadata	Description
Export upon saving	Activates/deactivates automated export of the selected Setup page to the experiment. The export is performed whenever the experiment is saved. For example, in the project management dialogs (e.g. Open Experiment), the parameters saved for the experiment can be displayed (see: "*Dialogs: Project and Experiment in Region 5* 115").
Setup page	Here you define what Setup page to export when saving the experiment.

Panel

General Options

Options - Options	Description
Multi-lingual text input	Enables textual input for computers with other language settings. For example, in this way, Panel pages can be pre-configured for different language settings. If the language set is configured for the respective text, it is displayed. Otherwise, the default language: "English" is displayed.

Navigation

Options - Options	Description
Always link new Widgets to Navigation bar	The navigation bar currently has no function.
Navigation through all Panel pages	The navigation bar currently has no function.
Post-processing mode as default	The navigation bar currently has no function.
Widgets navigation mode	The navigation bar currently has no function.

Options - View	Description
Datacut tools	The navigation bar currently has no function.
Post-processing tool	The navigation bar currently has no function.
Time controls	The navigation bar currently has no function.

Panel Widgets

Options - Widget Configuration	Description
Refresh rate of newly created Widgets	Specifies the refresh rate of Widgets which are newly created. If a Widget is created on a Panel page, it is assigned the refresh rate set here.

Project Management

General Options

Options - General	Description
Project view activated	To create more projects, you must activate the **Project view**. This option is deactivated by default. (Information on Projects 110). There is an extended view available of some dialogs 114. Multiple projects can be created and used. Only activate the **Project view** if you wish to use and manage multiple experiments in separate projects. If this function is deactivated, some dialogs are displayed in **simplified form** (e.g. "*Save experiment*" and "*Open experiment*" are **missing Project selection**, for instance).
Traceability of measurements	When Traceability is activated, all experiment settings belonging to the respective measurement are saved separately. Thus, for every measurement, it is possible to retrieve the configuration via the Data Browser. Note: Changes made to the setting only take effect on future measurements. See: • "*Data Browser*" > "*Context menu*" > "*Load measurement settings* 1227" • "*imc STUDIO (general)*": "*Information and tips*" > "*Data management*": ms-files 161
Show experiment template	In order to be able to use experiment templates, you must previously make the **experiment templates visible**. This option is deactivated by default. (Information on Experiment templates 110) An extended view of certain dialogs 114 is available. Experiment templates can be created and used. Only activate this function if you wish to create multiple similar experiments. If this function is deactivated, some dialogs are displayed in **simplified form** (e.g. "*New experiment*" and "*Edit project*" appear **without experiment template selection**).

HDD Settings

Options - General	Description
Database folder	Here you set where the "**Database**" is to be saved. (Information on "*Data management* 161 ") The database is not user-dependent. Note that every user who is set up in the system has writing and reading rights for this path.

Scripting

Script options

Reference

See "*Scripting*" > "*Script options* 1906".

Sequencer

General options

Optionen - General	Description
Reduce logbook entries	Reduces the number of logbook entries produced by the Sequencer. The following messages will be suppressed and not be reported to the logbook window or the logbook file: • Starting/Stopping command • Messages regarding loops (for/while) • Messages regarding if and switch

Setup

General

Options - Common	Description
Data storage state after preparation	The state of the data storage after preparation (prepare or reconfiguration) of the devices has been performed. • **As before preparation:** The state does not change in consequence of preparation. • **Data saving interrupted:** After preparation, the state is set to: "Data storage suspend". Data storage must be resumed as soon as the incoming measured data need to be saved. • **Recording data:** After preparation, the state is always set to: "Data storage running". This option does not replace activation of storage for each channel. Instead, this option regulates the condition of "***Suspend and Resume data storage*** 198". Please observe the notes on "*Preparation*". Once the button has been pressed, preparation is only performed if any change has been made. See "***Prepare and Reconfigure*** 195".
Parameter set export - Expanded mode	If set, you can control which columns 1170 will be exported upon parameterset export.

Options - First selected device	Description
Append device names to the channel names	If this option is activated, when a device is selected, the device name is always appended to the channel name, even when the first device is selected. If the option is deactivated, the device name is only appended as of the second device. This setting only takes effect the next time a device is selected, unless any device was selected previously.

Traceability of data

Options - Traceability	Description
Traceability of data	Groups of Setup-parameters which are saved as metadata in the Channel Properties. They can be read out again at a later point in time. The channel properties make reconstruction of the measurement's configuration possible.

Device options

Options - General	Description
Selection of the firmware version	If multiple imc DEVICES versions are installed on the PC, a specific version must be selected for the operation of each device. This option controls the selection strategy. If only one imc DEVICES version is installed, this setting has no effect. • **Manual:** With "**Manual**", the system always asks which firmware version to use when device are selected or an experiment is loaded. • **Automatic:** "**Automatic**" avoids firmware updates. The version currently running on the device is used preferentially. • **Always use latest:** With this selection, the most current firmware version compatible with this imc STUDIO version is always used.
Show warning for critical parameter changes	Show a warning when a critical parameter value changes, e.g. when changing the supply provided to all of a module's channels. Disabling this option can be particularly useful when using automated procedures such as Sequencer or Scripting, since the warning otherwise pauses the procedure until it has been confirmed.
Time limit for calling device control software [s]	Timeout in seconds for a call to the device control software. If this time elapses, the device control software automatically restarts. This can cause a parts of the configuration to be lost. If the devices are operated over a very slow connection, this value can be increased to ensure correct functioning. The minimum and the default value is 60 seconds.

Options - Virtual device clock	Description
Virtual device clock: The virtual clock synchronizes during running measurement with the Master-device's clock. This is necessary when functions are being used on the PC which require an exact time track (e.g. Video). In networks under high load, good synchronization (between the PC's virtual device clock and the device) is not always possible. When the demands on the network are too high, imc STUDIO posts a corresponding error message when an attempt is made to start measurement. It may be helpful to increase the setting for the Minimum accuracy. See also "Setup pages - Configuring device": "*Synchronization*" > "*Clock types*" > "*VRTC - Virtual clock on the PC* 307".	
Minimum accuracy [ms]	The option defines the maximum time in milliseconds that the PC's virtual clock may deviate from the device clock. This concerns mainly non imc devices, e.g. video. In case of video a value of 10 ms provides synchronisation of measurement data to video images at 100 frames per second. Too low a value prevents measurement start.

Sensors

Options - Sensor Characteristic Wizard	Description
Adjusting variable editable	Makes the option "Adjusting variable" editable in the Assistant.
Adjusting variable visible	If this option is activated, the option "Adjusting variable" is displayed in the Assistant. The values of the characteristic curve can be imported in reference to the adjusting variable.
Delay editable	Makes the option "Delay" editable in the Assistant.
Delay visible	If this option is activated, the option "Delay" is displayed in the Assistant.
Measurement variable editable	Makes the option "Measurement variable" editable in the Assistant.
Measurement variable visible	If this option is activated, the option "Measurement variable" is displayed in the Assistant. The values for the characteristic curve can be imported in reference to the measurement variable.

Variables

Datapool

Settings for memory use by the imc STUDIO Datapool.

Options - Speicher Optionen (PC)	Description
Split up pretrigger data into intervals	If the pretrigger's measured data are located in time across interval boundaries, it is possible to treat the measured data as follows: • **deactivated:** The pretrigger's data all are placed in the first interval (the interval in which the trigger fires) (*default setting*). • **activated:** The pretrigger's data are cut correctly. In this way, interval folders can be created subsequently. See also: "*Setup pages - Configuring device*" > "*Trigger and Events*" > ... > "*Pretrigger* 403"

Save/Export current Data

In addition to the optional automatic data saving, you can also perform targeted saving of the data currently in the PC (Current measurement) either during or subsequent to the measurement.

Reference

See also Setup pages - Configuring device: "*Storage options and directory structure*" > "*Controlling Data Storage*" > "*Targeted Data Saving, or Saving Subsequent to Measurement* 439".

Save current Data - Variable export

Define beforehand which file types are to be used for the export of the variables. The selection applies to:

- the command "*Export variable* 1758" and
- the "*Export current data* 439" menu action

The following applies for the command "*Export variable*": This is only a preset which is set for the command when the variable is selected. You can still modify the file type in the command later. The system will not adopt any file type in the command configuration which the variable does not support.

User-defined variables

These settings are used automatically upon import, conversion, or when setting user-defined variables.

Options - Automatic prefix on variable name according to scope	Description
Scopes (Experiment, Persistent, Project, Sequencer, Temporary)	If true, all variables of the scope will receive a prefix to their name upon creation or conversion.

Measurement Administration

Options - Variables	Description
Access to stored measurements	If this option is activated, saved measurements in the Data Browser are displayed.

7.2.5 Help menu

Internet

Menu item	Description
imc Website	Opens the imc website's Home page. Here, the user is provided with a quick way to contact imc , for example.

Activation

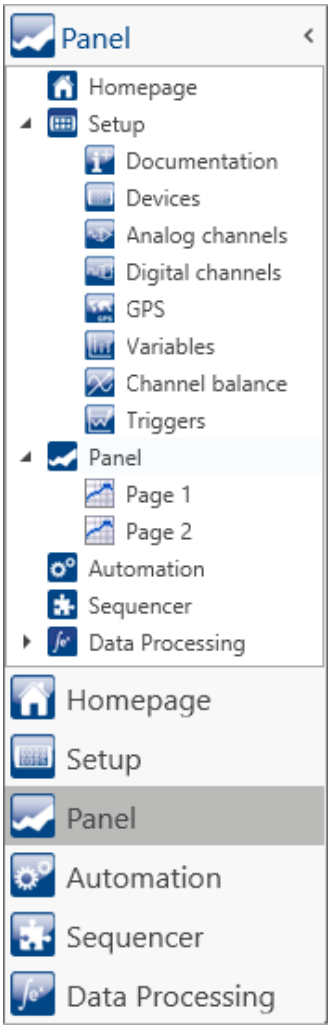
Menu item	Description
Product Configuration	Opens the Product configurator. This allows you to modify the product configuration in order to adapt it to the license you purchased. See "Changing the product configuration 29"
imc LICENSE Manager	Opens the imc LICENSE Manager. The software licenses are manged by means of the imc LICENSE Manager. If appropriate, customize the licenses here.

Help

Menu item	Description
Help	Opens the "Help" for imc STUDIO.
Additional Documents	Opens the program "imc Help and Documentation 70". Here you will find documents on the imc devices and other products.
About	Here, all the important "version information 39" on your installation of imc STUDIO is presented.

7.3 Navigation pane and Quick access toolbar

Navigations pane



Maximized Navigation Pane (example)

In the **Navigation pane**, the main windows of the installed plug-ins are displayed as a tree diagram. To open/change the main window, click on its corresponding entry.

The Navigation pane can be opened or closed, maximized and minimized.

- To **maximize/minimize** the Navigation pane, click in its top bar on the arrow symbol.
- To **open/close** the Navigation pane, click on the upper region of it below the arrow.

If the Navigation pane is maximized or opened, the main windows are additionally displayed as a tree diagram (see example shown).

Via the tree diagram, it is possible to jump to different main windows or directly to the pages containing the main windows.

You can hide the Navigation pane.

If you hide the Navigation pane, please note that your ability to navigate between the main windows. For this case, you should add the command: "*Browse workspace*" at suitable locations, for instance in the ribbon menu ("*User-defined buttons* 126").

To show or hide the Navigation pane use the dialog: "*Tool windows chooser* 141".

Quick access toolbar

The "Quick access toolbar" can be supplemented with various menu items. Which actions are available for this purpose doesn't depend on what tab is currently displayed in the menu ribbon.

Using the context menu, you can either add or remove menu actions.

- Adding: context menu of the menu item in the menu ribbon (note: the menu ribbon may not be minimized)
- Removing: context menu of the menu item in the toolbar

You may place the toolbar either above or below the menu ribbon.

Note — Saving the configuration

The **configuration of the "Quick access toolbar"** is saved 162 with the respective **view**.
The configuration **is only saved** whenever the **project is saved**.

7.4 Tool windows

Tool windows contain special elements for operating/viewing a main window. (For information on operating the tool window: "*Operation* 140")

Five tool windows
- The imc STUDIO frame's log book
- Four additional windows from the main window imc STUDIO Panel

One tool window belongs to the imc STUDIO frame and is always available:

- Logbook 142

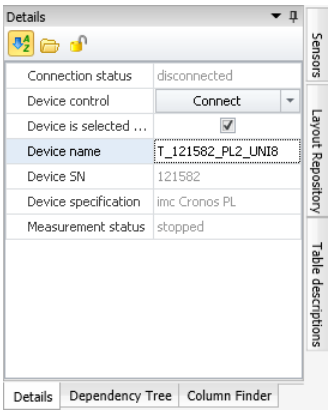
The *Logbook* is always displayed and can be minimized. The other main windows' tool windows (e.g. for the Panel) are displayed as soon as the respective main window is opened.

Note — Not all tool windows are visible

By default, all necessary tool windows are displayed. All others can be displayed 140, if you need them.

7.4.1 Operation

Tool windows contain special elements for operating and editing a main window.



Tool window (example)

Each main window has its own tool window, which is described in the documentation of the respective component. Tool windows can be moved and deleted.

By default, the tool windows are pinned to the main window (e.g. at the bottom or right margin).

The picture shows an opened tool window ("***Details***"). It is pinned and contains two additional tabs. If you click on one of these tabs, the corresponding tool window opens ("***Dependency Tree***" and "***Column Finder***").

Three other collapsed tool windows are seen at the right margin ("***Sensors***", "***Layout Repository***" and "***Table descriptions***").

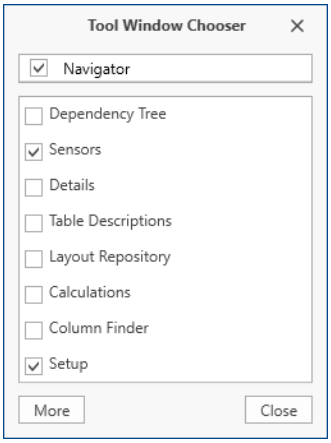
A tool window is opened when the mouse is swiped over it.

As long as a tool window is selected, or if the mouse is located over the tool window, the tool window remains open. Otherwise, it collapses to the side.

Showing/hiding the tool window

You can either show or hide tool windows. To do this, open the dialog: "*Tool Window Chooser*".

Ribbon	View
View > Choose Tool Windows ( )	Complete
Extra > Choose Tool Windows ( )	Standard

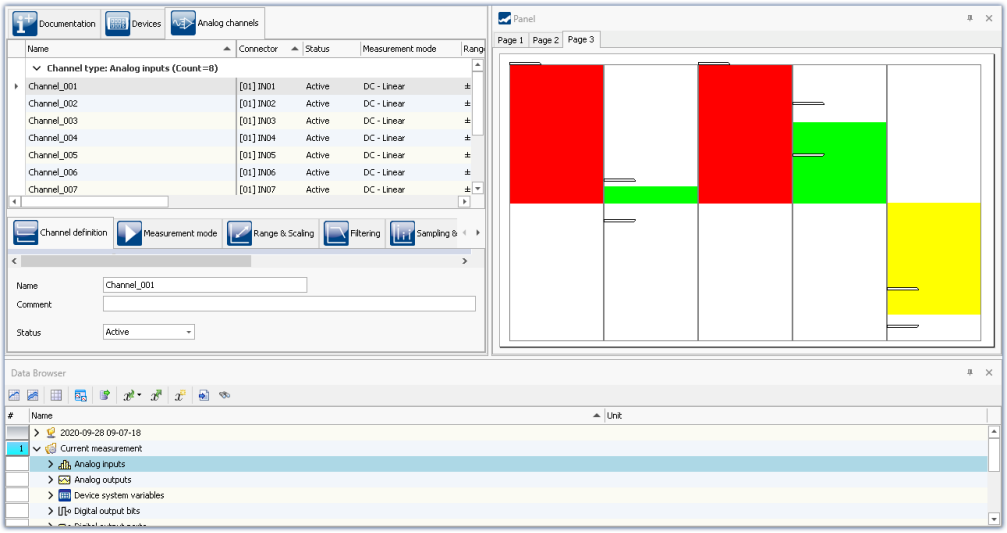


Dialog: Tool windows chooser

The dialog: "*Tool Window Chooser*" opens.

Set a checkmark in front of the tool windows which you wish to have displayed.

Selecting "*More*" provides you with comprehensive layouting capabilities for the view. You are able to place any tool window belonging to other components on the current page. For instance, you can display the Data Browser in the window of the Automation. Or you can show the Panel in the Setup.



Example: Panel and Data Browser are displayed together in the main window "Setup".

Other actions

Action	Description
Pinning the tool window	Tool windows can be pinned. Pinned tool windows remain open even when no longer selected. In order to pin a tool window, click on the thumbtack-button (📌).
Freely positioning the tool window	In order to freely position the pinned tool window, move the tool window's title bar using drag&drop to the desired position.
Docking the tool window	In order to dock the tool window, use the drag&drop technique by grabbing the tool window's by its title bar and moving it to the desired position. A Dock-symbol will appear at the locations which are possible. In the center of the targeted window (main window or tool window), a cross appears. To dock the tool window, release the mouse button with the mouse over the desired position. Dock at the top/bottom margin Dock at the left/right margin Insert as a tab in a different window (see example)

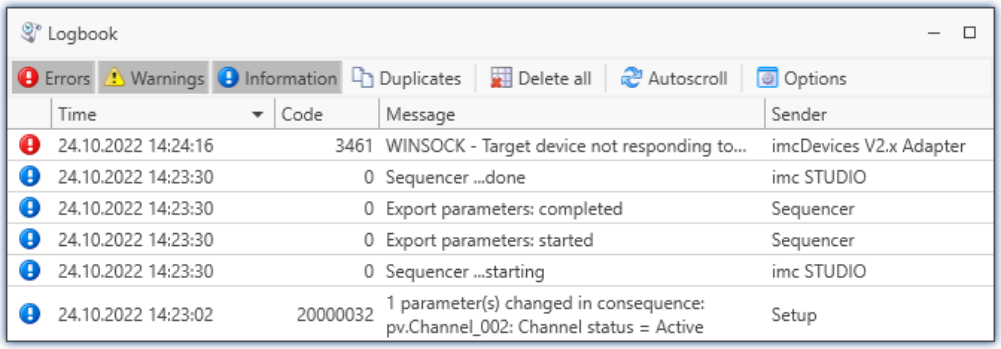
7.4.2 Logbook

In the tool window "*Logbook*", **messages of the categories**: "*Fatal*" (⛔), "*Error*" (❗), "*Warning*" (⚠) and "*Information*" (ℹ) are entered. The logbook entries **indicate what problems and errors exist** and provide hints on where to find them and how to remedy them. Some actions performed are documented.

E.g., a note is entered in the logbook for every command performed:

- performed successfully ("*Information*") or
- otherwise "*Error*" or "*Warning*"

By default, the Logbook is opened when an entry any of the categories "*Fatal*", "*Error*" or "*Warning*" occurs. By default "*Information*" category notes are entered into the logbook without any further effect.



Example of logbook entries

Each logbook entry consists of:

Parameter	Description
Symbol for the category	*Fatal* ( ), *Error* ( ), *Warning* ( ) and *Information* ( )
Time	Time the logbook entry occurred
Code	Logbook entry's error code
Message	Exact description of the logbook entry
Sender	The origin of the logbook entry

After a new imc STUDIO start, the logbook is empty. The logbook itself only displays messages which occurred since the last start. Older messages can be opened with the "*Logbook-Viewer*".

Note — **The logbook is saved**

For purposes of tracing history, the logbook is saved daily:
Default path for Windows 10: **C:\ProgramData\imc\imc STUDIO\Applications_1\log** (example)

Buttons / Context menu	Description
Filtering the logbook	Filtering the Logbook according to categories. To do this, click on the Filter-button: "*Errors*", "*Warnings*" or "*Information*" Errors | Warnings | Information *Logbook filter* Usually, all categories are selected. Filtering also takes effect on new messages Note that the filter has an effect when a new message appears. For this reason, this message may not be displayed. Default setting for the filters You can pre-configure the filter settings using the option 130 ("*Default settings for the filters in the logbook tool window*"). Thus for example, you can have certain information entries hidden by default. Such entries can later be displayed again if needed without affecting the default setting.
Show search panel	Search the logbook for entries and text passages Located in the context menu of the title bar. An input appears. Enter the desired search keyword here. All messages are displayed which contain the text entered. Search also takes effect on new messages Note that the **search works like a filter** and has an effect when a new message appears. For this reason, this message may not be displayed.

Buttons / Context menu	Description
Duplicates	Hiding duplicates In certain cases, the system attempts to perform the same action multiple times in succession. In consequence, multiple duplicate reports are displayed in the logbook. In order to keep the logbook clear and concise, it is possible to group the duplicates. The first and the last messages are always displayed, thus indicating the points in time for the first and last occurrences. In front of the displayed text for the grouped message, a number appears in parentheses. This number indicates how many times the message occurred.
Copy	Copying the selected message to the Clipboard Context menu "*Copy*": All information of the entry that can be seen in the logbook is now in the Clipboard.
Copy with details	Copying the selected message with internal information to the Clipboard For error messages, the stack trace is added to the copied text. This helps our tech support to analyze errors reported to us. Context menu "*Copy with details*": All of the entry's information is then copied to the Clipboard.
Delete all	Deleting the displayed messages "*Delete all*" clears the display Be aware that this only purge the display, but the logbook on the hard drive remains unaffected. As well, it is still possible to view the deleted messages in the "*Logbook-Viewer*".
Autoscroll	Activating/deactivating automatic scrolling When a new message is entered in the logbook, the display in the "*Autoscroll*" mode automatically jumps to the new entry. If you wish to investigate older messages, you can deactivate this mode. This happens automatically when a message is selected, or click on the "*Autoscroll*"-button. Re-activate the mode by: • pressing <Esc> • left-clicking the mouse over the selected entry while holding down the <Ctrl> key • clicking on the "*Autoscroll*"-button in the menu New messages are not automatically displayed visibly Be aware that when the "Autoscroll" mode is deactivated, you will not be automatically notified when new messages appear.
Logbook-Viewer	Starting the Logbook-Viewer and opening old logbooks By means of the "*Logbook-Viewer*", you can look at current, deleted and saved logbook entries. In addition to the familiar logbook-functions, it is possible to load saved logbooks.
Options	Options for the Logbook From within the logbook, you can open the Options for the Logbook 130.
Export	Logbook files are frequently needed by our technical support for error analysis purposes. In response to this button, all existing logbook files are zipped. They can then be sent to our technical support.

Useful options

Options - Logbook	Description
Delete Logbook-files [days]	After the specified number of days elapse, the logbook files are deleted automatically. The deletion verification test is performed upon starting the program and at 0:00 (12 AM).
Default settings for the filters in the logbook tool window	Category of messages for which the filter presets are to be made.
Open logbook in response to a report of category:	For this category, the logbook is opened and attached.

7.5 User administration and Access rights

In some cases, it is mandatory for each user to be able to identify themselves. This ensures that only authorized users can work with imc STUDIO.

Note

- By default, the User Administration is deactivated. In that state, each user has the rights belonging to the group "*imc Administrators*".

In imc STUDIO, a variety of actions are associated with particular user groups. Each user registered is assigned to a particular user group.

Affiliation with one of these user groups determines which imc STUDIO functions are available. For instance, a test engineer could configure experiments and hand them over to technicians to run. If the access rights are configured accordingly, then these users could perform the measurements and check and save the results. However, they would not be able to change the experiments.

The user group has extensive influence on the visibility and/or operability of menus, icons, etc..

Users' privileges are determined according to their respective group association. There are no special rights for an individual user.

The groups are ordered hierarchically (lower > higher):

	User is in the group "**imc Standard Users**"
	User is in the group "**imc Advanced Users**"
	User is in the group "**imc Administrators**"
	User is in the group "**imc Developers**" (only in the product edition: **imc STUDIO DEV**)

Note The user administration is saved with the project

- The User Administration settings are saved with the respective project.
- Changes take immediate effect, but are only saved when the project is saved.

Chapter overview

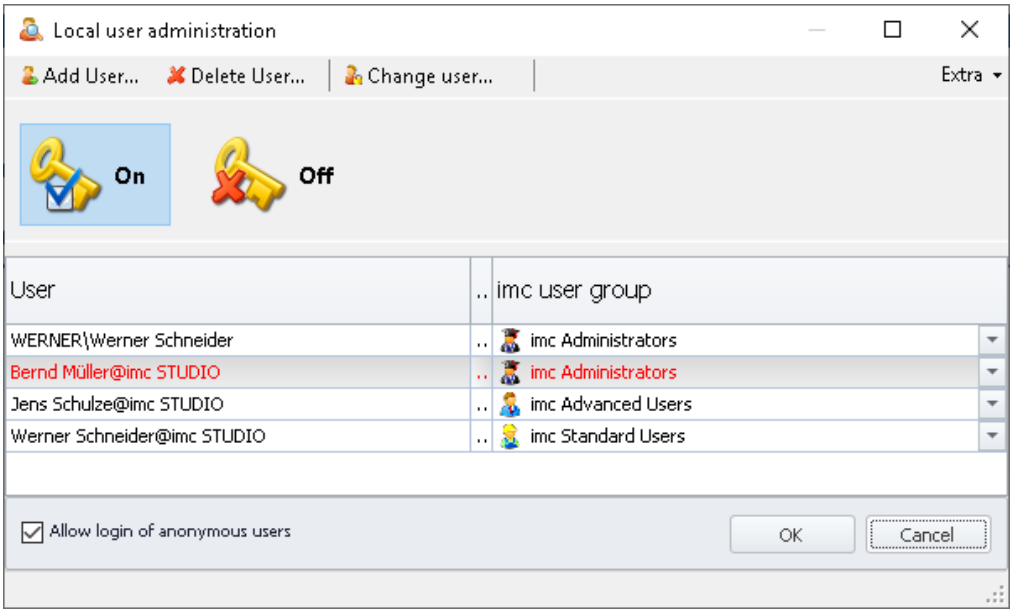
Synopsis	Section
The association of users to user groups is set in the "*User Administration*". Activating / deactivating the User Administration.	User Administration 147
Upon installation, default access rights are assigned. As a "*imc Administrators*", you can make very detailed configurations.	Setting access rights for actions 154
When the program starts, it determines to which user group the user logged into the PC belongs.	Startup behavior 150
You can switch users after the program start.	Logging on, logging off, and switching users 151

7.5.1 User administration

In this dialog you can activate/deactivate the user administration. You can set up users and assign them to the user groups. Thus, each user obtains rights corresponding to their respective group.

Open the dialog by selecting the ribbon menu item "*Administration*".

Ribbon	View
Extra > Administration ( )	Complete



User Administration
Assignment to imc user groups

Note Login is required in order to start the User Administration

- The User Administration settings can only be changed if a user having the associated rights is logged in.
- For this reason, a separate login process is necessary for starting the User Administration.

Activating / deactivating the User Administration

When the User Administration is (being) deactivated, every user has the rights associated with the group "*imc Administrators*".

- To activate the User Administration, click on the Activation-button ( )
- To deactivate the User Administration, click on the Deactivation-button ( )

User types

There are a variety of user-types:

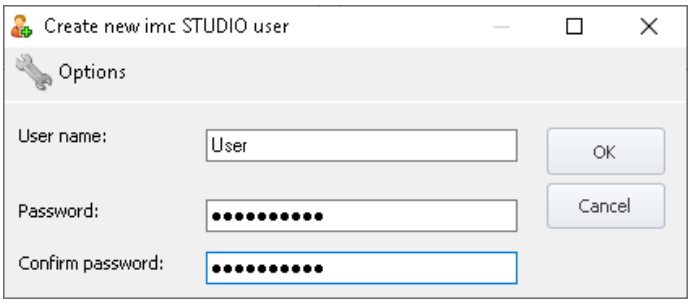
Project-internal imc STUDIO users	Users who can be assigned independently of the particular PC or domain. The user can be protected with a password.
Local computer accounts or groups	Windows or Domain account or group. The group memberships set under Windows or in the domain are not used. The user is protected by the password used for login to Windows 151.
Active Directory domain accounts or groups	

Adding users

Adding a user of the type "Project-internal imc STUDIO user":

- click on the button "*Add user*"
- select the user type "*Create an project-internal imc STUDIO user* ".

The following dialog appears:



Dialog for adding a user of the type "imc STUDIO internal user"

- Enter a user name
- Enter a password and confirm it.
- When the entry is correct, click on "*OK*". The user is added.

The new user's identity is structured as follows: *UserName@imc STUDIO*.

Note Minimum length of user name and password

The minimum length of the user name and of the password can be determined using the options in the title bar.

Adding a user of the type Local computer or Active Directory domain accounts or groups:

- Click on the button "*Add user*"
- Select a user type "*Add a Local Computer account or group*" or "*Add a domain account or group from Active Directory*"
- The Windows dialog for finding an account or group appears; follow its instructions
- Once a correct entry has been made, the user is added.

The new user's identity is structured as follows: *UserName@Complete Computer Name*.

Deleting / removing a user

To delete a user from the list:

- select the user,
- click on the button "*Delete User*",
- confirm the prompt by clicking on "*Yes*" if you have selected the correct user for deletion.

Assigning a user to an user group / Changing the user group

The groups are ordered hierarchically (lower > higher):

	User is in the group "**imc Standard Users**"
	User is in the group "**imc Advanced Users**"
	User is in the group "**imc Administrators**"
	User is in the group "**imc Developers**" (only in the product edition: **imc STUDIO DEV**)

To assign a user to an user group

- open the drop-down list of the column "*imc User Group*" and
- select the desired group

Note

The system must always contain at least one user belonging to the group "*imc Administrators*" or higher.

Anonymous users

Users who are not entered in the User Administration are referred to as "*Anonymous Users*".

If Anonymous Users are permitted, any user can log in with an arbitrary name. This user receives the rights belonging to the user group "*imc Standard Users*".

If, for instance, the PC-user is not set up as a user, at the time that person starts imc STUDIO, they are logged in as an Anonymous (with Startup behaviour 150: "*Windows Session*").

Prohibiting logging in as an Anonymous User

To prohibit Anonymous Users from logging in,

- delete the checkmark next to "*Allow login of anonymous users*".

Export/Import

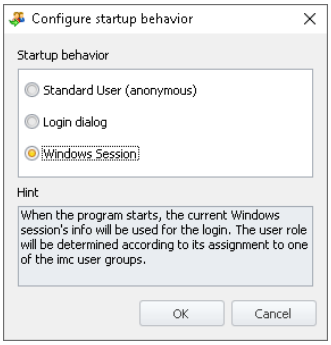
In order to be able to use the up-to-date user administration also in different projects or on other computers, you can export it and then import it at the intended destination.

- click on the button "*Extras*"
- select the desired action: Export or Import

7.5.2 Startup behavior

In this dialog, the startup behavior is set. Open the dialog by selecting the ribbon menu item "*Startup Behavior*".

Ribbon	View
Extra > Startup Behavior ( )	Complete



Startup behavior

Startup Behavior	Description
Standard User (anonymous)	imc STUDIO always starts with the access rights of the group "*imc Standard Users*"
Login Dialog	When starting the program, the Login 151 dialog is displayed.
Windows Session	The program uses the access rights of the user logged into the PC, in accordance with the affiliation with one of the user groups. That is the default setting after installation. For the assignment between the users and the user groups, see the section User administration 147. If the User Administration is activated, any PC-users not belonging to any user group 147 are assigned the rights of the group "*imc Standard Users*".

7.5.3 Logging on, logging off and switching users

The user name serves to identify and authenticate a user. This ensures that only authorized users work with imc STUDIO.

With startup behaviour "*Login Dialog* 150" when imc STUDIO starts the Log On window appears. As well, after imc STUDIO has started, you can switch users

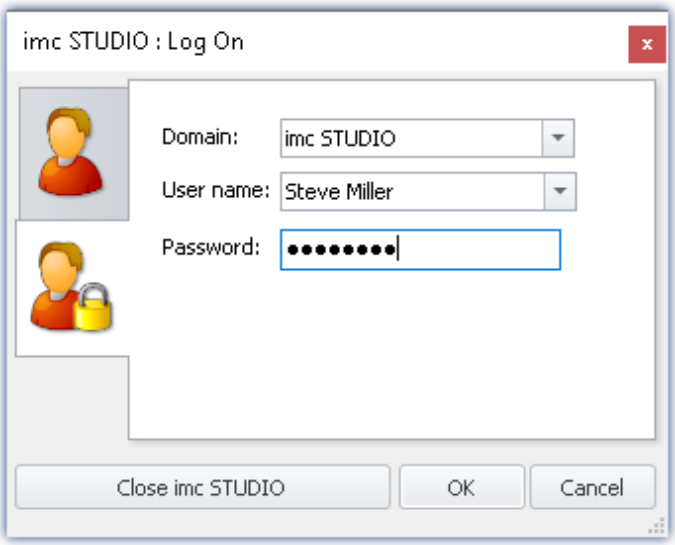
Ribbon	View
Extra > Login/Logout (/)	Complete
Home > Login/Logout (/)	Complete

If the **User Administration is deactivated**, logging on is always anonymous. Every user receives the rights belonging to the group "*imc Administrators*" for the duration of the session.

If the **User Administration is activated** the user can select:

• Log on an anonymous user	Only possible of Anonymous Users are permitted (see "User Administration 147"). In this case, anonymous users have the rights belonging to the group "*imc Standard Users*".
• Log on as registered user	

Logging on a registered user



Login Dialog
Logging on a registered user

Authentication of the registered user upon logging on references:

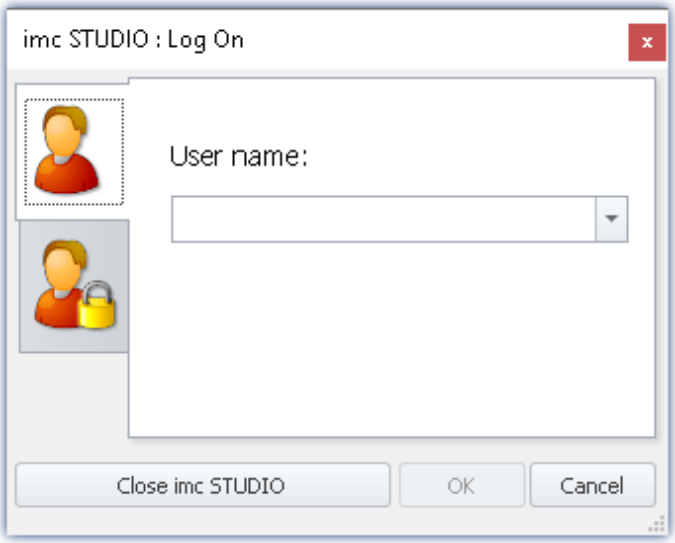
• the domain,	Select the domain: "*imc STUDIO*", "*Computer-name*" or "*Domain-name*"
• the User name and	Select a registered user (all users belonging to the selected domain are listed)
• the associated password.	If required, enter the correct password

Note Windows-Password

- For logged-on users of the PC, imc STUDIO uses the associated Windows password.
- The password entered is never saved by imc STUDIO. The password is transmitted to Windows as hash-code. Windows verifies its validity and returns the result to imc STUDIO.

Logging on an anonymous user

Authentication of the anonymous user references the user name. However, any arbitrary name can be used.



Login Dialog
Logging on an anonymous user

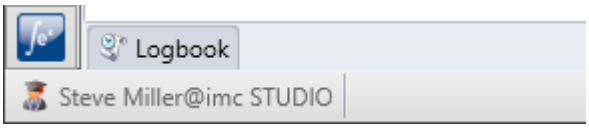
Logging a user off

When the user currently logged on logs off, the Login dialog appears.

The user is logged off. Cancel is not possible.

Status bar

The Status bar at the lower left indicates which user is currently logged on. To the left of the name is the symbol representing the respective user group.

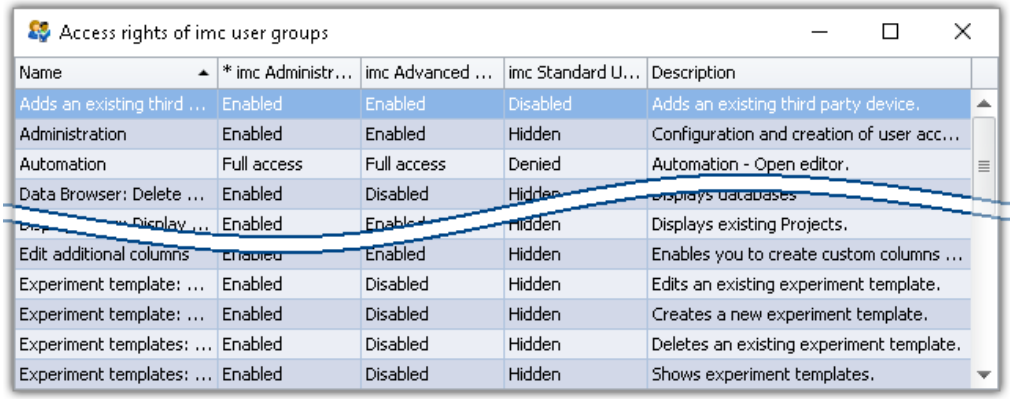


7.5.4 Setting access rights for actions

In the dialog "*Access rights of imc user group*" the rights belonging to the user groups are listed. The access rights can be edited according to the current requirements.

Open the dialog by selecting the ribbon menu item "*Setting Access Rights*".

Ribbon	View
Extra > Setting Access Rights ( )	Complete



Dialog for setting access rights

Any user in a higher group can restrict the rights of a user in a lower group, or assign him higher rights up to his own level. However, one cannot change one's own rights or those of a member of a higher group.

The following rights levels exist:

Rights level	Description
Hidden	not visible (not accessible for operation)
Disabled	visible, but blocked (not accessible for operation)
Enabled	visible and accessible
Denied	
Write protected	visible and accessible for operation by not editable
Full access	no restrictions

For some entries, the rights may differ from the list above.

Note — Rights: Saving an experiment

No changes can be saved unless the user who is logged in possesses experiment saving privileges. If a different experiment is loaded or if the program is closed while a user not having saving privileges is logged in, the changes will be discarded without any message.

Reference — See also:

Panel in fullscreen mode - Protection against changes 1546

7.6 Views

The program's GUI is adaptable, so that for example the windows and columns can be positioned and displayed according to the user's desires. **The layout of the GUI is** saved in so-called "*Views*". It is possible to create separate views for various requirements.

The following settings are saved along with a view:

Region	Description
Window arrangement	The imc STUDIO window's and the tool window's **position and size**
Menus	Layout of the **menu ribbon** and the **Quick access toolbar** 139
Layout	• The last main window opened • The tool window's layout (e.g. what metadata columns are displayed in the Data Browser) • The arrangement and configuration of the Setup pages (e.g. arrangement of columns, such as metadata columns)

Note — Views and projects

The relationship between saving the view and saving the project

The views are saved with their respective project. When a view is saved, it is necessary for the project to be saved. The system will notify the user accordingly.

Conversely, when a project is saved, the view is not automatically saved. Views are only saved whenever the action "*Save View (as)*" is intentionally selected. In order to avoid every little change to a view being saved whenever the project settings are saved, the views are ignored when specifically the project is saved.

Settings which apply globally to all views are saved with the Project

Please note that only the positions of individual elements is saved along with the views. The existence and configuration, for instance, of the Setup-columns (tables- and column descriptions) is saved along with the respective project. (see also "*What is saved where?* 162")

Since the project is always saved whenever the view is saved, the current configuration of the columns is also included when saving.

Overview of settings saved with the project, which affect the views

- Setup table descriptions and column descriptions (these include, for example, additional columns, such as metadata-columns, parameter set columns, ...) (tool window: Table Descriptions)
- saved Setup complete layouts (tool window: Layout Repository)

Note — Detecting changes to a view

If any major changes (columns added, menu changed, etc.) are made, the system explicitly recognizes this as a change to the view. When you exit the project by switching to a different project or by closing imc STUDIO, you are prompted to select whether to save the changes.

Minor changes, such as changed size of tool windows, are ignored.

Saving views

In order to save the current view, select the menu item "*Save View (as)*":

Ribbon	View
View > Save View / Save View as (/)	Complete
Extra > Save View / Save View as (/)	Standard

All settings belonging to the current view are saved. These include the items named above.

Note The views are saved with the project

The views are saved with their respective project. When a view is saved, it is necessary for the project to be saved. The system will notify the user accordingly.

Loading views

In order to load a saved view, select the menu item "*View*" in the drop-down list of the respective view. Here, all views available to the project are displayed.

After the selection the view is loaded.

The record of an experiment includes the view in which the experiment was saved. When the experiment is loaded, the view also loads automatically.

Deleting views

To delete a saved view, select the menu item "*Delete View*".

Ribbon	View
View > Delete View ( )	Complete
Extra > Delete View ( )	Standard

In the dialog which then opens, select the view to delete from the drop-down list. Confirm your selection by clicking on "*Save project*".

Restoring views

Views are not write-protected. They can be restored to their original state. The factory settings come with a copy of all views created as part of the program installation.

To restore a view from the factory settings, select the menu item "*Restore*".

Ribbon	View
View > Restore ( )	Complete
Extra > Restore ( )	Standard

In the dialog which then opens, select the desired views. They are imported from the factory settings. Conform your selection by clicking on "*OK*".

Reference See also

- Import of sensors 123
- Export/import of views 157
- Export/import of Setup table descriptions 273
- Export/import of Setup column descriptions 273
- What is saved where? 162

7.6.1 Exporting/importing views

When you export the view settings, a variety of project settings are stored in the export file.

Settings	Description
Saved views 155	• table descriptions and column descriptions (these include, for example, additional columns, such as "metadata columns", "parameter set columns", ...) (tool window: "*Table Descriptions*") • saved complete layouts (tool window: "*Layout Repository*")
Sensors	• user-defined characteristic curves/sensors (tool window: "*Sensors*")

Export of views, metadata columns, sensors, ...

By means of the menu item "*Import/Export*", you can export the view settings:

Ribbon	View
Project > Import/Export ( )	Complete
Extra > Import/Export ( )	Standard

- Choose option "*Export*" and then the list item "***Export views, meta data columns, sensors, ...***"
- Choose an appropriate file name and path for the view settings file

Note Save the view beforehand

Save the view beforehand if you have made any changes (see "*Views* 156").

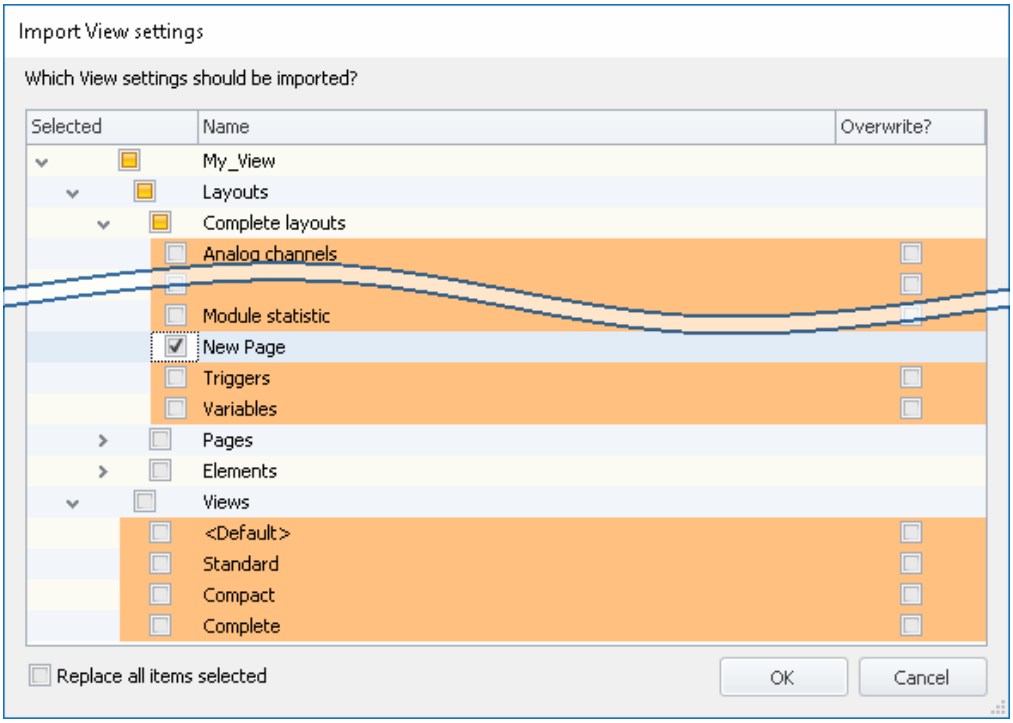
Import of views, metadata columns, ...

Using the menu item "*Import/Export*", you can **import the view settings to the current project**:

Ribbon	View
Project > Import/Export ( )	Complete
Extra > Import/Export ( )	Standard

- Choose option "*Import*" and then the list item "***Import views, meta data columns, ...***"
- In the file selection dialog, selected the saved view settings file

Thereafter you will see a dialog allowing you to import either all or individual view settings.



Entries having **colored shading already exist** in the current project. In this example, the complete layout "*New Page*" is not yet present there.

Checkbox	Description
Selected (left)	Check this box if you wish to **import the element**.
Overwrite? (right)	Check this box if the element in the project is to be **overwritten without any confirmation prompt**. If the box is not check-marked, a prompt appears to inquire the name for the element to be imported. (Unless "*Replace all items selected*" is check-marked.)
Replace all items selected	Corresponds to "*Overwrite?*" for all entries: Check this box if you want to **replace all elements without any confirmation prompt**.

You can select the following elements:

Element	Description
Layouts	All these elements belong to the Setup pages. Saved pages and saved designs, which can be displayed on the pages. For more information on the complete layouts, see the chapter: "*Table Description and Complete Layout* 274".
Views	Her you find the imc STUDIO-views. The views 155 apply for all components of imc STUDIO.

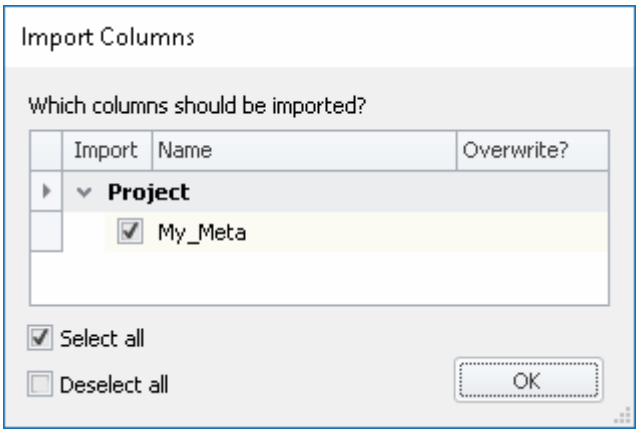
Note — New functions can be overwritten

Be aware that when making replacements, new functions may be lost if you import old views/columns.

Importing defined additional columns

An additional prompt appears if

- discrepancies between existing **Setup columns** were detected or
- the file contains **additional columns**.



Select which columns to import and/or any to replace.

If the **column already exists**, a checkbox appears under "*Overwrite?*". If the box is not check-marked, the **column is automatically renamed**. You can edit the name afterward. For more information, see the chapter: "*Creating and configuring additional columns* 251".

Reference See also

- Import of sensors 123
- Export/import of Setup table descriptions 273
- Export/import of Setup column descriptions 273

7.6.2 Customize ribbon menu

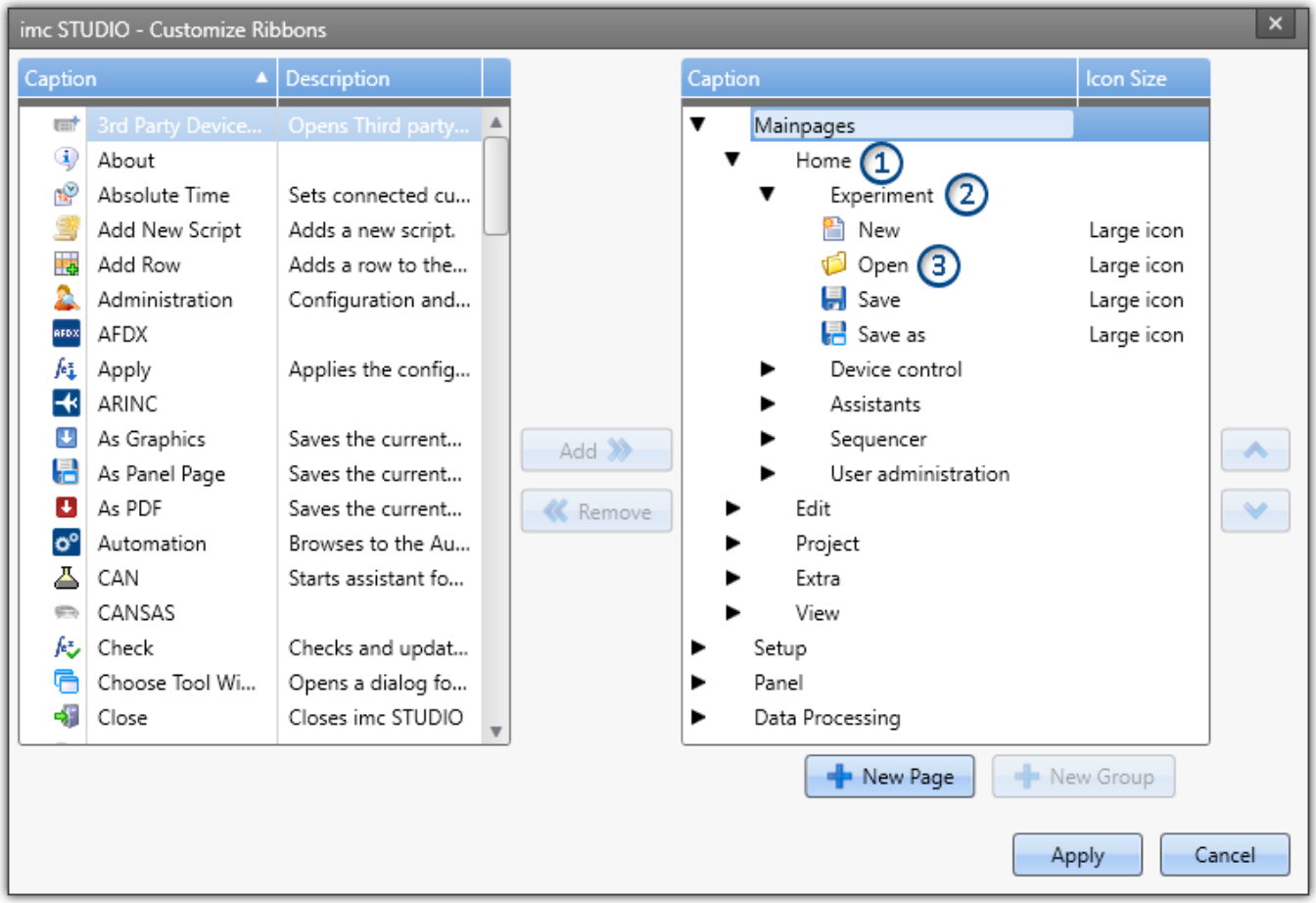
In order to provide quick access to the most important functions, you can customize the menu ribbon. The menu ribbon usually consists of multiple filecards (pages, panes) and multiple groups. The groups contain the menu actions. You can move or delete existing menu actions, or add new ones. Additionally, you are able to create your own custom groups and pages.

Each view has its own menu ribbon configuration. You can modify these configuration and save them along with the associated views.

In order to modify the menu ribbon, open the dialog: "*Customize Ribbon*".

Ribbon	View
View > Customize Ribbon Menu ( )	Complete

If this button is not present in your view, the dialog can also be opened by clicking on the corresponding symbol ( ) in the menu bar (top right).



In the left-side pane, you find all available menu actions. In the right-side pane, the menu actions belonging to the currently selected view are shown. The lists may also include actions which may not be visible in the user interface; e.g., actions only permitted for certain user roles/privileges, or actions which are only activated in conjunction with special product configurations.

Using the buttons: ✚, you can add new **pages** (1) or **groups** (2). To do this, first select the desired location in the right pane. Using the button: »», add the **action** (3) selected in the left pane at below the current position.

Delete the currently selected action, page or group from the list using the «« button.

The buttons ^ and v are used to change the action's position within the group. The **icon-size** in the menu ribbon can be modified in the column at the far right of the right pane, behind the respective action.

Reference Personalized menu items

You have the ability to create your own personal menu items. See:

- a command sequence ("*User-defined buttons* 126")
- a Setup page as a dialog (property: "*Show in Dialog as menu action*" - "*Properties of pages/complete layouts* 275" - Only in imc STUDIO Developer)

7.7 Information and tips

7.7.1 Data management

imc STUDIO uses a "*Database*" for saving projects. The database's path can be freely specified (see "*Experiments, Projects and the Database* 112").

All "**Projects**", "**Experiment templates**", "**Experiments**" and **measured data** are saved to the database.

The following files are created for a sample project:
(Project: "*MyProject*", Experiment template: "*MyTemplate*", Experiment: "*MyExp*")

`MyProject`	Project Name
`+---config`	

`|  |  MyProject.imcAppSettings`
`|  |  project.pcf`
`|  |`
	Project properties

`|  \---templates`
`|  |  MyTemplate.imcStudioTemplate`
`|  |  StandardTemplate.imcStudioTemplate`
`|  |`
	Experiment templates

`|  \---~MyProject`
`|  \---2022-01-18.bak`
`|  MyProject~001.imcAppSettings`
`|`
	Project backup files

`\---MyExp`	Experiment Name

`+---2022-01-19 14-00-00`
`|  129977989468984375.ms.lnk`
`|  Chanel_01.raw`
`|`
`+---2022-01-19 15-00-00`
`|  129977989628330078.ms.lnk`
`|  Chanel_01.raw`
`|`
	Saved measurements (the data structure is set in the device settings under Storage 292)
	• *.raw: measured data
	• *.ms.lnk: link to the appropriate experiment setting

`+---config`	

`|  |  MyExp.imcStudio`
`|  |  MyExp.Prepare_imcStudioArchiv`
`|  |  MyExp.Prepare.disconnect`
`|  |`
	• *.imcStudio: experiment-file
	• *.Prepare*: Necessary in order to be able to re-connect with the running measurement
	• and additional internal files

`|  +---.ms`
`|  |  129977989468984375.ms`
`|  |  129977989628330078.ms`
`|  |`
	Saved experiment (configuration) containing the saved measurements. See:
	• "*Data Browser*" > "*Context menu*" > "*Load measurement settings* 1227"
	• "*imc STUDIO (general)*": "*Options*" > "*Traceability of measurements* 132"

`|  \---~MyExp`
`|  \---2022-01-19.bak`
`|  MyExp~001.imcStudio`
`|  MyExp~002.imcStudio`
`|`
	Experiment's backup files

`\---Meta`	Own files 113 for the experiment, such as metadata, imc FAMOS-sequences, info-files

Note

Please note that not all files are listed here. For the internal administration of the database, several internal files are necessary.

7.7.2 What is saved where?

Every setting, option and parameter is saved. Some are saved with the loaded experiment, some with the project and others with the program. Below is a list of the important areas and where they are saved.

Legend:

Appl:	Application	**Seq:**	Sequencer	**Tab.d:**	Table description
Proj:	Project	**Exp:**	Experiment	**File:**	File system

Name	Description	Appl	Proj	Seq	Exp	View	Tab.d	File
Supplemental files	Files such as characteristic curves for device/ imc Inline FAMOS				Exp			
Selected device					Exp			
Known devices		Appl						
Connected sensors					Exp			
Default values	Setup pre-set values		Proj					
Repository (Automation)								File
Repository (Panel)								File
Sequencer	Sequence table in the Sequencer				Exp			
Sequencer-Events (Sequencer)	Events which apply to the loaded experiment			Seq				
Sequencer-Events (Experiment)	Events which apply to the loaded experiment				Exp			
Sequencer-Events (Project)	Events which apply to all of the project's experiments		Proj					
User Administration	Defined users		Proj					
Access rights	Privileges of user roles		Proj					
Show tabs (Panel-Fullscreen)	Visibility of the checkbox "Show tabs" in the Panel-Fullscreen		Proj					
Options (experiment)					Exp			
Options (project)			Proj					
Options (application)		Appl						
Views	window arrangements, menus, layouts, ...		Proj					
Menu ribbon (customizing)	Which button appears where					View		

Name	Description	Appl	Proj	Seq	Exp	View	Tab.d	File
Quick access toolbar	Which button appears where					View		
Column arrangement	What columns are displayed where					View		
User defined buttons (collection)	What buttons exist and their significance		Proj					
User defined buttons (customizing)	Which button appears where		Proj					
Table descriptions	Aggregation of the normal and additional columns		Proj					
Normal columns	What columns exist and their significance						Tab.b	
Additional columns	What columns exist and their significance						Tab.b	
Complete layouts	Completely designed pages		Proj					

7.7.3 Backup Copies – Data security mechanism for experiments

Each time the imc STUDIO experiment is saved, a backup copy of the previous "save-state" is also recorded. The following data storage depth arrangements are implemented:

- Per each calendar day, a maximum of the last 10 versions is saved.
- A maximum of 5 calendar days are saved.
- From the 3rd and previous calendar days, only the respective last version is saved.

Storage location - Backup-folder

The backup copies are located in a backup-folder within the Experiment-folder. There you will find a folder whose name has a prefixed tilde-character: "*~<ExperimentName>*". E.g. "*~Experiment_001*".

Note The Backup-folder is hidden

This Backup-folder is hidden. However, it is possible to enter its path into the address bar of the Windows Explorer in order to open the folder. Or, you can have the hidden folders displayed.

The Backup-folder contains one associated sub-folder for each work day: e.g. 2018-05-17.bak, 2018-05-16.bak, ...

These each contain the most recent saved states of the experiment. Below is an example with three and four saved versions.

MyExperiment\config\~MyExperiment\2018-05-16.bak\
- MyExperiment~001.imcStudio
- MyExperiment~002.imcStudio
- MyExperiment~003.imcStudio

MyExperiment\config\~MyExperiment\2018-05-17.bak\
- MyExperiment~001.imcStudio
- MyExperiment~002.imcStudio
- MyExperiment~003.imcStudio
- MyExperiment~004.imcStudio

Restoring a backup version

To restore a saved version, import the .imcStudio-file via the Project-dialog 115. First copy the file, for instance, to the Desktop, and rename it in a convenient way, e.g. MyExperiment~004.imcStudio -> MyExperiment.imcStudio.

Note Overwriting during import deletes the backups

Be aware that upon importing, the original experiment is overwritten if you give it exactly the same name. All additional backups are also deleted in consequence.

7.7.4 Guardian

Along with imc STUDIO, two system services are also installed: Guardian and WatchDog. These two services start upon starting the system.

WatchDog (GuardianWDService)

The WatchDog has the sole purpose starting the Guardian if it is not already running.

Guardian (GuardianService)

The Guardian handles a variety of tasks.

Task	Description
Preparing the database for imc STUDIO	The Data Browser displays the results of any measurements performed. When a measurement is run, the system informs imc STUDIO that a new measurement needs to be displayed. However, if any additional data sets are added at a later point in time, e.g. by means of a imc FAMOS-sequence, then the program needs to be notified of this. The Guardian monitors the folder and informs the program about every file which has been added. In consequence, these files are also available to you in the Data Browser.
Deleting elements from the database	The Guardian handles the deletion of elements from the database, e.g. measurements or whole experiments. For this purpose, imc STUDIO creates a ".deleteMe"-file in the respective folder. Subsequently the Guardian deletes the folder and its contents once the system is ready to do so. Example: Interval data saving with a limited measurement count is activated. In the measurement folders to be deleted, imc STUDIO creates a ".deleteMe"-file. The Guardian deletes the measurement as soon as the system give clearance to delete it, for instance, as soon as the measurement is no longer loaded.

Note — Unable to establish a connection with the Guardian process

imc STUDIO always monitors the service's accessibility so that you will be notified in case the Guardian is no longer running, or even if it had not started. If the Guardian is unable to respond for any reason, a corresponding message is posted in the logbook: "*Unable to establish a connection with the Guardian process.*"
If the Guardian becomes accessible again at a later point in time, you receive an additional message: "*The connection with the Guardian process has been restored.*"

Closing services

In some cases, the services are not desired and need to be closed. In this case, please first close the WatchDog-service and subsequently the Guardian-service. Be aware that the Guardian's tasks will no longer be performed; e.g. any interval limitation will no longer be applied.

Versionen

There is a separate Guardian and a separate WatchDog for the 64-bit version. In case of parallel installation with a Version 5 system, two services are running. The 64-bit variant has the suffix "*V2*".

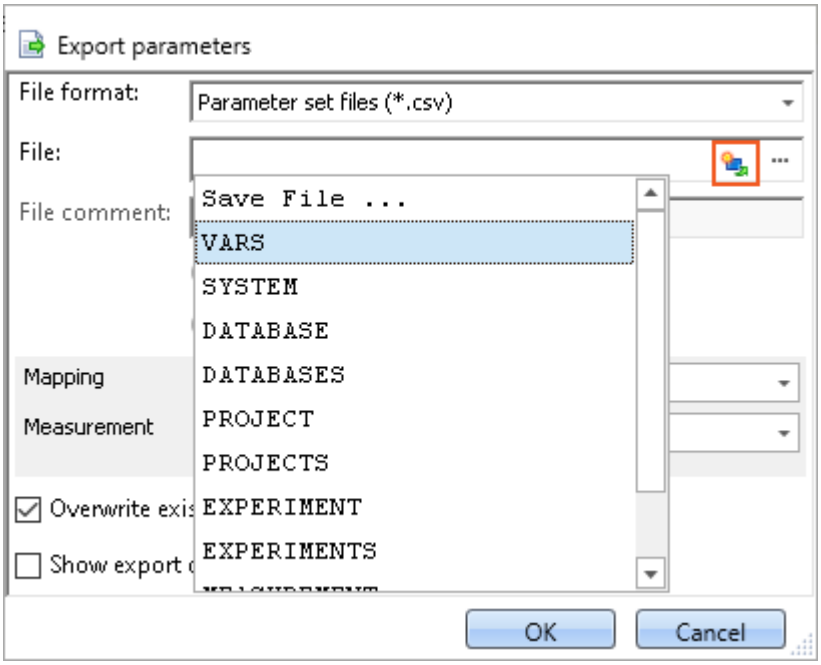
7.8 Placeholders

In imc STUDIO , you can use placeholders at various positions, which are **automatically recognized and resolved**.

With the help of placeholders, you can, for example, record a parameter set in the current experiment folder without knowing the path, or also display the variable's value in a text.

Placeholders can be used both in some commands as well as with Widgets. A list of all known placeholders is presented in the section "*list of placeholders*".

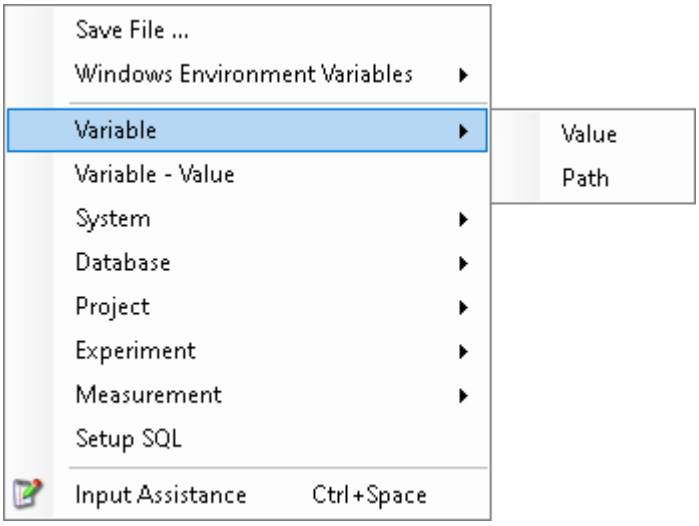
Placeholders are offered where the placeholder symbol ( ) is displayed. Not all placeholders can be used at all positions. To obtain a list of available placeholders, click the left mouse button over the placeholder symbol ( )



List of available placeholders for the comand "Export parameters"

Along with clicking the mouse over the placeholder symbol, there are other possibilities in the input box for having the list of available placeholders displayed:

- by means of the key combination CTRL + Space
- by means of the context menu (right mouse button)
- in the context menu by means of the item "*Input Assistance*"



Input box context menu

There is also the possibility of entering the desired placeholder entirely manually; however, it is not possible to see whether the placeholder is supported at this location.

In the next sections, the following information is available:

- Listing and description 167 of known placeholders
- Formatting 182 of the placeholders
- Environment variables of the OS 185 supported by imc STUDIO

7.8.1 List of placeholders

The following placeholders are described in this section:

- CONTROL 168
- DATABASE/DATABASES 169
- EXPERIMENT/EXPERIMENTS 170
- MEASUREMENT 173
- MEASUREMENTS 174
- PAGE 176
- PANEL 176
- PROJECT/PROJECTS 176
- SELCONTROL 178
- SETUP 179
- SYSTEM 180
- VAR/VARS 181

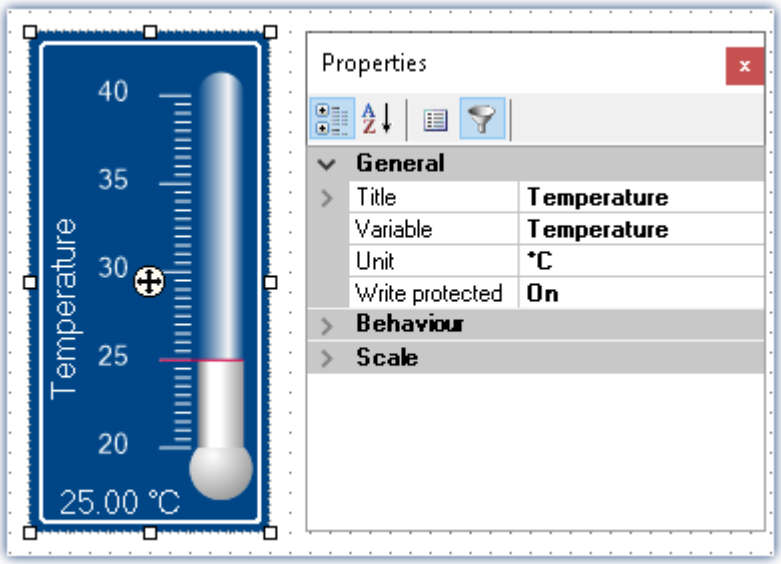
7.8.1.1 CONTROL

These placeholders are only available for Widgets.

Name	Context menu	Description
UNIT	Control - Unit	Unit of the current Widget, corresponding to the box Unit in the Properties-dialog. If there is no Unit, or if it is empty, the unit of the linked variable is adopted.
TITLE	Control - Title	Title of the current Widget, corresponding to the box *Title* in the *Properties*-dialog
VALUE	Control - Value	Current value of the variable, which is linked with the current Widget.

Example

The following example illustrates what the three placeholders mean.



Properties of a Widget

In this example, the placeholders listed above have the following values:

- <CONTROL.UNIT> = °C
- <CONTROL.TITLE> = Temperature
- <CONTROL.VALUE> = 25

7.8.1.2 DATABASE/DATABASES

This placeholder is available everywhere where filepaths are needed, e.g. for export and import commands.

Name	Context menu	Description
NAME	Database - Name	Name of the database (= name of the database folder)
PATH	Database - Path	Path to the DB folder

Here, the entries are used which are entered in the *Global Options*.

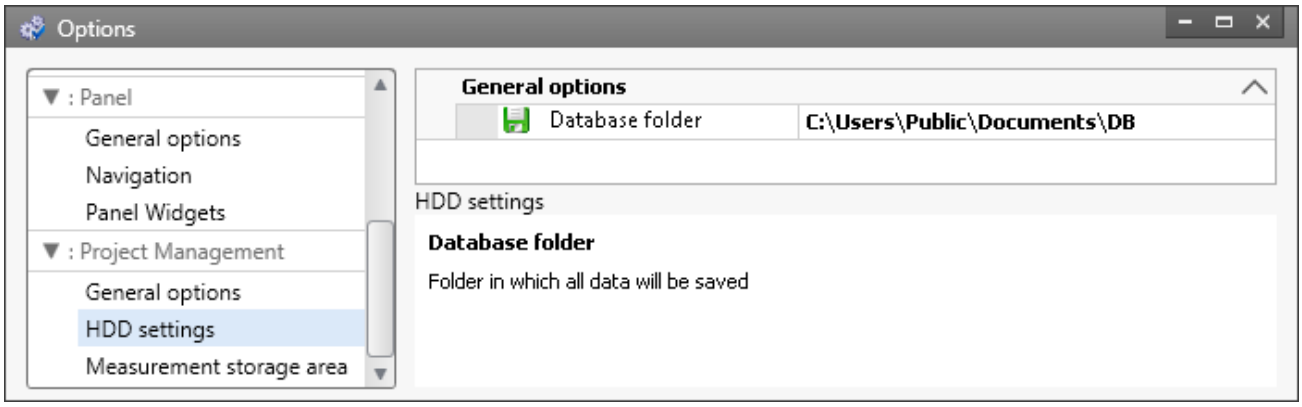
Note

The placeholder *DATABASE* always pertains to the current project database. If you wish to use the placeholder with another database, the placeholder *DATABASES* must be used, e.g.:
<DATABASES["Another_Database"].PATH>

The workings of the two placeholders is illustrated by the following to examples.

Example 1

The first example pertains to the unchanged imc STUDIO default settings, here under Windows 10:

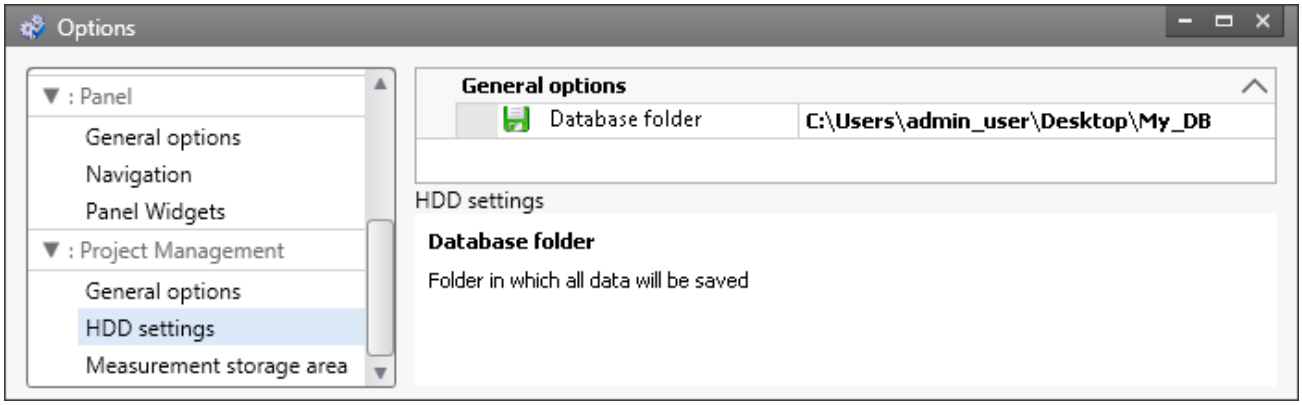


Default database path under Windows 10

- <DATABASE.NAME> = DB
- <DATABASE.PATH> = C:\Users\Public\Documents\DB

Example 2

The second example shows the resolved placeholders when the database path is changed:



Changed database path

- <DATABASE.NAME> = My_DB
- <DATABASE.PATH> = C:\Users\admin_user\Desktop\My_DB

To obtain the same values by using the placeholder DATABASES, the call must be structured as follows:

- <DATABASES["My_DB"].NAME> = My_DB
- <DATABASES["My_DB"].PATH> = C:\Users\admin_user\Desktop\My_DB

7.8.1.3 EXPERIMENT/EXPERIMENTS

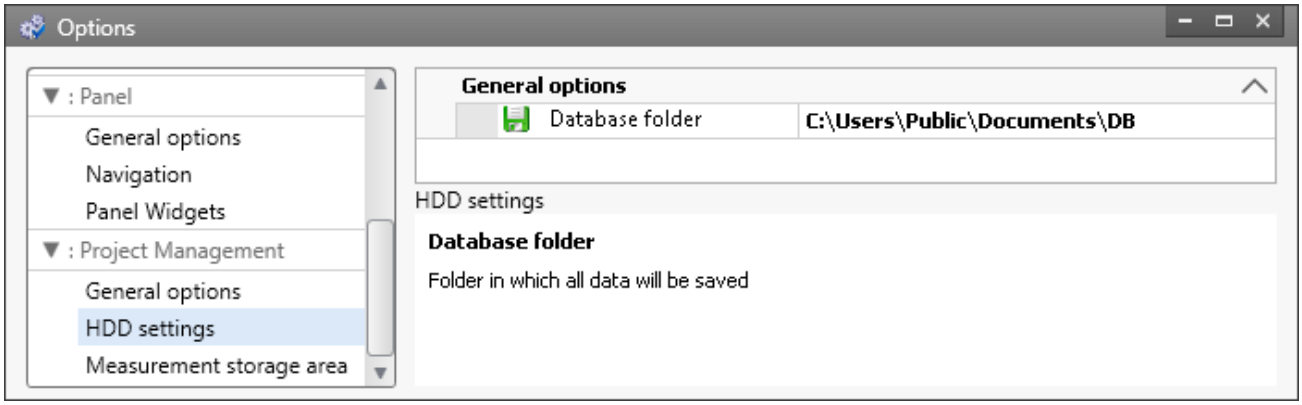
Name	Context menu	Description
CFG	Experiment - Configuration path	Path to the configuration folder ("*config*" folder) of the experiment
NAME	Experiment - Name	Name of experiment
PATH	Experiment - Path	Path to the experiment's saving folder
SETTINGS	Experiment - Configuration file	Experiment file including path (*.imcStudio)

Note

The placeholder *EXPERIMENT* always pertains to the current experiment. If you wish to use the placeholder with another experiment, the placeholder *EXPERIMENTS* must be used, e.g.:
<EXPERIMENTS["Another_Experiment"].PATH>

Example

The example is based on unchanged imc STUDIO default settings, here under Windows 10:



Default database path under Windows 10

The project's name is *StandardProject,* the experiment is called *My_Experiment*.
With these settings, the placeholders listed above have the following values:

Input	Result
<EXPERIMENT.CFG>	C:\Users\Public\Documents\DB\StandardProject\My_Experiment\config
<EXPERIMENT.NAME>	My_Experiment
<EXPERIMENT.PATH>	C:\Users\Public\Documents\DB\StandardProject\My_Experiment
<EXPERIMENT.SETTINGS>	C:\Users\Public\Documents\DB\StandardProject\My_Experiment\config\My_Experiment.imcStudio

To obtain the same values by using the placeholder "*EXPERIMENTS*", the call must be structured as follows:

Input	Result
<EXPERIMENTS["My_Experiment"].CFG>	C:\Users\Public\Documents\DB\StandardProject\My_Experiment\config
<EXPERIMENTS["My_Experiment"].NAME>	My_Experiment
<EXPERIMENTS["My_Experiment"].PATH>	C:\Users\Public\Documents\DB\StandardProject\My_Experiment
<EXPERIMENTS["My_Experiment"].SETTINGS>	C:\Users\Public\Documents\DB\StandardProject\My_Experiment\config\My_Experiment.imcStudio

Note Deviating System Behavior

Deviating system behavior in absence of the component: imc STUDIO Project Management:

If the component imc STUDIO Project Management is deactivated, no database is used to manage the experiments jointly. For this reason, the placeholder returns different results.

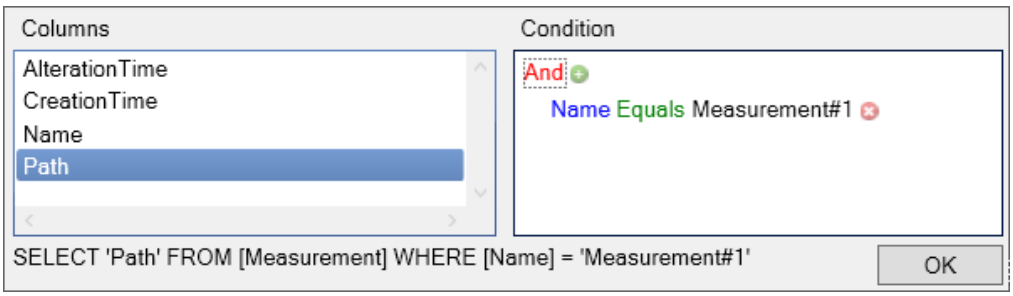
Name	Context menu	Description
CFG	-	*Not supported.*
NAME	-	*Not supported.*
PATH	Experiment - Path	Path to the storage directory of the configuration file ".imcStudio"/".imcExp"
SETTINGS	-	*Not supported.*

7.8.1.4 MEASUREMENT

With this placeholder, you can access properties and metadata belonging to a saved measurement, such as the directory path, or metadata saved along with the measurement.

Name	Context menu	Description
SQL	Measurement SQL	Access to data of a saved measurement

The placeholder reads properties and metadata belonging to the saved measurements. For this purpose, commands in the database language SQL are used. When the placeholder is entered, an assistant opens automatically, which handles the entering of the SQL command.



MEASUREMENT.SQL-Assistant

In the assistant, there is a choice among various **columns**, which can be associated with a **condition**:

Parameter	Description
Columns	Here, all available properties and metadata are displayed. Select the desired property; multi-selection is possible using the CTRL-key. In the default case, where there is no additional export of data, you can select among the following properties: • the last time the measurement was modified (AlterationTime), • the time the measurement was created (CreationTime), • the measurement's directory path (Path) and • the measurement's name (Name). This always returns the fixed name. If any metadata with the measurement were saved, they are listed here.
Condition	Here you specify which measurement's selected properties and metadata you wish to see. For example, in order to always see the metadata belonging to measurement #1, proceed as follows: • Click on the , which causes the condition to appear • Next, click on "AlterationTime" and select "Name". • Subsequently, select "Measurement#1" from the selection list behind "equals". If the condition is left empty, the properties/metadata of all measurements are displayed in succession (separated by ',‘).

Note Columns and Condition

In the **Columns**, select the **properties/metadata**, which you wish **to display**. In the **Condition**, specify **which measurement's properties/metadata** you wish to use.

Example

The selection made in the image above returns the directory path of the measurement which is currently designated as #1 (in the Data Browser).

7.8.1.5 MEASUREMENTS

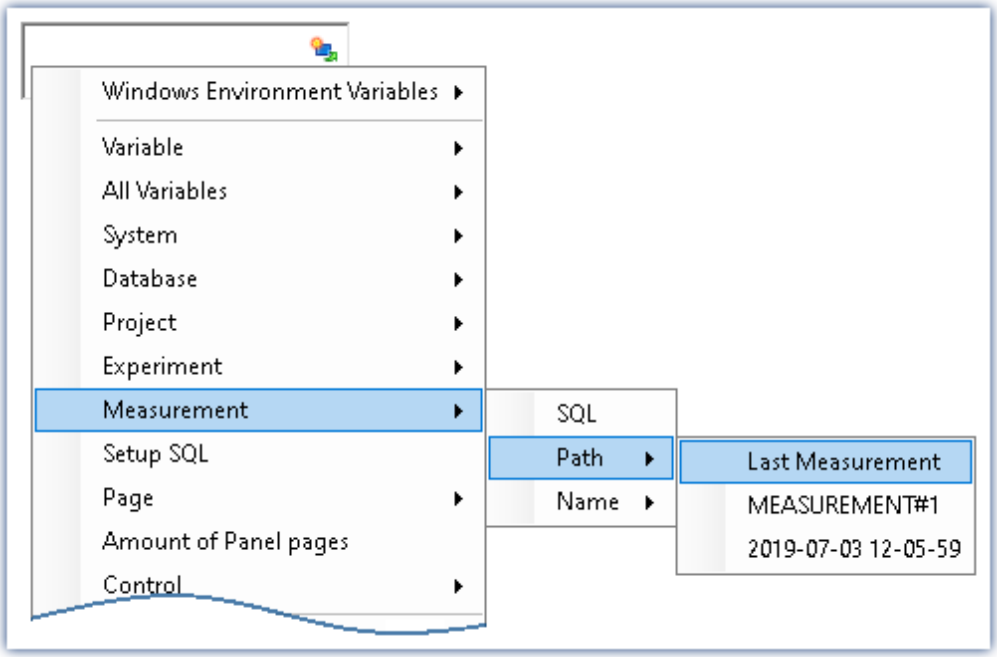
With this placeholder, you can get the path or name of a measurement; e.g. the folder path to the last saved measurement.

Name	Context menu	Description
PATH	Measurement - path	Returns a measurement's folder path
NAME	Measurement - name	Returns a measurement's name

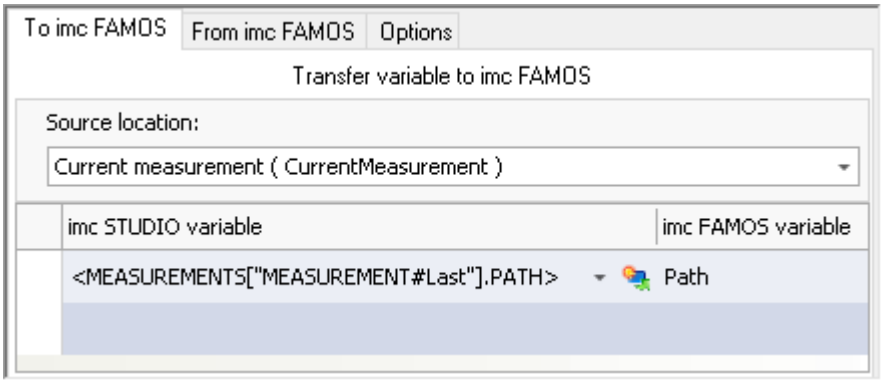
The following MEASUREMENTS-placeholders are available:

Placeholder	Description
<MEASUREMENTS["MEASUREMENT#LAST"].PATH>	Returns the data storage path to the measurement last saved.
<MEASUREMENTS["MEASUREMENT#LAST"].NAME>	Returns the name of the measurement last saved.
<MEASUREMENTS["MEASUREMENT#<Measurement number>"].PATH>	Returns the data storage path of the measurement having the respective measurement number; e.g., you can select a measurement in the Data Browser. For example, it can be assigned the number "1". You now have the ability to find the path to this measurement by means of the placeholder.
<MEASUREMENTS["MEASUREMENT#<Measurement number>"].NAME>	Returns the name of the measurement having the respective measurement number. For an example, see "PATH".
<MEASUREMENTS["<Measurement name>"].PATH>	Returns the data storage path of the measurement having the respective name.

Currently, you can't obtain the placeholder via the Placeholder-symbol, since there is no input support. Instead, use the context menu. In this menu, under "*Measurement*" there are various examples which you can select. Subsequently, you can modify the text according to your requirements.



Creating the placeholder via the context menu



Example: Transferring the data storage path to imc FAMOS

Reference The last measurement

More detailed information on the "last measurement" is presented in the chapter:
"*Storage options and directory structure*" > "*The last measurement* 441"

7.8.1.6 PAGE

This placeholder is only available for the Panel.

Name	Context menu	Description
NUMBER	Page - Number	Number of the current Panel page
VARS	Page - Variables	All variables which are linked with at least one Widget on the current page

Example

In the Panel, the following pages are generated:

- Start
- Meas
- Report

Existing pages in the Panel

For instance, if the placeholder is used on the page *Start*, then <PAGE.NUMBER> = 1, on the page *Report*, <PAGE.NUMBER> = 3.

If, for instance, a curve window is linked with *Channel_001* on the page *Start* and a standard meter is linked with the user-defined variable *Test*, then <PAGE.VARS> = Channel_001,Test.

7.8.1.7 PANEL

This placeholder is only available for the Panel.

Name	Context menu	Description
PAGECOUNT	Amount of Panel pages	Total number of pages in the Panel

Example

In the Panel, the following pages are generated:

- Start
- Meas
- Report

In which case, <PANEL.PAGECOUNT> = 3.

7.8.1.8 PROJECT/PROJECTS

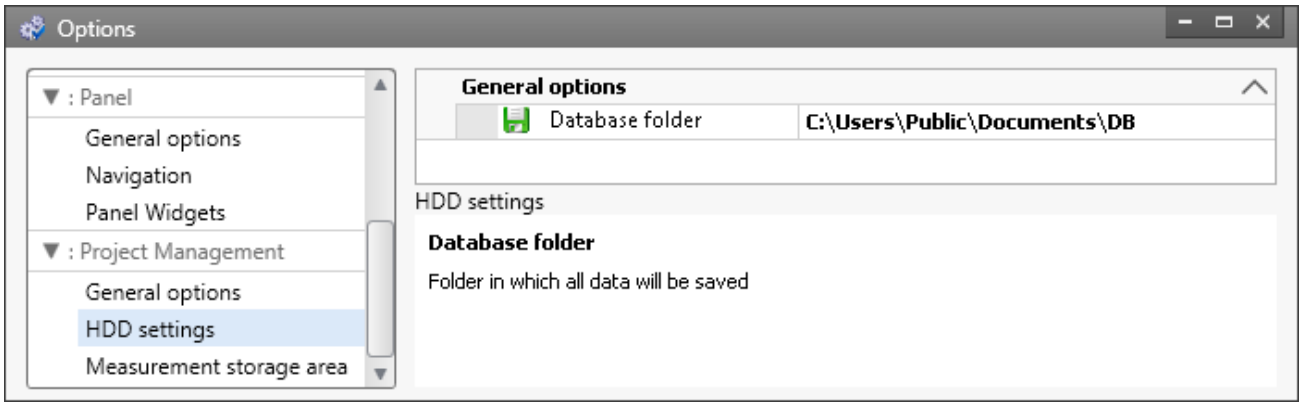
Name	Context menu	Description
CFG	Project - Configuration path	Path to the project's configuration folder (*config* folder)
NAME	Project - Name	Name of the project
PATH	Project - Path	Path to the project's saving folder
SETTINGS	Project - Configuration file	Settings for the project (*.imcAppSettings file)

Note

The placeholder *PROJECT* always pertains to the current project. If you wish to use the placeholder with another experiment, the placeholder *PROJECTS* must be used, e.g.: <PROJECTS["Another_Project"].PATH>

Example

This example is based on unchanged imc STUDIO, default settings, here under Windows 10:



Default database path under Windows 10

The project's name is *StandardProject.*

With these settings, the placeholders listed above have the following values:

Input	Result
<PROJECT.CFG>	C:\Users\Public\Documents\DB\StandardProject\config
<PROJECT.NAME>	StandardProject
<PROJECT.PATH>	C:\Users\Public\Documents\DB\StandardProject
<PROJECT.SETTINGS>	C:\Users\Public\Documents\DB\StandardProject\config\StandardProject.imcAppSettings

To obtain the same values using the placeholder *PROJECTS*, the call must be structured as follows:

Input	Result
<PROJECTS["StandardProject"].CFG>	C:\Users\Public\Documents\DB\StandardProject\config
<PROJECTS["StandardProject"].NAME>	StandardProject
<PROJECTS["StandardProject"].PATH>	C:\Users\Public\Documents\DB\StandardProject
<PROJECTS["StandardProject"].SETTINGS>	C:\Users\Public\Documents\DB\StandardProject\config\StandardProject.imcAppSettings

7.8.1.9 SELCONTROL

This placeholder is only available for the Panel.

Name	Context menu	Description
VARS	Variables of the selected control	List of all variables linked with the selected Widget

Example

A curve window is linked with the following channels:

- Channel_001
- Channel_002
- Channel_003

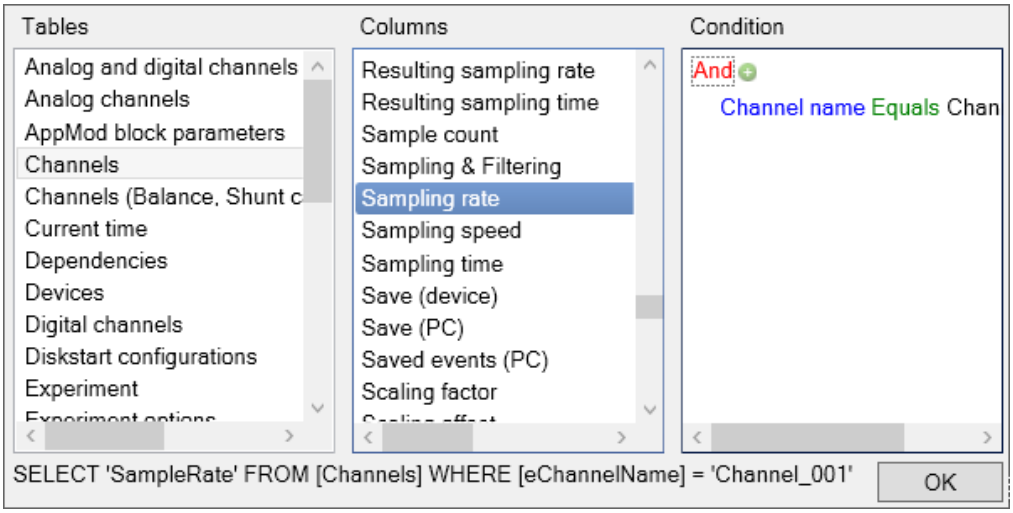
In which case, <SELCONTROL.VARS> = Channel_001,Channel_002,Channel_003.

7.8.1.10 SETUP

By means of this placeholder, you can access data from the Setup.

Name	Context menu	Description
SQL	Setup SQL	Access to current data from the Setup

The placeholder reads current parameter values from the Setup. Therefor, commands expressed in the SQL database language are used. When entering the placeholder, a wizard is automatically opened, which handles making the entries for the SQL command:



SETUP.SQL wizard

The wizard displays the following columns:

Parameter	Description
Tables	Here all table descriptions are listed, select the one desired. In order for the "*SETUP*"-placeholder to be able to get the current values from the Setup, it must query the table description. For more details on table descriptions and Setup pages, please observe the reference below.
Columns	Once your table description has been selected, all associated columns (parameters) which are available appear here. Select the parameter desired; it is also possible to select multiple parameters by using the CTRL-key.
Condition	Specify any condition here, for example a particular channel name, if desired. If the condition is left empty, all values belonging to the selected column, e.g. the sampling rates of all channels, are displayed.

Example

The selection which was made in the picture above returns the sampling rate of the channel "Channel_001" as a text string, e.g. "100 Hz". If the condition which had been entered is deleted, the full amount of sampling rates would be returned as an enumeration, e.g. "100 Hz, 1 kHz, 100 Hz,100 Hz, 1 kHz, , ,100 Hz, ...". Elements which do not possess any sampling rate, return an empty element (", ,").

Reference

Information on the relationship between table descriptions, complete layouts and Setup pages is provided in the section"*Table description and complete layout* 274".

Since metadata such as entries on the pages "*Documentation*", "*Annotation before measurement*" and "*Annotation after measurement*" belong to the Setup, these can also be displayed in the Panel or used in commands by this means.

Note

The placeholder reads by the SQL command **always the current value** from the Setup, i.e. the value is updated as soon as any change is made in the Setup.

7.8.1.11 SYSTEM

This placeholder enables the current date and the system's time to be used.

Name	Context menu	Description
DATE	System - Date - Standard	The operating system's current date. Format M/d/yyyy for English (USA) operating systems
	System - Date - Path suitable	The operating system's current date, format yyyy-MM-dd
TIME	System - Time - Standard	The operating system's current time, format hh:mm for English (USA) operating systems
	System - Time - Path suitable	The operating system's current time, format hh-mm-ss

Explanations of the formatting of the date 183 and the time 183 are provided in the section Formatting 182.

Example

Supposing the time June 14th, 2011, 2:30 p.m. is set in the (English(USA)) operating system. Then the two placeholders return in their default formatting:

- <SYSTEM.DATE> = 6/14/2011
- <SYSTEM.TIME> = 2:30 PM

The remaining formatting options are described in the section Formatting 182.

7.8.1.12 VAR/VARS

Name	Context menu	Description
COMMENT	Variable - Comment	Comment on the variable, e.g. the channel comment
FILE	Variable - File	File, in which the variable is saved
NAME	Variable - Name	Name of the variable
PATH	Variable - Path	Path in which the VARS.FILE is saved
PROPS	Variable - Properties	Several variable properties, e.g. category or metadata, which have been written into channel data.
UNIT	see YUNIT	see YUNIT
VALUE	Variable - Value	Current value of the variable, of a channel or a bit
XUNIT	Variable - X-unit	X - unit of the variable, e.g. s for seconds
YUNIT	Variable - Y-unit	Y - unit of the variable
YUNIT2	Variable - Y-unit2	Y-unit of the 2nd component (only relevant with complex data sets)
ZUNIT	Variable - Z-Unit	Z - unit of the variable

Note

The placeholder *VAR* always refers to the variable which is linked with the Widget. If you wish to access a different variable via the placeholder, the placeholder *VARS* is to be used, e.g.: <VARS["myVariable"].PATH>.

Example

A vibration in "*µ eps*" is measured over time (seconds). In imc Online FAMOS, an FFT of the input signal is performed.
The resulting virtual channel is saved together with the measurement channel on the PC.
Data storage settings:

- Path: C:\Users\Public\Documents\DB\StandardProject\My_Experiment
- Storage interval: End of measurement

The virtual channel's name in this example is "*myFFT*" and the comment is called "*Result of FFT*".
Thus, the placeholders listed above are resolved after the first measurement as follows:

Input	Result
<VARS ["myFFT"].COMMENT>	Result of FFT
<VARS["myFFT"].FILE>	myFFT.RAW
<VARS["myFFT"].NAME>	myFFT
<VARS["myFFT"].PATH>	C:\Users\Public\Documents\DB\StandardProject\My_Experiment\2019-12-18 09-37-14
<VARS["myFFT"].UNIT>	µ eps
<VARS["myFFT"].XUNIT>	s
<VARS["myFFT"].YUNIT>	µ eps
<VARS["myFFT"].ZUNIT>	Hz

<VARS["myFFT"].VALUE> is resolved, however in this example (FFT-calculation), it does not return a sensible value. Notes on formatting the placeholder VALUE are found under "*Formatting - VALUE* 184".

7.8.2 Formatting

With some placeholders, you have the possibility of specifying the formatting desired. The various formatting options are described as examples.

Usually, manual entry of the formatting string is necessary. The placeholder "*VALUE*" offers you support in making entries.

A formatting is possible with the following placeholders:

- PAGE.NUMBER/PANEL.PAGECOUNT 182
- SYSTEM.DATE 183
- SYSTEM.TIME 183
- VALUE 184

7.8.2.1 PAGE.NUMBER/PANEL.PAGECOUNT

The placeholders PAGE.NUMBER and PANEL.PAGECOUNT are integers where the amount of digits can be set.

If, for instance, <PANEL.PAGECOUNT> has the value 3, then this value may be outputted as follows:

- <PANEL.PAGECOUNT("00")> = 03
- <PANEL.PAGECOUNT("000")> = 003

These formatting strings can be used for <PAGE.NUMBER> in the same way.

7.8.2.2 SYSTEM.DATE

In the formatting string for the date, the following components can be used:

- d or dd = current date
- M or MM = current month
- yy or yyyy = current year
- Separator e.g. '.', ',', '-', '_'

Note

To use the placeholder when specifying the path as a string, use the separator "-". This character is allowed as a special character in all languages in pathnames.

Example

With the example of the date July 14th, 2019 (operating system English (USA)), the use of the formatting string is illustrated.

- <SYSTEM.DATE> = 7/14/2019
- <SYSTEM.DATE("yyyy-MM-dd")> = 2019-07-14
- <SYSTEM.DATE("yy-M-d")> = 19-7-14
- <SYSTEM.DATE("dd_MM_yy")> = 14_07_18

7.8.2.3 SYSTEM.TIME

In the formatting string for the time, the following components can be used:

- h or hh = current hour; 12h
- H or HH = current hour; 24h
- m or mm = current minute
- s or ss = current seconds
- Separator e.g. '.', ',', '-', '_'

Note

To use the placeholder when specifying the path as a string, use the separator "-". This character is allowed as a special character in all languages in pathnames.

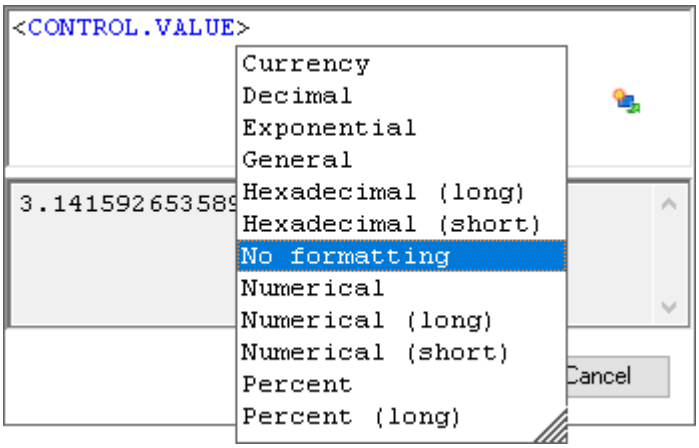
Example

With the example of the time 1:05:03 p.m. (operating system English (USA)), the use of the formatting string is illustrated.

- <SYSTEM.TIME> = 1:05
- <SYSTEM.TIME("hh-mm-ss")> = 01-05-03
- <SYSTEM.TIME("h-m-s")> = 1-5-3
- <SYSTEM.TIME("HH_mm_ss")> = 13_05_03

7.8.2.4 VALUE

Formatting for VALUE can either be entered by means of the input assistance (see figure below) or manually.



Input assistant for the formatting

The various formatting possibilities are explained in the following table on the basis of the variable pi=3.1415926535. The amount of digits before and after the decimal specified here can be replaced with any desired integer amount.

Description	Input assistant	Manual	Output
No formatting	No formatting	<VARS.["pi"].VALUE>	3.14159
Fixed point, 2 decimal digits	Numeric	<VARS.["pi"].VALUE("f2")> <VARS.["pi"].VALUE("0.00")>	3.14
Fixed point, 1 decimal digit	Numeric Short	<VARS.["pi"].VALUE("0.0")>	3.1
Fixed point, 6 decimal digits	Numeric Long	<VARS.["pi"].VALUE("0.000000")>	3.141593
Floating point, exponential notation, 2 decimal digits	Exponential	<VARS.["pi"].VALUE("e2")>	3.14e+000
Fixed point, 2 decimal digits, always output sign		<VARS.["pi"].VALUE("+#.##; -#.##")>	+3.14
Fixed point, 2 integer digits, 1 decimal digit, always output sign, no output of significant zeroes		<VARS.["pi"].VALUE("+00.0;-00.0")>	+3.14
Space before positive values, "-" before negative values		<VARS.["pi"].VALUE(" 0.00;-0.00")>	3.14
General	General	<VARS.["pi"].VALUE("g")>	3.1415926535
Fixed point, 4 decimal digits	Decimal	<VARS.["pi"].VALUE("0.0000")>	3.142
Hexadecimal	Hexadecimal Short	<VARS.["pi"].VALUE("x4")>	0003
	Hexadecimal Long	<VARS.["pi"].VALUE("x8")>	0003
Currency	Currency	<VARS.["pi"].VALUE("c2")>	3.14
Percent	Percent	<VARS.["pi"].VALUE("0.0%")>	314%
	Percent Long	<VARS.["pi"].VALUE("0.00%")>	314.16%

Instead of the decimal point comma can also be used. Then, the comma is displayed in the output, e,g,:

- <VARS.["pi"].VALUE("0,00")> = 3,14

With integer values, formatting is also possible. For instance, if a variable "*amount*" has the instantaneous value 4, then this value can be outputted also as 04, 004, etc.:

- <VARS.["amount"].VALUE("00")> = 04
- <VARS.["amount"].VALUE("000")> = 004

7.8.3 Environment variables of the OS

In imc STUDIO it is also possible to access the operating system's environment variables.

Tips on use:

- Set the variable to be resolved in angle brackets with percent symbols,
  as follows: <%USERNAME%>
- Observe capitalization. The variable must be spelled exactly as it is spelled by the operating system, e.g. *ProgramData*, *HOMEDRIVE* or *windir*.

8 Setup - Device (general)

Setup is the imc STUDIO component for uniform configuration and control of imc measurement devices.

Chapter overview

Synopsis	Section
Which devices are supported?	• Device Overview 187
Getting familiar with concepts concerning system operation	• Operation 231
What steps are necessary for performing a measurement? What actions are available?	• Performing a measurement: Procedure 241 • Ribbon 188
Applying characteristic curves and pre-made configurations to a channel	• Sensors 230
Designing custom columns and user interfaces	• Setup-Layout 247 • Creating and configuring additional columns 251

Reference Configuring the device and measurement

With the Setup pages , you can configure one or more measurement quickly and easily.
The extensive data saving, triggering and real-time functions are organized clearly for every device. The measurement parameters can be set individually for each channel. The measured data can be saved separately for each channel. Measurement channels can be monitored continually and only start recording upon the occurrence of specified events.
See: "*Setup pages - Configuring device* 288"

Note

If the documentation mentions a device, the comments apply similarly to other devices as well. All screenshots were created using the "*Complete*" view.

8.1 Device overview

imc STUDIO connects with the imc measurement devices e.g. via the local area network (LAN). For setup of the connection, the devices must be pre-configured (see "*Setting Up - Connect the device* 45").

Some of the capabilities discussed in this document only pertain to certain device models. The associated device groups are indicated at the respective locations. The groups are shown in the following table which can be used with imc STUDIO.

— not available ● standard ○ optional
CRXT imc CRONOS-XT CRFX imc CRONOS*flex* CRC imc CRONOS*compact*

imc device	SPARTAN BUSDAQ	BUSDAQ*flex*	SPARTAN-R SPARTAN-N	CRSL-N	CRC-400 CRFX-400	C1-N C-SERIE-N	C1-FD C-SERIE-FD	CRFX-2000	CRC-2000G	CRC-400GP	CRFX-2000G	CRFX-2000GP	CRXT	EOS	ARGUS*fit*
Driver package	imc DEVICES													imc DEVICE*core*	
Firmware group	**A**													**B**	
Device group	**A4**		**A5**					**A6**	**A7**					**B10**	**B11**
SN[1]	13		14					16	19					4120	416
TCP/IP Interface [MBit/s]	100		100					100	1000					1000	1000
Sampl.Rate[2] [kHz]	400		400					2000 / 400 [3]	2000 / 400 [3]	2000 / 400 [3]	2000	2000	2000	4000	5000
imc STUDIO Monitor supported	●		●					●	●					—	—
Connections[4]	4		4					4	4					—	—
Data Storage															
CF	●		●					—	—					—	—
Express Card	—		—					●	—					—	—
CFast	—		—					—	●					—	—
USB	—		—					●	●	●	●	●	—	—	—
microSD	—		—					—	—					—	●
Storage on network drive	●		●					●	●					—	—
Internal hard disk	(○)[5]	—	○	○	○	—	—	○	○					●	—
Synchronization															
DCF	●	●	●					●	●					—	—
IRIG-B	—	●	●					●	●					●	●
GPS	●	(●)[6]	●					●	●					—	●
NTP	—	●	●					●	●					●	●
PTP	—	—	—					—	—	●	—	●	●	—	—
Phase offset correction	—	●	●					●	●					●	●

1 Extend serial number range by four digits (three for imc EOS)
2 Max. aggregate sampling rate (see data sheet)
3 2000 via EtherCAT else 400
4 Number of imc STUDIO Monitor-connections or imc REMOTE (as of 14xxxx) connections
5 not available for imc BUSDAQ-2
6 not available for imc BUSDAQ*flex*-2-S

8.2 Ribbon

The ribbon of "imc STUDIO Setup".

8.2.1 Control

Table filter

Menu item	Description
Hide Passive Channels	The function helps to display only the channels that are relevant. Passive channels are hidden. The status of this setting is saved with the experiment. This means it remains activated until the button is clicked again.

Device Control

Menu item	Description
Process configuration 190	The current configuration is inspected for errors and made available to all plug-ins. For this purpose, no connection to the device is required. There are cases where certain settings are possible to make during operation while not being actually supported by the respective device type. In such a case, a corresponding error message will be posted (see documentation on imc STUDIO, chapter "Logbook 142").
Connect 190	imc STUDIO connects itself with all selected measurement devices (normally via LAN).
Prepare 195	Changed settings are analyzed and completely imported by the device.
Reconfigure 195	All settings are analyzed and completely imported by the device.
Balance 197	All active channels are balanced. This applies any settings which have been set on the Setup page "*Channel balance*".
Start measurement 197	The measurement is started for all devices.
Stop measurement 197	The measurement is stopped for all devices.
Suspend data storage 198	If data transfer to storage is activated, it is suspended for all devices.
Resume data storage 198	If data transfer to storage is activated, it is resumed for all devices.
Disconnect 190	imc STUDIO disconnects from all selected measurement devices.
Search for devices 199	The device search causes the system to search the network for any suitable devices. Ultimately, the devices found are listed.
Search for devices by IP/DNS 200	Search for devices by a user-defined IP address or DNS name.
3rd Party Device Management 200	Opens the "3rd Party Device Management" to add Third Party Devices to the device list.

imc WAVE

Menu item	Description
Calibration of IEPE sensors 202	Calibrating the microphones before the first measurement

8.2.1.1 Process configuration

The current configuration is inspected for errors and made available to all plug-ins. Connection to the devices is not required for this purpose.

Note

There are cases where certain settings are possible to make during operation while not being actually supported by the respective device type. In such a case, a corresponding error message will be posted.

Ribbon	View
Home > Process ( )	all
Setup-Control > Process ( )	Complete

8.2.1.2 Connect and disconnect

imc STUDIO connects itself with/disconnects itself from all selected measurement devices (normally via LAN).

Ribbon	View
Home > Connect ( )	all
Setup-Control > Connect ( )	Complete
Home > Disconnect ( )	all
Setup-Control > Disconnect ( )	Complete

Note

If the measurement device's software version is different than that of the product installation on the PC, the Firmware Update 63 follows next.

Change of the imc measurement device

When connection takes place, the system checks whether the previously registered device and the actually connected device match. There can be discrepancies when modular systems are connected. If a difference is detected, the current device setup needs to be determined. Subsequently, the changes are reflected in imc STUDIO.

Warning

For devices of the firmware group A 187

Please note that for this reason it may not be possible to apply the existing experiment settings. For instance, if the device previously had a bridge amplifier and it was replaced with a temperature amplifier

Note

For devices of the firmware group B 187

For transfer of the configuration from imc CANSAS*fit*- and imc ARGUS*fit*-modules which are attached to an imc ARGUS*fit* unit, the following limitation applies.

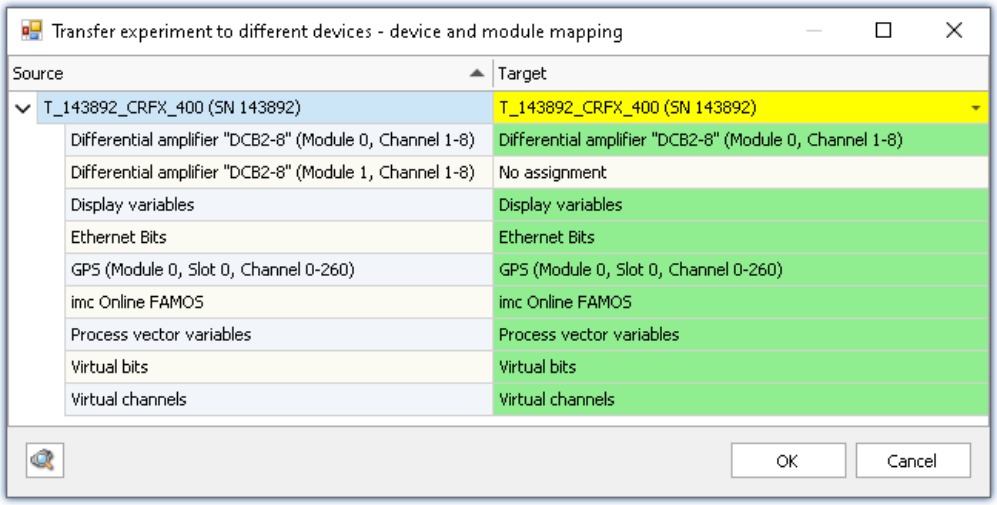
The configuration of a module can only be transferred when the following conditions are met:

- same module-type (e.g. ARGFT\UTI-6-SUP to ARGFT\UTI-6-SUP)
- same device family, meaning imc ARGUS*fit* only to imc ARGUS*fit* and imc CANSAS*fit* only to imc CANSAS*fit*
- same scope of functions (e.g. same connection terminal)

The module's hardware revision stated on the plaque does not matter.

When changes are detected, a prompt appears which asks you how to treat the current configuration. You can **discard** the configuration or **transfer** it to the "current, changed" device. In this case, transfer the configuration to the device.

A module assigning dialog appears. If just a module was added, you do not need to make any changes here. But if you have replaced a module or if internal hardware has been replaced as part of some servicing work, please verify the assignment.



Transferring in the case of a changed device configuration

At left you see the older device and at right the new one.

Below the respective device, you see a list of all of a device's "Modules" and components. What this list contains depends on the device's hardware configuration. Use the drop-down-list at right to select the appropriate modules.

The colors:

Green: All of the module's settings can be transferred.

Yellow: Not all of the module's settings can be transferred. For example, the amplifier type may be different. Please ensure that the target module is able to use the configuration required by the source module. E.g. a universal amplifier can use some of a temperature amplifier's settings.

Red: This assignment should not be selected.

What settings are applied:

In principle, all settings are applied, except for the **calibration values** (taring, bridge balancing, two-point scaling, ...). Settings which can not be applied are ignored and listed in the **log book** following the transfer.

Connected while the device is running a measurement

If an attempt is made to establish a connection with a device which is already running a measurement, you can either connect with the running measurement 192 or **stop the measurement** and connect with the device later.

Warning

When imc STUDIO establishes a connection with a running measurement, the current settings are discarded and the running experiment's settings are loaded.

8.2.1.2.1 Disconnect/Connect - with a measurement running

Disconnecting from the device during a running measurement - Disconnection file

If imc STUDIO is disconnected from the device during a running measurement (without stopping measurement), then a **disconnection file is saved on the PC**. This is loaded each time connection with the measurement is restored. The disconnection file reflects the **current state of the experiment** at the moment in time the **measurement starts**.

After "*Disconnect*" has been selected, the device resumes the measurement.

Note No connection restoration without an experiment

If the **experiment is deleted** form the PC with the disconnection file, then re-connection with the running measurement is **no longer possible**.

Re-connecting with the measuring device

Disconnection and re-connection are performed on the same computer. In the meanwhile, imc STUDIO can be exited (or the computer can be shut down).

Upon re-connecting with a device currently running a measurement, the system **checks** whether the associated **experiment and the disconnection file are on the PC**. Only then is it possible to restore the connection. If the appropriate files are not found on the PC when re-connecting, the connection procedure is canceled and an error message is posted.

If re-connection is possible, the **original experiment is re-loaded from the disconnection file**. Most of the time, a prompt to save data appears, since the currently loaded experiment will be discarded. (Exception: When no changes to the current experiment have been detected since disconnection. This is a rather rare case.)

Warning Measured data are lost under "Current measurement"

Since the experiment is loaded from the disconnection file, the measured data are emptied from "Current measurement". They are thus no longer available. By contrast, the saved measured data remain available. See "*Measured data and their storage during the disconnection phase*".

Note

Re-connection not via the Setup-table

Reconnection with the running measurement is only possible by means of the menu item "*Connect*". Not, by contrast, by means of the column "*Device control* 291".

Special case: another PC

If you start a measurement with PC A, then disconnect it from the device and later connect the device to PC B, this is only possible if the existing experiment in the database is identical (the disconnection file must also be copied).

If the project management is deactivated (no database used), the connection can also be re-established on other PCs on this condition: the experiment and the *.data folder are located on the same path in the PC as the one from which the measurement was started.

Measured data and their storage during the disconnection phase

As long as imc STUDIO is not connected to the device, it is not possible to view the current measurement data. Once connection has been restored, saving of data to new files resumes (new measurement folder).

If no gap in the data is detected (data overflow: e.g. RAM buffer duration exceeded), the system reads the RAM-buffer out. Otherwise, the measured data accruing between disconnection and re-connection are missing and are not saved. Workaround: Save the measured data additionally in the device.

Disconnect/Re-connect	RAM-buffer	Current measurement	Saved measured data
Re-connection **within** the RAM-buffer duration	RAM-buffer is **read out**.	The data accruing before disconnection are joined and displayed with the data from the RAM (disconnected time period) and with those accruing after re-connection. **No data loss and no events**.	The data **from the RAM** (disconnected time period) and those accruing **after re-connection** are saved in a new folder. Thus, the measured data are saved in two folders, without any data loss. • The first folder contains all data **up to disconnection**. • The second folder contains all data **from disconnection onward**.
Re-connection **within** the RAM-buffer duration imc STUDIO closed in interim	RAM-buffer is **read out**.	The data **from the RAM** (disconnected time period) and those accruing **after re-connection** are displayed. The data accruing before disconnection are discarded.	
Re-connection **after** the RAM-buffer duration	The RAM-buffer is **not read out**. Data loss occurs.	Depending on the setting for "*Event count*": • 1: only the data **after re-connection** are displayed, or • all: the data **before disconnection** are supplemented with a new event consisting of the data from **after re-connection**. No data from the disconnected time period are present.	The data accruing **after re-connection** are saved in a new folder. Thus, with loss of measured data, the data are saved in two folders. • The first folder contains all data **up to disconnection**. • The second folder contains all data **from re-connection** onward.
Re-connection **after** the RAM-buffer duration imc STUDIO closed in interim	The RAM-buffer is **not read out**. Data loss occurs.	The data accruing **after re-connection** are displayed. The data before re-connection are discarded.	

Note

If a measurement is performed in disconnected mode and you establish a connection to the device only **after completion** of the measurement, then the **measurement data are not in the Data Browser**.

Connecting to a device running a measurement, which was started by a Diskstart/Autostart procedure

When a Diskstart configuration is generated, an additional **disconnection file is saved on the PC**. Connection to the measuring device proceeds in the manner described above for re-connecting.

Editing an experiment created by connecting with a device

When connection to a device running a measurement is successful, the corresponding settings of all imc STUDIO components are loaded. These can be edited and used for re-configuring the device.

Warning Saving overwrites the experiment

Save the configuration, and thus overwrite the original experiment.

If you re-connect with the measurement at a later time, you automatically re-load the disconnection file. In consequence, you restore the old state.

8.2.1.2.2 Outage of the network connection

When the network connection between the PC and the device is lost, it often causes data to be lost also. Here, too, re-connection with the running measurement is performed. But this is to be distinguished from "*Disconnection and re-connection*".

Reference RAM Buffer Duration - Outage of the network connection

For more information on how to avoid loss of measured data and how to restore the connection, see the chapter: "*Setup pages - Configuring device*" > "*Storage options and directory structure*" > "*RAM Buffer Duration - Outage of the network connection* 427".

8.2.1.3 Prepare and Reconfigure

Before first starting the measurement, the configuration must be loaded in the device. The device is initialized automatically if the configuration in the device is not up-to-date. If the device contains the current configuration, no initialization is performed.

In order to prepare the device, use one of the two possibilities below:

Action	Description
Reconfigure	All settings are analyzed and completely imported by the device.
Prepare	Changed settings are analyzed and completely imported by the device. If no changes are detected, the device's configuration is already up-to-date. The actual preparation process is not carried out in that case. If you wish to have it carried out anyway, you must run "*Reconfiguration*".

Ribbon	View
Home > Prepare ( )	Complete
Setup-Control > Prepare ( )	Complete
Setup-Control > Reconfigure ( )	Complete

Note

- Under some circumstances the action "*Process configuration* 190" is automatic executed first, if this had not been done already.
- For preparation and reconfiguration, the PC must previously have set op a connection to the device (see "Connect and Disconnect 190").

FAQ

Question: Why don't I have the buttons "Prepare" and "Reconfigure"?

Answer: These two buttons are only present in the "*Complete*" view. In most cases, the device does not need to be manually initialized. It is sufficient for the device to be automatically initialized upon starting the measurement.

If you need the function anyway, you can either change the view, or add the button to your user interface (see "Customize Ribbon 159").

Question: What happens when multiple "writers" each try to set their own value during the preparation? For instance, if a value is set for the DAC via a Widget, and also in imc Online FAMOS in the control command: "OnInitAll".

Answer: To prevent signal jumps on a DAC-output channel, then during initialization, the system checks whether a DAC output channel is initialized in the "OnInitAll" in the imc Online FAMOS-code. If so, this value is used and any previously set value from the data pool (e.g. via a Widget) is ignored.

When "*preparing*" (reconfiguring) imc Online FAMOS wins and the value in the imc STUDIO variable is overwritten.

Question: The following error message is being posted: "The device has been shut down inaccurate in the past. Please check the uninterruptible power supply of your device!" What should I do?

Answer: When you shut down the device, it needs a few seconds to power down. During this time, the device is being powered by the internal UPS. If the UPS is defective, the device can not power down correctly and shuts down immediately.

Please check whether the device goes off immediately upon tripping the main switch, or only after one second or longer.

If the device goes off immediately, then presumably the UPS is not working correctly. Please get in touch with our tech support 8.

Reference

See also:

Executed once at the start: "*OnInitAll* 1044"

8.2.1.4 Balance - All active channels

The system balances all active channels. In the process, any settings appearing on the Setup page "*Channel balance*" under "*Balance* 377", e.g. Taring or Bridge, are applied.

Ribbon	View
Home > Balance ( )	all
Setup-Control > Balance ( )	Complete
Panel-Control > Balance ( )	Complete

8.2.1.5 Start and Stop measurement

The measurement is started/stopped for all devices.

Note

- It may be necessary to carry out the "preparation 195", unless this has already taken place.
- For starting and stopping the measurement, the PC needs to have established a connection with the device (see "Connect and Disconnect 190").

Ribbon	View
Home > Start ( )	all
Setup-Control > Start ( )	Complete
Home > Stop ( )	all
Setup-Control > Stop ( )	Complete

Additional ways to start/stop a measurement

- via the column "*Device control* 291" in the devices-table
- using the Sequencer command "*Execute device action*". See the example: "*Starting measurement* 1743".

In contrast to the menu items, these actions do not automatically affect all devices. Thus you can stop a device during measurement, modify it in some way, and restart it, without affecting the other devices. The only other effect is that from that point in time, measured data are saved in a new folder.

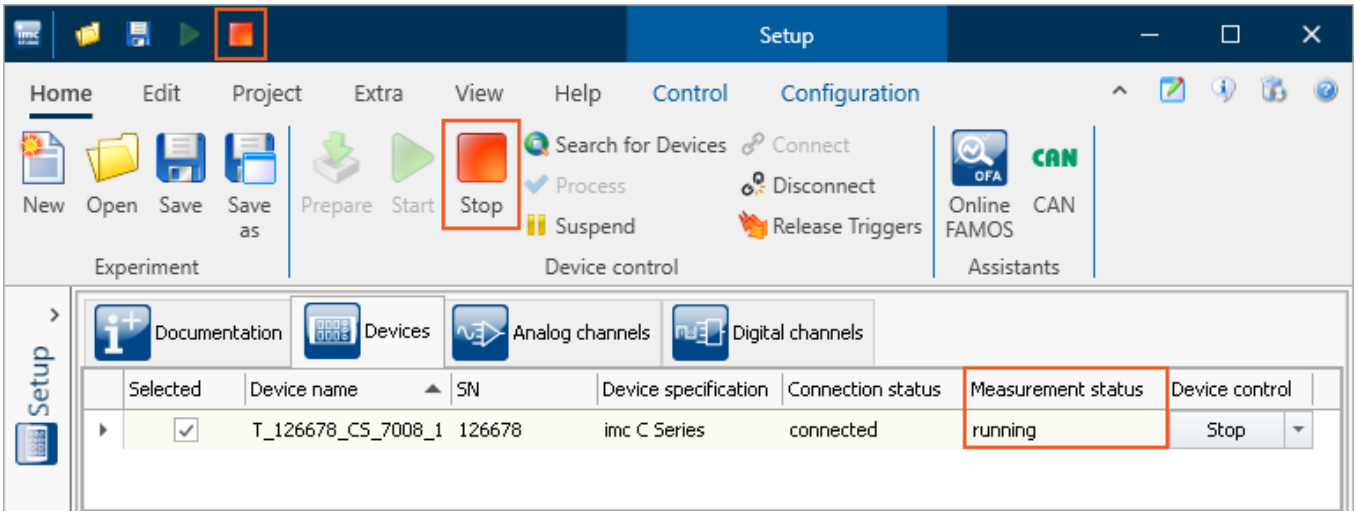
Also, these actions are not performed in synchronization with each other. The devices each either start or stop separately and as quickly as possible.

Measurement status (running, stopped)

The measurement's status is denoted e.g. by the stop-symbol (if imc STUDIO is connected with the device):

Icon	Description
	Measurement running (solid red)
	Measurement stopped (color depends on the Windows setting)

You can also see the status in the column "***Measurement status***" on the "**Devices**" page:



Column "Measurement status"

Note 6 flashes during measurement

During a running measurement, LED 6 flashes at a 1-second rhythm. This provides a simple visual indication of whether the measurement is running.

LED 6 does not flash,

- if it is used in the imc Online FAMOS source code,
- if flashing is deactivated in the imc Online FAMOS Options,
- if imc Online FAMOS is disabled.

8.2.1.6 Suspend and Resume data storage

If the saving of measured data is activated 370, then you are able to govern the data storage **during the running measurement**. You can **interrupt (suspend) data storage** and **resume it again** at a later time.

Ribbon	View
Home > Suspend data storage ( )	all
Setup-Control > Suspend data storage ( )	Complete
Home > Resume data storage ( )	all
Setup-Control > Resume data storage ( )	Complete

Effect

If you **suspend data storage**, the **files are closed** and appear in the Data Browser (as if the measurement had been concluded). Any **more data** which arrive or are calculated subsequently **are not saved**.

When you **resume data storage**, then **new files are created** in a **new measurement folder** (as if the measurement had just been started). Any data arriving before "resumption of data saving" are not saved.

This function is designed for **interrupting data saving for longer periods of time**. Please do **not use it to make "cuts" of data records** (concluding one measurement folder and immediately beginning a new one).

Note

- Data storage **con only be controlled** if it was previously **activated**.
- This **suspension/resumption** of data storage **is not synchronized** across the devices! Thus, the respective **data excerpt boundary points** of different devices/channels **may differ by a few measurement points**.

Scope

These actions **apply to all devices and to the PC**. No separation is possible. Any additional components which return data to be saved (such as 3rd-party devices) are also affected.

Note Video-files are not affected

The **saving of Video files** is **not affected** by this function. Video files are always recorded.

Reference Data storage state after preparation

Using the option "Data storage state after preparation", you can control the function's state. E.g. it is possible to make data storage generally active following "Preparation".
See the option: "*Setup*" > "*General*" > "*Data storage state after preparation* 134".

8.2.1.7 Search for devices

The device search causes the system to search the network for any suitable devices. How long this will take depends on how many devices are connected and on the network type. Ultimately, the devices found are listed and the view automatically switches to the "**Devices**" page of the plug-in **Setup**.

Ribbon	View
Home > Search for devices ( )	all
Setup-Control > Search for devices ( )	Complete

E.g., after the first launch of imc STUDIO, the device table 290 is empty. Before you can create an experiment, you must enter one or more devices into the table. To do this, perform a device search. The device table lists all devices found. The image below shows a typical situation:

Selected	Device name	SN	Device specification
	T_124835_C1_1_LEMO_ET	124835	imc C1-1 LEMO
	T_130039_busDAQ_X	130039	busDAQ-X
	T_130311_SPARTAN_U32_CAN	130311	imc SPARTAN

Results of device search (example)

Reference

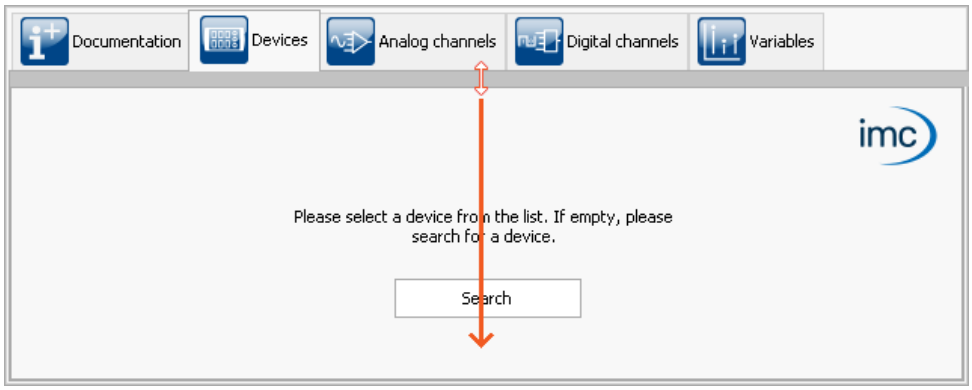
- How to correctly set **your device's network configuration** and what to make note of is presented in the chapter:
  "*Setting Up - Connect the device* 45"
- For more information on **selecting the device for the experiment**, see the chapter: "*Device Search - Known and Selected* 241"

FAQ

I am shown the message: "Unable to find any new devices."

Answer: In some cases, the Devices/Channel list can be hidden. It depends on the view and the Windows display settings applied.

Please check whether the list may only be covered over:



The slider is all the way at the top - The lower window covers the Devices list

8.2.1.8 Search for devices by IP/DNS

In a structured network (network with routers, Internet, ...), imc devices could not be integrated by means of a network search. With the knowledge of the IP address or of the domain name (DNS name), it is now possible to integrate a device into the "*Device Table* 290" and to establish a connection.

Ribbon	View
Setup-Control > Search for devices by IP/DNS ( )	Complete

Reference

More information is presented in the chapter: "*Setting Up - Connect the device* 45" > "*Connection via a direct address* 58"

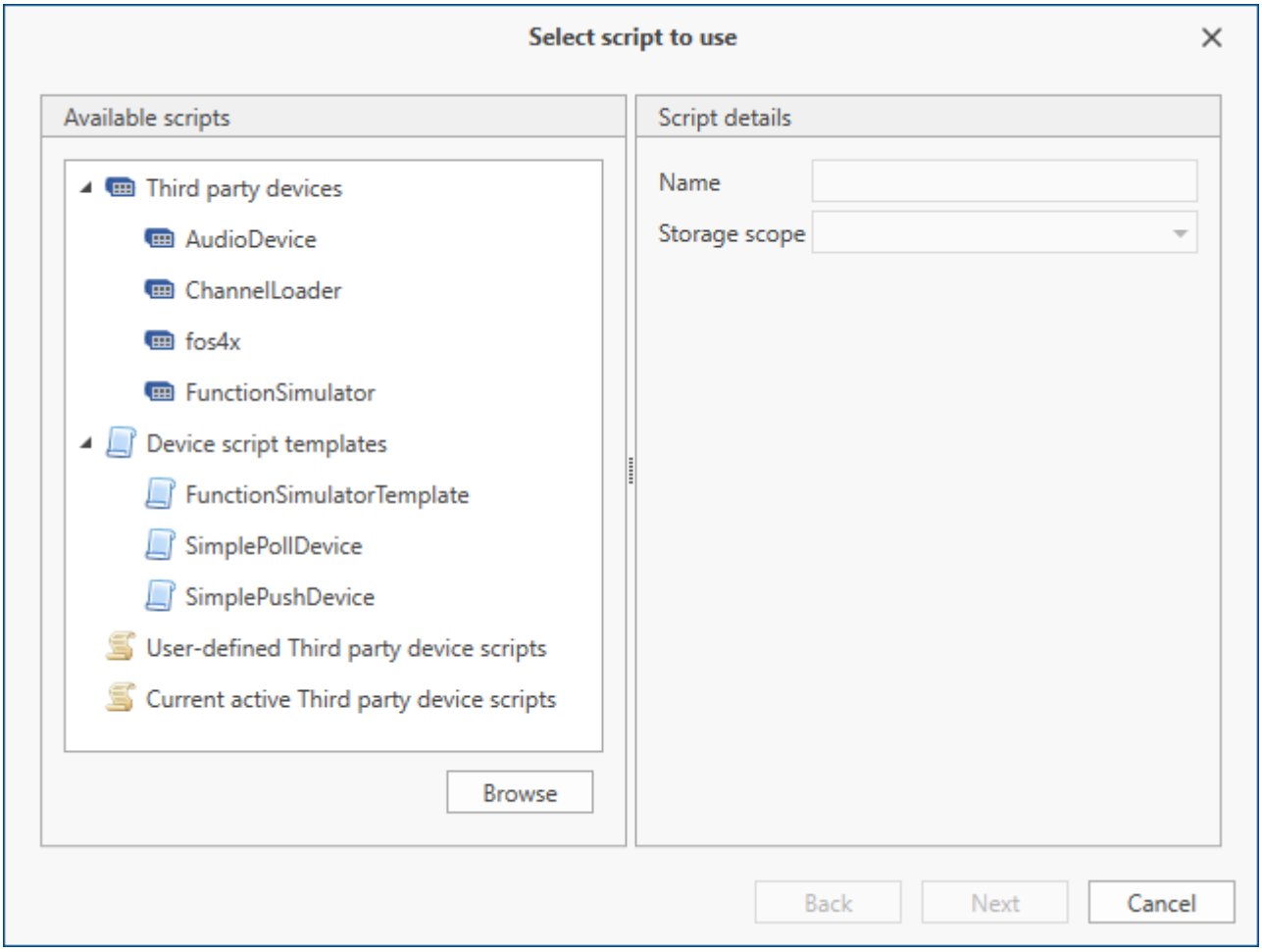
8.2.1.9 3rd party device management

Using the **3rd-Party Device Management**, it is possible to select executable 3rd-party device script templates, e.g. the Function-Simulator, the SimplePollDevice or the SimplePushDevice. Likewise, there are already pre-made 3rd-party devices. Along with the AudioDevice and the ChannelLoader, the fos4x is also available. fos4x requires an extra license.

Note

- The 3rd-Party Device Management is available in the editions *Developer*, *Professional* and *Standard*.
- For the use of 3rd-party devices, an activated **imc STUDIO 3PDI-Inclusive** or **imc STUDIO 3PDI-Exclusive** license is required.
- The **AudioDevice**, e.g. the audio interface **2-channel IEPE/ICP-signal conditioner (FH/ICP2-USB),** the **ChannelLoader** and the **FunctionSimulator** can be used without any **3PDI license**.
- For the templates a **3PDI-license** is needed.
- For the fos4x device an **imc STUDIO 3PDI-fos4x** license is required**.**

Ribbon	View
Setup-Control > 3rd Party Device Management ( )	Complete



3rd Party Device Management Dialog

When selecting a script-template, there is an option for subsequently opening the Script-Editor.

User-defined scripts can be added by clicking on "*Browse*".

In "*Current active Third party device scripts*" all Third Party Device Scripts are listed and can be deleted by selecting and clicking on "*Next*".

Reference

For more information, see Scripting - Third Party Device 1918.

AudioDevice

In order to use connected Audio-devices, select the device-script *AudioDevice* and click on *Next*. Subsequently, all Audio-devices appear in the device list. When one of the Audio-devices is selected, the computer's Audio-input channels appear on the page *Analog Channels*. In this way, when a measurement starts, the computer's Audio-inputs can be graphed and recorded.

For more information on this subject, see the chapter AudioDevice 723.

ChannelLoader

The *ChannelLoader* serves to play back data already recorded. To do this, select the *ChannelLoader* and click on *Next*. Subsequently, the device *ChannelLoader* appears in the device list. When *ChannelLoader* is selected, a file selection dialog with multi-selection appears. Here, multiple already-recorded data or imc FAMOS-data sets can be selected. On the page *Analog Channels*, the corresponding channels appear. When a measurement starts, then the selected data are played back cyclically at their respective sampling rates.
For more information on this subject, see the chapter ChannelLoader 724.

Device script templates

The devices from the script-templates simulate a sine-signal, or in the case of the *FunctionSimulator*, a number of different signals (trapezoid, sawtooth, rectangle, ...).

8.2.1.10 Calibration of IEPE sensors

Calibrating the microphones before the first measurement

Select the menu item "*Calibrate IEPE sensors*" to have the microphones calibrated.

Ribbon	View
Setup-Control > *Calibrate IEPE sensors* ( )	Complete

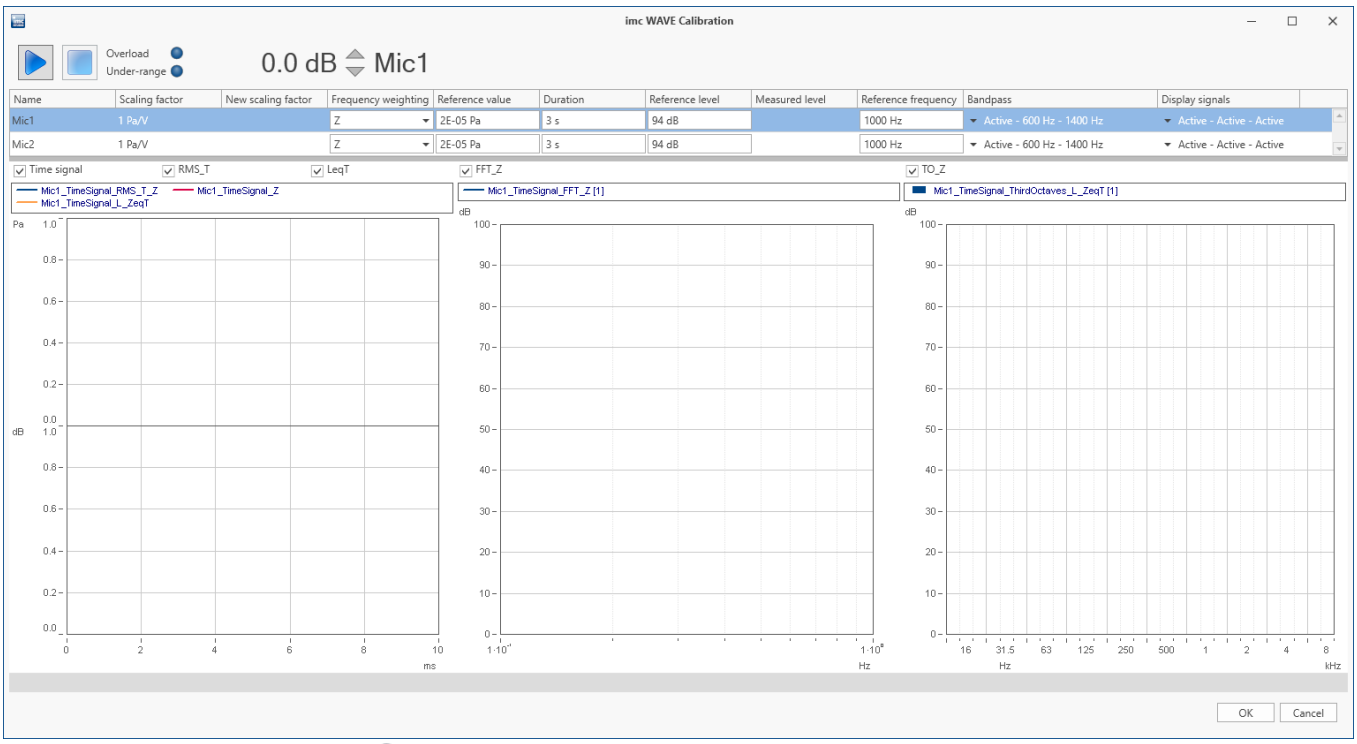
As an example, the signal sound level at a frequency of 1 kHz is adjusted to 94 db. Independent of the frequency weighting selected (A, B, C or Z), the gain is 0 dB at a frequency of 1 kHz.

At each of the measurement channels in succession, the calibrator's defined sound pressure level of either 94 dB or 114 dB is measured as the reference level (over the measurement duration specified). The scaling factor measured is adjusted for the microphone in the channel.

Procedure illustrated by an example:

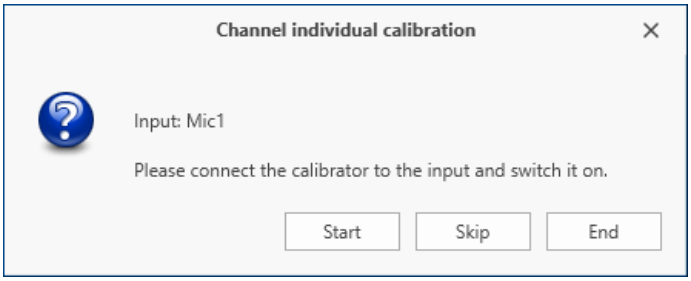
- Start the measurement
- Select the menu item: "*Calibrate IEPE sensors*"

The dialog "*imc WAVE Calibration*" opens:



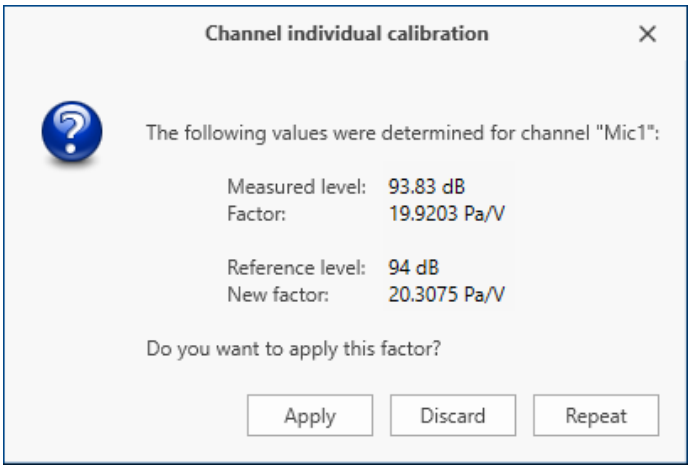
- Click on the Start button ( ).

Subsequently, for each channel in succession separately, a prompt to connect the calibrator and to start calibration appears.



- Connect the calibrator. Apply the reference level set and close the window by clicking "*Start*"

- After the measurement duration specified, the measurement result appears:



- Accept the new factor by clicking on "*Apply*"

Next, calibration for the next channel is performed. You are again prompted to connect the calibrator to the corresponding new input. Once you have balanced all channels or have concluded calibration, you can close the dialog by clicking "*OK*" to apply the values.

The respective scaling factor values are saved with the parameter: "*Factor* 351" (shown in the dialog "*Range & Scaling*"). The new values are applied **upon re-starting the measurement**.

8.2.2 Configuration

Note

In some dialogs (e.g. imc Online FAMOS), it is possible to make device-dependent settings. In that case, it is necessary to make a selection in the dialog of the device to which the settings pertain.

Assistants and Synthesizer

Menu item	Description
imc Online FAMOS	This starts the imc Online FAMOS dialog. For a detailed description, see the chapter: "*imc Online FAMOS and imc Inline FAMOS* 851".
imc Inline FAMOS	This starts the imc Inline FAMOS dialog. For a detailed description, see the chapter: "*imc Online FAMOS and imc Inline FAMOS* 851".
imc Display Editor	This starts the imc Display Editor dialog. For a detailed description, see the chapter: "*Setup - Advanced Device Functions (imc DEVICES)*" > "*imc Display Editor* 769".
imc CANSAS 210	This starts the imc CANSAS dialog.
Assistants	Here, a variety of setup assistants are offered. For a detailed description, see the chapter: "*Setup - Advanced Device Functions (imc DEVICES)*" > "*Fieldbusses* 474".
Synthesizer Configuration and Synthesizer Control Panel	Operation of the Synthesizer is described in a separate document.

Device configuration

Menu item	Description
Export Configuration 1161	This function provides a variety of exporting options. For most of the selection options, only the parameters (the content of the rows and columns) of the table on the currently open Setup page are exported.
Import Configuration 1163	This function enables the import of parameters. You can also select from which file to import which variable's values and/or settings.
Default Values 214	These are preset values for the configurations of the devices and channel. These values are applied upon selection of devices and when creating new device variables/channels or Data Processing variables.
Device interfaces 53	Displays the dialog for configuring the devices' (network-) interfaces.
Diskstart 206	Displays the dialog for creating Diskstart and Autostart experiments.
Device Properties 221	Displays the dialog showing the device's properties.
Module Properties 226	Displays the dialog showing the properties of the device's amplifiers/modules.
Supplemental Files 212	Shows the dialog to manage supplemental files. Here you can export, import and view existing additional files.

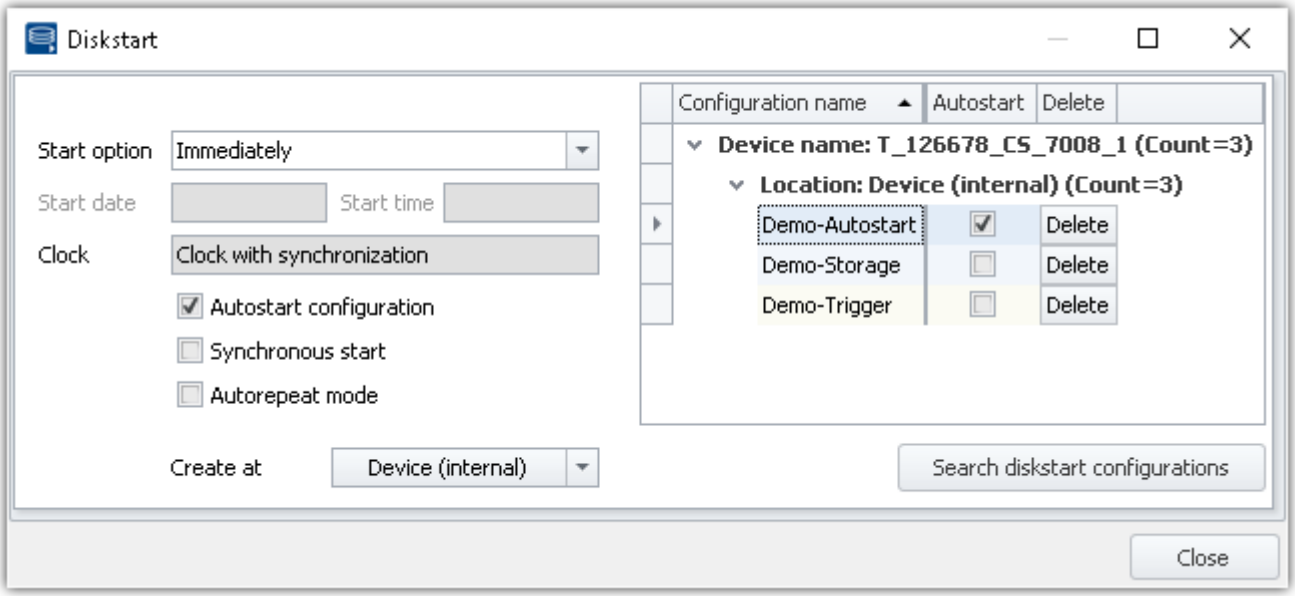
Edit

Menu item	Description
Create Process vector variable	This function creates a user-defined pv-variable for a device. The variable can be used in imc Online FAMOS, as if it were declared there in the `OnInitAll`. A variety of data types are available. Note for devices of the imc DEVICES group: • TI-Float: Optimized for imc Online FAMOS-processing • Float: Only for application in calculations For more info see section: "*Process vector variables* 1141"
Remove Rows	Deletes the selected pv-variable.

8.2.2.1 Diskstart

The measurement device is able to start a measurement without being connected to any PC. In most cases, this only makes sense of the data acquired are saved to the optional device hard disk. It is possible to save multiple, parallel Diskstart configurations in the device drive, but only one having the setting: "*Autostart configuration*".

Ribbon	View
Extra > Diskstart ( )	Standard
Setup-Configuration > Diskstart ( )	Complete



Diskstart without Autostart

Once the device has been activated, the Diskstart configuration can be selected by means of a connected Display unit. **For this, the optional Display unit is absolutely necessary.**

Diskstart with Autostart

If a Diskstart configuration having the property: "*Autostart configuration*" exists, then the associated measurement starts automatically as soon as the device is activated. This is done **without selection by means of a Display unit**. All other Diskstart configurations located on the device HD are thus no longer available for selection.

Exactly one Diskstart configuration having the property: "*Autostart configuration*" can be saved.

Start option

Start options are available both for the normal start (in response to the Start button), as well as for the Diskstart. The functionality is described in the chapter "*Timed start* 335" (the limitations and conditions are delineated).

Clock

The start condition "**Synchronous start**" ensures that the device only starts measurement once the device clock has been synchronized according to the synchonization settings 292.

Observe the "*maximum time to wait*", which is set in the device properties 224.

Autorepeat mode

When a Diskstart is performed, then after the end of a measurement, the measurement devices are able to start it again right away. This mode is called "*Autorepeat mode*".

Note — Note on Timed start with Autorepeat

In "*Autorepeat mode*", the duration between the measurements is not defined. For this reason, the Timed start only works for the first measurement. All subsequent measurements then start immediately.

Note — Autorepeat under imc STUDIO

In imc STUDIO, "*Autorepeat mode*" is replaced with a variety of other components, e.g. the Sequencer. The program can additionally be configured with an equivalent triggering setup.

For this reason, the "*Autorepeat mode*" only remains available for Diskstart.

Autonomous starting of a measurement makes imc STUDIO unable to connect with the running measurement without problems.

When imc STUDIO connects with the running measurement, the Autorepeat mode is deactivated. This means that the measurement is still concluded properly, but will not be re-started again afterwards.

Balance at device startup

Activate the property "*Balance at device startup*" (Setup page: "*Channel balance*" or "*Analog channels*") for the desired channels. If you wish, you can also display the parameter of the same name as an additional column by using the "*Column chooser* 248".

Parameter	Description		
Category: Channel	*long name*	*short name*	*column ID*
Balance at device startup	*Balance at device startup*	*Balance at startup*	*eBalanceAtDeviceStart*

This parameter has two purposes:

- Before a Diskstart/Autostart measurement starts, **balancing** of the selected channels is performed.
- The channels selected can be balanced using the **imc Online FAMOS function** `RunAutoBalance` 1008.

This balancing operation starts automatically before every Diskstart-measurement. The type of balancing performed is set on the following page: Setup page: "*Channel balance*" > "*Balance*". Available options: Taring or bridge balancing (depending on the hardware).

Note

The system performs a version check (to determine the firmware version). A Diskstart configuration is generated for precisely the current firmware version. if the device's firmware version is changed (e.g. by means of an update), then it is no longer possible to load the existing Diskstart configuration, or the configuration is canceled with the error message "Error: -132". Once the Diskstart configuration has been generated again, it can be opened again with the current Devices-software version.

8.2.2.1.1 Creating a Diskstart configuration

You can save an experiment as a Diskstart configuration

- directly in the device (on the device's hard drive ("*Device (internal)*" or "*Removable drive*")
- or initially on the PC ("*PC harddisk*"), if, for example, the measurement device is not available and the data are copied to the removable data carrier later.

Note

- If the storage location is: "*Removable drive*", then in general the complete imc STUDIO experiment is saved to the data carrier. The storage location: "*Device (internal)*" is not sufficiently large.
- When the option "*Synchronous start*" is activated, the device waits until it has been synchronized to the synchronization source specified. If the synchronization source is no longer present, the response depends on the device properties: the option "*Max. time to wait for synchronization*" > "0" starts the device immediately even without synchronization when the time elapses. For a value of "0", the waiting period is Unlimited; for "-1" it starts immediately and does not synchronize.

Precondition for both options:

- The desired device must be registered with the program.

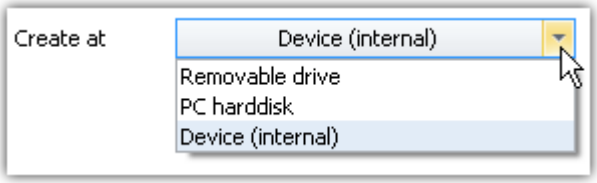
Preconditions for the storage locations: "*Device (intern)*" and "*Removable drive*":

- A connection must be established between the device and PC.
- The device must have sufficient memory.

Creating a Diskstart configuration

Proceed as follows to create a new Diskstart configuration:

- Ensure that there is a connection between the device and the PC (if the selected storage location is ("*Device (internal)*" or "*Removable drive*").
- Select the menu ribbon item "*Setup-Configuration*" > "*Diskstart*" ( ).
- Make the appropriate settings for the options *"Start option" and "Start date/-time"* 336.
- Activate the option: "*Autostart configuration* 206" if desired.
- Open the list box "*Create at*" and select the appropriate entry.



- Click on the button "*Create at*"

Note

- When the Diskstart configuration is created, an "ume-file" or "ums-file" (if with "*Autostart*") is generated on the basis of the current experiment settings. Saving the current experiment configuration is not necessary for this purpose. But there may be different experiments on the device hard drive from the PC.
- If "*PC harddisk*" is selected as the storage location, then an "ume"/"ums"-file is created on an arbitrarily selectable path. A corresponding experiment file ("**.imcSudio*") is also created.

Storage location: PC hard drive

If your measurement device is currently not available, you can set up a Diskstart by selecting the "*PC harddisk*" was the storage location.

Two files are created in the selected storage location:

- **Diskstart configuration file** for the device: "*.ume.zip" (without "*Autostart*") or "*.ums.zip" (with "*Autostart*")
- a corresponding **imc STUDIO experiment file**: "*.imcStudio"

You have thus created a configuration file which can be used for Diskstart.

The Diskstart configuration file can now be copied to a removable disk.

- Insert an appropriate removable drive into your PC.
- Create a **subfolder having the experiment's name** and copy both files to the subfolder created.

Example — Example without Autostart

Experiment name: "Experiment_001"

Device name: "dev001"

Drive name for the removable data storage medium: "D"

```
D:\Experiment_001\Dev001.ume.zip
D:\Experiment_001\Experiment_001.imcStudio
```

Note — Changing a Diskstart to a Diskstart with Autostart

It is possible to generate a Diskstart configuration with Autostart from an existing configuration without Autostart, without needing to create a new file.

To do this, change the file extension:

*.ume.zip (without "*Autostart*") --> *.ums.zip (with "*Autostart*")

8.2.2.1.2 Deleting a Diskstart

To delete a Diskstart configuration, proceed as follows:

- Ensure that a connection has been established between the device and the PC.
- Select the menu ribbon item "*Setup-Configuration*" > "*Diskstart*" ( ).
- If the desired configuration name does not appear, click on the button "*Find diskstart configurations*". All existing Diskstart-configurations will then appear in the table.
- Click on the button "*Delete*" next to the configuration to be deleted in the table.

8.2.2.1.3 Selecting and starting the Diskstart configuration

The Diskstart configuration is created and and saved to the device.

After activating the device, the removable drive and any hard drive are automatically searched for Diskstart configurations. If one or more Diskstart configurations are found, then a selection list plus some brief instructions are offered on the device display.

Selection of the Diskstart configuration is only accomplished by means of the device's display; no PC or connection to a PC is required.

Following selection of the Diskstart configuration and confirmation of the selection, the device can be configured and started at the push of a button.

8.2.2.1.4 Connecting with a running Diskstart measurement

Information on establishing a connection between a PC and a device running a Diskstart measurement are presented in the chapter: "Disconnect/Connect - with a measurement running 195".

8.2.2.2 imc CANSAS

This starts the imc CANSAS dialog.

Prerequisites

To see this entry it is necessary that

- the imc CANSAS configuration software is at least from version 1.4 R5 and
- the measurement device is equipped with a CAN interface of type CAN2.

Reference Manual

- The Help for the imc CANSAS configuration software can be opened by function key [F1] from the imc CANSAS assistant.
- As a manual in PDF format, it can be found on the installation media of imc CANSAS.

Integrating the imc CANSAS software with imc STUDIO

Certain functions and menu items aren't available for calling from imc STUDIO. For example, the imc CANSAS configuration is not administered as an MDB-database, since it is saved along with the experiment under imc STUDIO. XML-export/import is possible.

Access to the imc CANSAS hardware is provided via the measurement devices' CAN-Bus hardware. This communicates via Ethernet, so that all interfaces incl. imc-USB as well as the Interface-Dialog are blocked.

Note Do not make changes to the channel settings in imc STUDIO

Please **do not make any changes to the properties** of imc CANSAS-channels **by** means of the **imc STUDIO-software's** configuration pages. This includes not making such changes as: name, comment, sampling interval, unit and X-axis.

Instead make these settings in the imc CANSAS-software. If you do make any change in imc STUDIO, an error message will appear upon preparation.

Error correction: Please undo the changes (e.g. by means of the menu ribbon item "*Edit*" > "*Undo*"). Or, open imc CANSAS and then close it one time. All settings will be re-imported and the configuration will be corrected automatically.

When using imc STUDIO to configure the imc CANSAS modules, the modules are integrated by the device in a similar manner as amplifiers. This requires unique identifiers, since the process of preparing the measurement includes checking and setting the imc CANSAS module configurations. This substantially simplifies the operation of imc CANSAS modules, since the steps of exporting and importing CAN-Bus configurations are avoided.

© 2023 imc Test & Measurement GmbH

Note Units not displayed correctly

In some cases, the **unit** of imc CANSAS channels is not displayed correctly. For instance, the unit **"°C" becomes "?C"** and **"µm/m" becomes "?m/m"**.

This problem is caused by mixed **language settings** for the **format** in Windows.

Example:

- Windows display language: "*English (Germany)*" or even "*English (United States)*"
- Region: "*United States*"
- **Regional format**: "*English (Germany)*"

Workaround:

- Change Regional format to "*English (United States)*"
- It may also be necessary to change the numerical format manually in Windows to match the preference associated with the Regional format.

Correction:

Once the problem has occurred and been saved, it cannot be undone for the affected channels. The unit can be corrected, for example, by setting the measurement mode to "Voltage" and back again. This resets the unit.

Constraint

An Experiment involving imc CANSAS modules which were integrated into the system using the CAN-Assistant of imc STUDIO expects exactly the same imc CANSAS modules to be present when a measurement is prepared. Thus, if a previously integrated imc CANSAS module was removed from the measurement setup, then an error message is posted upon the next startup, since the missing module can no longer be addressed. It is only possible to start the measurement once the module has been removed from the setup using the CAN-Assistant. Even replacing a module with another which is identically configured will cause an error, since imc STUDIO will initially not recognize the replacement module. Measurement can only proceed once the new module has been exchanged and configured in the imc CANSAS interface. This means that replacement is not possible during a running measurement, even though the CAN-Bus technology supports it.

Remedy

This response can be avoided in either of two ways:

1. Direct configuration of the imc CANSAS modules using the PC-interface

If imc CANSAS modules are likely to be exchanged or removed in the course of a measurement project, it is worthwhile continuing to work with the imc CANSAS software and a CAN/PC interface. In that case, the configuration can be imported to imc STUDIO as a CBA file. Later, if any imc CANSAS module is replaced by another which is configured in the same way, the measurement procedure will not be adversely affected. If any imc CANSAS module is removed, the measurement will start without any error message being posted, but no data will be accumulated for that module's channels. This case can be monitored using either dedicated error checking for individual channels, or the Heartbeat function.

2. Setting parameters and creating CBA files

When exchanging, removing or adding a imc CANSAS module, the following steps are necessary:

a) Once all imc CANSAS modules have been configured in the imc CANSAS Assistant, open the CAN-Assistant and export the CAN-Bus configurations for each node as a CBA file.
b) Save the Experiment for later changes under a new name.
c) Open the imc CANSAS Assistant from within the imc STUDIO program and remove all imc CANSAS modules. The CAN-channels will then be removed in the CAN-Bus Assistant.
d) Open the CAN-Assistant again and import the previously created CBA file for the appropriate node.
e) Save the Experiment under a new name. The imc CANSAS channels will then be measured like before, as described in step 1. However, the checking of imc STUDIO is omitted.
f) This procedure doesn't create a database available in imc CANSAS. In order to reconfigure the imc CANSAS modules, the Experiment previously saved in step b) must be opened. Once adaptation of a imc CANSAS module has been carried out, its node in the CAN-Assistant must be saved again as a CBA file as described in step d). Then, open the Experiment saved under step e) and import the altered node into the CAN-Assistant.

8.2.2.3 Supplemental files

Supplemental files are **files saved in the device**. For instance, it is possible to **access** a characteristic curve supplemental file **with imc Online FAMOS**.

Additionally, you can also import a supplemental file **for a imc Inline FAMOS-task**. An imc Inline FAMOS-task is treated exactly the same as a device. Anything stated below about devices also applies to any imc Inline FAMOS-task.

The following supplemental file type are available:

- Characteristic curves and filter data for imc Online FAMOS / imc Inline FAMOS (*.dat)
- imc Online FAMOS source code (*.ofa)
- Synthesizer directory structures (*.dat)
- Messaging configurations (e-mail, SMS, UDP, ...) (*.msg)

Ribbon	View
Setup-Configuration > Supplemental Files	Complete

Note For imc ARGUS*fit*, the following differences apply

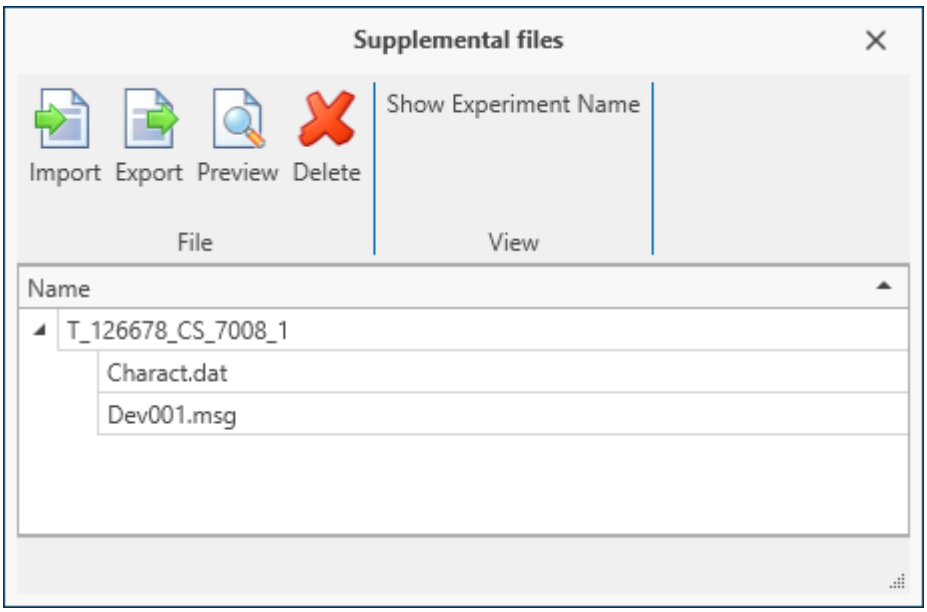
There is a separate dialog for imc ARGUS*fit* devices. The dialog is opened via the imc Online FAMOS dialog.

Menu "*Extra*" > "*Supplemental files*"

"*Characteristic curves and filter data for imc Online FAMOS*" can be imported via the dialog. A preview does not exist.

The Supplemental files dialog

In this dialog, all existing supplemental files are listed according to their associated selected device. In the example below, there is a device having a characteristic curve file (Charact.dat) and a messaging configuration file (Dev001.msg).



Dialog: Supplemental Files (example)

By means of this dialog, supplemental files can be:

- deleted,
- displayed (Preview),
- exported,
- and new files can be imported (also possible by means of the menu ribbon item "*Project*" > "*Import / Export*")

Import

- After clicking on "*import*", **select the file desired** from the file selection dialog.
- If you use **multiple devices** in the experiment, then after selection of the file, a device selection dialog appears. Here, you can **select to which devices to import** the file.

Imported supplemental files are saved in the Experiment-file. When the **measurement is prepared**, the supplemental files are **copied to the device memory** (except imc Inline FAMOS-supplemental files). In this way, it is possible for imc Online FAMOS, for example, to access the file without needing to be connected with the PC.

Renaming

Click the mouse on the name of a file to be imported. A cursor appears, and you can rename the file.

Preview

Using the function "*Preview*", an **external program** is started, which opens the file selected.

Changes which were made by means of the external program **can not be applied** automatically in the existing supplemental file. If you wish to save the changes made, save the file as a copy in a different folder, and then import this new file.

8.2.2.4 Default values

Default values are pre-set values for the configurations (parameters) of the devices and channels (e.g. saving to the device). It is possible to specify either individual parameters or whole groups of parameters.

The default values are applied when the device is selected and when creating new device variables/channels as well as Data Processing-variables (e.g. for virtual channels and Fieldbus channels).

When does it make sense to alter the default values?

When the default settings of the devices and channels are not appropriate for most of your applications.

Ribbon	View
Setup-Configuration > Default Values ( )	Complete

In this dialog, all available default values are listed.

Example:

- For all channels, the channel saving on the PC should be activated. It is possible to later deactivate data saving for each channel separately, but after having been set up, each channel should start off being saved.
- The circular buffer for the curve window should always be set to "*Undefined*" after selection of the device.
- The interval storage should always be activated after selection of the device.

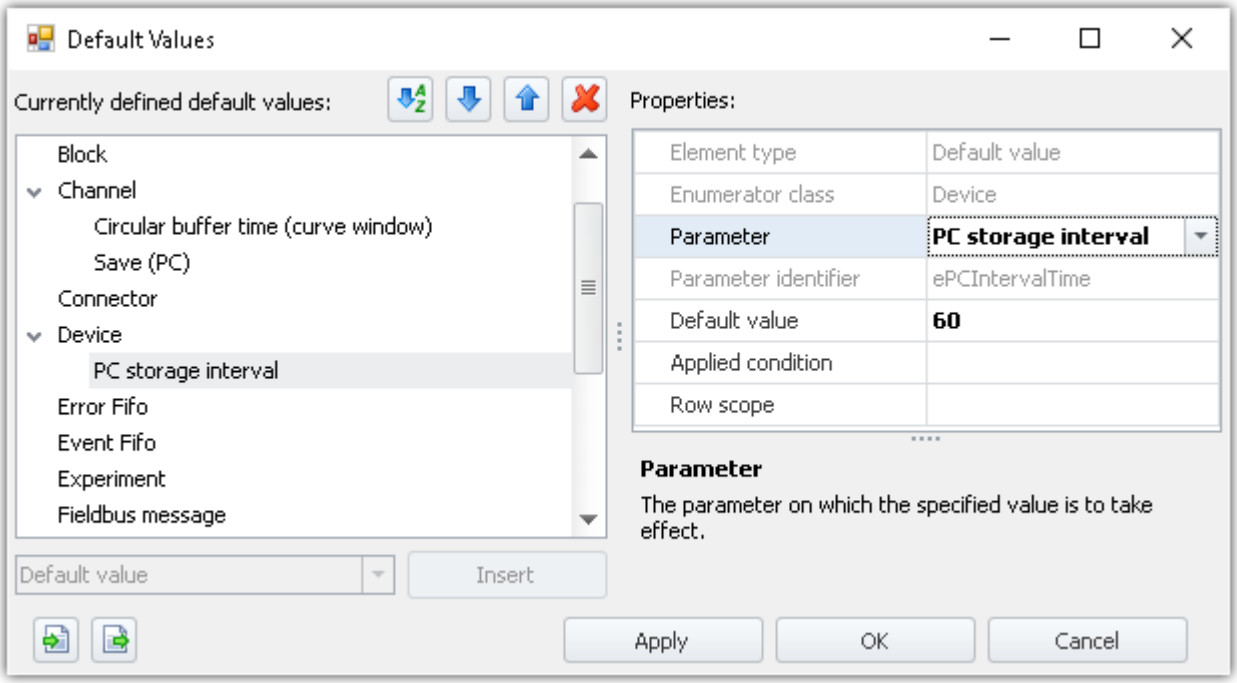
Note

Ensure that these settings are set initially (e.g. upon device selection). Subsequently, every parameter can be modified.

Reference Tip

To learn about a simple handling technique, observe the tip further below: "*Generating default values from the current settings* 217".

With the default settings, it is possible to set parameters in dependence on other parameters.



Example of default values: "PC storage interval" is set to "60 s"

Creating default values or groups of default values

In order to create "*Default values*" or "*groups of default values*", open the dialog: "*Default values*" (Ribbon "*Setup-Configuration*" > "*Default Values*").

- In the left-side portion of the dialog, select an appropriate Enumerator-class (application area of the parameters). A list of the classes and the areas they contain is presented in the chapter "*Enumerator-class* 246".
- In the drop-down selection list (lower left), select "Default value" or "Default value group".
- Click on "*Insert*" and the element is created at the location selected.
- In the "*Properties*" windows, you can set the target parameter, the default value to be set, and any other settings such as the applied condition, etc.

Properties

Property	Description
Element type	Makes the distinction between a default value and a group of default values. (write-protected)
Enumerator class	Specifies in which Enumerator-class the element belongs. (write-protected)
Parameter	The parameter to which the default value is to be applied. Select the parameter here.
Parameter identifier	The parameter's unique designation. (write-protected)
Default value	The value to which the specified parameter is to be set. E.g., when selecting a device or when creating virtual channels.
Applied condition	Here, you can define various logical conditions which allow default values to be set in certain cases. See also "*Order and dependencies of the default values* 216".
Row scope	Determines to which rows the default value is to be applied. Specifies row numbers, separated by commas, or ranges by using the minus sign. The first row has the index: "1". Example: 1-4: The first four parameters found are set. Thus for instance, an amplifier's first four channels can be modified, and the last four by specifying "5-8". The following is also possible: 1,6-8. The order does not necessarily match the table sorting arrangement displayed. For instance, after the analog channel "*[01] IN01*", the associated monitor channel appears. For this reason, use a well-defined "*Applied condition*" in order to set clear boundaries. For example, among others the channel type should be included.
Group name	The group mane displayed. The name can be entered to be particular to selected languages. The respective name is then displayed in accordance with the particular language setting. If the (non-English) name is empty, then the English name is always used.

Order and dependencies of the default values - Sorting and deleting

To **sort** entries, use the "up/down" arrow-symbols (/) or move them by means of Drag&Drop.

To **delete** entries, use the "X" symbol ( ).

Sequential order

The order of the entries affects the results. The default values are set from top to bottom in succession. Since the values can be set to depend on other parameters, this is important to take into account.

Example:

The balance action is to be set to "Bridge" for all channels having the coupling "Half bridge". However, beforehand all full bridges must be set as half bridges.

Setting up dependencies

The dependency is defined by means of the property "**Applied condition**" using the Filter Editor. Here, you can enter various logical conditions. In the example mentioned above, that would be the following logical condition:

In order for the dependencies to be language independent, they are resolved in the list into the internal parameters. In this case: ([eCoupling] = 5).

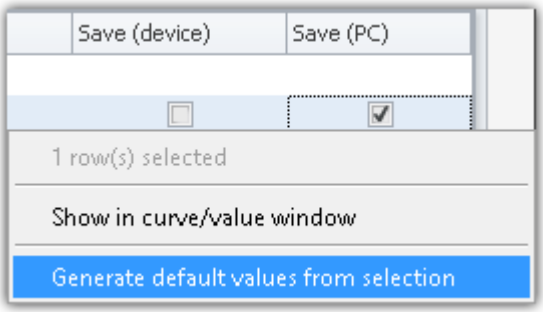
Dependencies can be defined either for each default value separately or for a whole group of default values. In this case, add a "Default value group" and add the respective parameters to the group. Define the application condition for the group. The individual default values do not require any application conditions of their own, but may also have separate conditions.

Generating default values from the current settings (tip)

It is often tedious and not so easy to set the default values correctly. Sometimes it is more reliable if the internal specifications are used to set the measurement range, for example.

In order to get familiar with the default values or to set up many parameters quickly, it is best to generate the default values from existing settings. By means of the channel table, you can define the selection as a default value.

To do this, open the context menu and select: "*Generate default values from selection*".



Context menu of a cell
Generate default values from selection

A dialog for setting the Default value group provides help in finding an appropriate condition. Select the condition for which the default value is to apply. The condition is entered in the generated group as the applicability condition.

Example:

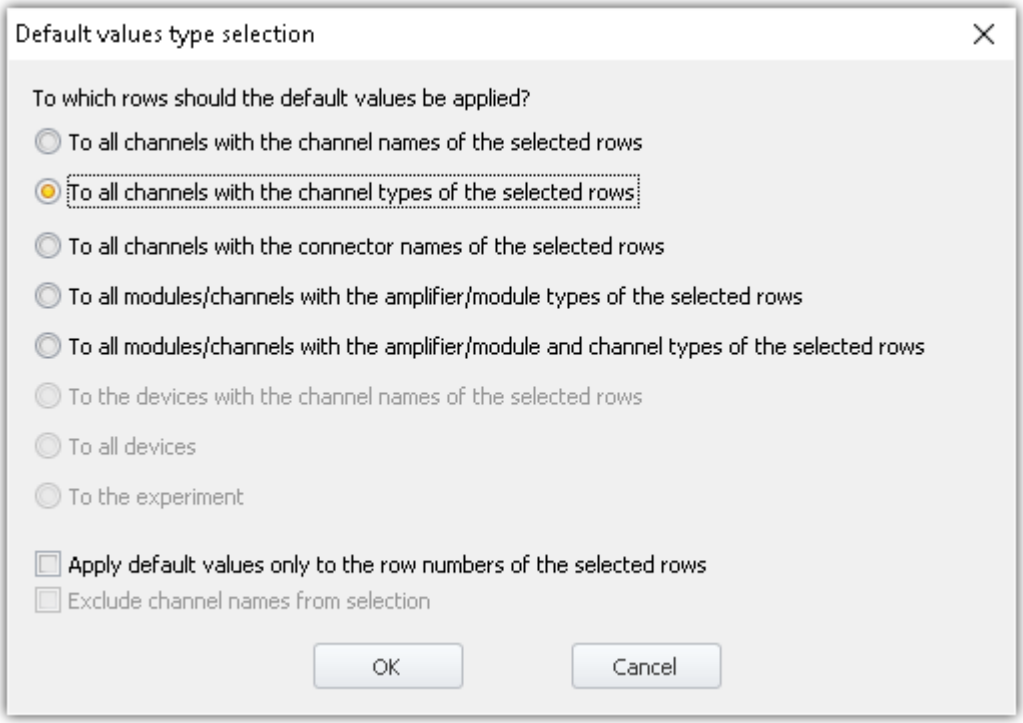
Make the following parameter settings for an analog channel:

Parameter	Value	Dialog (View: *Standard*)	Dialog (View: *Complete*)
Coupling	Half bridge	Measurement mode	Measurement mode
Input range	±10 "mV/V"	Measurement mode	Range & Scaling
Sampling rate	10 kHz	Sampling & Filtering	Sampling & Preprocessing
Save data (PC)	activate	Data transfer	Data transfer

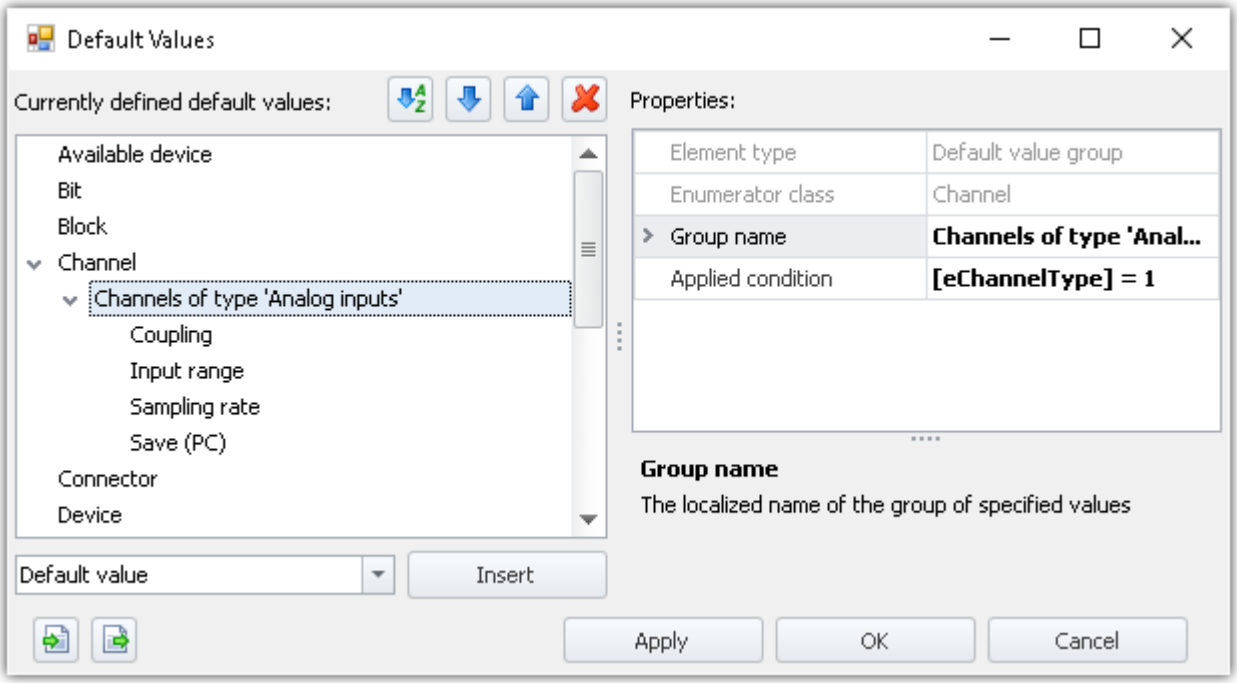
These settings are also found in the channel table in various columns. If you generate default values from a combined column 254 (one with multiple parameters), all of the column's parameter are set as the default value. In this case, either you have the desired parameter displayed 248 as a normal column and create the default value only from it, or you can later delete any default values not required.

Below, the second way is described:

- Select the (combined) columns: "Measurement mode", "Range & Scaling", "Sampling & Filtering" and "Save (PC)"
- Open the context menu and select: "*Generate default values from selection*"
- In the subsequent dialog for selecting the condition, select, for instance "*To all channels with the channel types of the selected rows*". Thus, a group is generated with the applicability condition: "*Channels of type 'Analog inputs'*"



- Open the dialog "Default values" and delete from the group generated all parameters which are not to be set.



Example: Setting default values for multiple parameters.
But only for the channels of type 'Analog inputs'

- Upon the next device selection, the analog channels are modified accordingly.

Note This example serves illustration purposes and is incomplete!

In any case, ensure whether every analog channel you use can really be used as a bridge amplifier. In that case, use the additional condition that only such channels are modified which can be set as a full bridge.

The pre-defined conditions

Conditions	Examples
To all channels with the channel names of the selected rows	"Channel name" equals "Channel_001"
To all channels with the channel types of the selected rows	"Channel type" equals "Analog inputs"
To all channels with the connector names of the selected rows	"Connector" equals "[01] IN01"
To all modules/channels with the amplifier/module types of the selected rows	"Module type" equals "Bridge, current and voltage amplifier and temperature measurement unit "UNI2-8""
To all modules/channels with the amplifier/module and channel types of the selected rows	"Module type" equals "Bridge, current and voltage amplifier and temperature measurement unit "UNI2-8"" AND "Channel type" equals "Analog inputs"
To the devices with the device names of the selected rows	"Device name" equals "imc_CS_7008_1"
To all devices	empty (thus, all devices are affected)
To the experiment	

Additional options	Description
Apply default values only to the row numbers of the selected rows	In the default values, the property: "Row scope" is set accordingly. E.g., the value "2" is entered if the second analog channel was selected.
Exclude channel names from selection	The channel name is not set as the default value even if it is selected.

Note Saving the default values

The **configuration of the default values** is saved with the respective **project**. Thus, the default values are available in all the project's experiments.
These settings are only saved if the "**project**" is also saved.

Transferring default values to other PCs or installations (import/export)

In order to transfer the default values, use the Import/Export icons(/).

Warning All existing entries will be deleted

If any default value already exist at the time of import, they will be deleted. After the import, only the imported default values are available.

8.2.2.5 Device properties

That dialog shows the configuration of the device. Most properties depend on the hardware. For example changing the type of the display is only useful, if the device is not equipped with an internal display.

Ribbon	View
Setup-Configuration > Device Properties	Complete

Please contact our tech support 8 before changing the settings (with the exception: "*Access protection*", "*Display*", "*Device time, Synchronization*").

Note

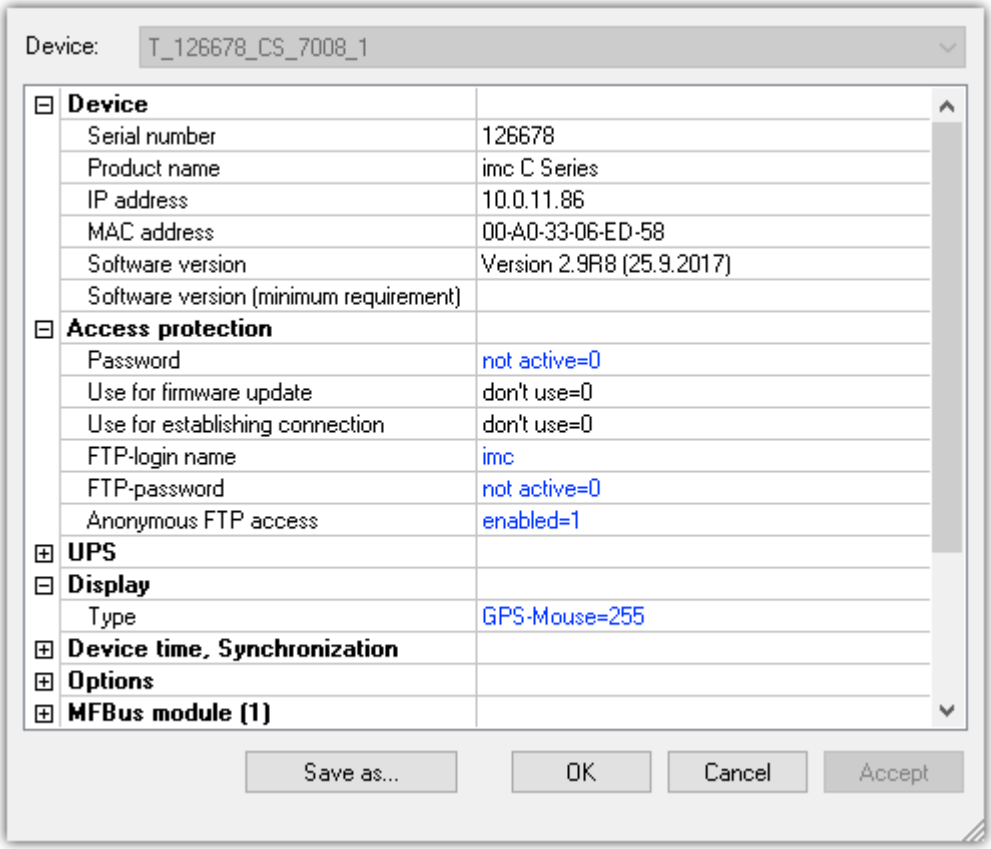
Restart

All changes of the settings need a restart of the system!

Log file

Whenever changed device properties are adopted, a text file is generated. All information on the device properties currently set is saved as soon as the button "*Apply*" in the Properties dialog is clicked.

- Storage location: Firmware folder of the device software, e.g. "*C:\Program Files (x86)\imc\imc_DEVICES_2.15R1\Firmware*".
- Name of the file: "*PropertiesLog_<Devicename>_yyyy-MM-dd_HH-mm-ss.txt*"



Device properties

Property: Device	Description
Serial number	Serial number of the device. Can't be changed.
Product name	Device types: imc C-SERIES, imc CRONOS*compact* etc.

Property: Device	Description
Software version	Version of the firmware.
Software version (minimum requirements)	The minimum required software version to run the device as configured. This prevents downgrading to a version which would not support parts of the device.
GPS receiver	Here, any available information on the GPS receiver, for example the identifier and the version, is displayed.

Property: Access protection	Description
Password	For protection against firmware update or connection setup. To delete the password, enter the valid password again.
Use for firmware update	Enable and Disable of firmware update 66.
Use for imc REMOTE WebServer	Enabling/disabling of operation via the device's Web-Interface.
Use for establishing connection	Enabling/disabling of the connection to the device (FTP, LAN, ...). When the device is newly integrated in consequence of a device search, it is necessary to enter the password twice. The first time is for establishing the connection in order to "*know*" the device, and the second time is for the purpose of entering the device in the list.
Certificate	The certificate needed for setup of a connection via https (e.g. via imc REMOTE). Here, select the certificate "imcCert.cpt", which is in the installation path under "*Firmware\Idif*", e.g. "*C:\Program Files (x86)\imc\imc_DEVICES_2.14R1\Firmware\Idif*". The certificate is always valid for one year and must be replaced upon expiration. The certificate is obtained from our tech support 8.
Expiration date	Expiration of the above mentioned certificate.
FTP-login name	Login name for FTP access
FTP-password	Password for FTP access
Anonymous FTP access	**enabled=1:** No prompt for "*FTP-login name*" and "*FTP-password*" **disabled=0:** The "*FTP-login name*" and "*FTP-password*" are used

Property: UPS	Description
Acquisition time after power fail [s]	Maximal measurement time after loss of supply voltage. The factory set time can be changed by the user. This entry has to be aligned with the hardware, thus may not be changed without consultation with imc. In case **the power of the accumulator is not sufficient**, data of devices will still be saved.

Property: Display	Description
Type	Type of display; alphanumerical, graphical and its resolution. The alphanumerical display is not supported for all devices. Devices having an internal display can only control one type. Technically an internal display is a "normal" display with a mounted cable. So even for these devices the display type can be changed. That connector could be used for a **GPS-mouse**, too. To change it, select the entry "*GPS-Mouse=255*". That option is not available for all devices. In case of doubt contact our tech support 8. After switching to GPS, all GPS signals can be received be the process vector variables.

Property: Device time and synchronization	Description
Time zone	Sets a time zone. In order to make it possible to assign a time to measured data, the measurement can only be prepared when a time zone is set for every imc device used.
Automatic daylight saving time switching	• **activated=1:** the device automatically switches to/from daylight saving time. • **deactivated=0:** the device uses standard time all year round!

NTP	• **Server (1), (2):** IP address or name of server
	• **max time discrepancy [ms]:** States the permitted time discrepancy from the NTP-Server in ms, up to which the device is still considered synchronized. When this discrepancy is surpassed, the message – "*Not synchronized*" or "*Synchronization in progress..*".. is displayed.
	For the sake of the measurement precision and comparability of measured data, we recommend increasing the value of this entry if the discrepancy is big. Otherwise it will be necessary to dispense with synchronization completely.
	• **Interval[s] (min;max):** Specifies the minimum and maximum time for the synchronization intervals, to which the following applies:
	  • Short intervals provide higher accuracy, but make more demands on the network.
	  • Long intervals may produce lower accuracy, but make less demands on the network.
	Accuracy:
	• If the external conditions (mainly temperature, network resource requirements) only fluctuate slightly, then the accuracy may be high even with long intervals. Shorter synchronization intervals should be selected if, for instance, frequent temperature fluctuations are to be expected.
	• NTP-Standard: min = 16 s, max = 1024 s. This value is also used if nothing was entered. This is adequate for constant temperatures, e.g. for installation of the devices in a cabinet.
	Example: NTP setting: min=8 s, max=64 s
	Conditions:
	• Devices were exposed to daytime-based temperature fluctuations, in other words, they were not installed in a cabinet.
	• Use of a company's internal NTP-server
	Result: Among 3 devices, there are signal timing discrepancies of 0.01 .. 0.2 ms. This value is far below the uncertainty level stipulated in the standard IEC 61000-4-30 and enables very good comparison of measurement signals.

Property: Device time and synchronization	Description
Max. time to wait for synchronisation [s]	• **0**: unlimited wait (default) • **-1**: don't wait • **xxx**: wait xxx seconds **Note:** • Select a waiting time which is certain to work for synchronizing with an operational synchronization input signal. Otherwise, after the waiting time the measurement will begin out of synch. However, this can only be subsequently recognized in the measured data if you record the Sync state along with the data by means of the imc Online FAMOS function `IsSyncronized`. • For a GPS Master with DCF-77 Slaves, a waiting time of 300s is recommended.
Synchronization signal input	Specifies a default clock to be used upon activation. If the value for synchronization signal is n.s.=0, the settings remain intact after activating the device. • 0 none: not defined. • 1 no synchronization signal: Synchronization signal input not evaluated • 2 SYNC synchronization signal: only DCF or IRIB-B is evaluated • 3 GPS synchronization signal: only GPS receiver evaluated • 4 SYNC or GPS synchronization signal: depending on which signal is applied first upon deactivation, either DCF, IRIB-B or GPS is utilized • 5 NTP synchron signal: Network Protocol used • 6 EtherCAT synchronization signal • 7 PTP: Precise These setting are applied in the experiment when connected with the device and there are no other settings regarding Synchronization 292 had been specified for the experiment.
Synchronization signal output	Specifies whether the device time is outputted as a DCF/IRIG-B signal.

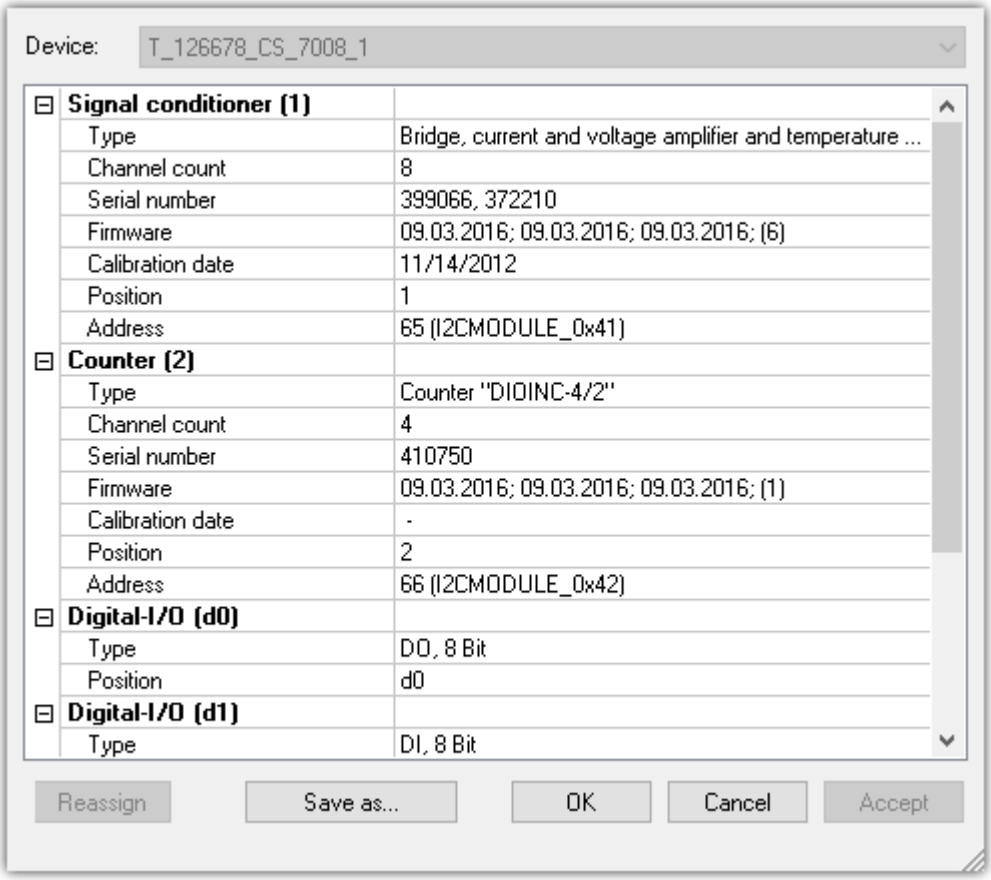
Property: Options	Description
Some software options come at an extra charge. These options are enabled by means of so-called **enabling codes**. The enabling code is a multi-digit number and is determined for each device on the basis of its serial number. Options can be enabled at factory and be available when the device is delivered (see the decal on the device frame new the serial number plaque). The options can be enabled after delivery. Enabled options are noted in the device's configuration (Flash-EEPROM) and thus are permanent and cannot be lost.	
imc Online FAMOS	Enable Online processing with imc Online FAMOS 851 base packet.
imc Online FAMOS Professional	This option comprises multiple subparts ("Optimization of the online programs by use of the on-chip memory", "Online Synchronous task", "Process vector"). The availability of these function option subparts, however, depends on the particular hardware configuration and is determined upon connecting with the device (see the chapter on "imc Online FAMOS 851").
Online classification	Enables functions of Online Class-counting Kit (see the chapter on imc Online FAMOS 851).
Online-Order tracking	Enables functions of Order Tracking Kit (see the chapter on "imc Online FAMOS 851").
CAN-Database Import	Enables Vector-database access, DBC-format; order option */VEC-DATB (license per device)

Property: Options	Description
imc STUDIO Monitor	Shows, if the devices is prepared for imc STUDIO Monitor. That entry can not be changed.
ECU-protocols	Activates the functions for CAN-bus subscribers which support the ECU protocol (see the chapter on "imc Online FAMOS 851" and the "imc Online/Inline FAMOS function reference 896").
Measurement time (T3) until suspend (s)	Only available to devices having a sleep-mode. Describes the minimum duration for which the suspend signal must be applied.
imc REMOTE	Activation of imc REMOTE LinkSecure or imc REMOTE WebServer. This requires an activation code which must be purchased along with the device. It is also possible to perform the activation subsequently. • **imc REMOTE LinkSecure:** Option for imc LINK providing secure access via an https connection • **imc REMOTE WebServer:** Enables imc REMOTE WebServer 822 which provides platform-independent remote access to imc measurement systems. If imc REMOTE WebServer is enabled in the device properties, a connection is permanently occupied. The maximum number of connections to the device depends on the device - see "*Device overview* 187".

8.2.2.6 Module properties

The dialog lists all amplifiers equipped with the device.

Ribbon	View
Setup-Configuration > Module Properties	Complete



Module properties

Along with the "*type designation*", "*serial number*", "*firmware*" and "*calibration date*", you will find the "*address*" (with bus designation) and the resulting "*position*" of the amplifier.

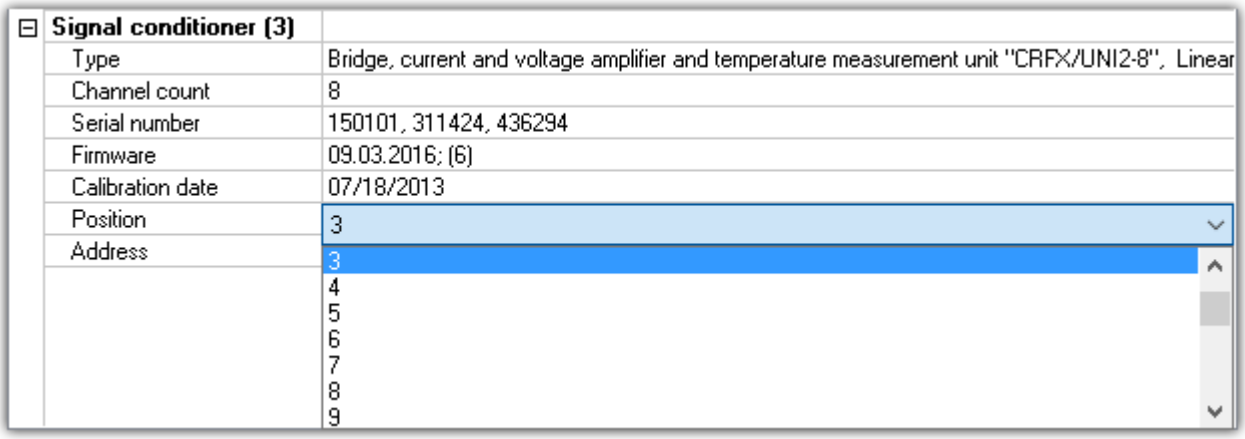
By using "*Save as...*" a file is created which lists the connected amplifiers

There are two groups of amplifiers:

- **Permanently installed amplifiers** use an address switch on the amplifier, which is set upon delivery of the system. With some devices, this can be changed by trained users (see "Device manual").
  A permanently installed amplifier is distinguished by the address designation: **I2CMODULE**.
- **Modules** (e.g. imc CRONOSflex or CRONOS-XT modules 343) are integrated with a master (e.g. CRFX/CRXT Base unit or with an EtherCAT Master (ECAT-DA) for imc CRONOS*compact*). The obtain their module address automatically. The numbering of the module occurs upon first connection and remains intact. Along with assignment of the module number (Position), a "ScanID" (identifier for the module) is also allocated. A 7-segment display represents the module address.
  A CRFX/CRXT module is distinguished by the address designation: **xbus**.

Assigning the module number

- **Reassign**: The button "*Reassign*" overwrites the existing numbering of the modules with an uninterrupted re-numbering.
- **Assign manually**: Enter the position manually. Click the number and choose a position from the list.



Assign manually

Warning

Note that changing the module configuration is equivalent to re-ordering the amplifiers. If a change in the order of module entries is made, any experiments present can no longer be used.

Removing CRFX/CRXT modules

If an module is removed, the device description retains the module numbering. A gap results in the numbering. Since the device description is saved along with the experiment, the old experiment can easily be reused in any later application of the same module.

Exchange of CRFX/CRXT modules

The device description identifies an module by its serial number. After exchanging an module with another of the same model, the new module is recognized as such by its serial number and receives a new address. In this case, correct the address (*Assign manually*), **before** resuming the use of existing experiments.

Note Notes on assigning the module number

- If an CRFX/CRXT module is alternately used with different imc devices, it saves the module address for each measurement device separately. The number of module addresses is limited to 63.
  If the number of possible module addresses is exceeded, the oldest entry is deleted.
  A module can thus have different module addresses in different device configurations. The use of the same module in different devices is thus possible without needing reassignment of the module address each time.
- If the ScanID matches the module address saved on the module is used.
- If there is a conflict in the ScanID (multiple modules having the same module address), then *Reassign* of the module addresses is performed.
- For all other modules (non-matching of the ScanID), the module addresses are reassigned.
- When assigning module addresses, the last address used is applied. The newly assigned addresses follow (are appended to the end of) the last address assigned.
- When the module addresses are reassigned, a new ScanID is made and all modules are assigned a new address.

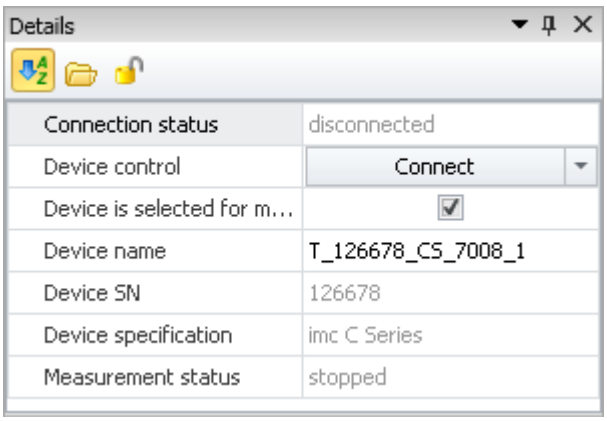
Note Notes on changing the device configuration

- After the device software connects with a device, it is at this time not possible to take any changes of the device configuration (removal and/or addition of modules) into account (e.g. event notification with subsequent adaptation of the settings dialogs).
- If the device configuration is changed, changes are only taken into account after separating from and reconnecting the device with the device software
- During a running measurement, it is not possible to respond to changes of the device configuration (no "hot plug")!

8.3 Tool Windows

8.3.1 Details

The content of the "*Details*" window depends on which page is selected. As in the "*tabular display* 231", you can also directly edit 239 parameters in this tool window.



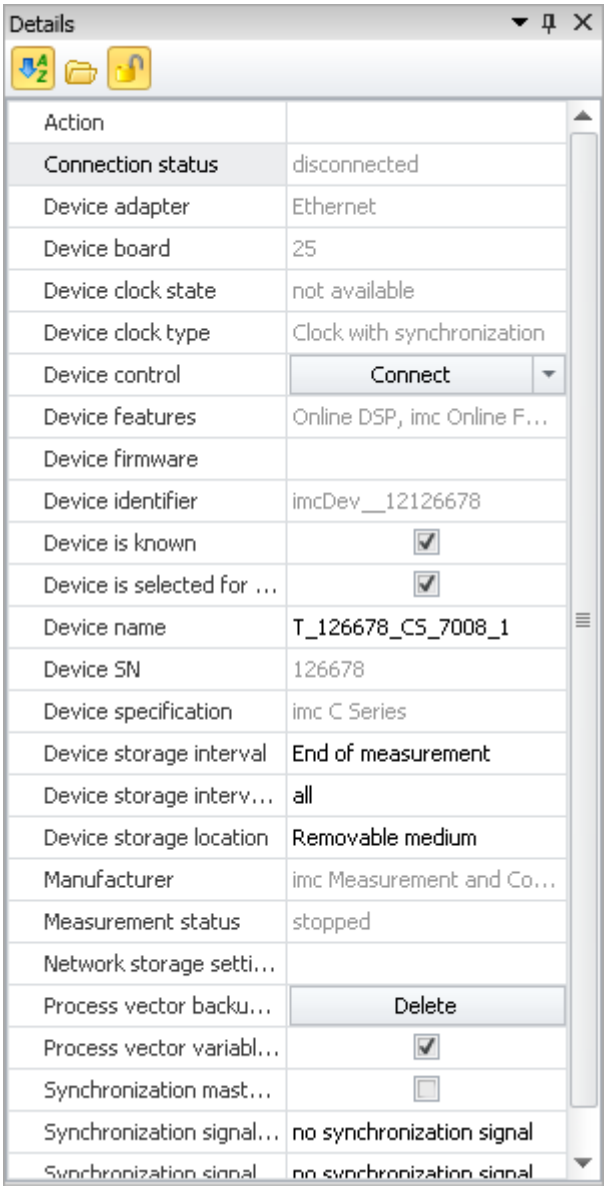
Tool window "Details" in the plug-in "imc STUDIO Setup" (example)

Hidden parameters

By default, the Details window sows a selected set of parameters which can also be seen in the tabular display and the dialogs. By clicking on the **padlock symbol** ( ), all other parameters are also displayed which belong to the selections in the tabular display ("hidden parameters").

Example

When you click on the padlock symbol, you see the much more extensive list of all device parameters. The image below shows only an excerpt:



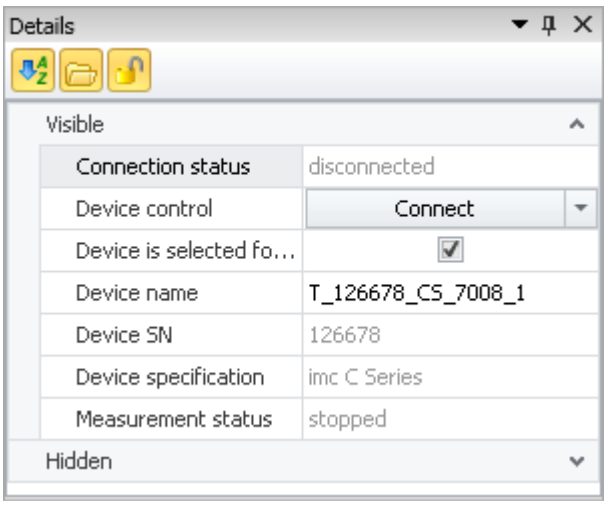
"Details" tool window for a selected device with display of the hidden parameters

In this display, you can also set parameters which are not available either in the tabular display or in the dialogs.

Grouping / sorting parameters

The parameters in the "*Details*" window can also be displayed in groups. To do this, click on the grouping symbol ( ). The parameters are then divided into the following groups:

- Visible
- Hidden



"Details" tool window in grouped display

The groups can be opened or closed individually (symbols:).

To sort the parameters alphabetically, click on the sorting symbol ( ). Click again to change the sorting order.

8.3.2 Sensors

The tool window "*Sensors*" presents all information on existing sensors and characteristic curves.

Here is where you find information on the sensors used. Additionally, you can create and edit "*user-defined characteristic curve*". If imc SENSORS is installed, the sensors from this database are shown and are available for use.

Reference

See "*Setup pages - Configuring device*" > "*Sensors, Characteristic Curves and TEDS* 729"

8.3.3 Layout repository

Complete layouts: Her you will find a list of all available Setup pages. In order to display a page, use Drag&Drop to move it to the tab-bar (or use the context menu - see "*Additional Pages* 407").

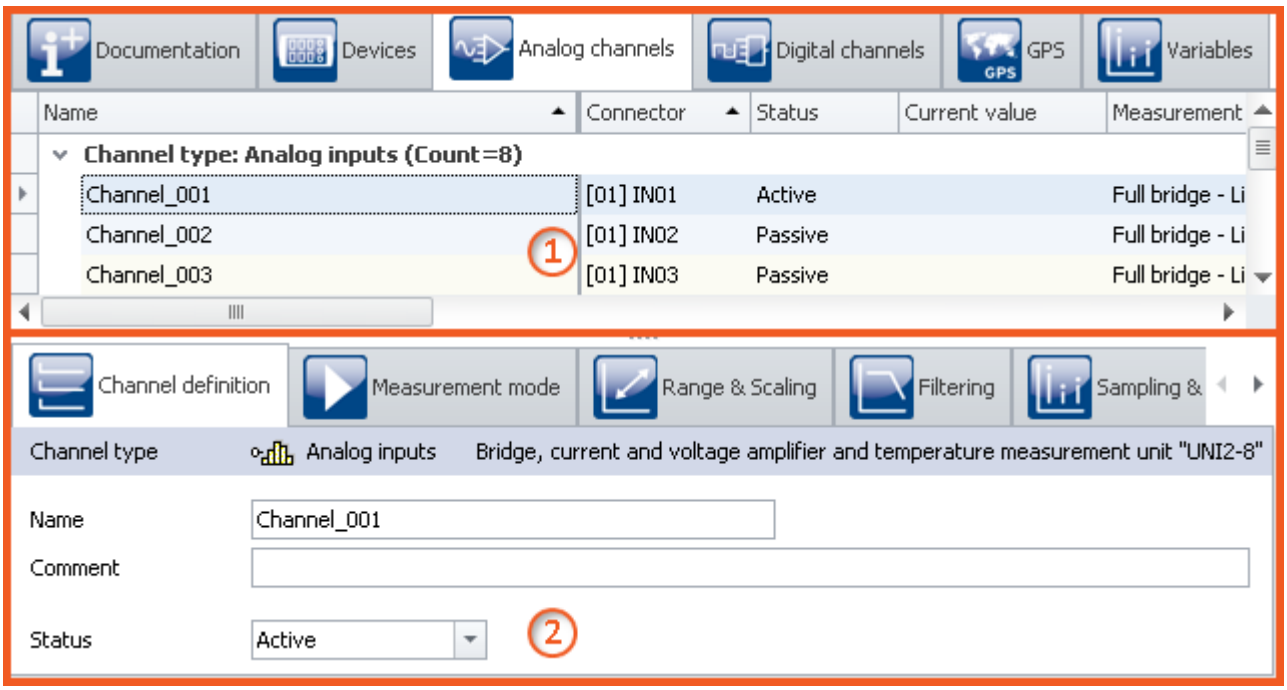
For the editions imc STUDIO Developer:

Complete layouts: Using the context menu, you can modify the settings of the saved pages (see "*Properties of Pages/Complete Layouts* 275").

Pages and Elements: Here you will find elements for modifying and designing your own personal pages (see "*Designing Pages* 276").

8.4 Operation

The Setup main window is subdivided into several predefined pages. Some pages are divided in two windows stacked vertically, as shown in the example below:



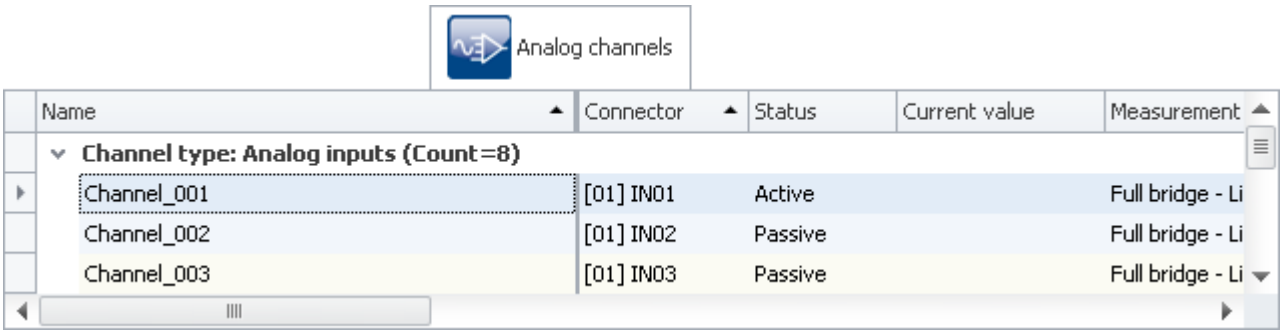
Tabular display and dialog (example)

Region		Description
1:	Tabular display 231	The top window shows a tabular (rows/columns/cells). In the table, e.g. all known devices or channels/variables are listed. It offers a quick overview of parameters and can be filled with many columns.
2:	Dialogs 238	The lower window shows special dialogs 238 for the most important parameters.
	Assistenten 240	The Assistants are an aid in configuring particular device components such as Fieldbusses or imc Online FAMOS.

Editing: You can modify the parameters by means of in-place editing 234 in the table, as well as in the dialog 239. To edit the height of the window, use the mouse to drag the dashed separator line.

8.4.1 Tabular display

In the tabular display, many parameters of the selected page are displayed in a segmented table.



Example of tabular "Analog channels"

You can edit 234 both individual parameters and sets of multiple parameters (see Selection 232) directly in the table cells. The availability and selection of settings depends on the page opened and on the selection in the table.

Other parameters: In principle, all parameters can be configured using the tabular. For this purpose, you can use the Column Chooser 235 to include the desired parameters in the table. Also, you can configure complex columns using the function Additional columns 251.

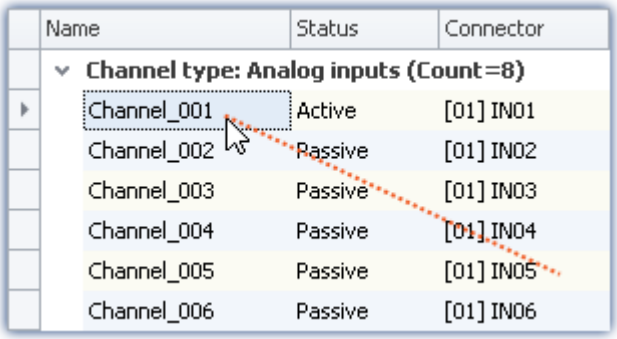
8.4.1.1 Selection

In the tabular display, you can edit both individual parameters as well as sets of multiple parameters **directly in table cell**.

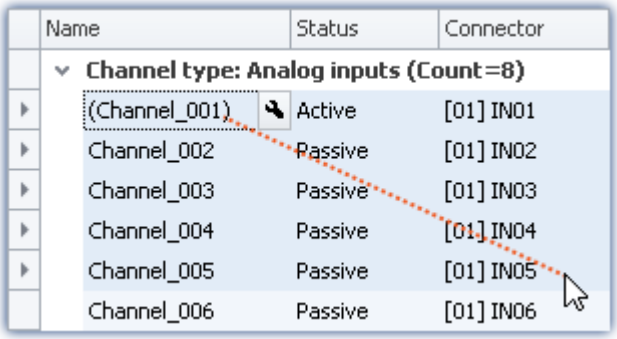
To do this, first select the row(s) which contain the parameters.

Selection with the mouse

To select contiguous cells, you only need to drag open a rectangle over the desired region with the mouse.



While holding down its button, drag the mouse over the region



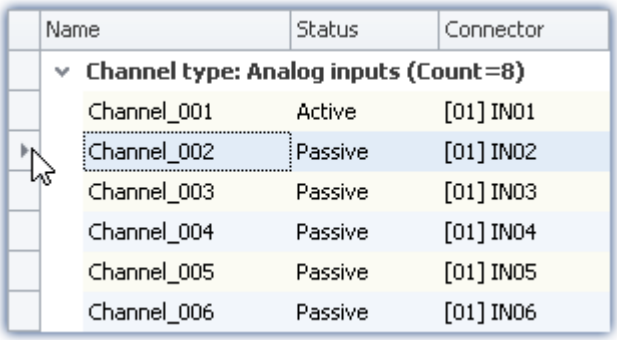
Release the mouse button at the position desired

Selection with mouse and keyboard

To do this, use the left margin column, as shown in the first image.

Selecting individual rows

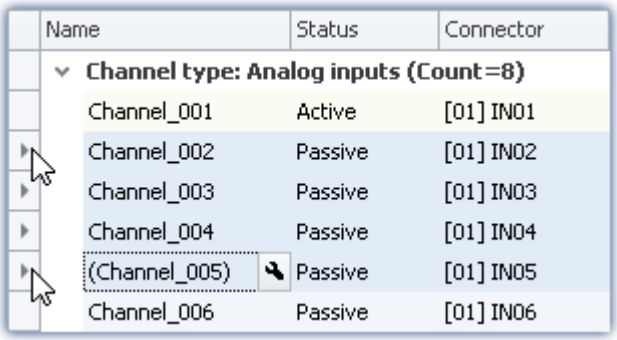
Click on the left edge of the desired entry:



Single line selected
(simply click on the edge of the table)

Selecting multiple, consecutive rows

Click on the first row which you wish to select. Then hold down the ***<Shift>*** key while clicking on the last row you wish to include in the selection. This selects all the rows from the first through the last.

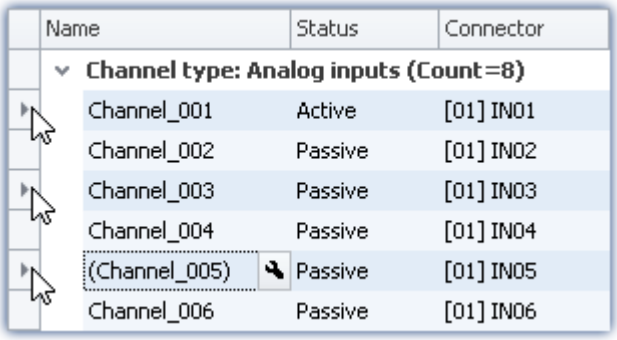


Selecting multiple consecutive lines
(hold down <Shift>-key)

You can also perform this procedure repeatedly (in multiple steps). To access more entries, you can use the scroll bar to move through the table. This means that you can also select lines in the table which are so far apart that they are not visible at the same time.

Selecting multiple rows separately

Click on the first row you wish to select. Then hold the ***<Ctrl>*** - key down and click on the next row which you wish to add to the selection. Each row you click on while holding down the ***<Ctrl>*** - key will be added to the selection.



Selecting multiple lines separately
(hold down <Ctrl>-key)

You can repeat this procedure (in multiple steps). To access more entries, you can use the scroll bar to move through the table. This means that you can also select rows in the table which are so far apart that they are not visible at the same time.

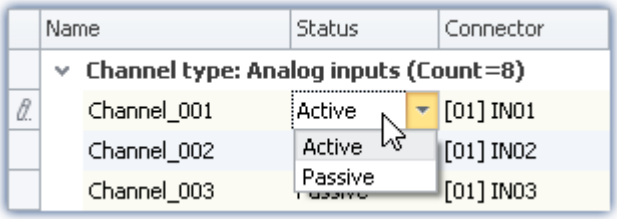
Cancel selection

To cancel a multiple selection, click on the left side of a single entry in the table

8.4.1.2 In-place editing

All parameters for which there is a column in the table can be edited in place (directly in the table). Note that there are dialogs 238 for selected parameters.

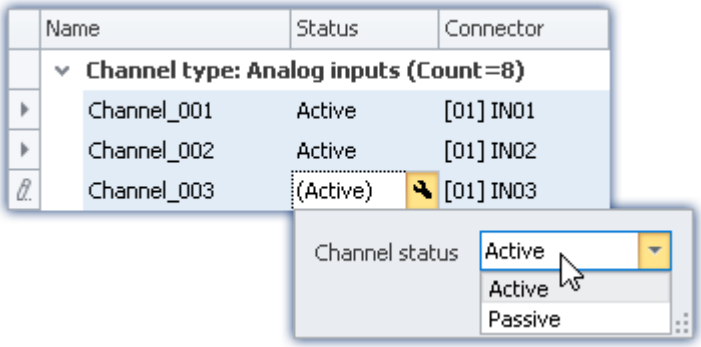
Editing single parameters



Editing single parameters: click on the row

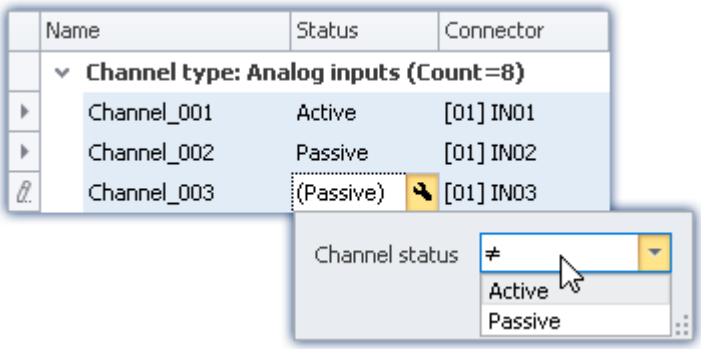
In order to edit individual parameters, click directly in the associated cell of the table. It is not necessary to select the line beforehand. After clicking, a control appears with which you can edit the parameters. The control's type depends on the particular parameter concerned. In the image above, you see a drop-down list for selecting the desired channel status (Active/Passive).

Editing multiple parameters



Control for multiple selection (example)

Select the desired row as explained in the previous chapter (Selection 232). Next, click on one of the rows where you wish to edit parameters. A control appears in which the corresponding parameter can be edited. If all of the parameter values are the same, the value is indicated; otherwise, an inequality - symbol (≠) is displayed:



Different parameters in a multiple selection procedure (example)

8.4.1.3 Configuring columns

By default, the various pages of **Setup** show a certain selection of columns (parameters). You can also configure the selection and order of the columns according to your own wishes (see "*Showing and moving columns* 248"). The column configuration is saved/opened with the view configured (see section "*imc STUDIO (general)*" > "*Views* 155").

Note Saving / restoring a configuration

In order to save a column layout, use the functions belonging to the View menu (see the manual imc STUDIO (general) > chapter "*Views* 155").

8.4.1.4 Sorting and grouping

Note Saving the sorting and grouping arrangements

The **sorting/grouping arrangements** are **saved with the views**. The views are **saved along with the project**. This means that any views are available for all of the project's experiments.

- See: "*What is saved where?* 162"
- See: "*Views* 155"

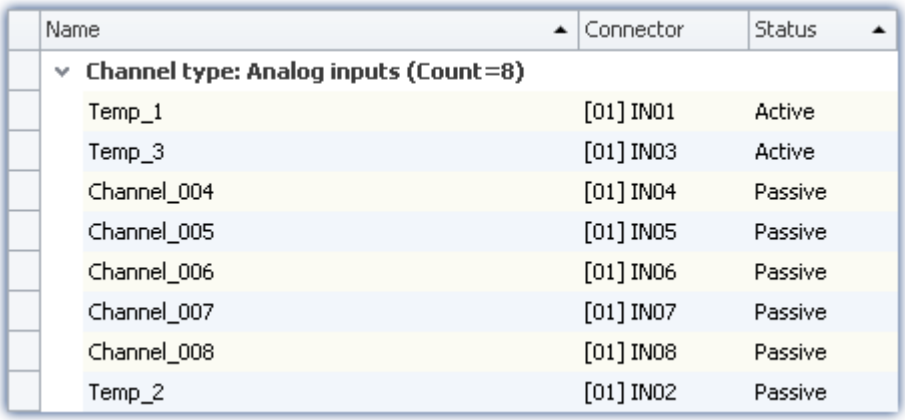
Sorting

Clicking the mouse over a column header causes the table to be sorted according to that column's content. The direction of the arrow at the right margin of the box indicates the sorting direction:

Reversing the direction: Each click of the mouse toggles the sorting direction to the respective opposite direction.

Undoing the sorting: Open the column header's context menu and select "*Clear Sorting*".

Sorting according to multiple columns: Activate the sorting according to an additional column by clicking the mouse while holding down the <SHIFT>-key.



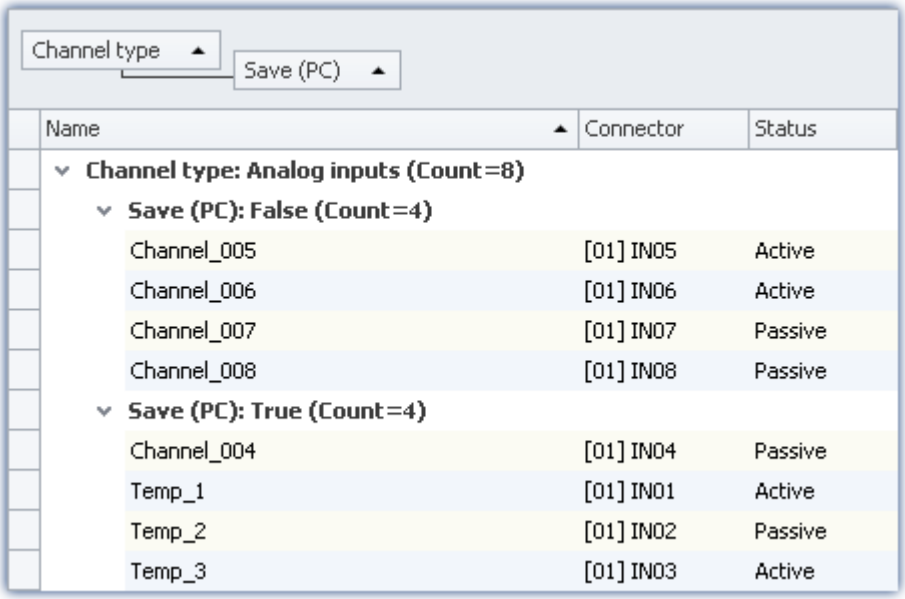
Name	Connector	Status
Channel type: Analog inputs (Count=8)		
Temp_1	[01] IN01	Active
Temp_3	[01] IN03	Active
Channel_004	[01] IN04	Passive
Channel_005	[01] IN05	Passive
Channel_006	[01] IN06	Passive
Channel_007	[01] IN07	Passive
Channel_008	[01] IN08	Passive
Temp_2	[01] IN02	Passive

The table is sorted according to the parameter "Status" and next by "Name".
Active channels are thus always at the top.
All active and passive channels are sorted alphabetically among themselves.

Grouping

By default, most tables are grouped according to one of the parameters. For instance, the channel tables are grouped according to the parameter: "*Channel type*". You can, if desired, cancel the grouping or modify it.

To do this, open the context menu in the upper left corner of the table and select the item "*Group By Box*". The box is then displayed above the table. Here, you can add table columns using Drag&Drop.



Example of grouping according to the channel type and next by the saving status.

Disadvantage: Any columns according to which groups were formed can no longer be displayed in the table. If this is required however, then create an additional column of the type : "*Combined column* 251". This workaround lets you show the parameter after all. (Creating/editing of the columns is only possible with at minimum the imc STUDIO PRO 29 product versions.)

8.4.1.5 Filtering columns

Note — Saving the filter settings

The **filter setting** is **saved with the views**. The views are **saved along with the project**. This means that any views are available for all of the project's experiments.

- See: "*What is saved where?* 162"
- See: "*Views* 155"

To select a filter you use the header of the column to which the filter is to be applied. When you move the mouse over the column header, the filter-symbol appears at right.

Clicking the mouse over the filter-symbol causes a selection menu to appear. Here you can select a value from among the current settings, which is to be the only one still displayed. Or, you can define your own filter ("*Custom*").

Once you have defined a filter, a filter bar appears on the bottom.

Filter-bar

Here, the filter currently set is shown. In this example, all columns which use either "*Full bridge*" or "*Half bridge*" coupling are used.

Removing filters: You can remove a filter by clicking on the "X", or by using the filter-symbol in the table header: select "*All*".

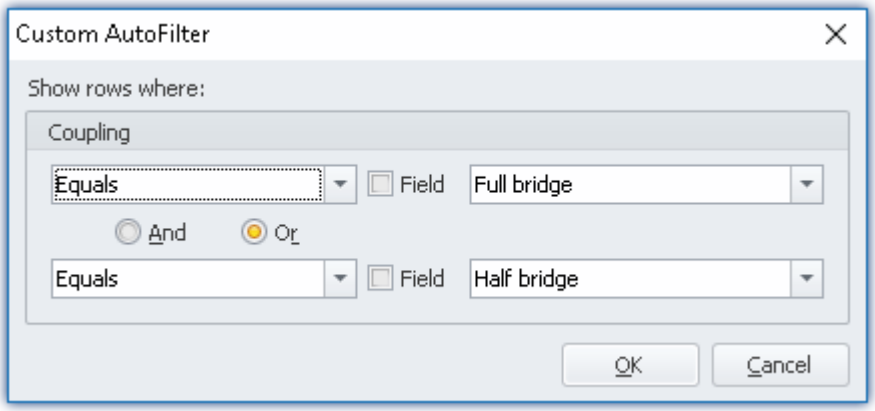
Activating/deactivating: Using the checkbox in the bar, you can activate/deactivate the filter without removing it.

Changing filters: Clicking the mouse over the Drop-Down symbol causes a list to appear, which offers the most recently used filters for selection. Here, you can select a different filter.

User-defined (custom) filters

If you can't find the appropriate value by means of the filter-symbol, or if you require different conditions, you can implement "*Custom*" entries.

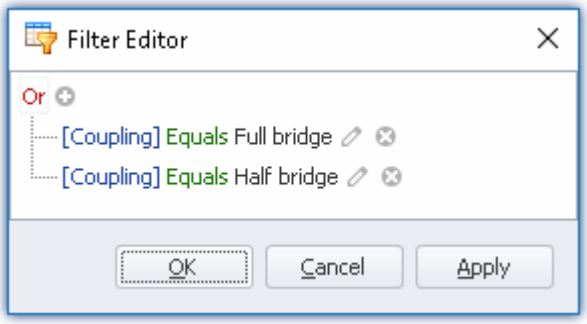
The following dialog appears. In this dialog, configure the filter as desired and then click on "OK". Note that the filter only works with exact values/texts.



Custom filter

Editing filters

In order to define complex filters, click in the Filter-bar on "*Edit filter*". The following dialog window then opens:



General Filter-dialog

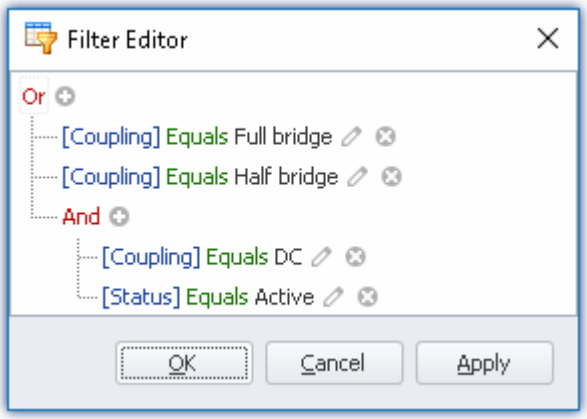
Using this dialog window, you are able to formulate complex combinations of filter conditions.

Changing the group's operator: When you click on the operator which is already set for the group, a list of operators then appears, from which you can select the appropriate one: "And", "Or", ...

Adding a condition: By clicking on ⊕, you can add any desired number of filter conditions to the operator group. In doing so, be sure to follow the pattern of placing the parameter (e.g. coupling) at the first position in the condition statement, then the condition itself in the second position (e.g. "equals" or "Does not equal", ...), and finally at the third position the "value".

Deleting: Clicking on ⊗ lets you delete filter conditions.

Adding a group with an additional operator: Click on the operator set for the group. In the list which then appears, select "*Add group*". The new group can have a different operator type and can in turn contain any kind of conditions/groups.

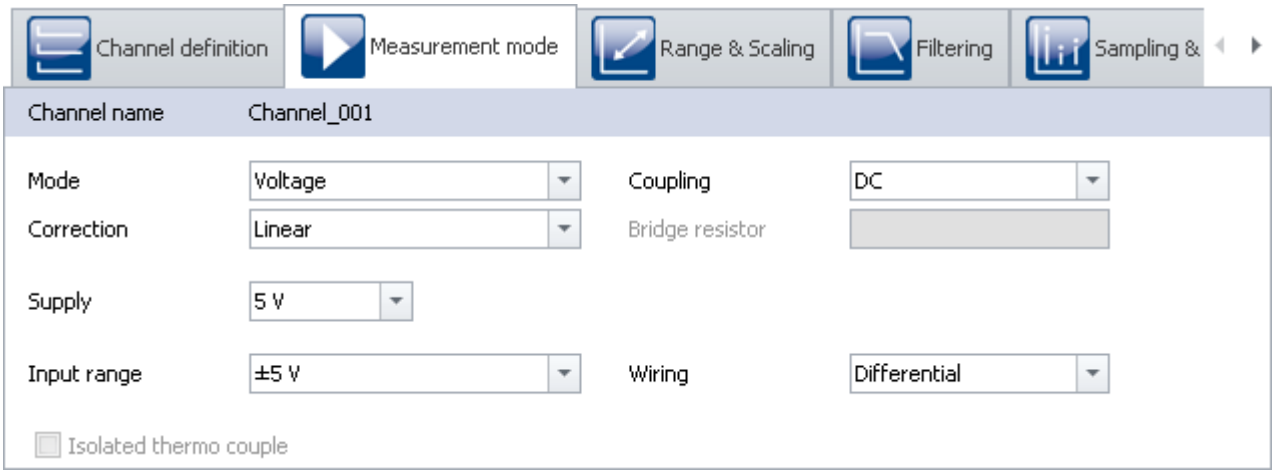


Filter setting with two groups
Coupling = "Full bridge" OR coupling = "Half bridge" OR
(coupling = "DC" AND Status = "Active")

Dependency on other parameters: Instead of entering a value according to which to filter, you can also select another parameter. This means you are able to add dependency relationships to other settings. To do this, click on the pencil-symbol or on the box-symbol next to the value to change.

8.4.2 Dialogs

In this section, the most important parameters of special dialogs are offered.



Example of "Measurement mode" dialog

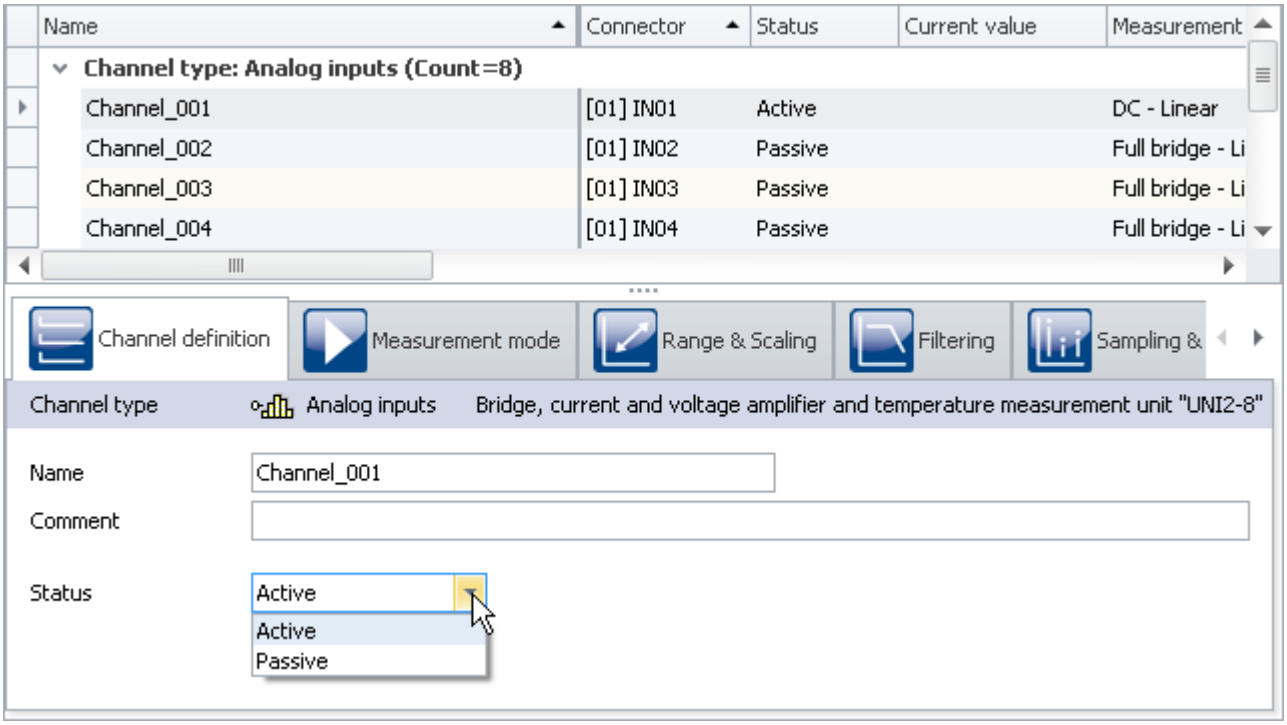
Selected parameters are presented here, distributed over multiple dialogs. The dialogs are designed in a clearly structured manner. As with the tabular display, you can also directly edit 239 parameters in the dialogs.

Availability and selection of the dialogs and the settings depends on what page is open and on the selection in the table.

8.4.2.1 Direct editing

You can directly edit (change) all parameters displayed in the dialogs.

Editing individual parameters



Editing individual parameters (example)

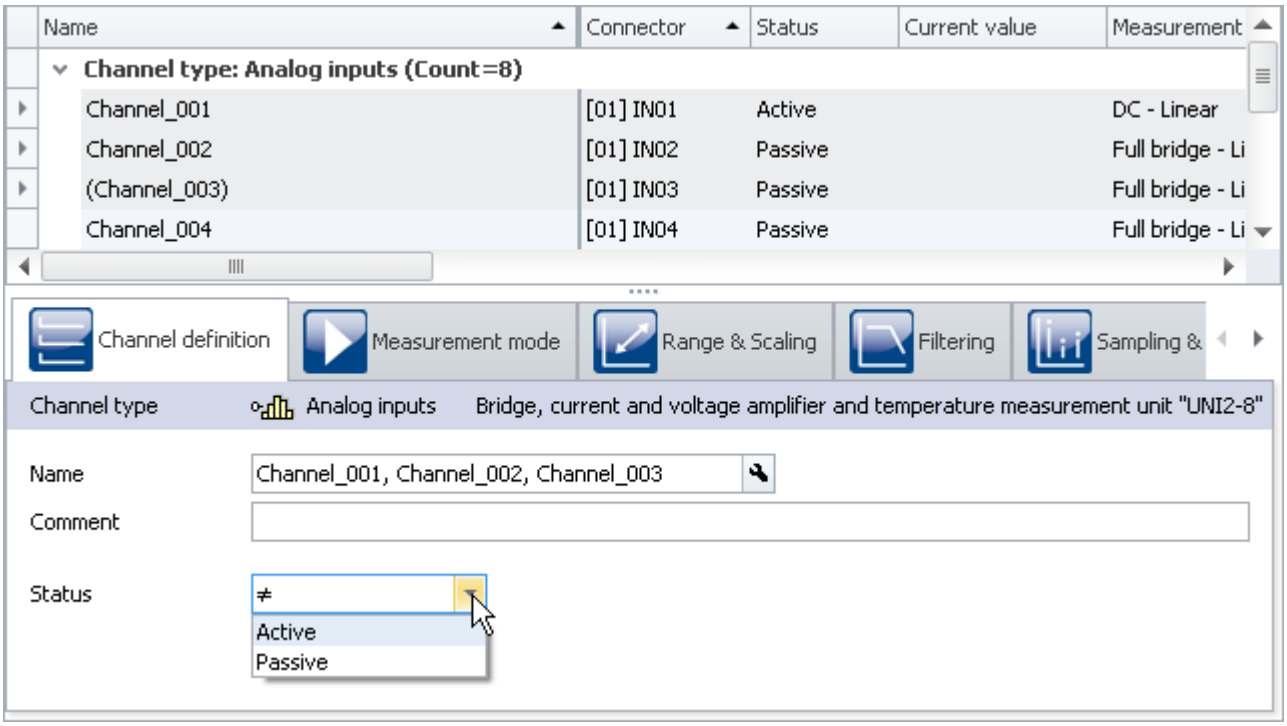
To edit a parameter, first select the corresponding table cell in the tabular display (see Selection 232). The dialogs associated with the selection are indicated. Open the dialog in which the parameter to be changed is found.

Click on the control with which the parameter can be edited (changed). The control's type depends on the parameter. In the image above, you see a drop-down selection list for setting the channel status (active/passive).

Editing multiple parameters

When a parameter in in a dialog is edited, then this change causes changes to all rows in the tabular display selected. If all parameters have the same value, this is displayed, otherwise an "unequal" symbol is shown (≠).

In the image shown below as an example, you see three selected channels and the dialog *channel definition*. If you select the **status** *active* in the drop-down dialog, the status for all three channels is set to *active*. The results are shown immediately in the channel table.



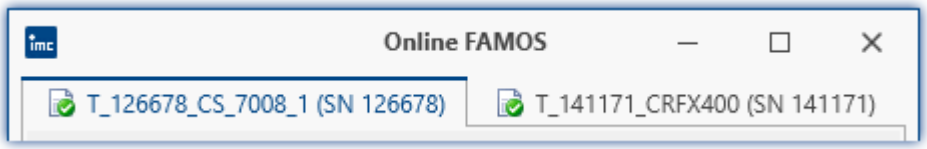
Non-matching parameters in a multiple selection (example)

8.4.3 Assistants

The Assistants are an aid in configuring particular device components such as Fieldbusses or imc Online FAMOS. The methods of operating and configuring the respective Assistants are presented in the associated chapters.

All Assistants are shown in a dialog frame by means of which you can select the device affected. If only one device is involved in the experiment, there is nothing for you to do; that device is automatically selected. But if there are multiple devices in the list, then use the tabs in the upper region of the dialog to select the device you wish to configure.

For each device a separate tab appears.



The symbols appearing in front of the tabs provide an indication of any name conflicts (red X-symbol). A green check-mark indicates that there is no name conflict.

8.5 Performing a measurement: Procedure

This chapter describes how the measurement procedure (the experiment) is controlled. Many functions can be accessed via the Ribbon **Home** and all via the Ribbon **Setup** 188.

Action		Description
	Connect signal leads	
	Searching for and selecting devices 241	Before you can create an experiment, you must adopt one or more devices into the device table. To do this, run a device search and selection.
	Firmware Update 63	The software only works with devices having the right firmware. Upon the connection/preparation, the versions are compared. If they don't match, the firmware update is carried out.
	Configure devices 288	Each measurement setup can require a variety measurement device settings. Configure your device and each measurement channel according to their respective requirements. For this purpose, various software Assistants (wizards) are provided.
	Process configuration 190	Current configuration is inspected for errors and made available to all the plug-ins. No connection with the device is required for this.
	Connect 190	imc STUDIO connects itself with all measurement devices selected (generally via LAN).
	Balance	For amplifier channels: balance the channels. When a system having amplifier channels is switched on, or the gain, gain function or filter function is changed, the offset needs to be balanced.
	Prepare 195	The current configuration is exported to the device.
	Start measurement 197	The measurement is started for all devices.
	Stop measurement 197	The measurement is stopped for all devices.

8.5.1 Device Search - known and selected

Term	Description
Search for devices	In a device search, the system searches the network for all suitable devices. The "*device table* 290" lists all devices found.
Device is known	"*Device is known*" means that the device is already available for selection as soon as the software starts. The device's setup is familiar to the software and an experiment can created without needing to connect with the device.
Selected	"*Selected*" means that a device is to be used for the current experiment.

Making a device known

To make a device known, you must previously have launched a device search for it. Perform a "*device search* 199" ( ). The system then searches for devices in the network.

Ribbon	View
Home > Search for devices ( )	all
Setup-Control > Search for devices ( )	Complete

Finally, the devices found are listed. Te following image shows a typical display:

Selected	Device name	SN	Device specification
	T_124835_C1_1_LEMO_ET	124835	imc C1-1 LEMO
	T_130039_busDAQ_X	130039	busDAQ-X
	T_130311_SPARTAN_U32_CAN	130311	imc SPARTAN

Device search results (example)

The devices listed now are not yet known!

To make a device which has been found "*known*", you can do one of the following:

- "*select*" the device (this means that the device is immediately selected for the experiment)
- select the disabled parameter "*Device known*" (e.g. in the tool window "*Details* 228").

Once the device has been made "*known*", its setup is determined.

Note — Note on determining the device configuration

For the purpose of successfully determining how the device is configured, a temporary connection to the device is established.

If this is not possible, the device is not made known to the system. A corresponding entry appears in the log book. Possible reasons include:

- incorrect firmware version
- the device is in Measurement mode (which means that the device is currently performing a measurement)
- the device has established an active connection to a different computer

Now that the device is "*known*", it is available for selection, without the need for a search, the next time the program starts.

Note — Special case

Registering the device with the system by loading experiments

When an experiment is loaded in which an unknown (unregistered) device is used, the device is automatically adopted as a registered (known) device.

FAQ

Question: The device in an experiment has a different device configuration than my registered device. Will the registered device be replaced with it?

Answer: The registered device is only replaced when you establish a connection with the device. If an experiment with a registered device is loaded, but its configuration is different (e.g. CRFX with different modules), the registered device is initially not overwritten. The Device Table temporarily indicates a "new" device. When you restart imc STUDIO, the original device appears once again in the table.
Only once you establish a connection with the device will the device list be updated. The previous state is discarded.

Device selection

To be selected for an experiment, a device must be "*selected*".

Selected	Device name	SN	Device specification
☑	T_124835_C1_1_LEMO_ET	124835	imc C1-1 LEMO
☐	T_130039_busDAQ_X	130039	busDAQ-X
☐	T_130311_SPARTAN_U32_CAN	130311	imc SPARTAN

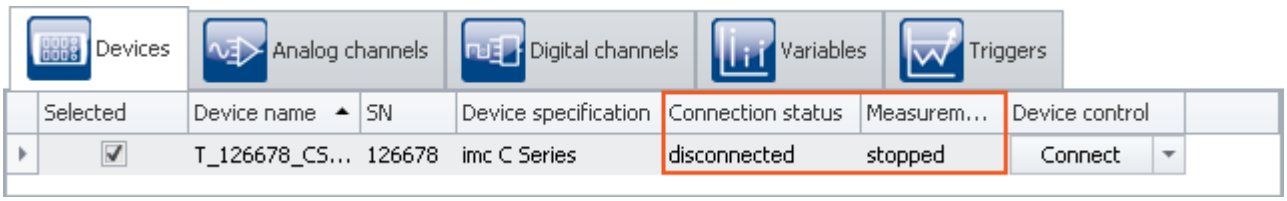
By clicking on the desired devices "*Selected*" box, the device becomes ready for the experiment. You can also select multiple devices for your experiment.

You are not yet connected with the device!

If your device was not previously known, it is made *known* the first time the it is selected.

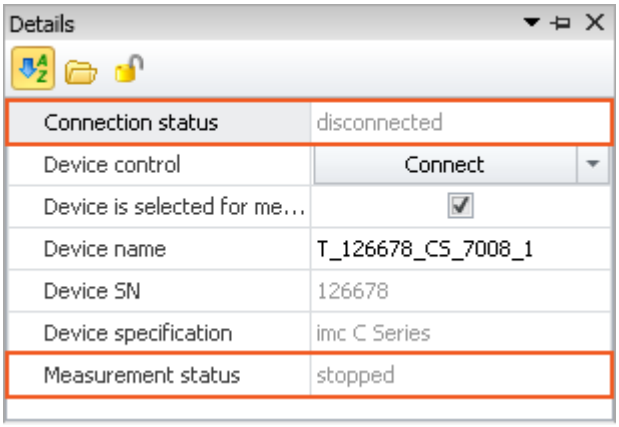
Status display

After selecting the device, the **status** of the "***Connection*** "and of the "***Measurement***" is indicated in corresponding columns:



Status indication in the columns "Connection status" and "Measurement status"

The same information is also available in the "*Details*" toolbox ("*Measurement status*", "*Connection status*"):



Status display in the "Details" window

8.6 Metadata

Saving metadata with the channel

Metadata can be saved directly in the channel file. No separate file is created. This means the relevant information is saved **in all channels** and thus **in all channel files** on the PC hard drive.

Column selection

In the Options "*Setup*" > "*Traceability of data*", you have the choice of which groups of parameters to save.

Ribbon	View
Extra > Options ( )	all

Option	Description
Traceability of data 134	Groups of Setup-parameters which are saved as metadata in the Channel Properties. They can be read out again at a later point in time. The channel properties make reconstruction of the measurement's configuration possible.

Various groups of Setup-parameters are available for selection, which can also be saved. The tooltip for each of these selections contains an exact list of all parameters. Here is an overview:

Groups	Description
Basic information	Device serial number, channel number, module serial number
Hardware information	Among others connector, channel type, module number, module type, amplifier calibration date
Channel settings	Among others measurement mode, coupling, offset, factor, bridge parameters, inc.-parameters, filter parameters
Balance information	Balance date, balance compensation 1/2, balance time
Metadata	All user-defined metadata columns 260, which had been created for the channel Channels can record any data which can be interpreted as either text or a number. Something which cannot be saved with the channels is embedded files (images/PDFs). For instance, when a file is saves as "Link" or as "Link + in experiment" 261, the path to the file is saved as metadata in the channel. If this is not the case, the parameter remains empty in the channel.

Notes

Saved parameters - adapted to the channel type

Only parameters relevant to the channel type are saved. Analog channels contain different parameters than incremental counter channels.

Changes during measurement

During **creation of the file**, the metadata for the channel are specified. For this reason, any changes made to the metadata **during the measurement are not saved to the file**.
The exception is when data saving is resumed in a new file. In this case, the changed metadata are applied to the new file. This pertains to "Interval saving", for instance, and in the case of "*Suspend/resume data storage*".

Reading out metadata

In order to read out the metadata saved with the channel, proceed as follows:

- **imc FAMOS**: Use the command `UserPropText?` (for texts) or `UserPropValue?` (for numbers).

For the purpose of parameter selection, the the metadata's language-independent column ID is required.

FAQ

Question: How do I find out the metadata's column ID?

Answer: Open a saved channel containing metadata in the free-floating curve window (double-click on the channel in the Data Browser).

Next, select the menu item "*Configuration*" > "*More Channels*". Jump to the tab "*Channel info*".

Here you will find the available saved metadata with their respective column IDs under the heading "*User-defined properties*".

Additionally, the column ID is included in the list appearing in the description of the respective parameters; e.g. "*Measurement mode* 349".

8.7 Information and tips

8.7.1 Enumeration classes

In an Enumeration class, the associations to "Object-groups" (e.g. analog channels, Display variables, devices, ...) are defined.

It is important to make an appropriate selection for the Default Values 214 for example, in order to have the correct targets for the defaults.

Enumerator class	Internal designation	Description
Experiment	eExperiment	Experiment metadata
Device	eDevice	Device settings
Module	eModule	Amplifier parameters
Bit	eSubChannel	DI-/DO-bits (incl. Mon-DI-Bits), virtual bits, Ethernet bits, Field-bus bits (MFBDIO), pv-variables (of DI-ports, analog-, incremental counter- or Fieldbus channels)
Channel	eChannel	Analog channels, DIO-ports, other pv-variables, virtual channels, Display variables, ...
Trigger	eTrigger	Trigger settings (trigger name, trigger linkage, event number, trigger, trigger count)
Parameter	eParameter	AppMod block parameter (only the parameter value can be set)

8.7.2 Setup-layout

This section describes the Setup-interface's designing options.

Chapter overview

Synopsis	Section
Showing and arranging existing columns	Showing and moving columns 248
Personally defined columns	Creating and configuring additional columns 251
Preventing dialogs from closing until after an entry has been made	Using Mandatory Boxes 269
Transferring columns to other PCs or to other projects	Export/import of column/table descriptions 273
Modifying and configuring pages	Table Description and Complete Layout 274 Properties of the Pages/Complete Layouts 275 Designing pages 276

8.7.2.1 Showing and moving columns

By default, the various Setup pages show a certain selection of columns (parameters). You can also **configure the selection and order** of the columns according to your own wishes. The column configuration is saved/opened with the view configured.

Note Saving the column selection

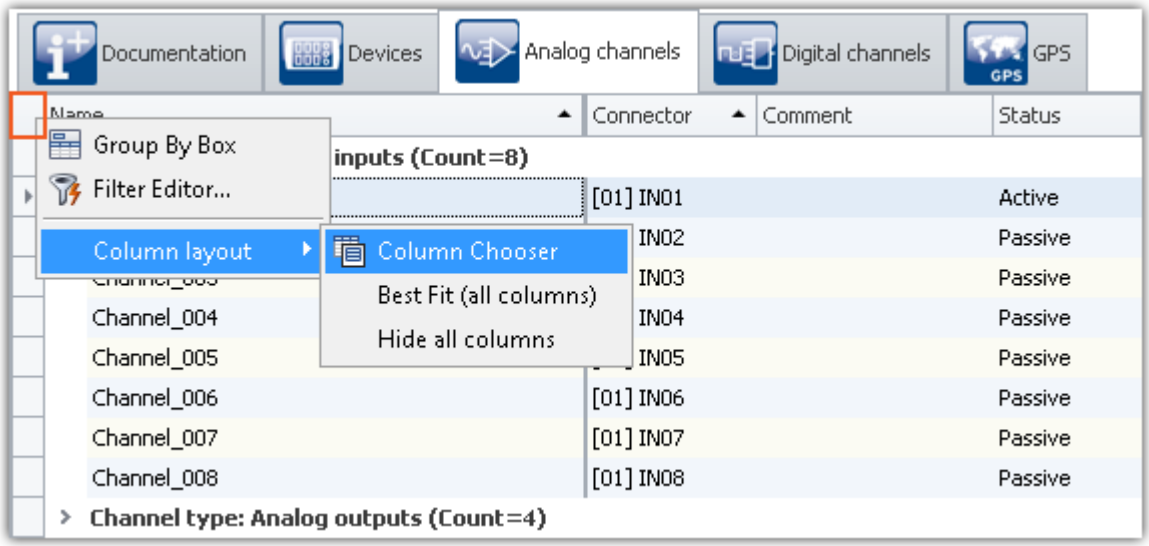
The **position and size of the column** is **saved with the views**. The views are **saved along with the project**. This means that any views are available for all of the project's experiments.

- See: "*What is saved where?* 162"
- See: "*Views* 155"

The pages are constructed in a variety of styles. The various design elements have different methods for configuration.

Display in a "Table": e.g. the pages "Devices", "Analog channels", ...

Opening "Column Chooser"

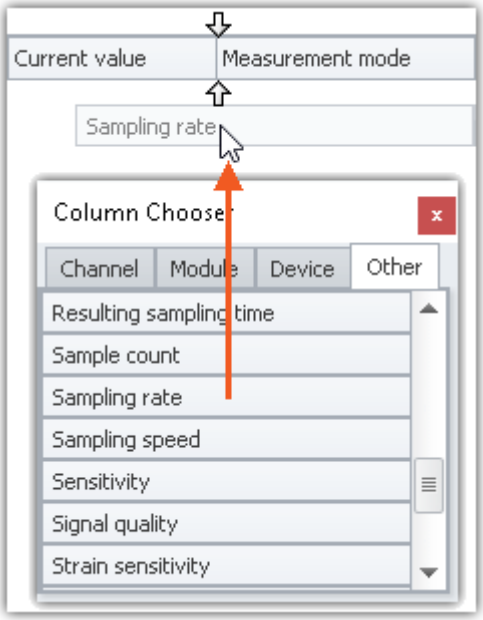


Opening the table's context menu

In order to insert columns, open the "*Column Chooser*" dialog. To do this, open the context menu in the upper left corner of the table. Select "*Column Chooser*".

Inserting columns

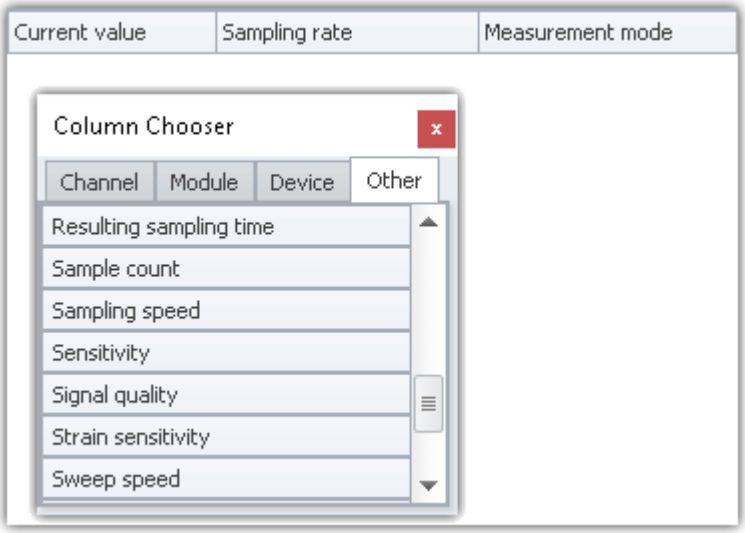
The dialog "*Column Chooser*" contains a list of al columns which can be inserted. In order to insert a column, **drag it from the list** to the column's header, as shown in the image below.



Dragging a column

The position of the arrows (before the column has been placed) **indicates** whether the column is to be inserted. In this example, the column "Sampling rate" is inserted to the right next to "Current value".

Once you have placed the column, it is inserted at the position chosen.



The column positioned

Moving columns (changing position)

To change a column's position, **use Drag&Drop to move it** to the title bar at the desired position.

Deleting columns

In order to delete a column, use one of the two following possibilities:

- Open the "*Column Chooser*" dialog and drag the column to the dialog window
- Drag the column to an empty area next to the table as shown below:

Deleting a column by Drag&Drop

The "*Column Chooser*" dialog does not need to be open.

Display in a "Detail-Table": e.g. the pages "Documentation", "Experiment", ...

Opening "Column Chooser"

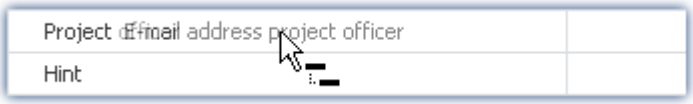
In order to insert columns, open the dialog "*Column Chooser*". To do this, open the **context menu of an existing column title** or of any empty area. Select the item "*Column Chooser*".

Inserting a column

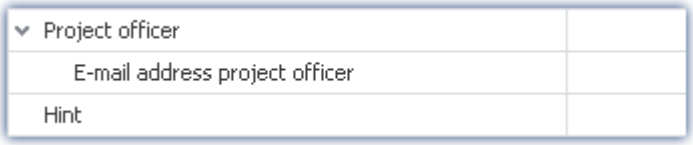
The dialog "*Column Chooser*" contains a list of all columns which can be inserted. In order to insert a column, **drag it from the list** into the existing table.

The **mouse cursor** indicates where the column will be inserted.

As a branch of an existing group

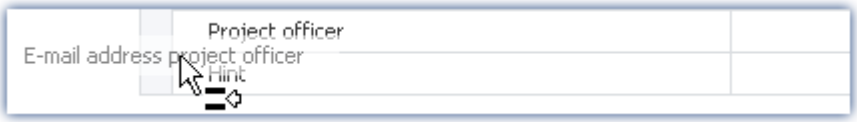


Insertion as a branch in an existing column



The next higher level column becomes a grouping element

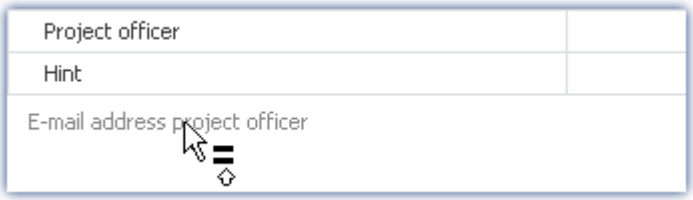
Between existing columns



Isertion between two existing columns
Be aware that the cursor needs to be rather far to the left

The column is inserted in between

After the last column



Insertion after the last existing column

The column is inserted at the end

Moving columns (changing position)

First open the dialog "*Column Chooser*". In order to change the position of a column, **move the title** by means of Drag&Drop to the **desired position** (in the same manner as inserting the column).

Deleting columns

First open the dialog "*Column Chooser*". To delete a column, **move the title** by means of Drag&Drop to the **dialog**.

Note — Saving / restoring a configuration

In order to save a column layout, use the functions belonging to the View menu (see the manual imc STUDIO (general) > chapter "*Views* 155").

8.7.2.2 Creating and configuring additional columns

Note — Prerequisite: imc STUDIO Professional

Creating/editing columns is only supported as of the product versions imc STUDIO PRO 29.

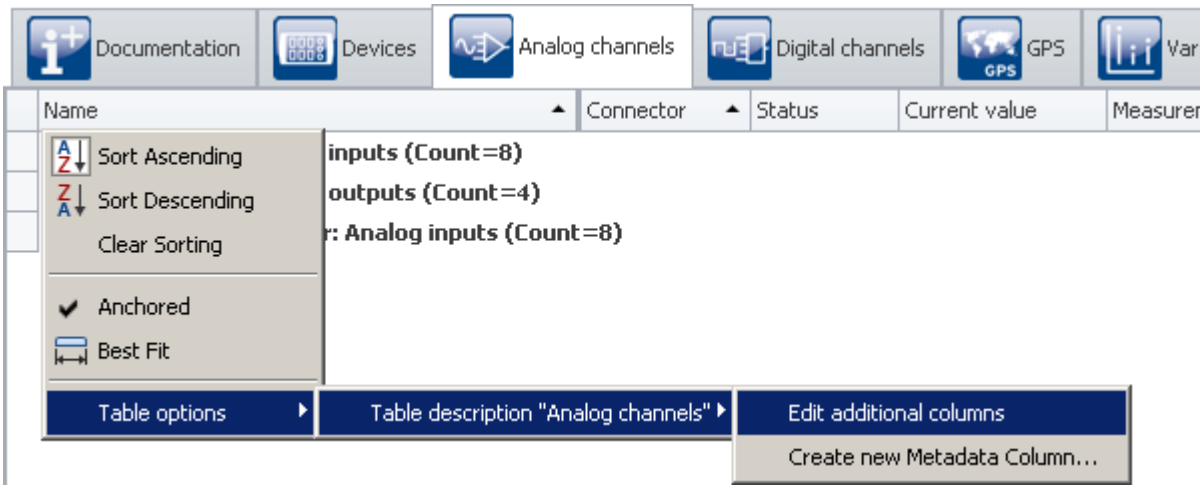
"*Additional columns*" contains controls for **improving and adapting** the **operation and documentation** of your measurement results in a wide variety of ways.

Type	Description
Combined column 254	Allows combination, operation and display of multiple parameters in a single column.
Metadata column 260	Allows display of any desired information and documents (also from third-party) applications.
Parameter set column 262	Fixed parameter sets (combinations of settings) can be saved in a column and set with a single click of a button.
Parameter set import column 267	Pre-defined, comprehensive parameter set files (*.csv) can be loaded by means of the drop-down list.

Additional columns are defined in two steps. **First select the type** of extra column and set certain basic properties such as the title and designation. Then for the "*Combined column*" and the "*Parameter set column*", select **the corresponding columns** and for the "*Metadata column*" select **the data type** or data source (file).

Creating the columns with the Collection editor

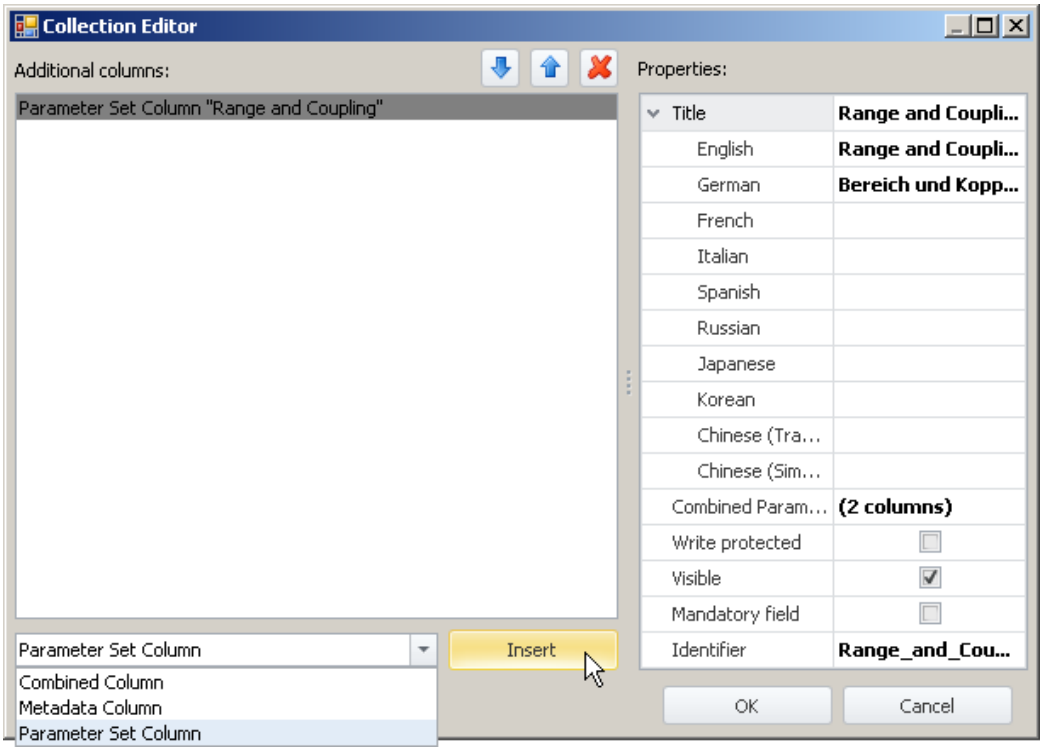
To create additional columns, open the **context menu of the column headers** and select "*Table options*" > "*Table description "<Name>"* " > "*Edit additional columns*".



"Edit additional columns" in the context menu of the column headers
Here also: Context menu of the table header

This function is available on every Setup page (Documentation (here without table), Devices, Channels, ...).

The **Collection Editor** opens:



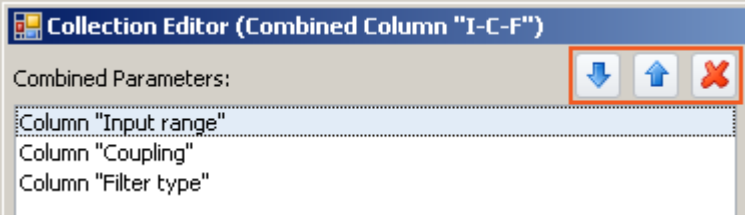
Collection Editor (example)

- In the drop-down list (at bottom left), **select the type of the additional column**.
- Click on "*Insert*" and **the column is adopted into the list**.
- In the "*Properties*" window, set the column's **appearance / functionality**.

For a detailed example, see the chapter "*Combined column* 254" or "*Parameter set column* 262".

Sorting/deleting columns and parameters in the Collection Editor

To **sort entries**, use the "Up/Down"-arrow icons (/). To delete entries, use the "X" icon ( ).



Collection Editor (example shows sorting)
List of parameters of a "combined column"

For a detailed example, see the chapter "*Combined column* 254"".

Common properties

Some properties are shared by all Additional columns:

Properties	Description
Title	The column title as it appears in the table. The title can be set separately for selected languages. Depending on the language setting, the corresponding title is displayed. If the title is empty, the English version is used. Special characters and spaces may be used. A comma in a column name should be avoided.
Write-protected	The column content cannot be edited in the column.
Visible	If you select this option, the column is defined but not visible. This makes sense for instance when you pre-define different columns and later display / select them individually.
Identifier	This identifier is used within the program to uniquely distinguish the column. By default, the title is used for it. If this identifier is already assigned, a distinguishing suffix is automatically appended ("_1", "_2", ...). **Special characters (except "_") and spaces may not be used.**

Note Saving the columns

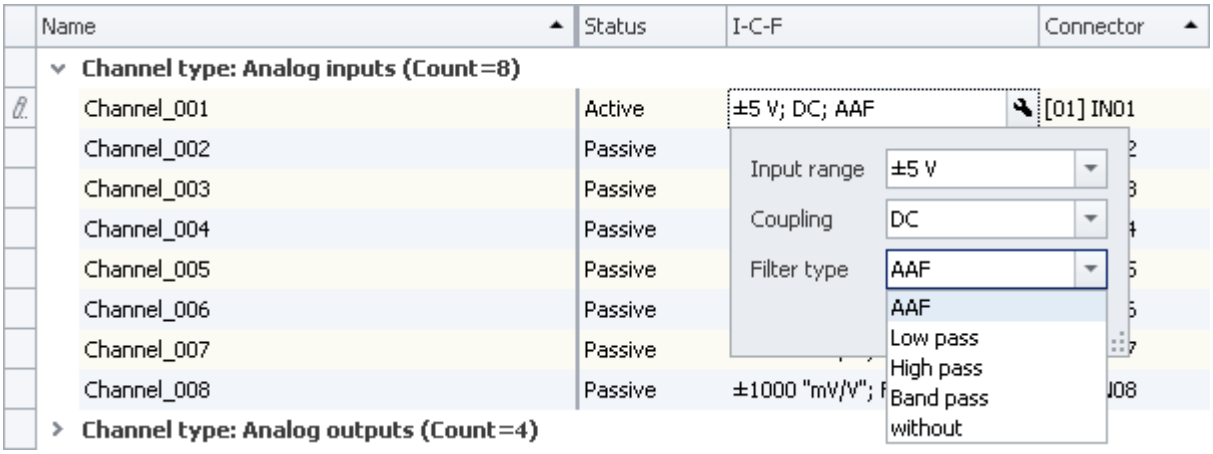
The **Additional column's configuration is saved** in the respective **table description**. The table descriptions are **saved along with the project**. This means that any additional columns defined are available for all of the project's experiments.

- See: "*What is saved where?* 162"
- See: "*Views* 155"

8.7.2.2.1 Combined column

The **Combined column** joins multiple parameters in one column.

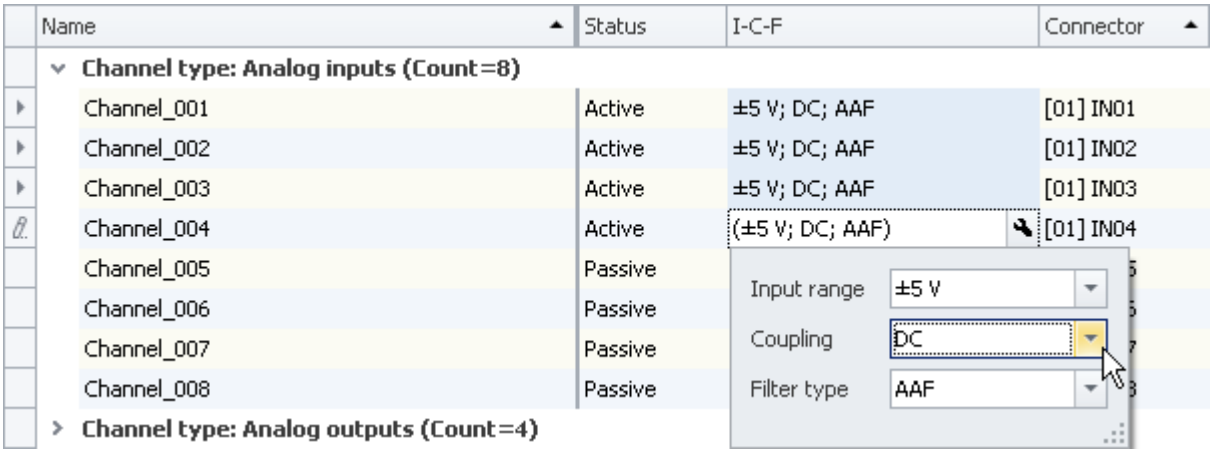
This can make sense, for instance when a measurement is often performed and a certain set of parameters must be set before every measurement. You can then set the parameters in this column and no longer need to open a series of various columns/index cards.



Combined column
Example of column: I-C-F

Configuring

In order to configure the parameters, click on them in the table cell (it is also possible to select multiple cells at once; see the chapter *Tabular Display* > *Selection* 232). The controls for the parameters are displayed and at the edge you see the wrench icon (🔧):



Combined column with multiple selected cells
Example of column: I-C-F

If you change the coupling in the image above from "**DC**" to "**Half bridge**", for instance, the change applies to all selected cells. If the cells all had different couplings before opening the combined column, then the inequality symbol (≠) is shown in the box for the coupling.

To close the combined column, click again on the wrench icon (🔧)

Properties

For shared properties, see: Additional Columns 253.

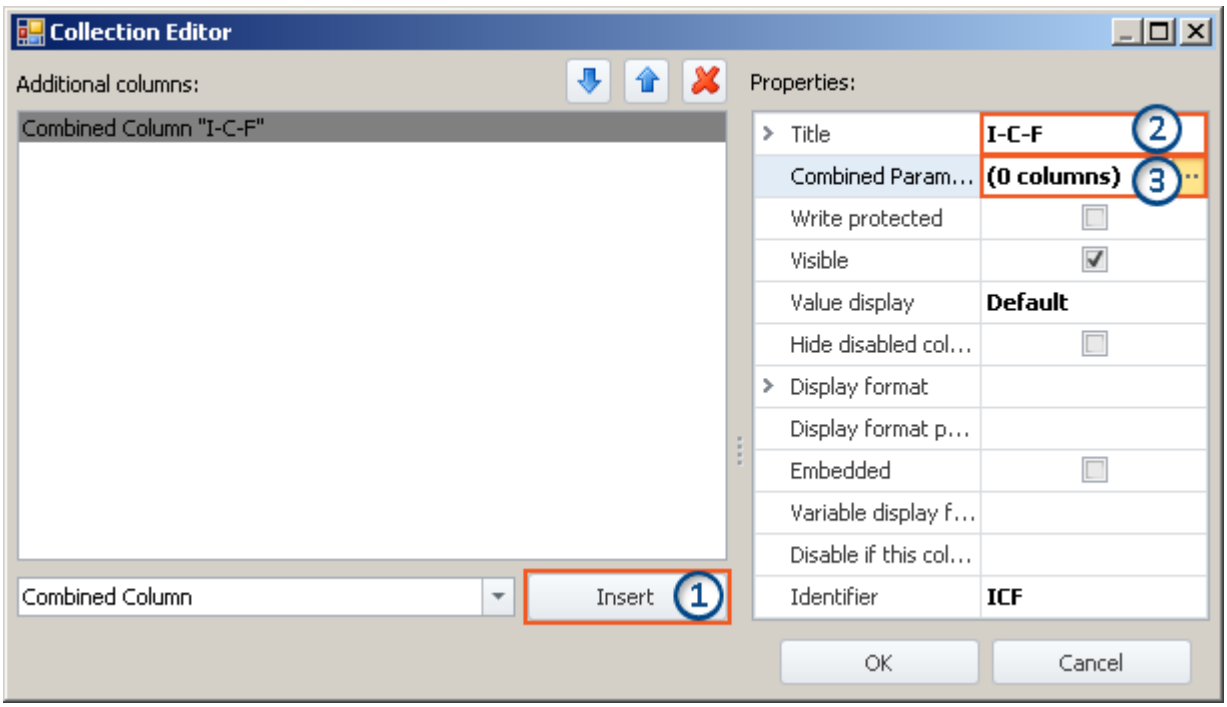
Type	Description
Combined Parameters	List of the parameters which are to be set by means of the column.

Type	Description
Value display	Display format in the cell. The values belonging to the combined columns are always listed in sequence in a row. By means of the following options, the format can be modified. • **Default**: Displays only the value of the combined columns e.g. Channel_001; Active • **SimpleFormat**: Selected parameters can be shown/hidden. Separators or additional texts can be freely defined. • **DependentFormat**: Selected parameters can be shown/hidden according to the status of one parameter. Separators or additional texts can be freely defined. • **ShortCaption**: Displays the short form of the parameter name before each value e.g. Name: Channel_001; Status: Active • **LongCaption**: Displays the long form of the parameter name before each value e.g. Channel name: Channel_001; Channel status: Active
Hide disabled columns from caption	In the cell, the space for deactivated columns is omitted. For instance, in a column with the parameters: Filter type + Filter cut-off frequency 1 + Filter cut-off frequency 2 and parameter values: Low pass, 20 kHz with Hide *on*: Low pass; 20 kHz with Hide *off*: Low pass; 20 kHz; or for the parameter value: AAF with Hide *on*: AAF with Hide *off*: AAF; ;
Display format	By means of the display format, you can set which parameters are visible in the cell. Example with four parameters: 0: Filter type 1: Filter characteristic 2: Filter cut-off frequency 1 3: Filter cut-off frequency 2 Display format: {0} - {2} - {3} Result: e.g. Band pass - 10 Hz - 1 kHz You can also use other texts and separators: Display format: Filter: {0} > {2} > {3} Result: e.g. Filter: Band pass > 10 Hz > 1 kHz Only usable with the value display type: **SimpleFormat**.

<table>
<tr><th>Type</th><th>Description</th></tr>
<tr><td>Display format parameter
and
Variable display format</td><td>A variable display format is defined in accordance with a parameter's value.
Example: Column: Sampling & Filtering. Here, the measurement duration is only displayed if it is not set to Undefined.
The column has seven parameters.
0: Sampling rate
1: Duration
2: Filter characteristic
3: Filter type
4: ...
Display format parameter: Duration
<table>
<tr><th>Variable display format</th><th>Properties</th></tr>
<tr><td>0: "{0} - {3}"</td><td>ParameterValue: 0
PrintoutFormat: {0} - {3}</td></tr>
<tr><td>: "{0} - {3} - {1}"</td><td>ParameterValue:
PrintoutFormat: {0} - {3} - {1}</td></tr>
</table>
Parameter = 0 (Duration = Undefined)
Show the 0th and 3rd value with separator "-": e.g. 100 Hz - Low pass
Parameter = 10 s (Duration = 10 s)
Show the 0th and 3rd value with separator "-": e.g. 100 Hz - Low pass - 10 s
In this way, other display formats can also be defined.
Only usable with the value display: DependentFormat.</td></tr>
<tr><td>Embedded</td><td>The parameters can be edited directly in the cell, instead of in the separate window. An example is presented on the page: Triggers, the combined column: Properties.
For this function, it is necessary to alter the row height in order to make all parameters visible (only available as of imc STUDIO Developer).</td></tr>
<tr><td>Disable if this column is empty</td><td>If the selected column is empty (e.g. an empty metadata-column), it is not possible to view or edit the parameters belonging to the combined column. In order to delete the view, click on the <Backspace> key.</td></tr>
</table>

Example of a Combined column

Start the Collection editor as explained in the previous chapter Additional columns 251.



Creating a combined column (example)

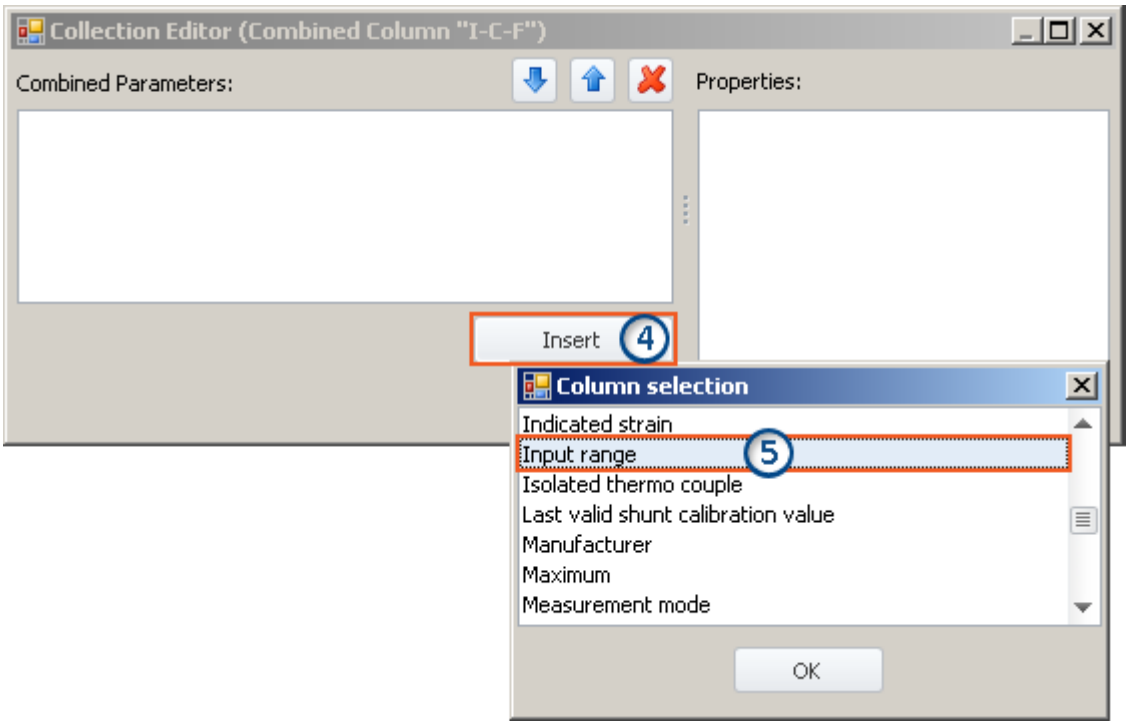
1) In the drop-down list, select the entry ***Combined Column*** and click on ***Insert***. This creates a nameless (title) combined column.

2) Enter the desired title in the input box ***Title***, e.g. "I-C-F" and hit the **<Enter>**-key (see image above). This entry is automatically used as a unique ***Identifier***. Note that special characters (except "_") and spaces can not be used as identifiers. Remove all "-" in this example.

Adding parameters (columns)

3) Click in the row ***Combined Parameters*** in the Properties window and click on the button with the three dots (⋯).

This opens **an additional window belonging to the Collection editor**. Here you set which columns/parameters are to be combined.



Collection editor, second window, adding column "Input range"

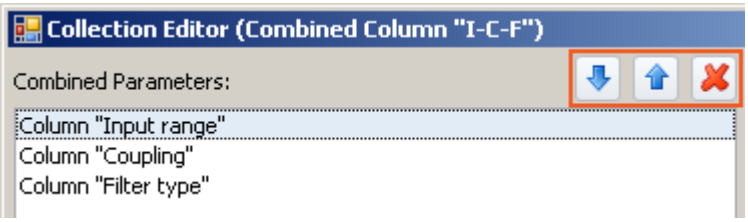
4) Click on ***Insert***: An additional window with a list of all available parameters opens.

5) Select from the list the desired parameter(s) (for this example: **Input range**, **Coupling** and **Filter type**).

Sort

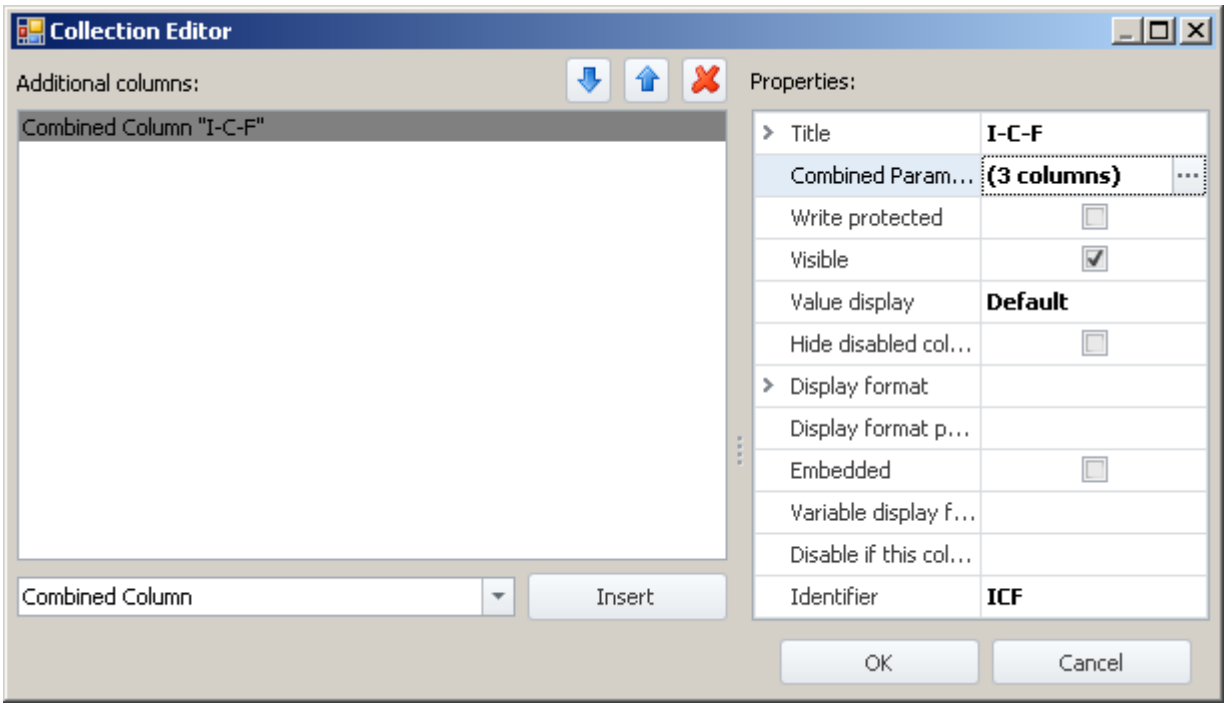
To sort entries, use the "Up/Down"-arrow icons (/).

Sort the order of "Input range", "Coupling", "Filter type", as shown below:



List of the combined parameters (after sorting)

Close the second window by clicking ***OK***. You ten see the first window with the configured column "**I-C-F**" and the note indicating that "(3 columns)" have been combined:



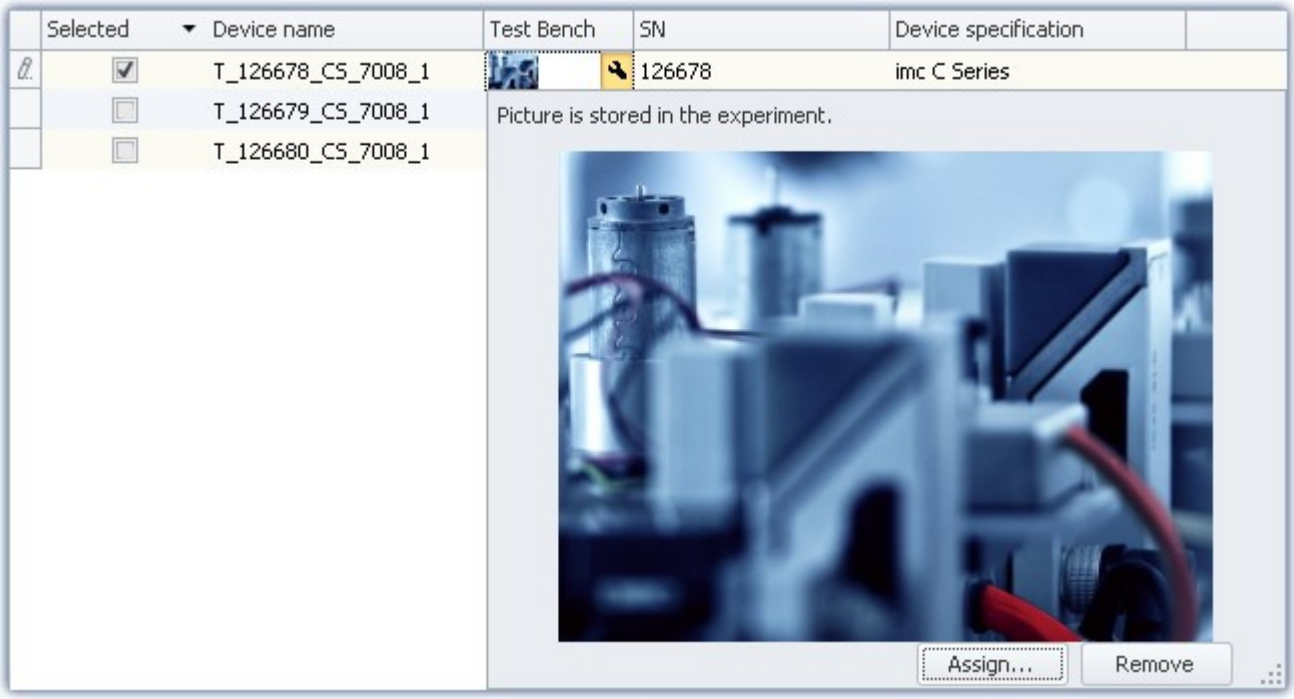
Collection editor, first window, combined column "I-C-F" (3 columns)

Close the window by clicking ***OK***. You now have a new column in the channel list, which shows the three parameters together and with the help of which you can configure them together.

8.7.2.2.2 Metadata column

In a **Metadata column**, you can display any desired information or documents. They can be pictures (e.g. a picture of how to place the test object into the test rig), or a display of the date and time, or descriptive texts, or anything similar.

The metadata can be entered/edited at any time, for instance, even after a measurement. The natural use of metadata is for test/inspection documentation. The metadata content is saved along with the experiment, except for the types "Picture" and "Document". For these types, you can choose whether the file contents are saved with the experiment or only as a link.



Example of a metadata column

Metadata types

The following types are available:

Type	Description
Single-line text	Shows a single-line text in the column. To enter the text, click on the column.
Multiline text	Shows a multi-line text in the column. To read or edit the complete text, click on the cell.
Picture	Shows a picture (graphic) from a file. In the cell, the picture is minimized as a *Preview* or the associated icon is displayed as a *Placeholder*. In order to see the picture in full scale, click on the cell (see picture above)
Document	Shows a document of any desired type. In the cell, the associated icon is displayed as a *Placeholder*. Some document types, such as PDF and some graphics, can be viewed directly when you click in the cell. In order to open the document with the associated default program, click on the button: *Open*.
Text from list	Shows a selected text from a list of texts. To select a different text from the list, click on the cell. You create the list when making the column definitions (Property: "Text list").
Date	Shows a control having Date format. To enter the date, click on the cell. Click on the wrench icon (🔧) to select a date.

Type	Description
Time	Shows a control having Time format (Hour:Minute:Second). To enter the time, click on the cell.
Logical value	Shows a logical value in the form of a checkbox (☑ / ☐). This lets you symbolize any binary state "true/false" ("on/off", "Error/OK"). To change the state, click on the checkbox.
Directory	Shows a link to a directory. Click on the wrench icon (🔧) in order to start the folder selection dialog.

Properties

For shared properties, see: Additional Columns 253.

Type	Description
Text list	List of the texts available for selection. To edit the list, click on the button with the three dots (···). Only usable for the type: Text from list.
Show placeholder	If *on*: instead of the preview, the icon corresponding to the file type is displayed. If *off*: a preview of the content is displayed • Type **Picture**: minimizes the picture • Type **Multiline text**: the first row Only usable for the types: **Picture**, **Multiline text** and **Document** (here, always *on*).
Storage type	Determines how the external file is integrated/linked. • **In the experiment**: The file is incorporated into the experiment. The column thus has no link to the original file. If the original file is altered, the change is ignored. This ensures, however, that the original file is always visible in the experiment. • **Link**: Only the link to the file is saved in the column. Please ensure that the file is retained. • **Link + in experiment**: The file is incorporated in the Experiment-file and the link exists in addition. As long as the file exists, it can be opened via the link. As soon as the file is no longer present, the incorporated file is used. If the file is saved **in the experiment**, the Experiment-file is correspondingly larger. If possible, the **link** should be used for large files. Only usable for the types: **Picture** and **Document**.
Default value	Value which is displayed before any value has been entered. Only usable for the types: **Single-line text**.

Type	Description
Entry format	Input format for a numerical input. • 0: Mandatory • #: Optional • Decimal separator: decimal period "." (please don't use decimal comma ",") When the entry format is empty, any arbitrary text can be entered. If the entry format is used, the only possible entry is al number conforming to the specified format. Only usable for the type: **Single-line text**. Example: 000.0## 001.2 123.4 123456.789 Example: 0.00 1.23 123.45 0.12
Mandatory box	Mandatory if the Setup page is embedded in a dialog. The dialog can only be closed once the entry has been made. In the following dialogs, mandatory box are evaluated: • Command: "Show Panel page as dialog" • Command: "Display Setup dialog" An example is presented in the chapter: Using Mandatory Boxes 269.
Show most recently use values	When active, a list of the most recently entered values can be opened in the column. Each entry in the list can be selected and used. Only usable for the type: **Single-line text**.
Most recently use values	For viewing and editing the list of most recently used values.

8.7.2.2.3 Parameter set column

The ***Parameter set column*** offers the ability to join multiple settings with permanently set values to a ***parameter set***. Then, if the parameter sets are pre-defined, the end user must only select the parameter set corresponding to the desired measurement. This saves the user the need to know the entire measurement hardware and its complete settings options.

Creating a column

- Create a parameter set column (Title, Identifier).
- Set which columns (parameters) are joined together in the parameter set.
- This concludes the definition of the parameter set column.

Creating/saving parameters

- Next, make the appropriate individual parameter settings.
- Then click in the Parameter set column and select "- New parameter set -".

Name	Activate	Status	Save (device)	Save (PC)
Channel type: Analog inputs (Count=8)				
Channel_001	Active, without storage	Active	☐	☐
Channel_002		Passive	☐	☐
Channel_003	Active, storage PC	Active	☐	☑
Channel_004		Active	☑	☑
Channel_005	Active, without storage	Passive	☐	☐
Channel_006	Active, storage PC	Passive	☐	☐
Channel_007	- New parameter set -	Passive	☐	☐
Channel_008		Passive	☐	☐

Example: Parameter set column with the parameters: Channel status, Save (device) and Save (PC).

- Assign a name for the parameter set.

In the parameter, all column settings are saved which are associated with the parameter set column.

In the example above, if a new parameter set belonging to the selected column is created, the following settings are saved:

- Channel status: Active
- Save (device): true
- Save (PC): true

For this column, an appropriate name can be assigned: e.g. *Active, storage PC and device*.

Note Saving the parameter

The **configuration of the parameters** is saved 162 in the respective **supplemental column**. The column itself is saved with the project. Thus, the parameters defined are available in all the project's experiments.
The parameters **are only saved** when the **project is saved**.

Selecting a parameter set

When you select a parameter set from the list, all columns adopt the settings saved which are associated with the Parameter set column.

Name	Activate	Status	Save (device)	Save (PC)
Channel type: Analog inputs (Count=8)				
Channel_001	Active, without storage	Active	☐	☐
Channel_002		Passive	☐	☐
Channel_003	Active, storage PC	Active	☐	☑
Channel_004	Active, storage PC and device	Active	☑	☑
Channel_005		Passive	☐	☐
Channel_006	Active, without storage	Passive	☐	☐
Channel_007	Active, storage PC	Passive	☐	☐
Channel_008	Active, storage PC and device - New parameter set -	Passive	☐	☐

In the example, if the parameter set *Active, storage PC and device* is selected for Channel_005, the following settings are adopted:

- Channel status: Active
- Save (device): true
- Save (PC): true

These settings are adopted because they were previously so defined (see: **Creating/saving parameters**)

As long as no further columns are linked with the Parameter set column, all other parameters remain unchanged.

Renaming/deleting a parameter set

To rename or delete a parameter set, open the selection list in a cell and then open the context menu of the particular item.

Next, select *Rename parameter set* or *Delete parameter set*.

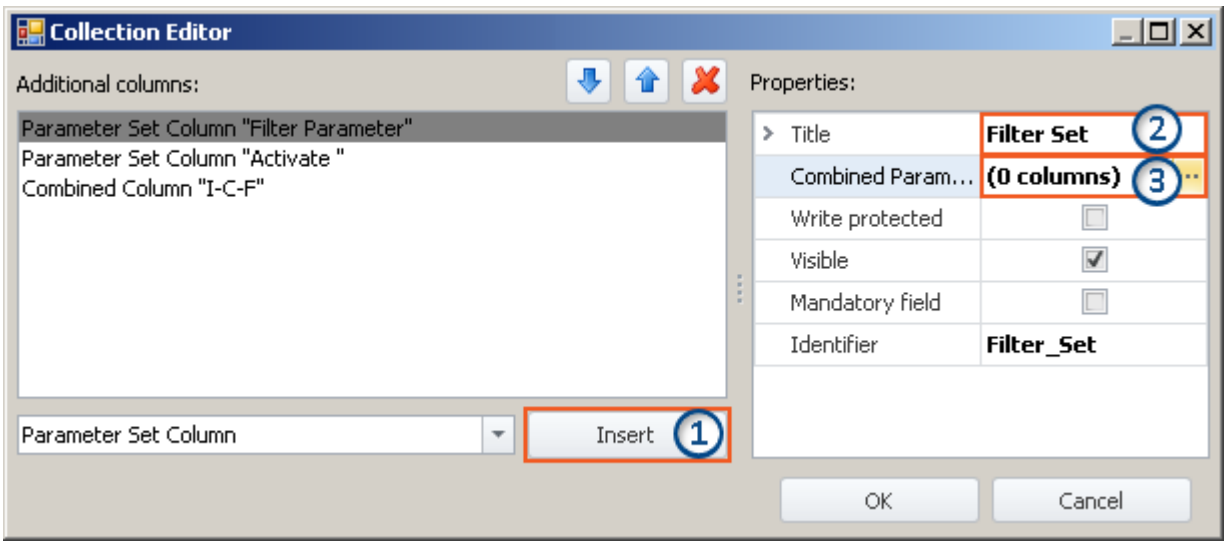
Properties

For shared properties, see: Additional columns 253.

Type	Description
Combined Parameters	List of the parameters which can be set via the column.
Mandatory box	Mandatory when the Setup page is embedded on a Panel page as a dialog. It is only possible to close the dialog once the entry has been made. An example is presented in the description of the command: Show Panel page as a dialog 1732.

Example of a Parameter set column

Start the Collection editor as explained in the chapter Additional columns 251.



Creating a parameter set column (example)

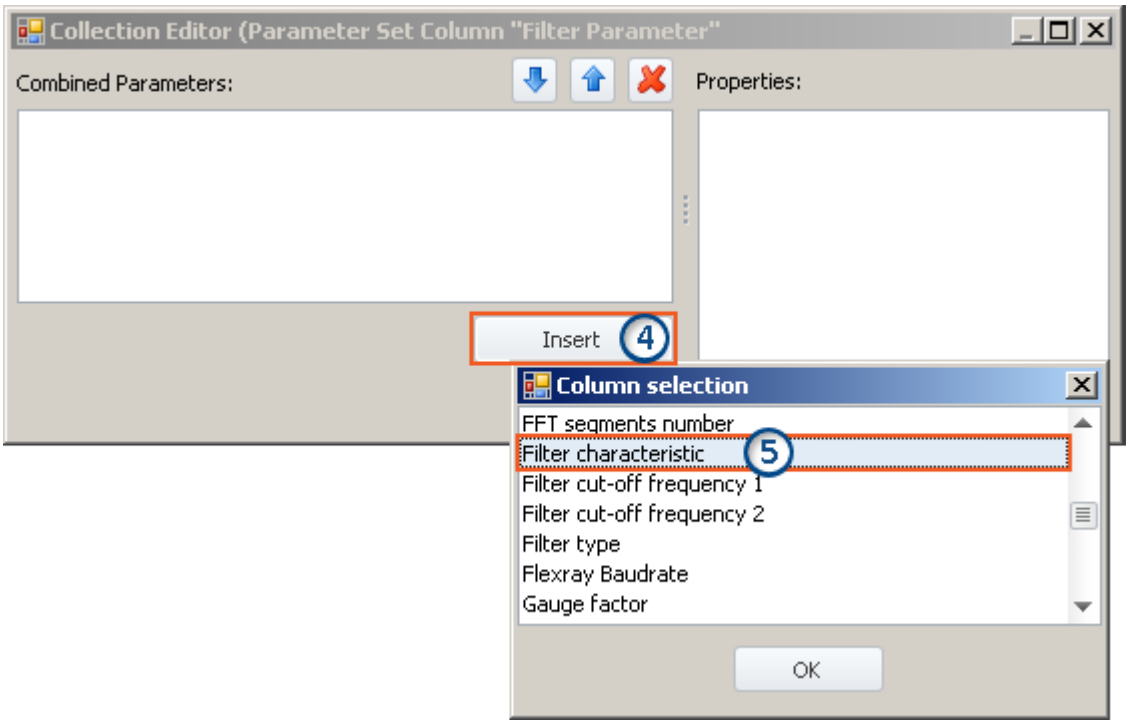
1) In the drop-down list, select the entry ***Parameter Set*** and click on ***Insert***. This creates a nameless (title) parameter set column.

2) Enter the desired title in the input box ***Title***, e.g. "Filter Parameter" and hit the **<Enter>**-key (see image above). This entry is automatically used as a unique ***Identifier***. Note that special characters (except "_") and spaces should not be used in identifiers. Delete all " " in this example. Instead you can use: "_".

Adding parameters (columns)

3) Click in the row ***Combined Parameters*** in the Properties window and then on the button with the three dots(···).

This opens **a further window belonging to the Collection editor**. Here you set which columns/parameters are to be combined.



Collection editor, second window, adding column "Filter characteristic"

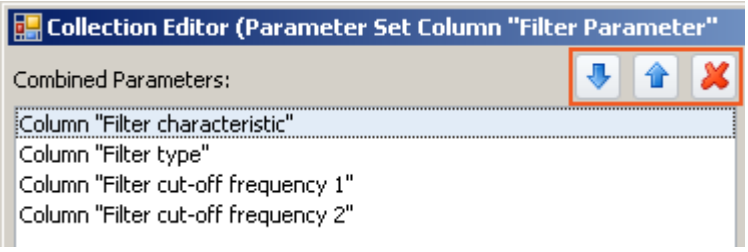
4) Click on ***Insert***: An additional window with a list of all available parameters opens.

5) From the list, select the desired parameter(s) (in this example: **Filter characteristic**, **Filter cut-off frequency 1**, **Filter cut-off frequency 2** and **Filter type**).

Sort

To sort entries, use the "Up/Down"-arrow icons (/).

Sort the order of "Filter characteristic", "Filter cut-off frequency 1", "Filter cut-off frequency 2", "Filter type", as shown below:



List of the combined parameters (after sorting)

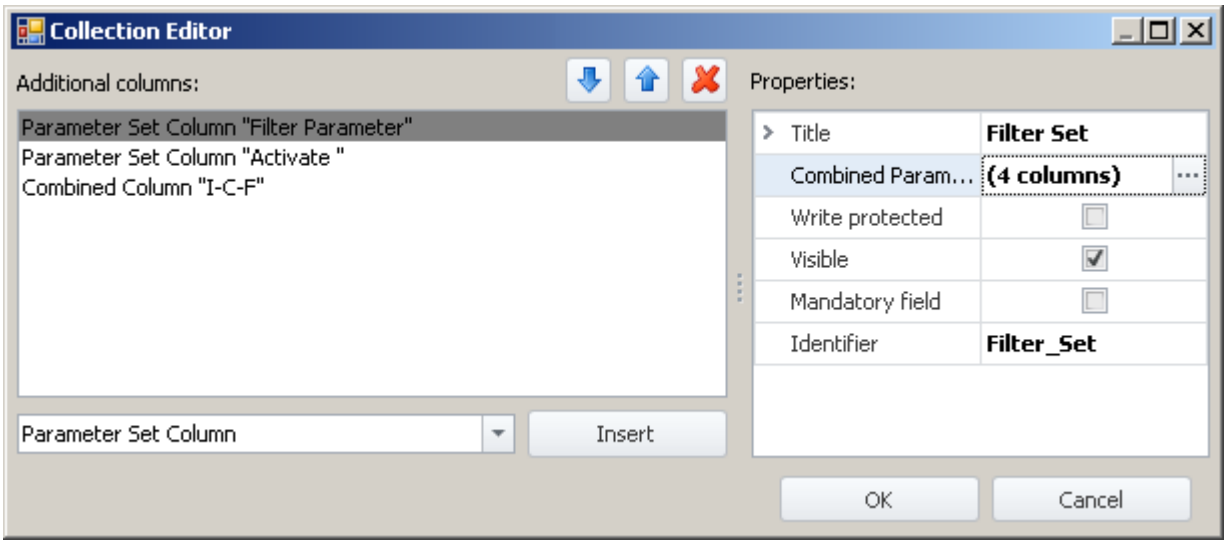
Note Why the sorting is important

The sorting determines the order in which the parameters are set.

For instance, the column: *Filter cut-off frequency 2* can only be assigned a value if the column is "active". In other words, if the column is required for the filter type used, as is the case with Bandpass, for example.

In consequence, it is necessary to set the parameter values in the order which their respective dependencies requires. The filter type must be set before the cut-off frequency.

Close the second window by clicking ***OK***. You ten see the first window with the configured column "**Filter Parameter**" and the note indicating that "(4 columns)" have been combined:



Collection editor, first window, parameter set column "Filter Parameter" (4 columns)

Close the window by clicking ***OK***. You now have a new column in the channel list, by means of which you can save and load the parameters of the four columns.

8.7.2.2.4 Parameter set import column

By using the "*Parameter Set Import Column*", it is possible to quickly load a variety of settings. Selecting particular entries automatically selects the associated saved parameter set (*.csv).

Operation technique, and distinction from the Parameter Set Column:

The **Parameter Set Column** 262 is bound to its associated columns. By contrast, the **Parameter Set Import Column** operates like the command: **Parameter set import** 1172. A parameter set is imported and all settings belonging to it are made.

Parameters which are not in the file retain their values.

Example of a Parameter Set Import Column

At a certain test station, two engine types are routinely tested. For each type, there are 3 test procedures, having varying designations (metadata) and channel settings.

For each of these six "measurement states", one parameter set (*.csv) is generated. The following is a sample file:

Company name	Test object number	Test part number	Test station photo	Hint
My company	X1	X1_123	.\Picture1.png	speed test 1
Channel name	Save (PC)	Sampling time	Duration	
Channel_001	True	1 ms	40 s	

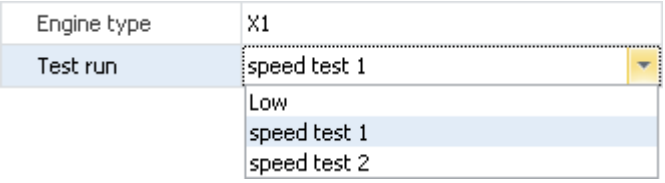
These files are saved in folders as follows:

```
Engine types
+---X1
|   |   speed test 1.csv
|   |
|   +---speed test 1.data
|   |   \---DevSetup
|   |           Picture1.png
|   |
|   |   speed test 2.csv
|   |
|   +---speed test 2.data
|   |   \---DevSetup
|   |           Picture2.png
|   |
|   |   low.csv
|   |
|   \---low.data
|       \---DevSetup
|               Picture3.png
|
+---X2
    |   speed test 3.csv
    |
    |   speed test 5.csv
    |
    |   low 3.csv
    |
    \---low 3.data
        \---DevSetup
                Picture4.png
```

The Engine Types X1 and X2 are folders in which the csv-files are located. Along with the csv-file, there can be a folder of the same name with the extension "*.data" containing additional files. These files can be used as linked metadata (e.g. pictures).

On the Setup-page, two columns are set up to conform to this folder structure.

- First column: First folder level: In this example, the engine type can be selected here (folder name: "X1" or "X2")
- Last column: File level: In this example, the test procedure can be selected here (name of the existing csv-files)



If an entry is selected in the first column, the content of the second column adapts according to what csv-files are present.

If an entry in the second column is selected, the csv-file is imported.

Properties

For shared properties, see: Additional Columns 253.

Type	Description
ImportRoot	Path to the first folder level
Column captions	Optional list: Designations of the column titles • No specification: columns have the default name: Parameter Set Level 1,2,3,... • Specification provided: Columns are assigned the pre-defined name. See example: *Engine type* and *Test run* To edit the list, click on the button with the three dots ( ).

Note

- Multiple subfolders are also possible. For each additional subfolder, a separate column is created. However, a csv-file is imported only to the lowest level.
- Parameter Set Import uses for **Mapping: Import mapping by name**. When channel parameters are imported, the names of the channels must match the names in the csv-file.
- Parameter Set Import performs importation across Table Descriptions. This means that the csv-file can contain parameters for a variety of different Setup pages. In this example, there are parameters for the pages:
  - Documentation (table description: Experiment)
  - Analog channels (table description: Analog Channels, Digital Channels and Channels, ...)
- The column type is only present on the Setup pages: *Documentation* or *Experiment*.

8.7.2.3 Using mandatory boxes

When Setup pages are called in dialogs, it is possible to prevent the dialogs from being closed. In this case, it is only possible to close them once the important "mandatory entries" have been made.

Only metadata columns can be defined as so-called "Mandatory boxes".

It is not possible to use this function to verify any other entries (for instance, any entries made via Widgets or entries of device/channel-parameters). The verification of the validity of entries is based on what type and what format you had previously selected for the metadata column. So, for the type "*Folder*", for example, there is a check of whether the character \ is included, unless a user-defined format was used.

A mandatory box exclusively affects the behavior of the following functions:

- the command "Show Panel page as dialog"
- the command "Display Setup dialog"

It has no effect on the normal Setup pages.

To illustrate this function and its use, the following examples are presented.

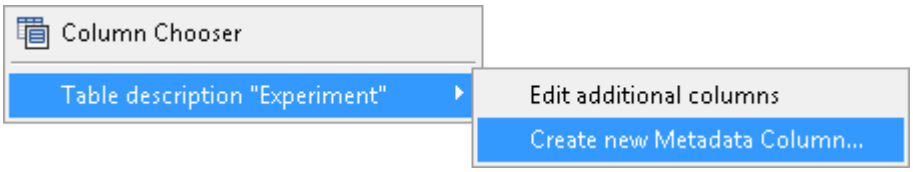
Example: Creating a Panel page as a dialog with a mandatory box

In this example, a new metadata column "*User*" is added to the already existing Setup page "*Documentation*" and declared as a mandatory box. Next, a Panel page is created on which the Setup page "*Documentation*" is displayed. This Panel page is called as a dialog, in such a way that the dialog can only be closed by clicking "OK" if there is an entry in the mandatory box.

Note: The page "*Documentation*" displays the columns of the table description named "*Experiment*".

Open the page "*Documentation*" in the **Setup**

Open the context menu over the white area and select "*Table description "Experiment'*"" > "Create new Metadata Column...".



Context menu of the "Documentation" page (Table description: "Experiment")

A dialog opens, which is named "*Collection editor*". Click on the button "Insert". A column with no title initially opens.

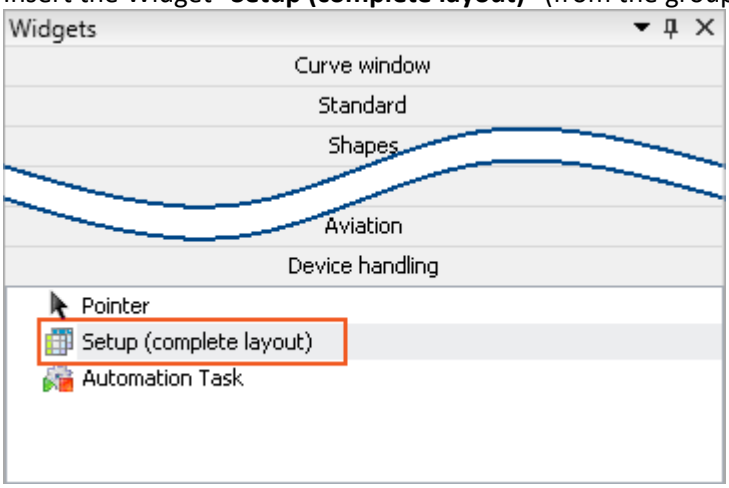
Enter the following **properties**:

- Title: User
- Type: Single-line text (unchanged)
- Mandatory box: ☑

All other settings remain intact. Close the dialog with "OK". The new column is now present on the page "*Documentation*".

Next, go to **Panel**.

Insert the Widget "**Setup (complete layout)**" (from the group "*Device handling*") into the Panel page.



Device handling - Widget "Setup (complete layout)"

Open the Widget's properties and select the **Layout**: "*Documentation*".

Note

If the new column is not visible, perform the following steps:
To display it, open the context menu in the Widget and select the menu item "Column Chooser". There, select the "User" (Tab: "Rows") column and use Drag&Drop to move it to the Widget.

Jump to **Sequencer** and there, add the command "*Show Panel page as dialog*".

Configure the command:

- Panel page to be displayed: Select the previously configured page
- Display dialog buttons: Yes
- OK-button: Active: if all entries are valid

All other settings remain intact. Close the configuration by clicking "OK"

Next, **start** the **Sequencer**. The dialog opens and only closes again when "*User*" is assigned a **valid value** (here: text).

Example: Defining the existing column "Test engineer" as a mandatory box

On the Setup page "Documentation", you find the metadata column "*Test engineer*". We will define it as a mandatory box.

Note: The page "*Documentation*" displays the columns of the table description named "*Experiment*".

Note

It is not possible to edit existing columns with the Standard version of imc STUDIO!

In order to access existing metadata columns, open the properties of the table description "*Experiment*":

- Jump to the Setup
- If necessary, start by turning on the visibility of the tool window "*Table Descriptions*" (ribbon: "*View*" > "*Choose Tool Windows*"). The tool window is not available in the Standard version of imc STUDIO.
- Open the context menu of the table description "*Experiment*" and select "*Properties*".

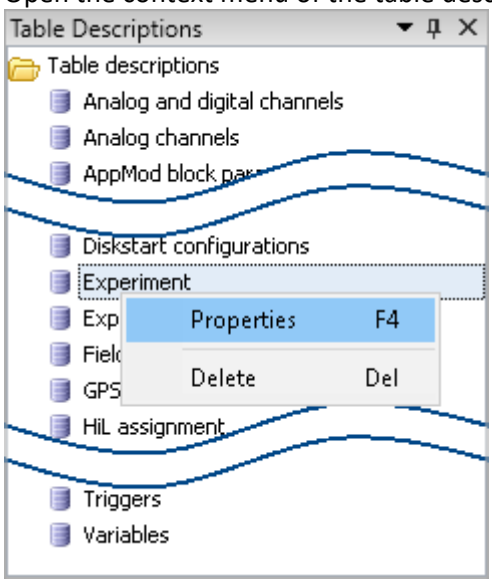
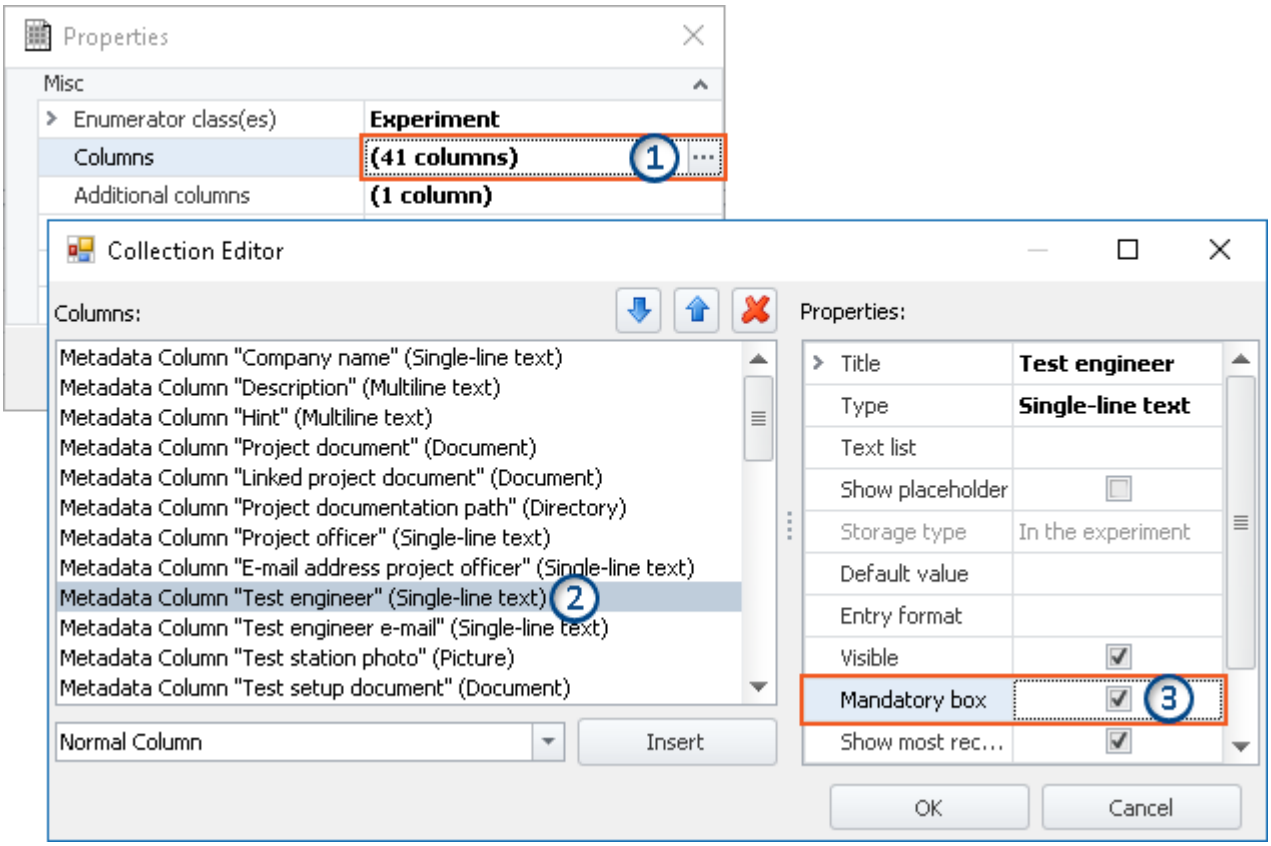


Table Descriptions tool window

Open the list of existing columns:

- In the Properties dialog, left-click the mouse over the column named "*Columns*" (1).
- Open the list by clicking on the button (...). A dialog named "*Collection Editor*" appears, which contains the properties of the individual metadata columns.



Properties and list editor for the table Experiment

Define the column "*Test engineer*" as a mandatory box:

- In the listing editor, select the column "*Test engineer*" (2)
- Put a check mark in the box for "*Mandatory box*" (3)
- Close the dialog with "OK".

Next, display the page "*Documentation*" in a dialog. The dialog only closes again when "*Test engineer*" is assigned a **valid value** (here: text).

8.7.2.4 Export/import of column/table descriptions

You are able to export/import complete table descriptions and individual column descriptions in a targeted way. For instance, you could transfer defined additional columns from one table description to another one. Or, you can exchange table descriptions without needing to import the entire view.

Exporting column/table descriptions

Using the menu item "*Import/Export*", you can **save the column and table description**:

Ribbon	View
Project > Import/Export ( )	Complete
Extra > Import/Export ( )	Standard

- Select the option "*Export*", and in the list, select the entry "*Export of Setup table description*" or "*Export of Setup column descriptions*"
- From the list offered, select the desired elements (a table description or one or more column descriptions belonging to a table description)
- Select an appropriate file name and path for the export file

Note Save the view beforehand

Save the view beforehand if you have made any changes (see "*Views* 156").

Import of column/table descriptions

Using the menu item "*Import/Export*", you can **import the column - and table descriptions**:

Ribbon	View
Project > Import/Export ( )	Complete
Extra > Import/Export ( )	Standard

Importing a Setup table description

- Select the option "*Import*" and from the list, the entry "*Import of Setup table description*"
- In the file selection dialog, select the saved file

The table description is imported along with all columns it contains. If any table description having the same "*ID*" exists, a prompt appears. Here, you can either replace the existing description, or rename the "*ID*" of the new table description.

Please not that the "*ID*" is only the internal ID. The displayed name (e.g. in the tool window: "*Table description*") is not changed. If you have renamed the "*ID*", it is now possible that two table descriptions having the same displayed name, but different IDs, exist. This situation is not recommended.

Note New functions can be overwritten

Please be aware that when making replacements, new functions may be lost if you import old views/columns.

Importing Setup column descriptions

- Select the option "*Import*" and from the list, the entry "*Import of Setup column descriptions*"
- In the file selection dialog, select the saved file
- From the list offered, select the target table description

All columns are imported to the selected table description. Table descriptions are not offered if they have columns with the same "*IDs*".

Reference See also

Exporting/importing Views 157

8.7.2.5 Configuring and designing the pages

8.7.2.5.1 Table description and Complete layout

Terms	Description
Table description	Collection of normal and additional columns
Column description	Configuration of a column in the table description
Complete Layout	Finished designed Setup pages. The page is filled with elements which are linked with table descriptions or columns belonging to table descriptions.
View	Arrangements of windows, menus, layouts, column arrangements, ...

The "*Table description*" **contains a variety of columns**. Based on these columns, the individual Setup pages are created. To do this, **the desired columns** belonging to one or more table descriptions are **selected and ordered**. The **new page** created can be **saved** as a "*Complete layout*" (as of imc STUDIO Professional) and thus made available e.g. for the metadata export.

In the "*Complete Layout*", only the information regarding the "visual" appearance is saved:

- which columns are displayed, and
- how and where they are displayed.

The columns' **values** are linked to the "*Table description*". This ensures that the values in all Setup pages are updated simultaneously.

Example

The column "*Channel name*" is used both on the page "*Analog channels*" and on the page "*Channel balance*". Now, if you change the channel name on one of these pages, it is automatically updated on the other page, since it affects the same column which simply appears on two different pages.

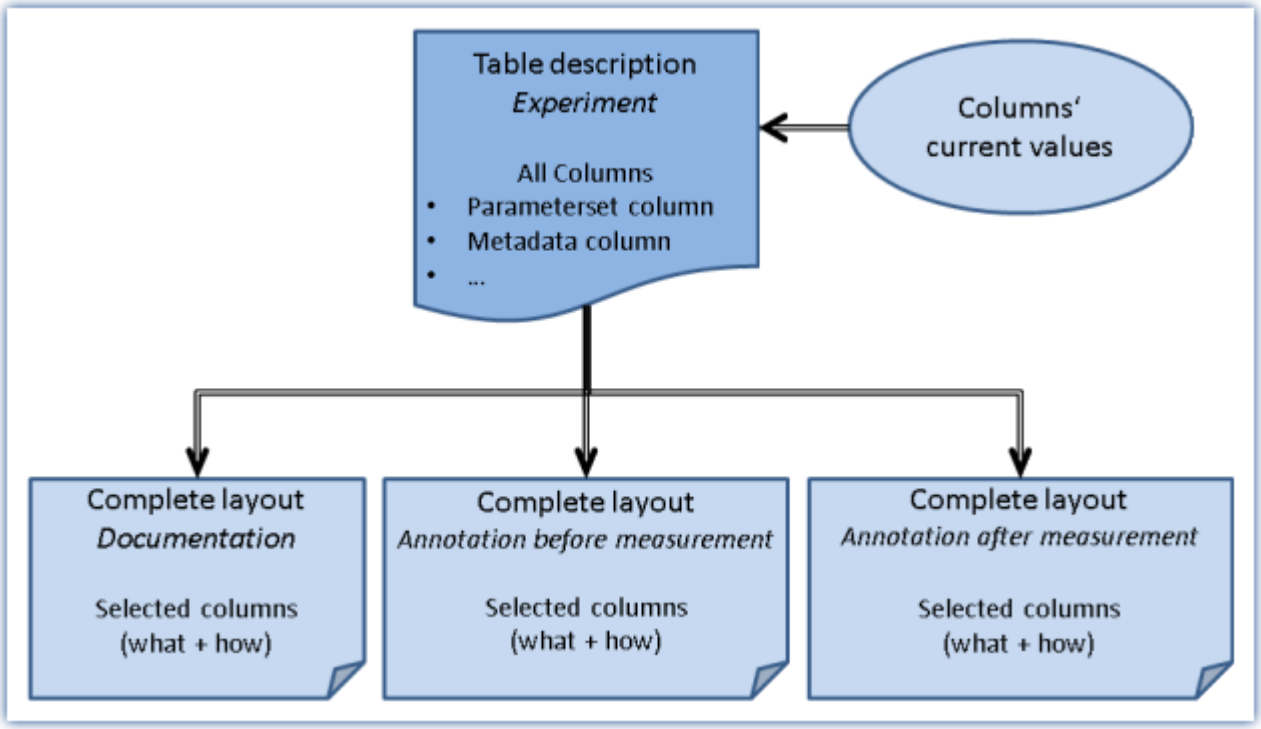


Table description and complete layout

8.7.2.5.2 Properties of the pages/Complete layouts

Note Prerequisite: imc STUDIO Developer

This function is only available in the product versions imc STUDIO DEV 29.

By means of the tool window: "*Layout Repository* 230", you can modify the pages' settings. To do this, open a page's context menu (Complete layouts) and select "*Properties*".

Property	Description
Filename	Write protected - internal name of the page
Name	Display name in in the tab bar
Image	Displayed icon in front of the name (to delete a selected image, press the keyboard combination: <CTRL> + <Del>)
Metadata template	Currently without function
Dialog callable as menu action	Activates the page for the call from the menu ribbon. A new menu item is created, which calls the page as a dialog.

8.7.2.5.3 Designing pages

Note Prerequisite: imc STUDIO Developer

The Design mode is only available for use from the product versions imc STUDIO DEV 29.

More fundamental **changes to existing pages**, or the **creation of new pages** are possible by means of the Setup-"*Design Mode*".

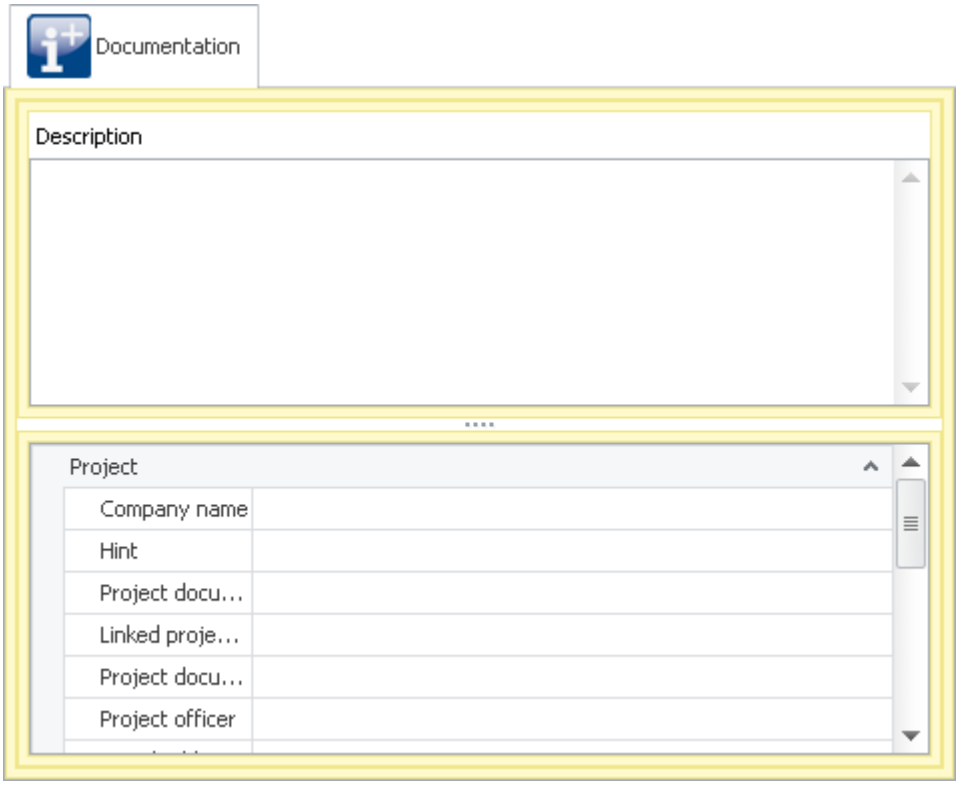
As a condition for being able to use "*Design Mode*", detailed knowledge of the interdependencies of table descriptions and the elements on the page is needed. Modification of the existing pages is only recommended for applications in which the existing pages are not sufficient.

Terms	Description
Design Mode	When "*Design Mode*" is active, the page elements can be modified.
Page	(in the tool window: "*Layout Repository*") Empty pages without content. At minimum, pages always have a frame. But they can also have multiple frame, which can be separated by a slider, for example.
Frame	Free area into which design elements can be inserted.
Elements	(in the tool window: "*Layout Repository*") Elements which can be filled with columns (columns from table descriptions). An element can contain additional elements, which, for instance, can be linked with other table descriptions, like the higher-level element.
Designer	External tool for enhancing existing elements.

Design Mode

To start the "*Design Mode*" of the Setup-component, select "*Design Mode*" in the context menu of an existing page.

The current page contains yellow "*frames*" for each of the usable spaces.



Example: the Documentation page
On the outer borders, the page's yellow frame is seen.
This contains two additional yellow frames. These are filled with elements.
The elements are linked with a table description.

In order to fill "*Frames*" with columns, an "*element*" must be inserted. You can also insert elements from the tool window "*Layout Repository*".

Example

Create a new page via the context menu of the tab bar: "*Create New Page*".

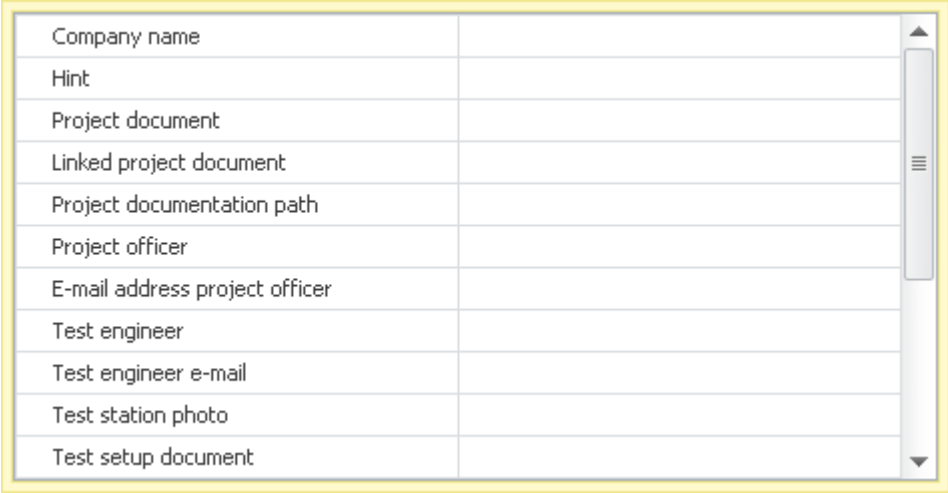
Open the new page by clicking on the mouse. An empty page with a frame becomes visible.

Insert an element from the tool window "*Layout Repository*", e.g.

- **Table:** this corresponds to the design of the pages "Devices", "Analog channels", ...
- **Detail-Table:** this corresponds to the design of the pages "Documentation", "Experiment", ...

The view only changes if the element already possesses additional design elements. The page remains empty after insertion of the detail-table, for example.

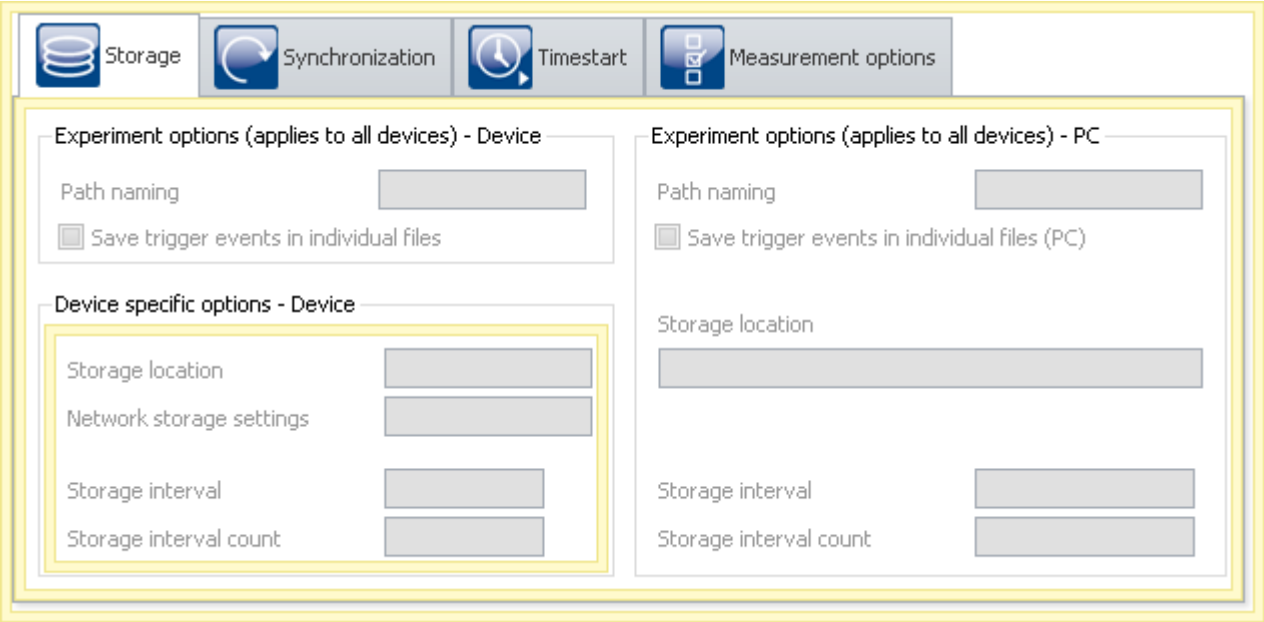
Next, you can link the element with a table description. Drag the appropriate table description from the tool window of the same name to the element. If you have selected a table-element, all of the columns are displayed.



Example of a new page with the element "Detail-Table" and the table description: "Experiment" Conforms to the lower portion of the page: "Documentation"

Nested elements

In order to assign nested elements to table descriptions, start at the lowest level.



In this example, when one assigns a table description to the middle frame, then the same table description is also assigned to the small frame ("*Device specific options*"). Only then can you assign a different table description to the small frame.

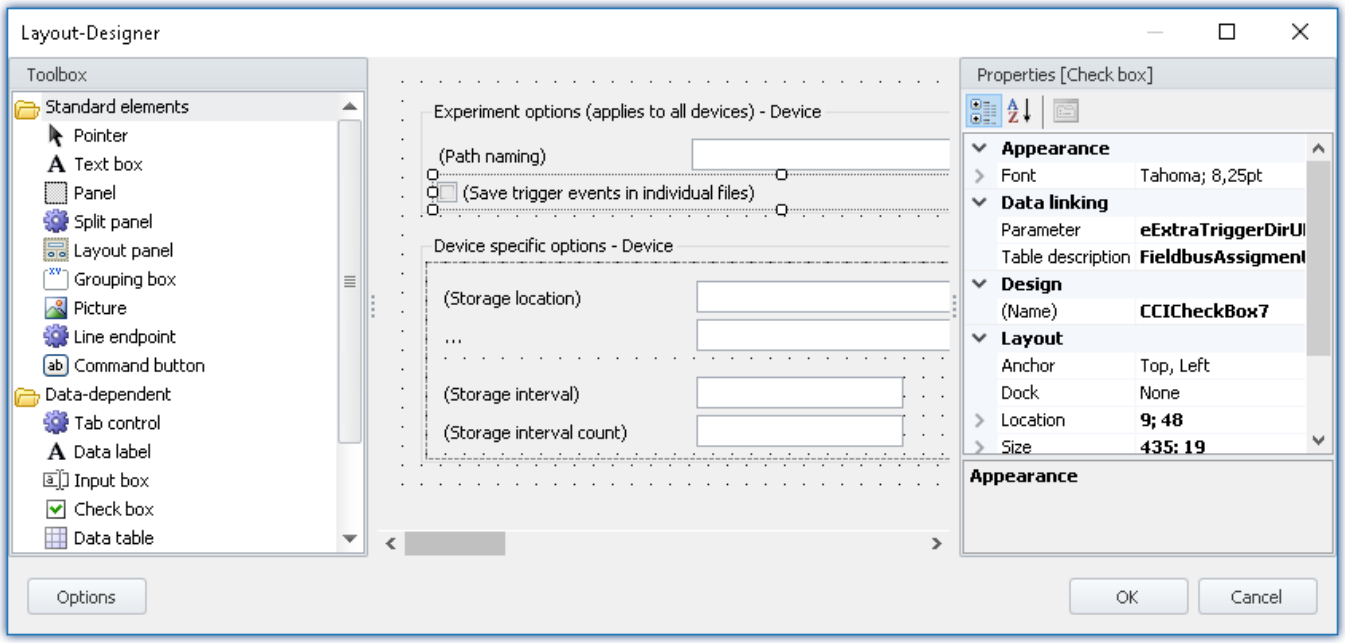
Designer

Warning Use of the "*Designer*"

Use of the "*Designer*" is expressly not recommended and is performed at the user's own risk. Any functions lost can only be restored by replacing the entire pages (from the "*Layout Repository*" or by resetting the view to the factory settings).

Open the Designer via a frame's context menu. Select "*Edit in Designer*". The contents of the selected frame are displayed in the "Layout-Designer".

In the Designer, you can add more elements and assign exact parameters to these.



FAQ

Why are some columns missing from the Parameter-list?

Answer: The list "*Parameter*" only shows "*Normal columns*". An example of these are parameters which come straight from the device.

All other columns are unfamiliar to the "*Designer*" and are thus no presented in the list, e.g.

- Calculated columns (such as the "Sampling rate")
- Combined columns (such as "ScalingPoint1" ("Point 1" of two-point scaling)
- Metadata columns, parameter set columns, mirrored columns, ...

Solution: You can enter the column's name in the input box, despite that it is not available as a selection in the list. Enter the exact column designation.

The linkage with the table description is made by means of the input box of the same name, or by means of Drag&Drop to the finished page.

8.7.3 Transferring an experiment to other devices

You can transfer a complete experiment to other devices. In the process, all possible settings are transferred along to the other devices. The transfer can be targeted for specific device modules; for this purpose, there is a software Assistant providing aid in making the desired assignment.

There are multiple ways to open the Assistant:

1. Transferring a complete experiment
2. Switching devices in an open experiment
3. Unable to set up connection to the device

Note

For devices of the firmware group B 187

For transfer of the configuration from imc CANSAS*fit*- and imc ARGUS*fit*-modules which are attached to an imc ARGUS*fit* unit, the following limitation applies.

The configuration of a module can only be transferred when the following conditions are met:

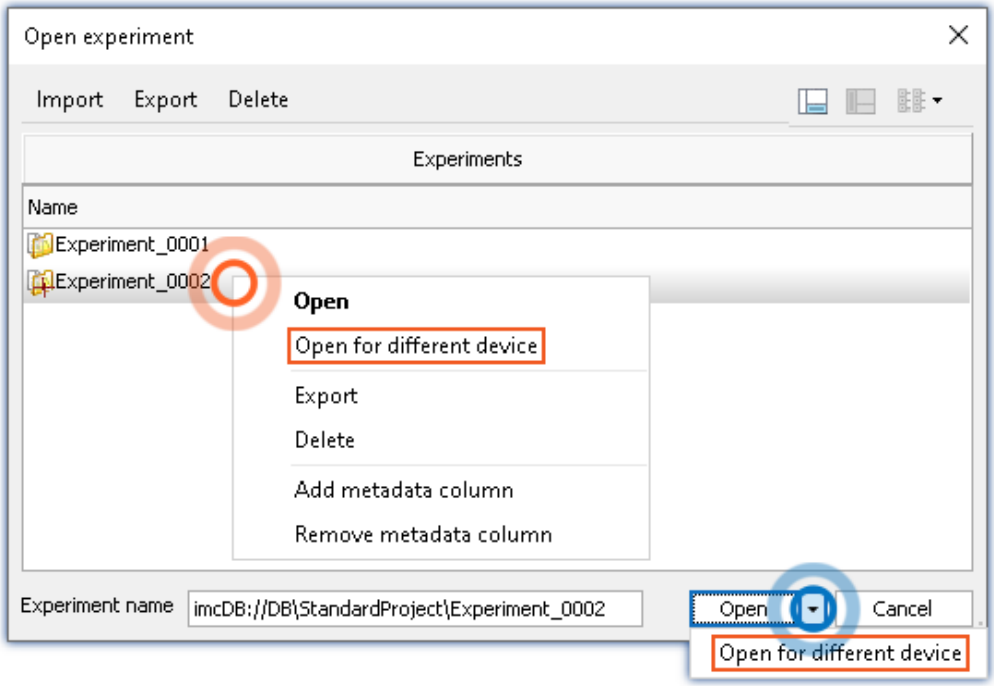
- same module-type (e.g. ARGFT\UTI-6-SUP to ARGFT\UTI-6-SUP)
- same device family, meaning imc ARGUS*fit* only to imc ARGUS*fit* and imc CANSAS*fit* only to imc CANSAS*fit*
- same scope of functions (e.g. same connection terminal)

The module's hardware revision stated on the plaque does not matter.

1. Transferring a complete experiment

Using the **open experiment dialog**:

- Open the experiment's context menu, or the drop-down list of the "*Open*"-button, and select "*Open for different device*".



This function is also available via the menu ribbon "*Project*" > "*Import / Export*", under "*Transfer experiment to other devices*".

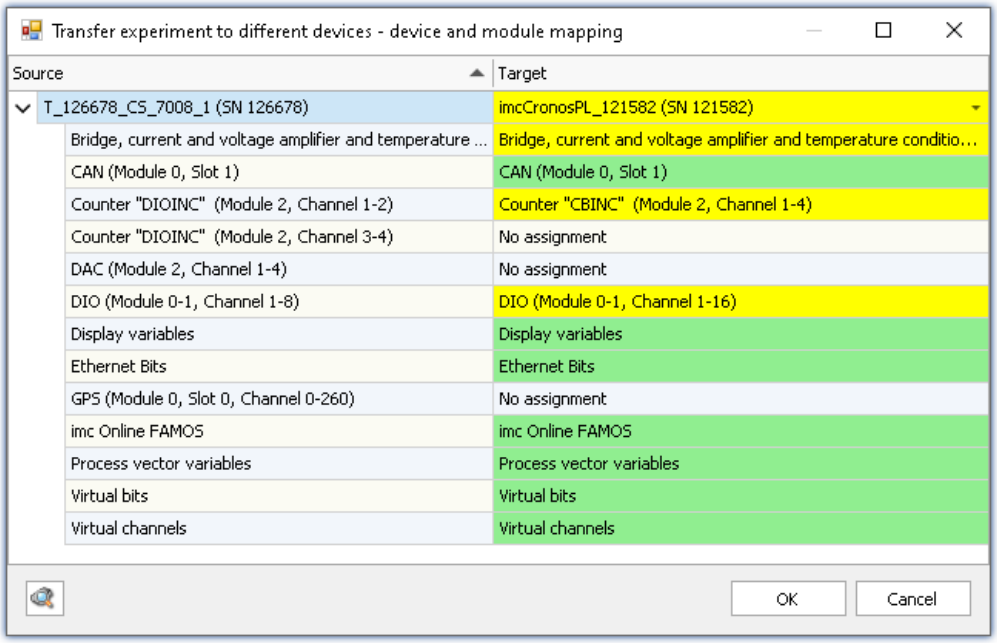
2. Switching devices in an open experiment

- Open an experiment
- Go to the Setup page: "*Devices*"
  - Select the device to be transferred (column: "*Selected*")
  - A dialog appears: Transfer the setting by means of the button: "*Transfer*"
  - In the next step, you can search for the device if it is not already part of the experiment. Otherwise, skip this step
- The Assistant opens

3. Unable to set up connection to the device

- If it is not possible to set up the connection to the device, it is possible to use a different device. The corresponding dialog for this purpose appears.

Configure device and module mapping



Transferring to different device
Example: Different device types

At left, there is a list of all devices belonging to the experiment (or only one device if it was de-selected). At right is a list of target devices; this side is initially empty and must be configured.

Below each device's entry is a list of all of its respective "modules" and components. What this list contains depends on the device's hardware configuration.

To set the target device desired, select a device from the drop-down list which is in the right-hand column on the row displaying the source device. As soon as a device is selected, the device is analyzed. The Assistant generates a suggested allocation (see example above).

When the devices are of the same model and hardware configuration, it is possible to use the default setting. If the experiment is to be transferred to a different kind of device, verify the suggested settings.

The colors:

Green: All of the module's settings can be transferred.

Yellow: Not all of the module's settings can be transferred. For example, the amplifier type may be different.

Please ensure that the target module is able to use the configuration required by the source module. E.g. a universal amplifier can use some of a temperature amplifier's settings.

Red: This assignment should not be selected.

What settings are applied:

In principle, all settings are applied, except for the **calibration values** (taring, bridge balancing, two-point scaling, ...). Settings which can not be applied are ignored and listed in the **log book** following the transfer.

As well, the **channel names** are adopted.

If multiple devices are used in an experiment, the channels are automatically assigned names with the device name as a prefix. If some of these channels were not renamed, the name remains intact after the transfer. The device name within the channel name then no longer reflects the name of the new device.

Automated analysis

When a target device is selected, the analysis starts. The result is the suggested assignment. This can be initiated for all devices automatically, as soon as the dialog is opened. To do this, click on this button:

When the button is clicked, all devices registered with the system are analyzed. Once this step has been concluded, it is possible to select a device.

Advantage: In the drop-down-list, it is immediately visible which devices are suitable for the transfer.

Disadvantage: **All known** devices are analyzed, requiring a corresponding amount of time, according to how many there are.

Reference Replacing modules

A similar dialog is displayed when one of a device's modules is replaced. See "*Change of the imc measurement device* 190".

8.7.4 Import of an imc DEVICES experiment

You can transfer a complete imc DEVICES experiment to imc STUDIO. In the process, all possible settings are transferred.

- Open the ***Project*** menu ribbon, select the command ***Import / Export***
- Select the option ***Import*** and next the list entry ***Import of an imc DEVICES Experiment***
- Select the desired experiment

FAQ

Question: In imc STUDIO,there are no default values for device variables like in imc DEVICES. What is the best way to handle this?

Answer: When an imc DEVICES experiment with default values is imported, a corresponding notification dialog appears. The default values are temporarily applied as the current values for the variables.

The current values are not saved along with the experiment!

- Way 1:
  Use the commands: "*Export Variable* 1758" + "*Refill variable* 1760"
  Export the variables which are important for the procedure. You can have the export-file imported to the corresponding location automatically.
  Use the command: "Export Variable" to effect the export.
  For the import, a button on a Panel page, for example, or the Sequencer with the command sequence: "*Refill variable* " + "*Execute menu action: Start (Start Measurement)*"
- Way 2:
  Use the command: "*Set variables* 1764". Here, the values can unfortunately not be adopted automatically. Configure the command manually.
  To set the variable, use a button on a Panel page, for example, or the Sequencer with the command sequence: "*Set variables*" + "*Execute menu action: Start (Start Measurement)*".

Question: How do I get the free-floating curve windows from imc DEVICES into the imc STUDIO Panel pages?

Answer: In imc DEVICES experiments, free-floating curve windows 1517 are used to view the measured data. In order that the curve window configurations not be lost when importing to imc STUDIO, the curve windows are also displayed in imc STUDIO. You can transfer 1515 the configuration of these curve windows to embedded curve windows (copy: `CTRL+SHIFT+C` | paste: `CTRL+SHIFT+V`). Thus, the configurations are also available in imc STUDIO.

Question: How do I make settings for "Autorepeat mode" in imc STUDIO?

Answer: In imc STUDIO, there are no "Autorepeat mode" in the sense in which they were used in imc DEVICES. Instead, please use the Sequencer to repeatedly restart the measurement, or modify the triggermachine in such a way that the measurement is not concluded, but instead a trigger repeatedly starts up again.

Question: Where do I find the action: "Balance active channels"

Answer: To adjust all active channels, use the command: "Execute device action" with the following configuration:

Enumerator class: Channel

Action column: Balance action

Action: Currently selected (thus, the imc DEVICES function is configured)

Filter type: All rows (passive channels are ignored!)

Note Known problems

The circular buffer contents are not imported correctly (imc DEVICES Version 2.7x - (build prior to 3. FEB. 2011).

The circular buffer memory set is not imported correctly from experiments which were created with the imc DEVICES version 2.7x (build date prior to 2/3/2011).

For all channels, the circular buffer memory for data storage is set to 1 min.

Solution: Open these experiments with the imc DEVICES version, which is installed with this imc STUDIO version, and save the experiment again.

After import, the device configuration is missing (imc DEVICES Version 2.6x and older).

imc DEVICES experiments belonging to Version 2.6x and older do not yet contain the device description file (*.umi). However, the import mechanism requires this file.

Solution 1: Before importing, copy the appropriate umi-file into the Experiment-folder.

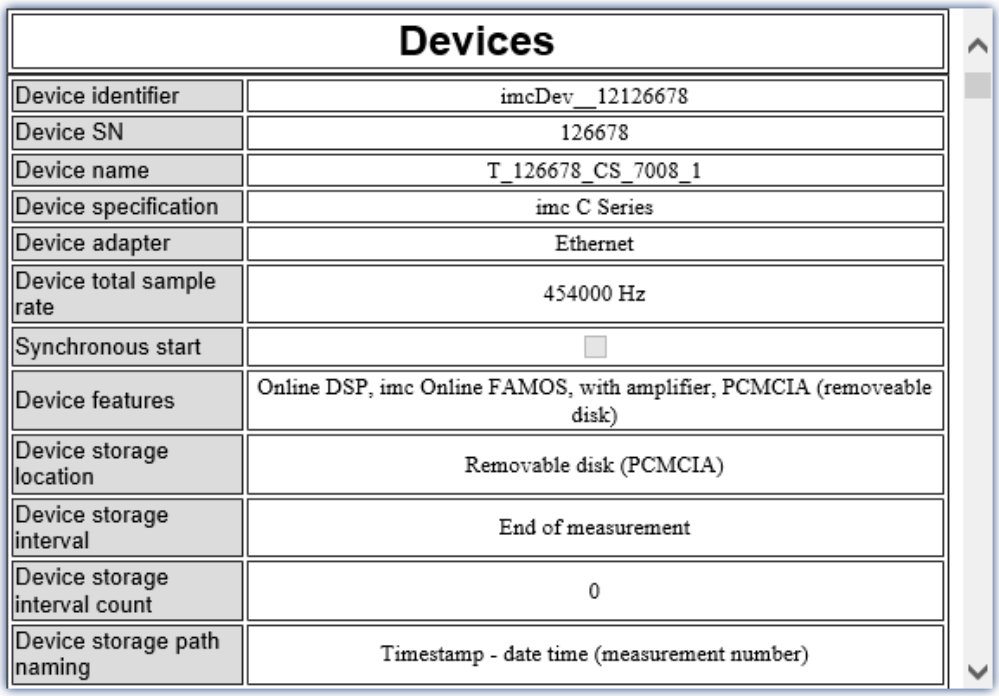
Solution 2: Open these experiments with the imc DEVICES version which is installed with this imc STUDIO version and save the experiment again **under a new name**. Import the newly created experiment.

8.7.5 Report of the Setup-configuration

You can **create a report** of the **current Setup-configuration**. The report is opened in the Browser and can be **saved or printed** there. Start the report via the menu ribbon if Setup is open.

Ribbon	View
Edit > Print ( )	Complete
Edit > Print Preview ( )	Complete
Home > Print ( )	Standard

A list of many device and channel configurations, as well as of the imc Online FAMOS-source text is shown.



Devices	
Device identifier	imcDev__12126678
Device SN	126678
Device name	T_126678_CS_7008_1
Device specification	imc C Series
Device adapter	Ethernet
Device total sample rate	454000 Hz
Synchronous start	
Device features	Online DSP, imc Online FAMOS, with amplifier, PCMCIA (removeable disk)
Device storage location	Removable disk (PCMCIA)
Device storage interval	End of measurement
Device storage interval count	0
Device storage path naming	Timestamp - date time (measurement number)

Example of a report
The first columns of the device configuration. The channel configuration appears further below.

8.8 Troubleshooting

Note

Only qualified technicians[1] are allowed to make repairs on the device! Unauthorized opening or incorrect repair of the device may greatly endanger the user (electric shock, fire hazard). Devices which have been altered or tampered will no longer comply with their license and may not be used. In case of accident (e.g. damage to housing, terminals, modules or power supply, or exposure to liquids or foreign substances), turn the device off immediately, unplug the power cord and inform our tech support 8.

1 Authorized / qualified personnel refers to persons familiar with the setup, installation, commissioning, and operation of the product and who hold certification for their respective skills.

- The device cannot be switched on 286
- Error message when starting measurement 286
- Device not found or measurement data is not displayed 286
- Hardware defects 287
- Error description 287

The Device cannot be Switched on

- Check the power supply and the fuses (below the power switch).
- After switching off the unit, wait at least 10 seconds before switching it back on.
- Switch the system off and disconnect all lines except the power supply. Then try again to switch the unit on.

Error Message when Starting Measurement

If error messages appear when starting the device, please check the following first:

- Check the connection between the unit and your PC (see "*Setting Up - Connect the device* 45").
- Next, check that the correct connection type has been specified. (see device model plaque).
- Turn the device off and then on again. Select Ribbon "*Home*" > "*Disconnect*" followed by "*Connect*".

The latter command reads the hardware configuration into the software.

Device not found or Measurement Data is not displayed

Error Message 103 - Check the device <=> PC connection and the power supply.

With Ethernet-interface: If your device is not located after using *Device search*, see the notes in "*Setting Up - Connect the device* 45".

Hardware Defects

If measurements return implausible results, follow these instructions to determine whether the hardware is defective:

1. Switch off device and disconnect all external power and signal input lines from the device.
2. Reconnect the power supply and switch device on: Does it boot correctly?
3. Connect the device to the PC (in accordance with the interface type)
4. Start the imc STUDIO software
5. Select the device
6. Select Ribbon *Home* (or *Setup-Control*) > *Connect*
7. Select Ribbon *Home* (or *Setup-Control*) > *Prepare*
8. Start measurement (measurement duration 10 s)
9. Select channel and display as curve
10. Restart measurement (measurement duration 10 s)

Error Description

To help locate defects when sending a unit back for repairs, please enclose a description of the error(s) encountered and, if applicable, a graph of expected versus incorrectly returned signal courses, as well as the following information for our tech support 8:

- Device serial number
- Software version used, export of the version information (About)
- Windows operating system version
- Experiment file from the database: <Experiment>.imcStudio

9 Setup pages - Configuring device

Each measurement setup can have a variety of requirements in terms of measurement device settings. Configure the device and each measurement channel according to its requirements. For this purpose, various software Assistants (wizards) are provided.

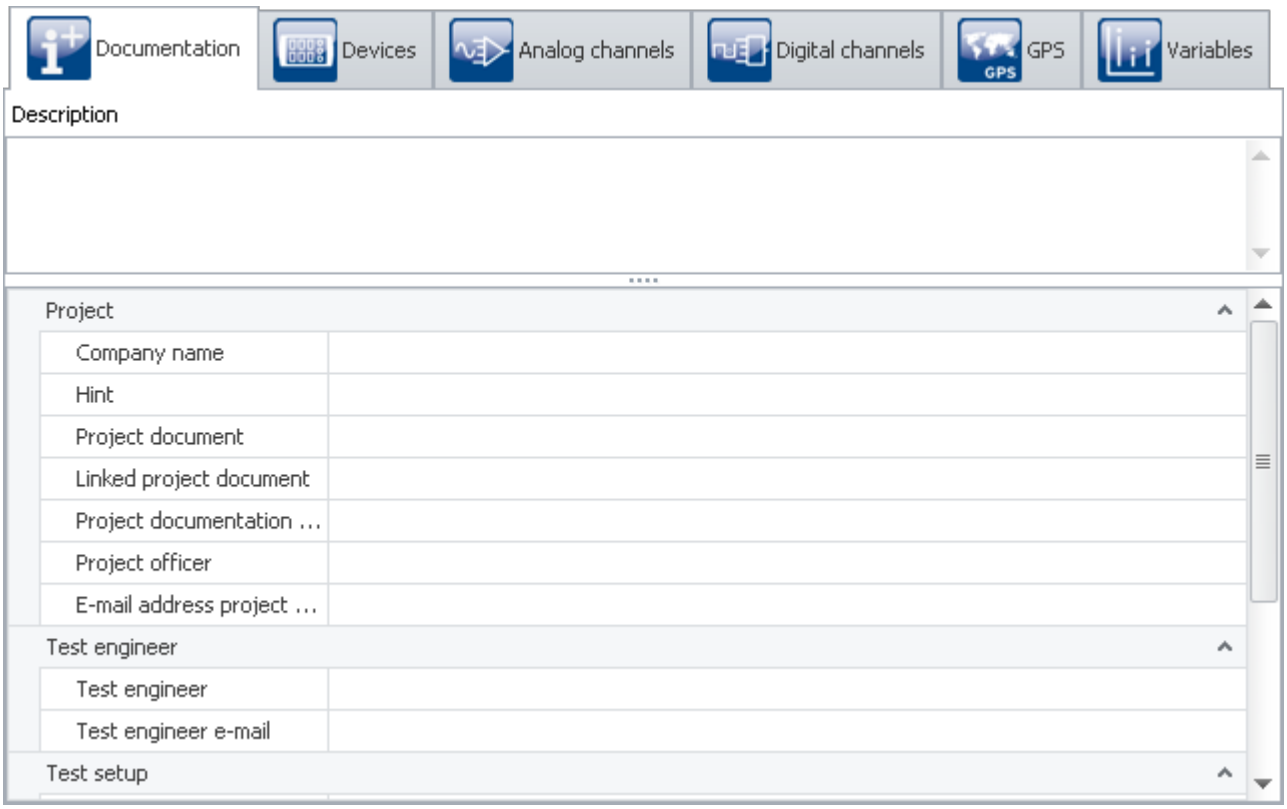
The following chapters describe the individual parameters of all pages (Setup pages) for configuring the measurement device.

Page	Description
Documentation 289	On this page, you can supply the experiment with **descriptions** and **metadata**.
Devices 289	Here, select the device for the measurement. Configure the data storage 292, synchronization 292 and the timed start 335, among other settings.
Analog channels 339	Here, configure the channel-specific parameters. Among others, those for the **analog inputs** and **virtual channels**.
Digital channels 339	Here, configure the channel-specific parameters. Among others, those for the **digital inputs/outputs** and **incremental counter channels**.
GPS 339	Here, configure the channel-specific parameters relating to the **GPS-channels**.
Variables 339	Here, configure the channel-specific parameters relating to the **Display variables** and **process vector variables**.
Channel balance 377	On this page, you can have various **adjustment** and **calibration types** performed.
Triggers 384	Here, configure the **Trigger-Machine** in order to perform targeted starting and stopping of channels.
TEDS 729	Here, import **TEDS** information in order to obtain precise configuration of the channels.
imc SIMPLEX 742	Sensors for imc DEVICEcore devices
Application module 409	Page for the configuration of **imc Application Module**.
Storage options and directory structure 411	Saving measured data on the PC and in the device
Device hard disk, removable drive 443	This section describes how to handle the storage media of the imc measurement devices and how to use them with imc STUDIO.
Fieldbusses 474	Operation and configuration of the various Fieldbus-systems

The menu ribbon offers various Assistants and additional dialogs 204 for configuration purposes.

9.1 Documentation (experiment description)

On the page "*Documentation*", you can supply the experiment with descriptions and metadata.



The page is subdivided into two areas: the **Table** below and a **Description**.

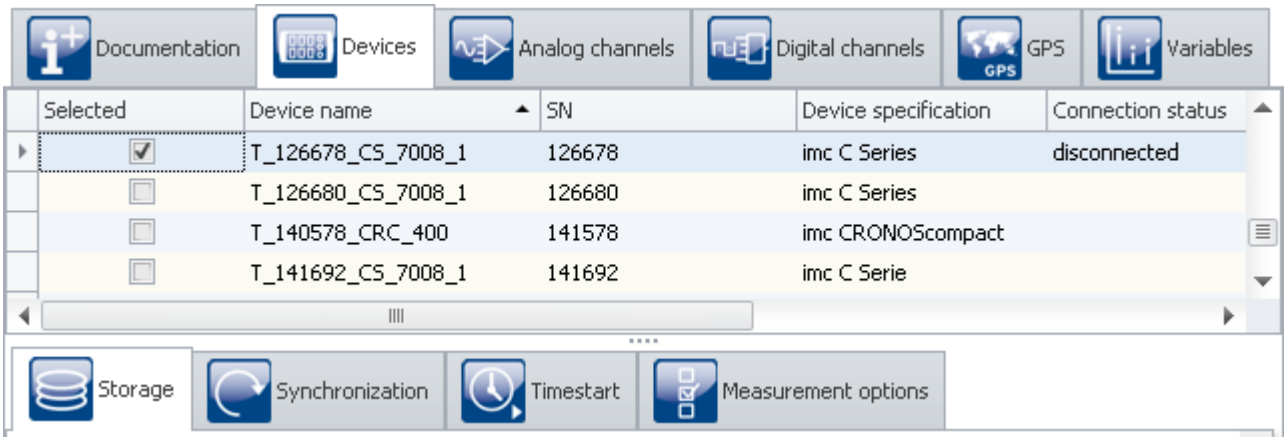
In the table, you can enter information pertinent to the experiment.

The table can be filled with multiple columns (see Configuring columns 250). By opening the **Column Chooser**, you can additionally move columns, group them together, or delete them.

In the box **Description**, you can provide your experiment with up to multiple lines of explicit description.

9.2 Configuring devices

On the page **Devices**, you can set all device-specific parameters.



"Devices" page in the plug-in "imc STUDIO Setup"

The page is subdivided into two areas: the "*Device table*" and the "*Dialogs*".

In the Devices table, all known devices are listed. It offers quick access to many settings. In the lower area, you can select various dialogs. These offer clearly-presented access to the properties.

The availability and selection of the settings depends on the device type, which determines what settings options the parameters can have.

The following dialogs may be present:

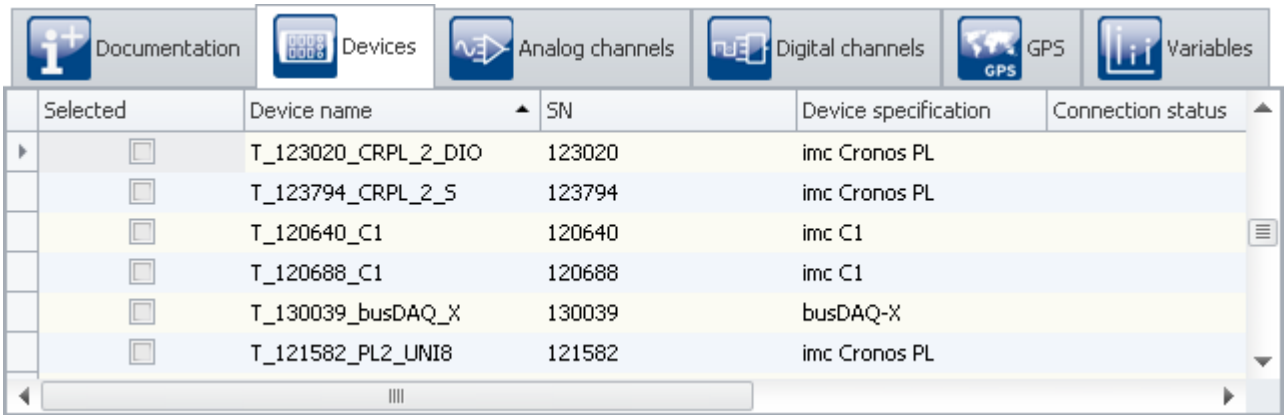
Page	Description
Storage 292	Settings for saving measured data in the device and on the PC.
Synchronization 292	Settings for synchronization to a clock or between multiple measurement devices
Timed start 335	This enables you to start the measurement at a defined point in time.
Measurement options 338	Various additional settings.

Reference See also additional info on:

- Operation and configuration of the tabular display 231 and the dialogs 238
- Showing and moving columns 248
- Creating and configuring additional columns 251

9.2.1 Device table

In this area, the **known** 241 **devices** are displayed in a structured table, as seen in the example image below.



Device-Table (example)

To be able to **use a device** for the current experiment, you must first find and select 241 it. To **adapt the devices' settings**, select one or more devices. After that, you either edit the table cells 234 directly or select the respective dialog 238 for the desired parameters.

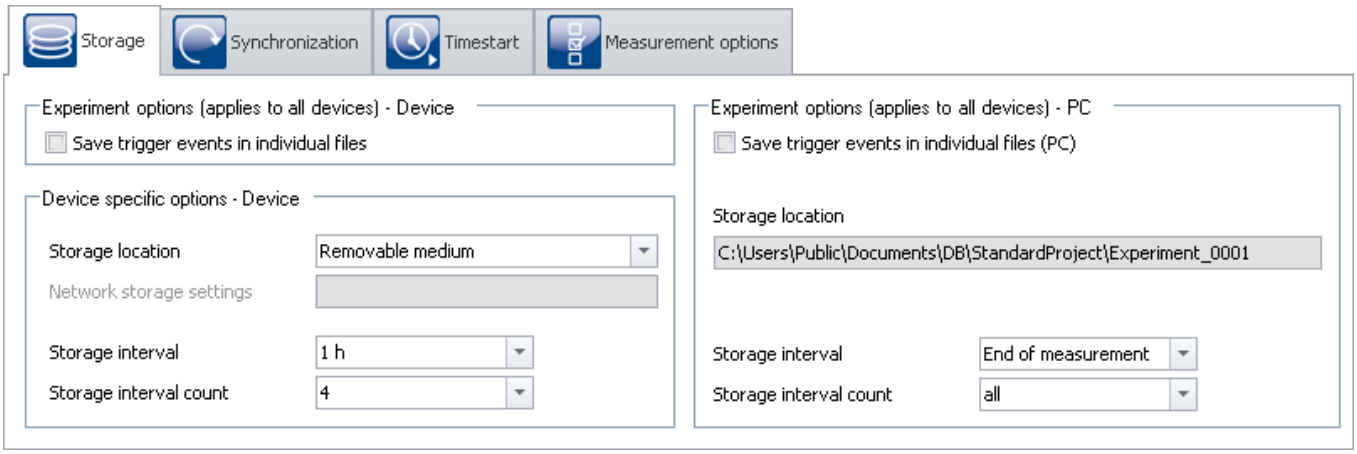
Parameter	Description		
	long name	*short name*	*column ID*
Selected	*Device is selected for measurement*	*Selected*	*Dev_SelForMeasurement*
	Clicking on the "*Selected*" box which corresponds to the desired device makes that device available for the experiment. You can also select multiple devices for your experiment. See "*Device Search - Known and Selected* 241"		

Parameter	Description		
	long name	*short name*	*column ID*
Device name	*Device name*	*Name*	*eDeviceNickname*
	The displayed name of the device. You can modify the name if desired. The device's name must be unique; i.e. no two devices may have the same name. Maximum number of characters: 24. In order to be able to modify the name, first establish a "*Connection*" with the device. After having made the change, perform the "*Prepare*", so that the change is saved in the device. The device's name is also saved with the experiment. As soon as you load an experiment with a different device name and prepare it, the changed name is overwritten.		
SN	*Device SN*	*SN*	*eDeviceSN*
	Unique identification number for the imc devices.		
Device specification	*Device specification*	*Specification*	*eDeviceProduct*
	imc device type. Reference this name, for example, to obtain the associated device documentation.		
Connection status	*Connection status*	*Connection*	*eConnectionStatus*
	Indicates the current connection status: "*connected*" or "*disconnected*".		
Measurement status	*Measurement status*	*Measurement*	*eMeasurementStatus*
	Indicates the current status of the measurement: "*stopped*", "*running*" or "*reconfigured*".		
Device control	*Device control*	*Device control*	*eDeviceControlAction*
	You can select an action via the selection list. By clicking the mouse over the button, you can cause the device to run the action. In contrast to menu items, this action only applies to the device selected. Thus you can stop a device during measurement, modify it in some way, and restart it, without affecting the other devices. The only other effect is that from that point in time, measured data are saved in a new folder. Disconnection and re-connection with the running measurement are not possible using this button. For that purpose, use the menu item "*Connect* 192".		

9.2.2 Storage

Reference

This chapter contains a brief introduction to saving channel signal data. More detailed descriptions of all functions are presented in the section: "*Storage options and directory structure* 411".



"Storage" dialog

This dialog set s the options for saving the measured data to the device and to the PC. You can decide how to name folders and where to place them. You can also set at what intervals to save the measured data.

Parameter	Description
Save trigger events in individual files	See "*Save trigger events in individual files* 434"
	In order to display the measured data associated with the events, add the column "*Event time* 1215" manually in the Data Browser.
Storage interval / Storage interval count	See "*Saving interval* 423"
	When interval storage is active, all **folders are deleted except for the specified number** of intervals!

9.2.3 Synchronization

This chapter describes the different options for synchronizing imc measurement devices.

This amounts to a **unified time base for different devices** whose measured data are compared. Synchronized devices can return **measured data which are exactly matched to each other in time**.

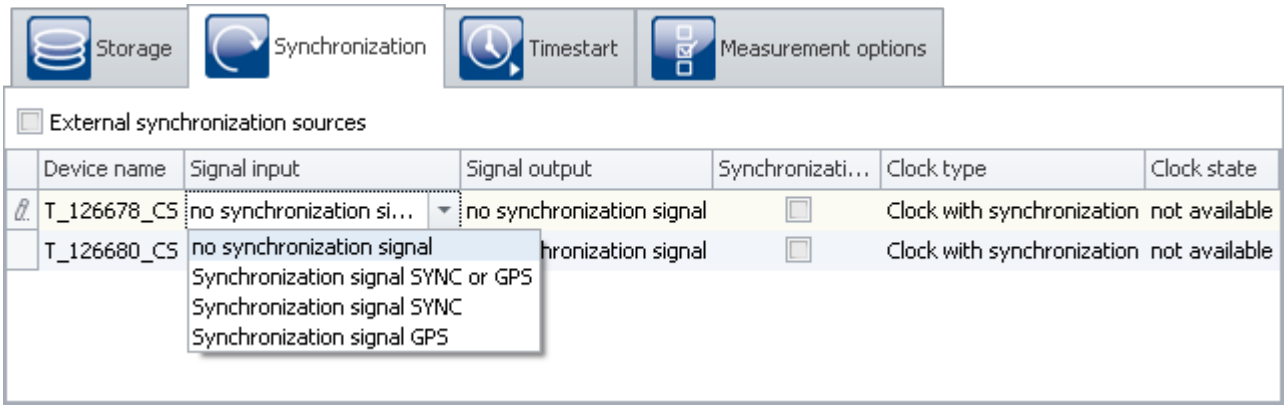
Reference Differences at runtime

Differences in the processing speeds between amplifiers, fieldbusses, etc., may also cause synchronization discrepancies. This topic is not covered here. Wherever it can, the system compensates for each of these devices' processing speed differentials.

Please find differences of the processing speed between measurement amplifiers in the corresponding device manual, chapter: "..." > "*Synchronicity*" > "*Delay*".

Setting synchronization

All **synchronization settings** are to be made in the dialog: "*Synchronization*" (on the Setup page "*Devices*").



In this dialog, you are able to **synchronize multiple devices with each other** or set a device according to an **external clock**. Synchronized devices can return **measured data which are exactly matched to each other in time**.

In order for the devices to be synchronized with each other, they must be **connected with the synchronization signal generator**. This can be either an external master or an imc device. Some models (e.g. DCF/IRIG-B) are connected with each other via the **SYNC terminal**. Others are synchronized via the **network** (e.g. NTP/PTP) or via **GPS**.

Parameter	Description
External synchronization sources	The devices are synchronized with an external synchronization source (e.g. DCF77, IRIG-B, ...). They have **no "*signal output*" of their own** and thus can not be the "*synchronization master*" in consequence. The checkmark's only effect is that no other "*Masters*" can be activated.
Synchronization master	If **no external synchronization source** is used and **only one device** is to be used **as the synchronization signal source**, put a checkmark here. This setting assistance is only relevant for IRIG-B and DCF77 and must not be activated for NTP and PTP synchronisation. The checkmark's only effect is that no other "*Masters*" can be activated.
Output	Here, select the synchronization signal according to which the other devices are synchronized. There are different output signals depending on the particular device. • E.g. DCF or IRIG-B. Devices which are synchronized with this device must be set for "*Signal input*" to either "*Synchronization signal SYNC*" or "*Synchronization signal SYNC or GPS* 315". • PTP serves to synchronize clocks within distributed computer systems. It forms a self-organizing hierarchical structure. In accordance with the specified settings, an imc device only becomes the PTP-master if there is no "better" PTP-master available. See the chapter: "*External timer: PTP* 318". • PTP-Master only 319: If the network includes no PTP-server true to absolute time, you can synchronize one imc device to a hardware synchronization signal (e.g. GPS). When you define that device as the PTP-master, it will no longer be synchronized by any other PTP-subscriber.
Input	Here, select the incoming synchronization signal, which depends on the synchronization signal source. For DFC and IRIG-B, please select "*SYNC*".

Note Simultaneously starting all devices

Channels of synchronized devices have correct absolute time. **However, the devices still start serially.**
To prevent this, select

- "*Synchronous start*" under "*Timed start* 335"
- an appropriate "*Start options* 336". E.g. "*Automatic timed start*".

See: "*Simultaneous start of all devices* 337"

Guidepost

Summary	Section
What is synchronization needed for? What advantages does it have?	What's the purpose of synchronization - and how does it work? 294
Brief explanation of terms. For instance: synchronization, timed start, time zone, simultaneous starting of all devices, Synchronous start (synchronized, but not simultaneous starting)	Definitions 297
Functioning and exact description of the various clock types. Internal device clocks, external time emitter, and the PC-clock.	Clock types 298
Various applications of synchronization. How are which clocks used? How do the devices need to be interconnected? What settings are to be made for imc STUDIO? Is there anything to pay attention to?	Synchronization varieties 311
What issues are involved when the synchronization signal is interrupted? What time discrepancies will occur? How to re-synchronize the signal?	Phase offset correction 322
What special issues are there in conjunction with time zones?	Working with time zones 322
If all devices are connected with an external Master: What happens if, for instance, the signal is lost? ...	What happens when? 327

9.2.3.1 What's the purpose of synchronization - and how does it work?

If you are only using one device, you may wonder what synchronization is good for. After all, your data already are associated with a correct time. For most users, it's sufficient to be able to compare measured signals with each other as long as they all were captured within the same device.

Synchronization is necessary if:

- **multiple devices** are measuring the same object simultaneously
- the **absolute time** is relevant

If **data** are captured **by different devices**, it should be possible to **track their chronological relationships** just as well as those of data originating in one device. Absolute time usually doesn't play any major role in this context.

Absolute time becomes important when **signals originating very far apart** are captured by separate devices. When such data are associated with an absolute time-stamp, they can be analyzed in conjunction with each other.

The synchronization process

After "Preparation", the devices apply whatever synchronization settings have been made.

Only **after the first preparation procedure**, the devices **synchronize** to the respective sources. The **device clock is adjusted** (phase), the **frequency is put under steady control** (frequency synchronized). When adopting the time, the **time zone** set in the device **is taken into account** (e.g. conversion of the GPS-time (UTC) to the time zone set).

Each synchronization type has special distinctions. With some, for instance, the PC clock time is adopted by the devices, with others not (for example, when an external Master is used). More information is presented in the chapter: "*VRTC - Virtual clock on the PC* 307".

When **starting a measurement**, its **start time** is either **determined** automatically (simple start, automatic timed start), or is specified explicitly (timed start). As soon as the start time has been determined, the device clock may not be set again (on this topic, see also the note further below, "*Time deviations*"). This also determines whether the measurement time is expressed in standard or daylight saving time.

Synchronization signal interruption (only for devices of group A5 187 and higher):
When the synchronization signal is available again after an interruption, the system determines how far the device's time deviates from the reference time. This phase error is corrected by adjusting the device clock rate until the device is once again absolutely synchronized to the reference time (on this topic, see: "*Phase error correction* 322").

Note Time deviations

If you click on "Start Measurement" ( ) before the synchronization has been completed, it is no longer possible to correct any deviation by the device clock from the reference clock.

If **no time deviations** may exist, **then wait to start the measurement** until all devices are indicated as "*Synchronized*". Or force a **simultaneous starting of all devices** 337 if you are using more than one device.

How does the deviation happen?

The starting time is set as soon as you click on "*Start measurement*" ( ) . This applies to every style of start, such as "*immediately*" or starting at a "*defined time*".

If the start time has been specified, the device clock will no longer be set (by a time adjustment to adopt the absolute time). In spite of this, the device will report that it is (frequency-) synchronized! (Clock status and imc Online FAMOS "`IsSynchronized`")

Thus, the frequency (clock rate) is synchronized, but there is an offset from the reference time.

Upon the next preparation procedure, the system tries again to adopt the absolute time given by the external reference clock. Toward this end, the synchronization is temporarily abandoned and then re-established. Starting of the next measurement should only begin after the re-synchronization has been performed. Otherwise, this process will be repeated upon each preparation procedure!

Determining the deviation from the reference clock in conjunction with NTP- and PTP-synchronization

The device clock's remaining deviation from the reference clock is represented by the variable "*pv.State.ResidualSyncTimeDeviation*". This value is determined when "*Start Measurement*" is pressed. At that moment the current deviation is determined.

- "0.0" : "Initial value", "no deviation" or "Unable to determine value" (see below).
- ">0" : Deviation in seconds

No statement for any other synchronization types

With the other synchronization types (such as GPS, DCF, ...), the value cannot be determined and is always set to "0.0"!

Note Cases where the device clock is not adjusted

When the PC prepares a measurement, it also sets the device's time.
The device time is **not** set by the PC if:

- a valid synchronization signal is already applied (in particular when the device is synchronized), or
- the device's synchronization output (*SYNC*) is already running (otherwise, all devices must restart the synchronization), or
- "NTP" or "PTP" is selected as the synchronization signal

9.2.3.2 Definitions

Synchronization:

Devices are synchronized if their time bases are rigidly linked. This is achieved in either of two ways:

- *Use of an external time signal:* If the time signal of an exact time standard (absolute time encoder: e.g. DCF, GPS, NTP, ...) is available, it can be used to synchronize the devices. The time bases of the devices are then not only synchronized to each other but also to the time standard used.
- *Use of a device's own time base:* The time signal of one of the devices (master) is used to synchronize other devices (slaves). The time bases of the devices are only synchronized with each other, but not with absolute time.

Mixed-operation is possible, such that one device is synchronized with an external source and all other devices synchronize to the (Master) device.

Timed start:

The start is triggered by a particular clock at a pre-defined point in time. Each device has its own clock. If multiple devices are to start simultaneously, the precision of the start time depends on the precision of the synchronization.

Synchronous start (synchronized, but not simultaneous starting):

This operating mode only works with multiple devices, and each device's own clock triggers the start. For this purpose, all clocks must be synchronized. The master clock is one of the following:

- an externally synchronizable clock
- a device's clock module, which has been declared as the Master, or
- each device uses its own time source,
- ...

Simultaneous start of all devices:

When Synchronization is activated, simultaneous start of all devices is possible. For this purpose, certain conditions must be met. Among others, an appropriate "*Start options* 336" must be selected. E.g. "Automatic timed start".

With the option "immediately", the devices measure synchronously, and are started quickly, but serially. The measured data are then only synchronized with each other in the absolute time display.

Additionally, the option: "*Synchronous start* 335" must be activated.

See: "*Simultaneous start of all devices* 337"

Time zone and switch to/from daylight saving time:

The device software is able to take the switch to/from daylight saving time or between time zones into account. The setting is made in the device properties 223: "*Device time, synchronization*" > "*Time zone*" and "*Automatic daylight saving time switching*".

Ribbon	View
Setup-Configuration > Device Properties	Complete

Therms	Description
STD	**ST**andar**D** time
DST	**D**aylight **S**aving **T**ime
DST-state	DST or STD
DST-switch	switch from DST to STD or vice versa
UTC	Coordinated Universal Time (Greenwich meridian); no DST-switching!
localtime	the time prevalent in the respective time zone with regard to the current DST-state
RTC	RealTimeClock; internal device clock
SyncRTC 299	Synchronizable RealTimeClock of imc devices
VRTC 307	Virtual clock on the PC

9.2.3.3 Clock types

Internal clocks

Clock type	imc measurement device
SyncRTC 299	For all device types: Synchronizable RealTimeClock of imc devices
VRTC 307	Virtual clock on the PC

External clocks

Clock type	imc measurement device
DCF77 300	For devices belonging to the firmware group A 187
IRIG-B 299	For devices of the group A5 and higher 187 and devices belonging to the firmware group B 187
GPS 300	For devices belonging to the firmware group A 187 and devices of the group B11 187
NTP 300	For devices of the group A5 and higher 187 and devices belonging to the firmware group B 187
PTP 301	For devices of the group A7 187 with the suffix "-GP", as well as CRXT

9.2.3.3.1 Internal Clocks: Device Clock - SyncRTC

SyncRTC: **Sync**hronizable **R**eal **T**ime **C**lock.

The SyncRTC is the device's internal clock.

It is used to determine the start time of the first measurement after initialization, for example. Once this start time has been determined, the SyncRTC's absolute time plays no further role until the next measurement's initialization. Any time readings are derived by adding the sampling times to the start time, and the sampling times are derived from the SyncRTC's frequency.

If for instance the clock jumps (time change for daylight savings, etc.) this does not affect the measurement.

There are a variety of ways to synchronize the SyncRTC with external time sources. By doing this, it is ensured to both use the correct absolute time and the correct frequency.

Most devices* come with an external "*SYNC*" terminal and a GPS-input for synchronization with other clocks. The SYNC terminal can be configured as your choice of either an input or output. You can select either DCF77 or IRIG-B 002 (if supported by the device) as the signal format.

As the input signals, you can use other imc device signals or external time sources (e.g. GPS-clocks from Meinberg, Hopf, etc.).

Depending on the device type, additional synchronization mechanisms are also available. See: Clock types: External clocks 298.

*Note: Certain terminals are not included with some device and housing types!

9.2.3.3.2 External Clocks: IRIG-B, DCF

IRIG-B

The IRIG-B signal is applied via the SYNC socket. The measurement device processes a LOW* active 5V TTL signal and supports IRIG-B002, B000, B001, B003. As well, inverted signals are recognized and evaluated by the hardware.

For all models, the pulse rate is 100 pulses per second, DC Level Shift (DCLS), width coded, no carrier. The last digit indicates the correct information:

B000	BCD, CF, SBS	BCD - Binary Coded Decimal, Code of the time (HH,MM,SS,DDD)
B001	BCD, CF	SBS - Straight Binary Second of day, daily second (0....86400)
B002	BCD	CF - Control Functions, depending on application
B003	BCD, SBS	

The models listed here can be connected for the input signal. However, the measurement device only applies the BCD information!

Duration of synchronization: With a typical synchronization time of approx. 20 s, the IRIG-B is the fastest model available.

Note

- For the synchronization signal input, only "*Synchronization signal SYNC*" may be set.
- The time information is expected to correspond to the device's local time!
- **During signal output** (IRIG-B002) only the **BCD information** will be applied, this means no CF and no SBS information will be applied.

*: Due to the greater steepness of edges, LOW active signals achieve better precision than HIGH active signals.

DCF77

The DCF77 signal is applied via the SYNC socket. The measurement device processes a LOW* active 5V TTL signal (1Hz pulse). As well, inverted signals are recognized and evaluated by the hardware.

Duration of synchronization: The complete time is encoded within one minute and begins at the start of the next minute. For this reason the minimum duration for the synchronization process is at least one minute!

Note

- For the synchronization signal input, only "*Synchronization signal SYNC*" may be set.

*: Due to the greater steepness of edges, LOW active signals achieve better precision than HIGH active signals.

9.2.3.3.3 External Clocks: GPS

The GPS (**G**lobal **P**ositioning **S**ystem) receiver is connected to the "*GPS*" socket. Depending on the device type, the GPS connection is either a 9-pin DSUB or a 7-pin LEMO connector. The supply is provided by the measurement device. In this way, the GPS receiver finds its positions as soon as the device is activated.

The GPS signal, received all around the world, returns, among other things, Greenwich Mean Time (GMT)

Duration of synchronization process: The minimum duration for synchronization depends on the reception and on the last location of use. Upon first use, the receiver needs a few minutes until multiple satellites and its own position have been located. Upon the next start, the last position data are used, and the satellites are found found more quickly.

With a direct line of sight to the sky, the receiver is typically ready to transmit time and position data to the device after one minute.

9.2.3.3.4 External Clocks: NTP

The Network Time Protocol (NTP) is a standard for synchronizing clocks in computer systems and uses the transport protocol UDP. It was specially developed to enable reliable distribution of time information via networks having variable package runtimes.

Precision: NTPv4 can synchronize a local system's time via the public Internet at a precision of approx. 10 milliseconds. In local networks under ideal conditions, precision levels of 200 microseconds and even better are possible.

Duration of synchronization process: Synchronization to an NTP server is a lengthy process. At intervals, the device exchanges a data package with the server. With the default settings and a large time divergence it can take up to 3h to achieve a precision level in the range of 20ms, or up to 12h until the deviation is <5ms.

In order for synchronization to begin immediately upon system activation, NTP can be set up as the measurement device's default synchronization input (see "*device properties* 221").

There you can also reduce the interval time in order to achieve better precision and faster synchronization. "Min-Interval" counteracts large divergence; as the values improve, the time interval increases up to "Max-Interval".

After the first synchronization procedure, the device should remain activated for at least one hour. Only then will the frequency deviation determined be saved in the device. When the device is restarted, the frequency deviation is known and the synchronization process proceeds correspondingly faster.

The measurement device supports up to two NTP servers.

Application in imc STUDIO

On the Setup page "*Devices*" in the dialog "*Synchronization*", **the signal input is configured**. For NTP-capable devices, the "*Synchronization signal NTP*" is available there, among others.

When the **signal input "*Synchronization signal NTP*"** is selected, an additional tab with the name "NTP" appears. Here, you can configure NTP for the devices. This tab only appears when the associated device is selected. Some of the parameters can also be configured as settings in the device properties 221.

You can insert the following parameters in the table or partly configure them under the "NTP" tab:

- Synchronization NTP server 1 and 2
- Synchronization NTP deviation time max.
- NTP synchronization interval
- Synchronization waiting time on self start max. (applies to all synchronization types)

For making NTP settings, the following parameter belonging to the device properties 221 is also relevant:

- observance of the "*daylight saving's time*" transitions

Note

- If the phasing between two voltages in different devices is to be compared, only the Master is to be synchronized to NTP. This synchronizes additional devices via SYNC (DCF77/IRIG-B), see NTP time via LAN in Master/Slave setup 316.
- The quality of the synchronization can be monitored with the help of the pv-variable "pv.SyncTimeDeviation". It returns the estimated deviation of the NTP-server' clock time in seconds. The momentary value is not relevant! only after observing its course over longer periods is it possible to get an impression of the synchronization quality!
- How constant the synchronization is depends strongly on temperature fluctuations in the device! Every °C can temporarily cause a sizable deviation. High precision is only achieved under constant temperature conditions (e.g. in switching cabinets).

9.2.3.3.5 External Clocks: PTP

The Precision Time Protocol (PTP) serves synchronize clocks within distributed computer systems. Compared with NTP, its achievable precision is significantly higher.

Characteristics	Description
Prerequisites and limitations	PTP-capable device 187 are required. PTP-capable network equipment (switches) and optionally a PTP time server are required. The synchronization is confined to the local network. No possibility for synchronization via the Internet is provided.
Structure	PTP creates a self-organizing hierarchical structure, in which each device listens for information from the current time server throughout the network. If it does not identify any server, or if its own information is superior to that of the current server, it sends its own information to the network. The current server as well as all clients recognize that there is a new and better server and use the new server from then on. The previous current server stops sending its information. In principle, any PTP-capable imc device can also serve as a time server. By means of the configuration option "Slave only", it is possible to prevent a device from making itself available as the time server.
Precision	Under good conditions, PTP achieves a precision to the extent of well less than 1µs.

Characteristics	Description
Duration of synchronization process	The synchronization takes about 20-30 seconds after preparing the measurement. Depending on the set PTP parameter, the synchronization can last longer. The sync process takes longer if the measurement was started before the device could sync it's time with the time server. The offset in time will then be balanced over a longer period of time by adjusting the device's internal clock (SyncRTC).
Time zone	For the use of PTP it is necessary for the time zone to be set in appropriately for one's own needs, or else there may be an unintended time offset between the imc device and other devices or the PC.

Note

The parameters "*First step threshold*" and "*Step threshold*" can be used to influence the behavior before and after the start of the measurement (see the description of the parameters).

Application in imc STUDIO

On the Setup page "*Devices*" in the dialog "*Synchronization*", **the signal input is configured**. For PTP-capable devices, the "*Synchronization signal PTP*" is available there, among others.

When the **signal input "*Synchronization signal PTP*"** is selected, an additional tab with the name "PTP" appears. Here, you can configure PTP for the devices. This tab only appears when the associated device is selected.

Presets

Using various parameters it is possible to modify the protocol to meet your requirements. imc STUDIO offers a number of presets for these parameters. Additionally, you are able to edit every parameter.

Preset	Description
IEEE 1588 2008	Standard defined by IEEE (default setting)
fos4X	Configuration for allowing synchronization with the fos4X devices, which can be used in imc STUDIO as Third Party devices.
User-defined	Enables configuration of all PTP-parameters. If "user-defined" is selected, all parameters are displayed.

Note

The values set are discarded as soon as another preset is selected.

Note

- The PTP parameters are saved along with the experiment.
- For the parameter "Signal output", there is no option like "Synchronization signal PTP".
- If the parameter "External synchronization source" is activated int he dialog "Synchronization", then for all PTP-devices "Slave only" is activated.

Comparison: NTP vs. PTP

	NTP	PTP
Precision	10 ms via internet 200 µs in the local network	below 1 µs (down to 100 ns achievable)
Settling	up to 3 h, for a 20 ms precision level 300	approx. 20-30 s
Structure	internet time server supported specify time server	for local networks only self-organizing: the best server wins
Hardware	standard hardware	special devices: group A7 and higher with the suffix "-GP", as well as CRXT. PTP compatible Network switch required

PTP - Parameter

Among parameters there is a distinction between "Device-specific" and "Experiment options (for all devices)". "Device-specific" always applies to the device currently selected in the Devices table.

If you selected the preset "User-defined", you are able to modify all PTP parameters according to your requirements. The parameters are displayed as soon as "User-defined"is selected.

The following parameters can either be modified or are specified by the preset (the description of the parameters is provided further below):

Device-specific options				
Parameter	**Min**	**Max**	**IEEE 1588 2008 (Default)**	**fos4X**
Slave only	0	1	0	0
Priority 1	0	255	128	127
Priority 2	0	255	128	127
User description				
Delay asymmetry	-1000000	1000000	0	0
UDP TTL	1	255	1	1

Experiment options (for all devices)				
Parameter	**Min**	**Max**	**IEEE 1588 2008 (Default)**	**fos4X**
Domain number	0	255	0	0
Announce interval	-3	5	1	1
Sync interval	-6	5	0	-2
Delay request interval	-6	5	0	0
Announce receipt timeout	2	10	3	3
Delay mechanism	Auto, E2E, hybridE2E, P2P		E2E	E2E

Experiment options (for all devices)				
Parameter	**Min**	**Max**	**IEEE 1588 2008 (Default)**	**fos4X**
Network transport	UDPv4, UDPv6, L2/IEEE 802.3		UDPv4	UDPv6
TAI-UTC offset	-32768	32767	36	36
UDPv6 scope	0x02	0x0E	0x0E	0x02
Transport specific	0x00	0x0F	0x00	0x00
Step threshold	0.0	DOUBLE_MAX	0.0	0.0
First step threshold	0.0	DOUBLE_MAX	0.00002	0.00002
PTP destination MAC address	01:1B:19:00:00:00			
P2P destination MAC address	01:80:C2:00:00:0E			

Description

Device-specific options	
Parameter	**Description**
Slave only	Prevents the device from providing its clock time to other subscribers. This parameter can be set for all Preset-variants.
Priority 1	Sets the priority according to which the devices choose a server. The subscriber with the lowest value becomes the server. Priority 1 is the strongest among criteria!
Priority 2	Sets the priority according to which the devices choose a server. The subscriber with the lowest value becomes the server. Priority 2 is a weak criterion! Priority 1 and the server quality characteristics specified by the device manufacturer outweigh the parameter Priority 2, which is only evaluated if all other values are identical.
User description	Identification of the device for PTP-Management programs.
Delay asymmetry	With this value, a deterministic time offset resulting from asymmetric packet run times in the network can be compensated. For this purpose, exact information or measurements are necessary. If you use appropriate PTP-switches, it should not be necessary to adjust this parameter.
UDP TTL	Lifespan of the PTP-packets in the network. If a PTP-network is to be operated beyond the confines of one router, this value must be increased accordingly. This value is only taken into account if UDPv4 was set as the network transport.

The following parameters match for all devices involved. This applies not only to the devices from imc but to all switches, PTP-servers and other devices!

Experiment options (for all devices)	
Parameter	**Description**
Domain number	Using the domain number it is possible to create multiple independent PTP-domains. Devices having differing domain numbers do not influence each other.

Experiment options (for all devices)	
Parameter	**Description**
Announce interval	This interval determines at what intervals the current time server sends its quality information to the network. The value is stated as a power of 2. The value range of -3 to 5 covers interval times of 1/8s to 32s.
Sync interval	This interval determines at what intervals the current time server sends its time information to the network. This value is stated as a power of 2. The value range of -6 to 5 covers interval times of 1/64s to 32s. When the value is less than -3, imc devices can no longer send data at the speed required. These settings are only offered for reasons of compatibility and should be avoided. Note that large values lead to long intervals and thus slow down the synchronization and decrease its precision!
Delay request interval	This interval specifies at what intervals the PTP-slaves determine the packet run times in the network. The value is stated as a power of 2. The value range of -6 to 5 covers interval times of 1/64s to 32s. When the value is less than -3, imc devices can no longer send data at the speed required. These settings are only offered for reasons of compatibility and should be avoided. Note that large values lead to long intervals and thus slow down the synchronization and decrease its precision!
Announce receipt timeout	Specifies how many "Announce Intervals" a device waits until it assumes that the current PTP-master no longer exists and takes appropriate measures. Example: Announce interval = 1, Announce receipt timeout = 3 -> The time is: 2^1s * 3 = 6s.
Delay mechanism	For measurement of the packet run times, multiple procedures are available. Select the procedure here which you wish to use. **E2E:** With the End-To-End procedure, the packet run time along the entire route between the PTP-Slave and the PTP-Master. This procedure is the one most commonly supported and theoretically also works with switches which are not PTP-capable. For this reason it is the default. However, it is strongly to be recommended NOT using switches unless they explicitly support PTP, otherwise unpredictable behavior can result which can impair the synchronization by any unforeseeable amount. The procedure can be used without any problems in conjunction with appropriate switches. **P2P:** The Peer-To-Peer procedure can exclusively be used with P2P-capable infrastructure. Here, the packet run times are determined between two directly connected nodes (device and switch or switch and switch). On the way between the master and slave, one correction value from each switch traversed is incremented in the packet by the run-time determined and the dwell time in the switch. The receiver can calculate and incorporate the total run-time. This procedure has advantages when there are multiple switches between the master and the slaves. **Auto:** With this setting, "E2E" is used initially. As soon as a "P2P"-packet is recognized, the setting switches to "P2P".

Experiment options (for all devices)	
Parameter	**Description**
	hybridE2E: This is an improved form of "E2E" and reduces the network load since it uses Unicast packets instead of multicast packets. However, it is not supported by all commercially available devices.
Network transport	Depending on the infrastructure used, select UDPv4, UDPv6 or L2/IEEE802.3.
TAI-UTC offset	By default PTP uses atomic time (Temps Atomique International, TAI) as the time base. In the Announce packets, along with the server's quality characteristics, the offset between atomic time and UTC is also transmitted. In order that the device can distribute the correct offset, the current value must be entered here.
UDPv6 scope	Sets the lifespan of the PTP-packets if UDPv6 is set as the "Network transport".
Transport specific	Only needed if you use both imc devices and other devices in a shared PTP-network and the other devices here use a value which is not 0x00. Make the same setting here.
Step threshold	Sets the threshold value as of which a time deviation can no longer be compensated for by adjusting the device clock. This value is only taken into account once the time may no longer jump in the device clocks, which means after the start time of the first measurement after the initialization is set. If the value is 0, any deviation however large is compensated, which can take very long when deviations are large. If the value is not equal to 0 but smaller than the deviation from the PTP-Master, the frequency of the clock rate signals in the device, and thus also of all signal samples, is synchronized to the frequency of the PTP-Master. By contrast, the time deviation is NOT compensated for, but instead kept constant. In this way, it is possible to avoid the signal sampling either accelerating or slowing down over the time during which the deviation is compensated. This can be helpful for measurements in which the exactness of the sampling frequency is more important than the exactness of the absolute time.
First step threshold	This value works in the same way as "Step threshold", but is only used between the initialization of a measurement and when the start time of the the first measurement is determined after the initialization. Note: With the default settings, every time deviation after a measurement is initialized is eliminated within a few seconds. For very small deviations, this is accomplished by brief temporary modification of the clock frequency; for larger deviations, the device clock is directly adjusted. However, if a measurement is started before this procedure was completed, then regardless of how large the deviation is, it is compensated by modifying the clock frequencies in the device.
PTP destination MAC address	Only change this value if you use a different MAC-address in your infrastructure.
P2P destination MAC address	Only change this value if you use a different MAC-address in your infrastructure.

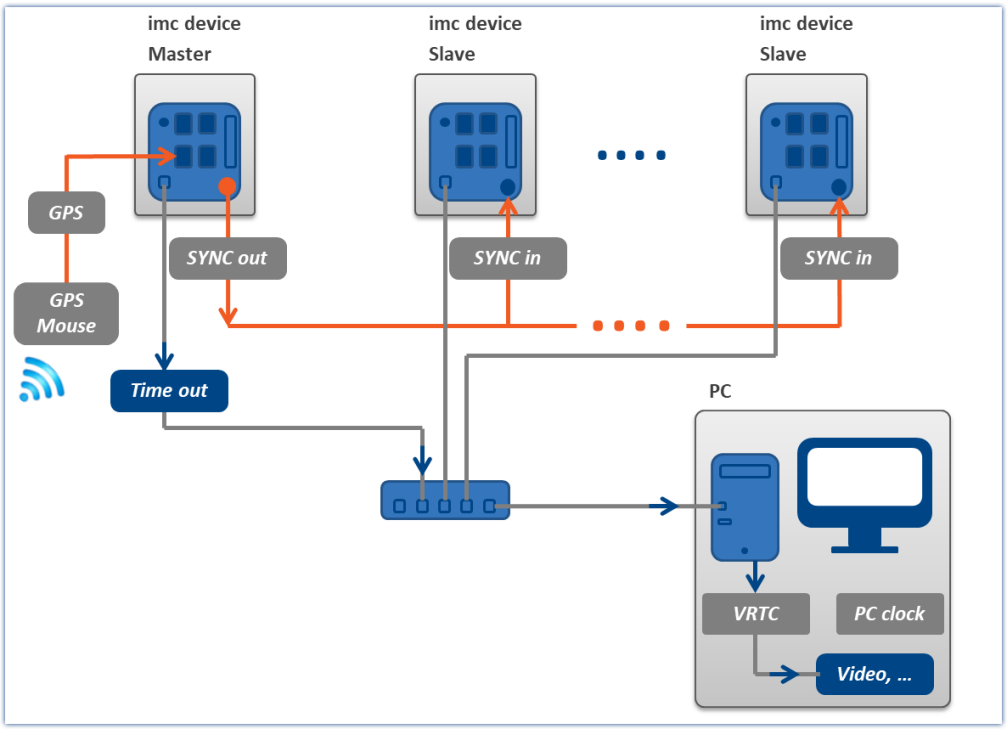
9.2.3.3.6 VRTC - Virtual clock on the PC

For the purposes of precise time output on the PC and synchronization of device results with PC results, a **virtual clock** (VRTC) is used.

The subscribers on the PC always use the virtual clock's time. Such a subscriber can be, for example, the clock on a Panel page or a Video-signal.

Outline: How does the virtual device clock work?

During the running measurement, the virtual clock synchronizes itself to the Master imc device's clock. This is needed if functions are used on the PC side which require an exact time track (e.g. Video, report channels and third-party devices such as Fos4X).



Example of synchronizing the VRTC via the Master device

On heavily burdened networks, good synchronization (of the PC's virtual device clock with the device) is not always possible. When the network load is too high, imc STUDIO posts a corresponding error message when an attempt is made to start the measurement.

It may help to increase the minimum accuracy.

Minimum accuracy:

The option defines the maximum time in milliseconds that the PC's virtual clock may deviate from the device clock. This concerns mainly non-imc devices, e.g. video. In case of video a value of 10 ms provides synchronization of measurement data to video images at 100 frames per second. Too low a value prevents measurement start.

The "*Minimum accuracy*" is located among the options under: "*Extra*" ribbon > "*Options*" in the region: "*Setup*" > "*Device options*" > "*Virtual device clock* 135".

FAQ

Question: Which clock time is used when preparing the devices?

Answer: This depends on what synchronization is set for the devices:

- Without an external synchronization master, the device clocks are **set to the VRTC's time** upon preparing (all device clocks, or in a synchronization, those of the master devices).
- If you have an external synchronization master, then when performing preparation, synchronize the device clocks with the synchronization master. The virtual clock is set **according to the master device's clock**.

Subsequently, the VRTC aligns itself to the device clock.

To find more tips and notes on variations, see the section "*Details on workings of VRTC*". In particular for synchronizing NTP- and PTP, or on the topic of the behavior during measurement.

Question: Is there a distinction between the PC clock and the VRTC on the PC?

Answer: Yes. The VRTC runs on the PC and under some circumstances it can assume the PC's clock time (see previous question). However, it has its own high-resolution counter which enables it to provide significantly more precise time readings than the PC time. The largest deviation occurs in the case of external synchronization. In that case, the VRTC adopts the device clock time upon preparation.

Resolution and deviation before and after start of measurement

During the measurement, the VRTC aligns itself to the device clock. The clock rate of the devices has a resolution of 1 ms or better.

The deviations differ among the devices **up until the moment of measurement start**:

- With **devices of group A5** 187 **and higher**, at the moment the measurement starts the VRTC **typically has a 1 ms deviation**.
- With **devices of group A4** 187 **and higher**, at the moment the measurement starts the VRTC **typically has a 15 ms deviation**.

Over the course of multiple seconds after starting measurement, the deviation improves. This should be noted about all VRTC subscribers, e.g. Video, report channels and third-party devices such as Fos4X.

Whenever possible, define a device of group A5 187 or higher as the synchronization master. Especially when you are using multiple different devices within a measurement setup.

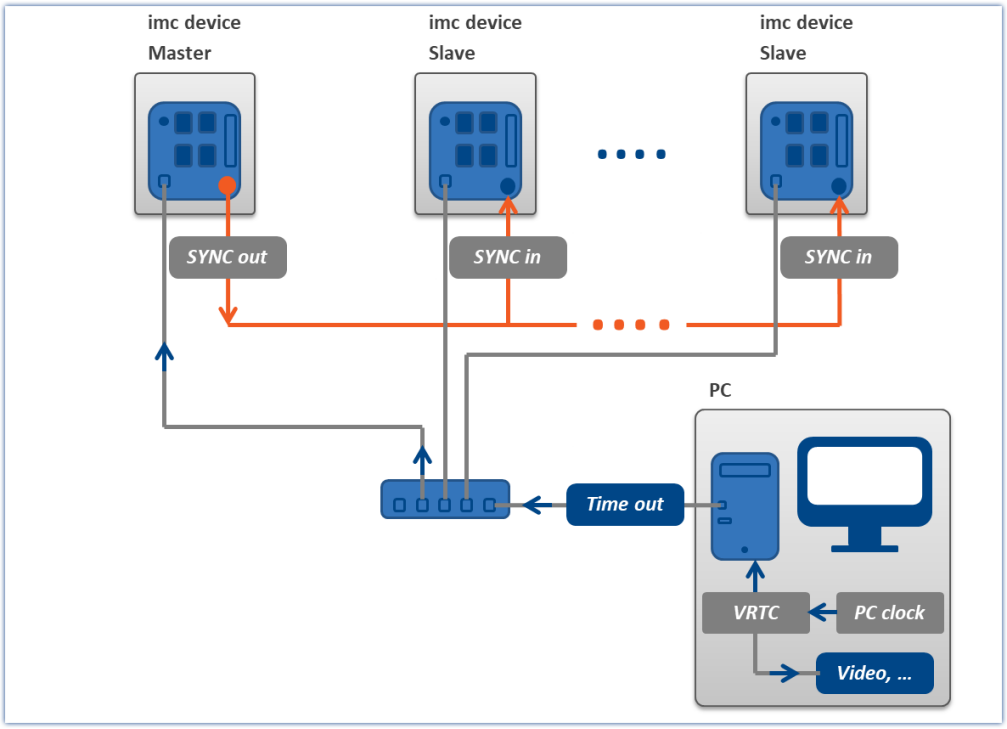
Details on workings of VRTC

When imc STUDIO starts, the VRTC adopts the PC's clock time.

If no measurement is running (or no other components require the VRTC) the system behaves as follows: The time between the PC and VRTC is calibrated every 10 s. Since there are no subscribers, the time adjustment simply jumps abruptly (in either positive or negative direction).

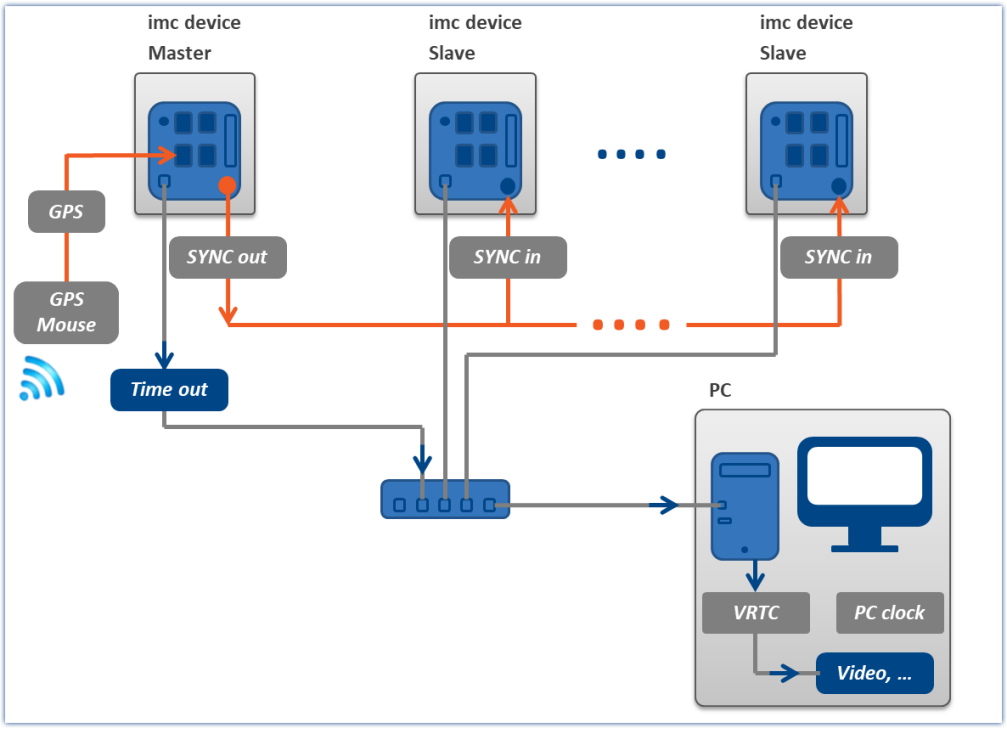
Behavior when preparing (all devices) - Setting the device clock:

- When there is no external synchronization master, then upon preparation, the device clocks are synched to the VRTC clock one single time (all devices' clocks, or in case of synchronization, those of the Master devices).



One-time-only synchronization during preparation, without any external Master

- If you have an external synchronization master, then when performing preparation, the device clocks synchronize with the synchronization master. The virtual clock is set **according to the master device's clock** a single time (see below the exception regarding NTP- and PTP-synchronization).



One-time-only synchronization during preparation, with an external Master

Behavior during measurement:

- The imc devices synchronize in accordance with their synchronization settings. These operate independently of the VRTC.
- Behavior of the VRTC
  - **No synchronization** of the devices mutually or **synchronization via DCF, IRIG B**:
    The VRTC aligns itself to the Master device's clock (if there is no Master device, then the first one in the list).
    The time alignment always increases monotonically. No abrupt skips are made.
  - **NTP**, **PTP**:
    There is no calibration of the VRTC to the device clock! Please synchronize the PC to the same source.
    The VRTC aligns itself to the PC's time.

Note — The precision level depends on the operating system

The precision of the PC's internal clock rate plays a large role here. Newer operating systems have higher precision. With Win 7, one can assume a precision level of up to 1 µs. With Win 10, up to 100 ns.

Note — Recommended methods

- Whenever possible, use defined Master devices.
- Whenever possible, define a device of group A5 187 or higher as the Master. Especially when you are using multiple different devices within a measurement setup.
- When using NTP oder PTP, always synchronize the PC also with the source.

9.2.3.4 Synchronization varieties

In the following examples, two types of synchronization are presented.

Variety	Description
Master/Slave setup	All devices are synchronized mutually via "SYNC". The clock rate is provided by a Master device (which can be synchronized to an external clock, for example). All devices are connected via the SYNC jack. No terminal resistor is needed.
Separate synchronization of all devices	Individual synchronization of each device. If connection between devices via a SYNC line is not possible, then each device can be synchronized separately to an external clock.

Note Outage of the synchronization source

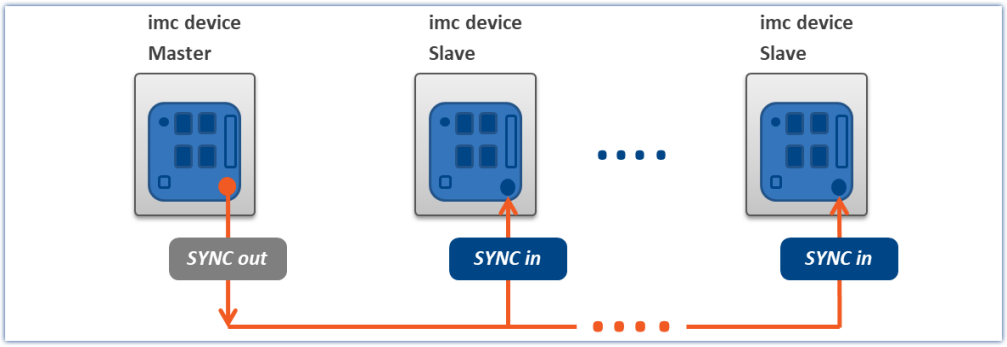
- If there is an **outage of the Master**, all devices continue to work according to their own internal clocks and drift apart in accordance with the precision stated in the technical data of the device.
- If there is an **outage of the external clock** (e.g. GPS or NTP/PTP), the Master continues according to its internal clock. However, the devices remain synchronized to each other.

9.2.3.4.1 IRIG-B or DCF77 - no external timer

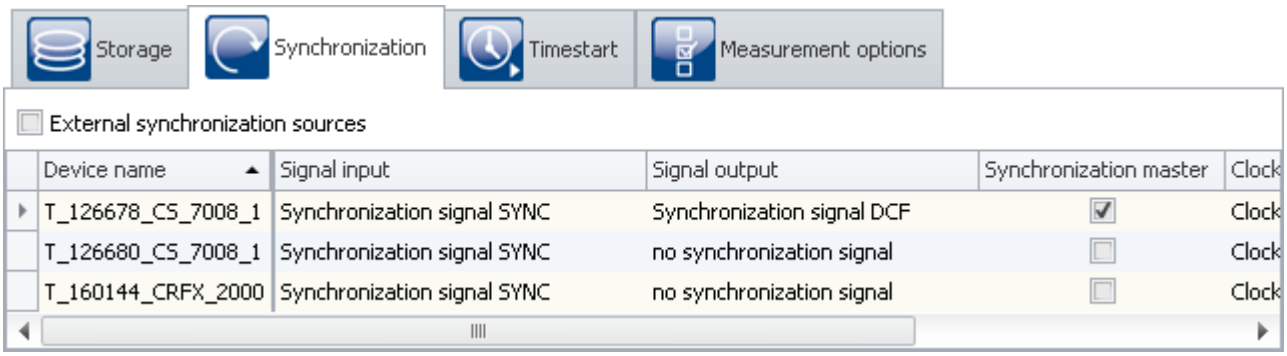
Master/Slave setup

Illustrated by example of: DCF77

The devices are synchronized with each other, but not synchronized to absolute time as provided by an external clock.



Example for synchronization with master and slave devices



Settings in the plug-in Setup on the page "Devices" in the dialog "Synchronization"

Note SYNC for DCF and IRIG-B

With DCF77 and IRIG-B, select "*SYNC*" as the signal input. The hardware automatically recognizes which type is involved.

Note

- The Master also synchronizes itself to its own signal, so that it also reports that it is synchronized. For this reason, its synchronization input is set to "*Synchronization signal SYNC*".
- A slave clock receiving an external clock signal is never set by PC.
- The master clock is set to PC time during preparation (see "*VRTC - Virtual clock on the PC* 307").
  This also happens, for instance, after activating the devices [1].
  The positive result of this is that when working with the devices, it is necessary to wait for the synchronization only once.
- In case the master device is equipped with a GPS receiver that should be ignored, the *Signal input* must be set to "*Synchronization signal SYNC*". Otherwise, the device might synchronize to the GPS-signal.

1: This doesn't apply if the the default synchronization setting is defined in the device properties. In that case, the time is only adjusted if the signal output is subsequently set to "No synchronization signal" on the Synchronizations-page. After the adjustment, run the initialization once and then set the desired synchronization.

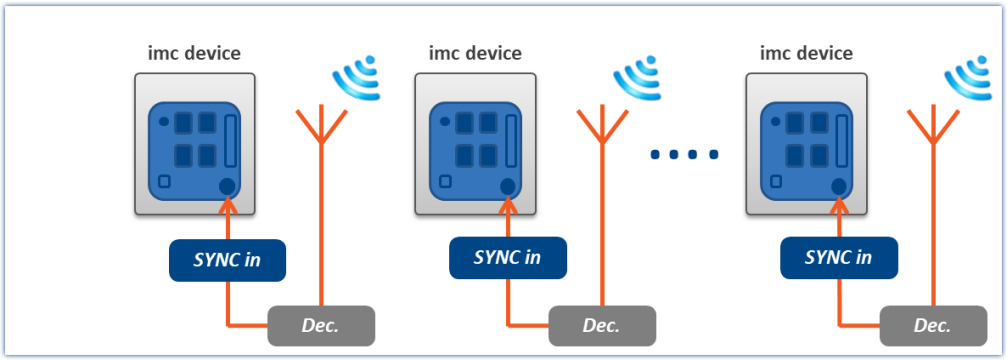
9.2.3.4.2 IRIG-B or DCF77 - external timer

Separate synchronization of all devices

Illustrated by example of: DCF77

Each device is synchronized to absolute DCF77 time and thus all devices are mutually synchronized.

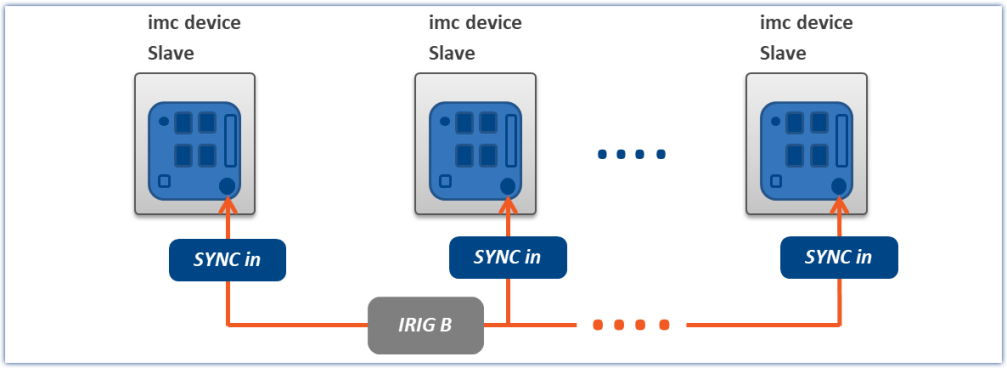
The antenna signal must be converted to by a decoder to TTL level.



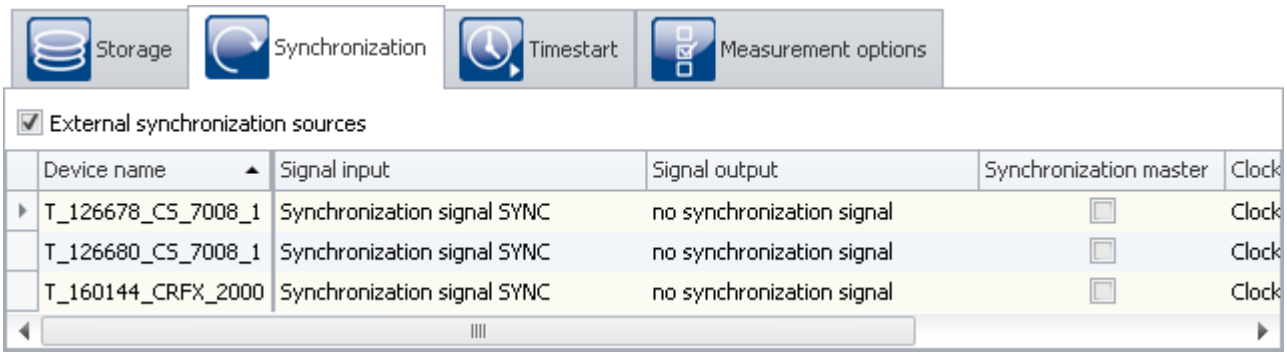
Example of individual synchronization with DCF77

Illustrated by example of: IRIG-B

Each device is synchronized to absolute IRIG-B time and thus all devices are mutually synchronized.



Example of individual synchronization with IRIG-B



Settings in the plug-in Setup on the page "Devices" in the dialog "Synchronization"

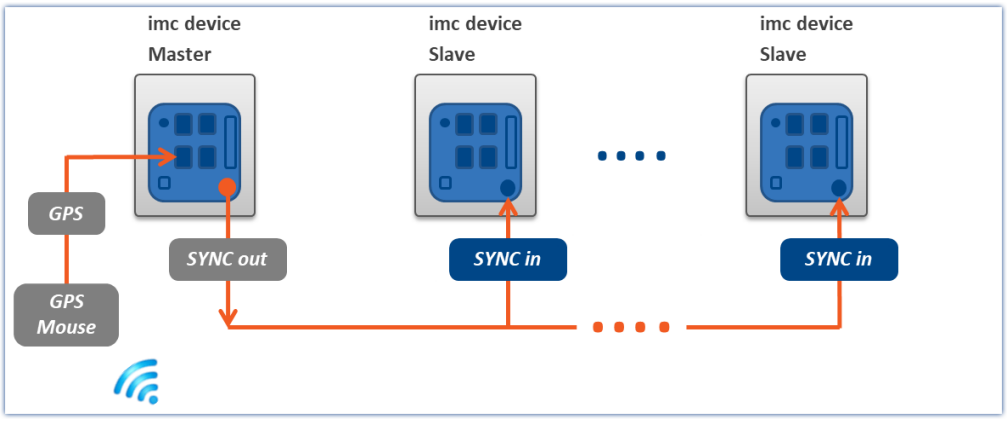
Note SYNC for DCF and IRIG-B

With DCF77 and IRIG-B, select "*SYNC*" as the signal input. The hardware automatically recognizes which type is involved.

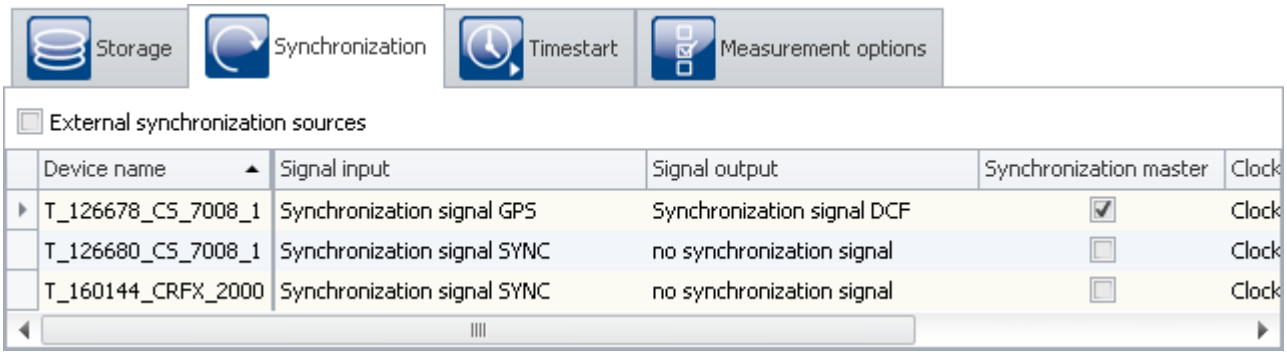
9.2.3.4.3 GPS receiver as timer

Master/Slave setup

The devices are synchronized with each other and synchronized to absolute GPS time.



Master device synchronized by GPS and slave devices by SYNC



Settings in the plug-in Setup on the page "Devices" in the dialog "Synchronization"

Different response for GPS-synchronization + SYNC-output

- **For devices of group A4** 187: If DCF-output is activated for a device and simultaneously GPS is set as the synchronization input, the DCF-output only as long as the device is synchronized to the GPS-source. If the GPS-source fails, the DCF-output is temporarily deactivated!
- **For devices of groups A5-A7** 187 **and devices belonging to the firmware group B** 187**:**
  SYNC-output begins upon the first synchronization with GPS. If there is an outage of the GPS-signal, the SYNC-signal continues to be outputted anyway. All devices connected remain synchronized to the Master. Once the GPS-signal resumes, the resulting phase offset to the Master is corrected, see Phase offset correction 322. In the process, all connected SYNC Slaves are automatically adjusted to have the same phase.

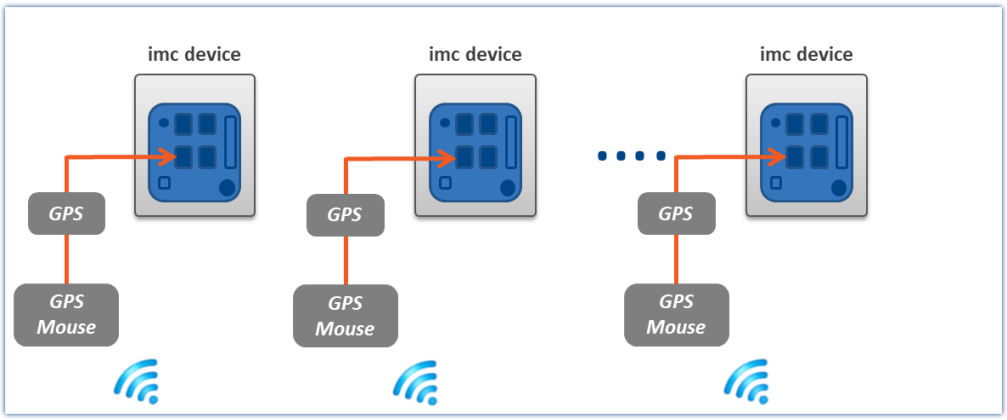
Different response for GPS- and SYNC synchronization + SYNC-output

- **For devices of group A4** 187: DCF-output begins immediately. If a GPS-signal is applied, the system synchronizes to it. If the GPS-signal has a temporary outage or if it is no longer available after preparation of the measurement, the device synchronizes to its own DCF-output. The GPS-signal is then no longer taken into account for the running measurement!
- **For devices of groups A5-A7** 187:
  SYNC-output begins immediately. Any GPS-signal applied always supersedes the device's own SYNC-signal as the synchronization source! If no GPS-signal is applied, the device synchronizes to its own SYNC-output. As soon as a GPS-signal is available, the device synchronizes to the GPS. The SYNC-output is briefly interrupted, so that all SYNC-Slaves adopt the GPS time. Following the first synchronization with GPS, the SYNC-output remains intact even if the GPS-signal fails!
  If the GPS-signal fails again, the synchronizes to its own SYNC-signal. If the external GPS-signal returns, the device synchronizes to the device by correcting the resulting phase offset 322. This automatically also adjusts all connected SYNC-Slaves to the phase.

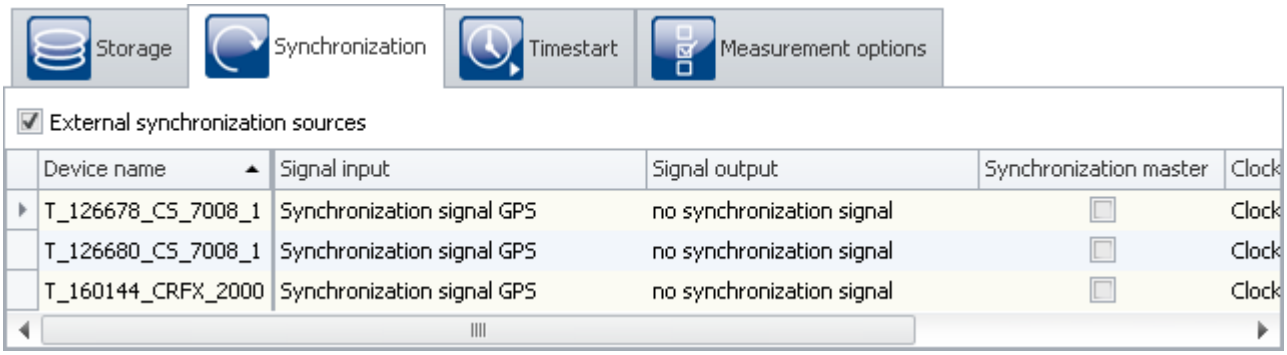
Device name	Signal input	Signal output	Synchronization master
T_126678_CS_7008_1	Synchronization signal SYNC or GPS	Synchronization signal DCF	☑
T_126680_CS_7008_1	Synchronization signal SYNC	no synchronization signal	☐
T_160144_CRFX_2000	Synchronization signal SYNC	no synchronization signal	☐

Separate synchronization of all devices

Each device is synchronized to absolute GPS time and thus all devices are mutually synchronized.



Example of individual synchronization with GPS

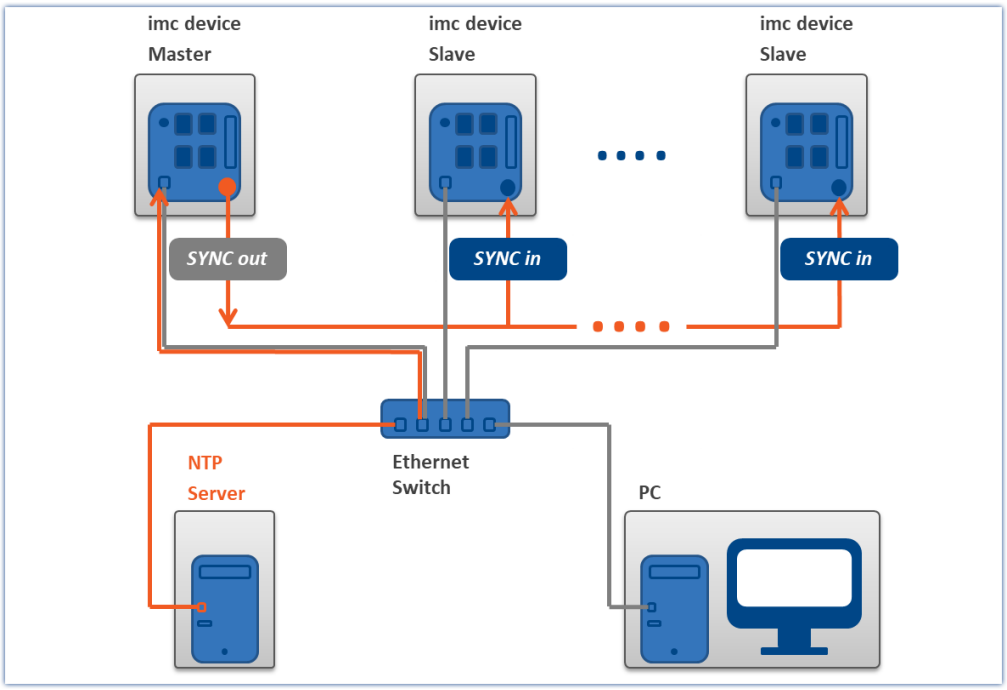


Settings in the plug-in Setup on the page "Devices" in the dialog "Synchronization"

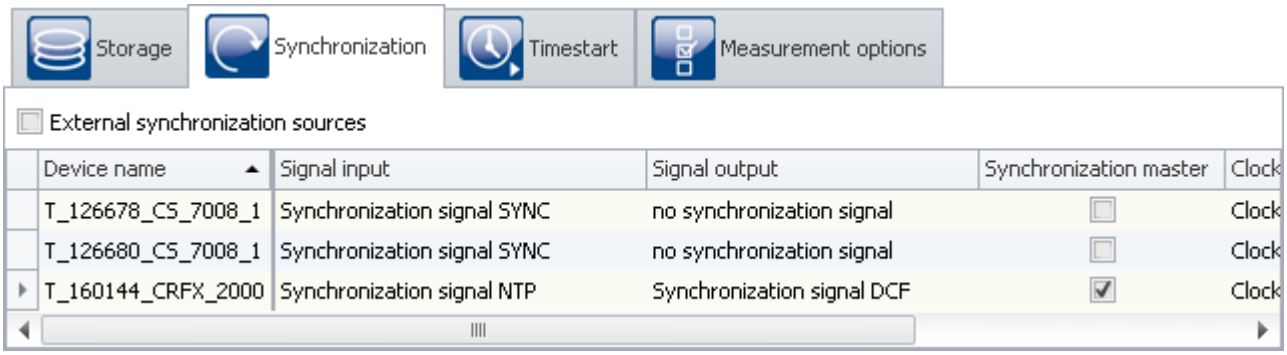
9.2.3.4.4 NTP

Master/Slave setup

The devices are synchronized with each other and synchronized to absolute NTP time.



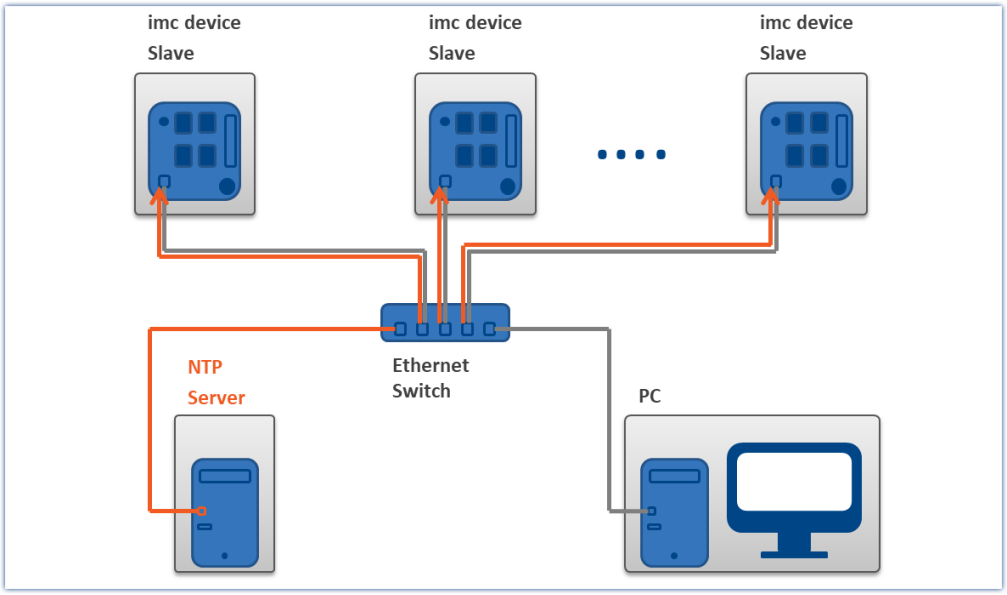
Master device synchronized by NTP and slave devices



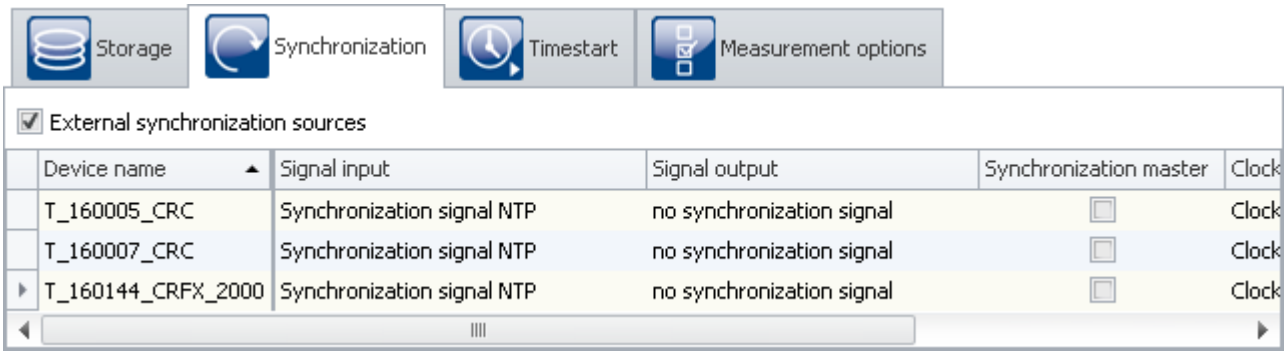
Settings in the plug-in Setup on the page "Devices" in the dialog "Synchronization"

Separate synchronization of all devices

Each device is synchronized to absolute NTP time. All devices are thus mutually synchronized, although to a limited extent (see External clock: NTP 300).



Example of individual synchronization with NTP

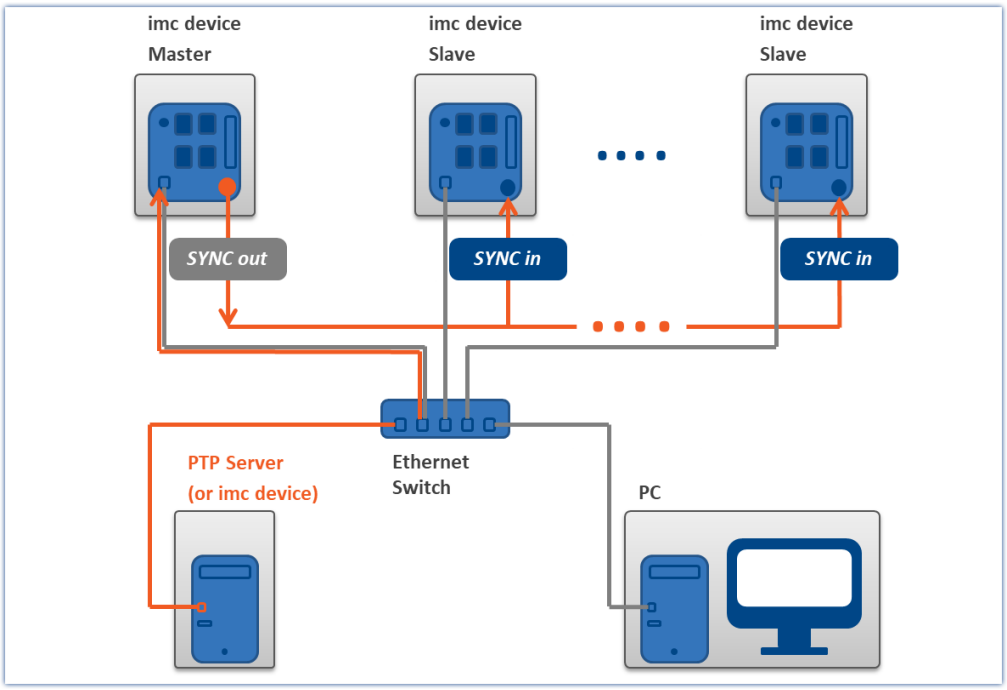


Settings in the plug-in Setup on the page "Devices" in the dialog "Synchronization"

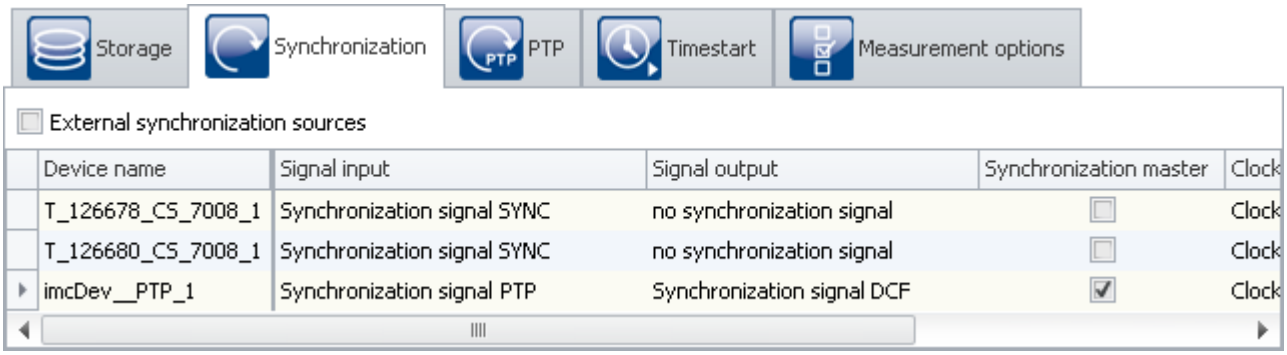
9.2.3.4.5 PTP

Master/Slave setup

The devices are synchronized with each other and synchronized to absolute PTP time.



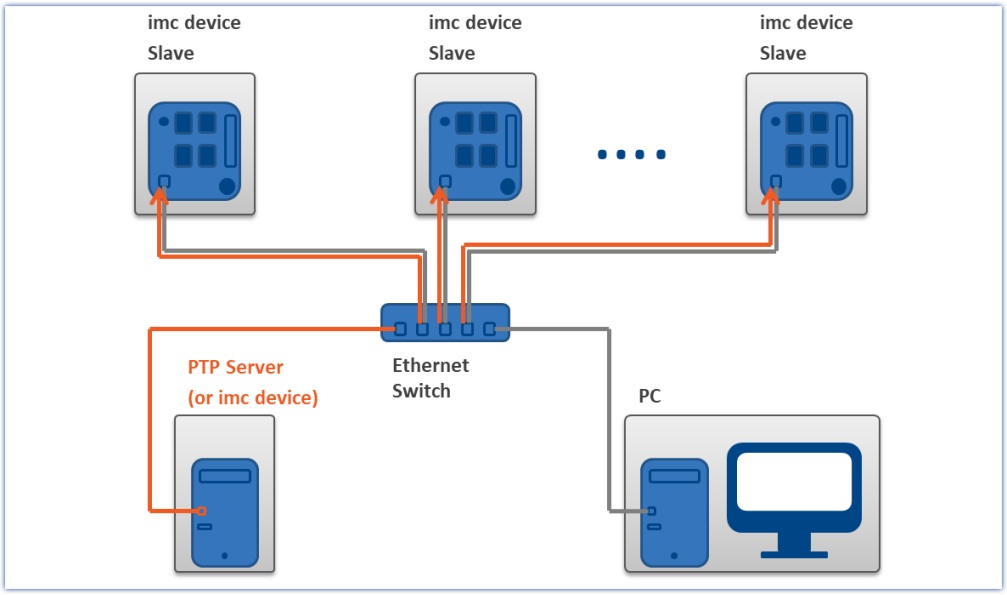
Master device synchronized by PTP and slave devices



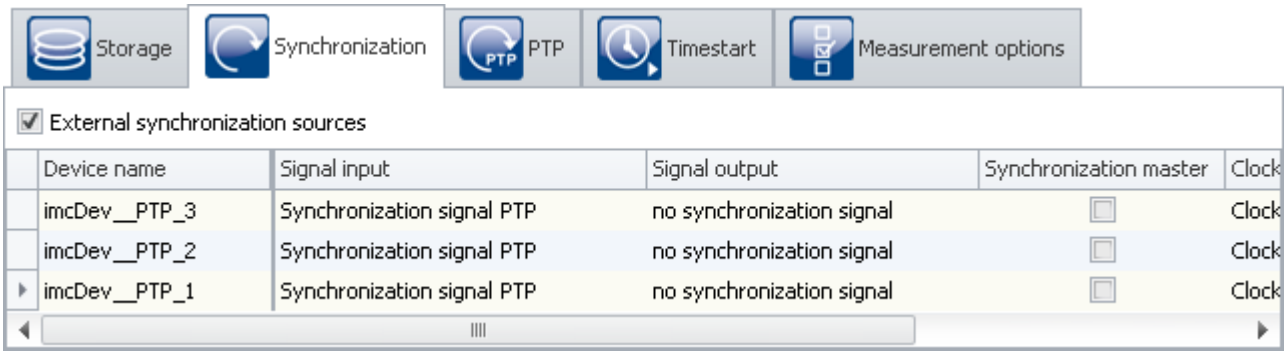
Settings in the Setup on the page "Devices" in the dialog "Synchronization"

Separate synchronization of all devices

Each device is synchronized to absolute PTP time. All devices are thus mutually synchronized, although to a limited extent (see External clock: PTP 301).



Example of individual synchronization with PTP



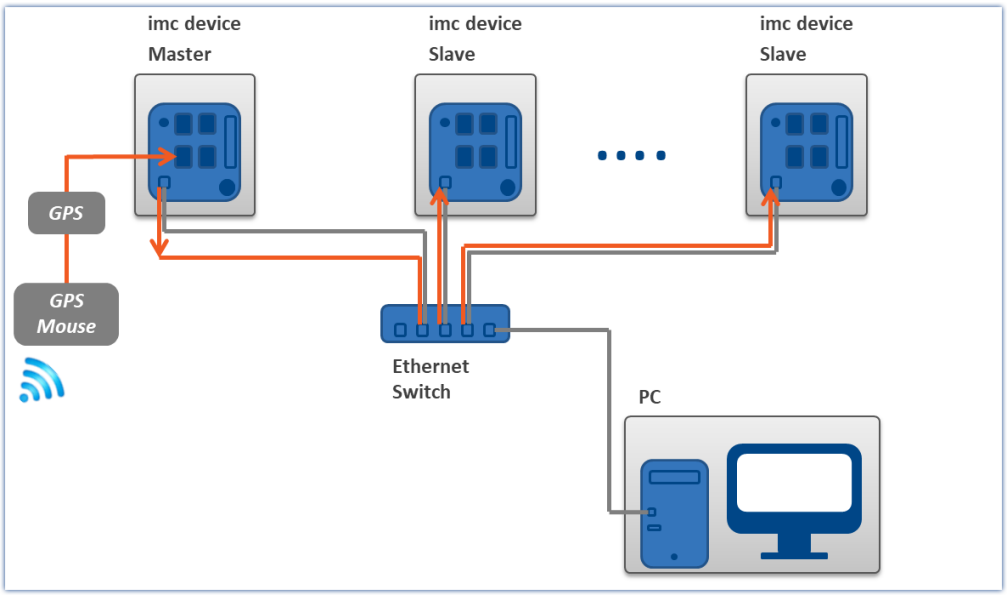
Settings in the Setup on the page "Devices" in the dialog "Synchronization"

9.2.3.4.6 PTP-Master only

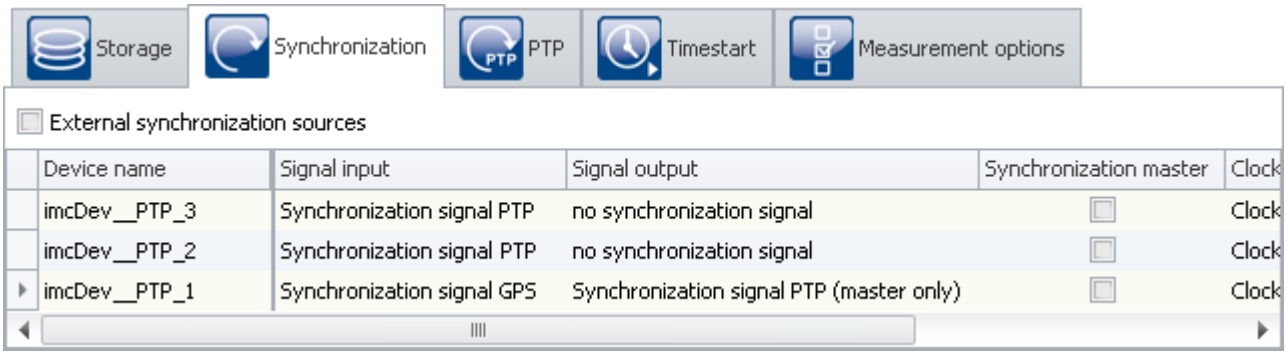
Due to the presets, an imc device only becomes the PTP-Master if no "better" PTP-master is available. See chapter: "*External clocks: PTP* 301".

Any unsynchronized imc device does not have **precise absolute time**. These devices should not become the master in a PTP-network. The PTP-parameters are selected accordingly. Any **imc device is identifies as a "bad" clock**. **Other subscribers which are true to absolute time** should accordingly be identified as a "good" PTP-master.

If the network includes **no PTP-server true to absolute time**, you can **synchronize** one **imc device** to a hardware synchronization signal (e.g. GPS). When you define that **device as the PTP-master**, it will no longer be synchronized by any other PTP-subscriber.



A PTP-master clock synchronizs itself to the GPS-clock. All other PTP-subscribers in the network can be synchronized to this master.



Settings in the Setup on the page "Devices" in the dialog "Synchronization"

Set the **signal input of the Master device** appropriately, e.g. GPS, SYNC, ...

Set the **signal output to "*Synchronization signal PTP (Master-Only)*"**. All other imc devices in the experiment **having "*Synchronization signal PTP*" as the signal input** are **automatically set to "*Slave-only*"**.

The synchronization procedure

The master **initially does not return time information** via PTP. Only after the "Master" **synchronizes to the external signal** does it return time information via the PTP-work.

- If the **only PTP-subscribers are imc devices**, they synchronize themselves with the Master specified.
- If there are **additional PTP-subscribers**, then the imc slave-devices may already be synchronized to the other subscribers. The **devices report that they are synchronized**, although they are **not synchronized with the imc-Master device**.

As soon as the **master device returns the time information**, the other **devices synchronize themselves to it**. How this is done depends on the parameters. The **devices select the subscriber of best quality** (see "*External clock: PTP* 301"). As soon as any device is defined as the "*PTP (Master-Only)*", it is assigned **a slightly higher quality** than the other imc devices. Thus, other imc devices, which are not involved with the experiment but which are subscribers in the network, can also synchronize themselves to the master.

Note Network structure and further notes

- Please always be aware of your PTP-network's setup in order to avoid any unintended effects. If another subscriber has higher quality, then the slave-devices may have an "absolute" time which is different than the master-device's.
- A "*Synchronization master*" must not be selected in this variant.
- Please observe the notes in the chapter Precision 334 and Constraints of the Synchronization 335, especially regarding *phase* and *RAM buffer duration*.

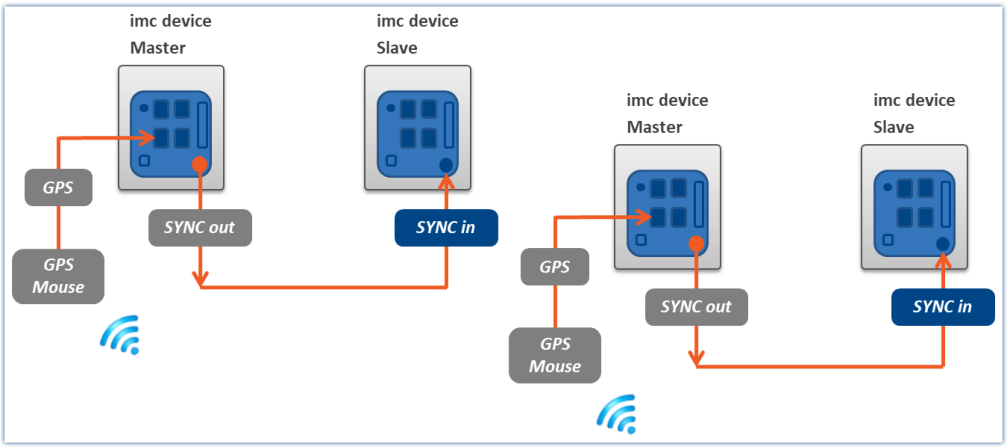
PTP-Master only, and DCF or IRIG-B output

As an additional way to output the time information as the PTP-master, the device can also be defined as the DCF- or IRIG-B-Master.

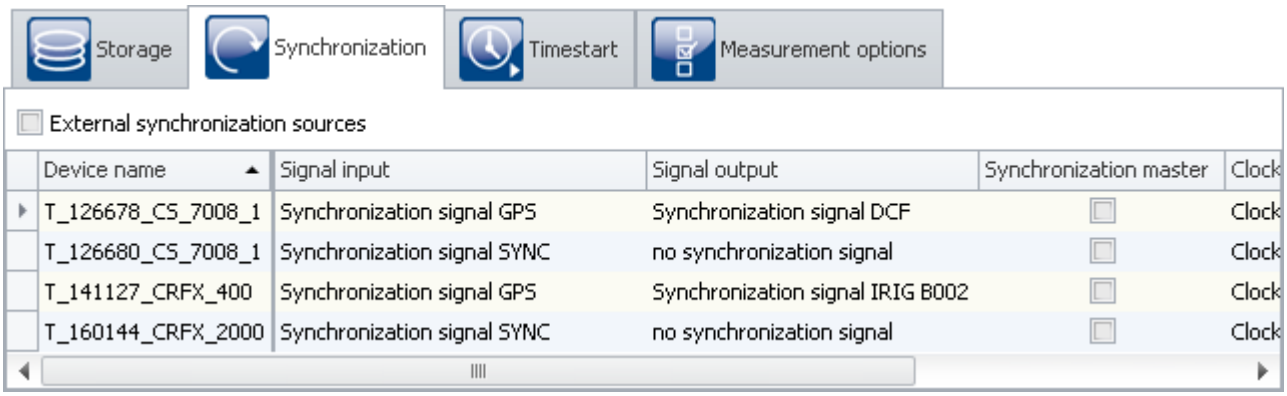
Set the **signal output to "*Synchronization signal PTP (Master-Only) and DCF/IRIG B002*"**.

9.2.3.4.7 Synchronized start with multi-master-devices

If it is not possible to connect multiple devices by a SYNC line, it is possible to divide the devices into multiple Master/Slave groups:



Example for two Master/Slave combinations



Settings in the Setup on the page "Devices" in the dialog "Synchronization"

Note

Since in multi-Master/Slave operation there are multiple masters outputting the SYNC signal, the SYNC outputs of the various masters may not be connected with each other.

9.2.3.5 Phase offset correction

For devices of the group A4 187:

With devices of group A4, following an interruption of the synchronization signal only the frequency error is corrected (i.e. the frequency of the SyncRTC and of the signal sampling is synchronized with the time source). Any phase error is not corrected. The offset remains intact until the next initialization.

For devices of the group A5 and higher 187 and devices belonging to the firmware group B 187:

If the measurement device had been synchronized to a (time) source and if it receives this same source again after an interruption, the phase offset which emerged in the meantime is corrected. Signals captured by different devices are then synchronized just as well after resumption of reception as before the interruption.

If an implausible phase offset is calculated upon re-synchronization, the system assumes that the time source is different than before. In this case, no phase correction is performed during the running measurement, instead only the frequency error is corrected (i.e. the frequency of the SyncRTC and of the signal sampling is synchronized with the time source). A completely new synchronization procedure is performed when the measurement is re-initialized. The same applies in the case of a transition from DCF to GPS etc. .

9.2.3.6 Working with time zones

The abbreviations pertinent to the topic of time zones are presented in the section Definitions 298.

The time zone information includes the shift from the local standard time STD to UTC, as well as the difference between STD and DST and the respective time offsets.

Only when complete time zone information for two zones is supplied can the times be converted to local time.

One special case is a "double" hour in which the time is set back. Here, it is not enough to know in which time zone the time is stated. It is also necessary to know whether the time is time is stated in STD or DST.

If a planned DST-switch is not performed, the time may be correctly indicated due to explicit specification of the DST-state!

Example

- Time zone for Germany, 05.07. 14:00 STD
- This moment is actually during daylight saving time and should be stated as 15:00 DST
- However, the conversion to UTC in both cases correctly returns 13:00 UTC

What are time zones good for?

In order to make it possible to assign a time to measured data, the measurement can only be prepared when a time zone is set for every imc device used.

A device would assume that UTC time applies if it is synchronized using GPS, NTP, etc. This can be resolved by specifying a time zone. Then the device applies the time zone specified, regardless of what synchronizations source is used.

The DST switch is activated in the device properties 223: "*Device time, synchronization*" > "*Time zone*" and "*Automatic daylight saving time switching*".

Ribbon	View
Setup-Configuration > Device Properties	Complete

The setting "*deactivated*" means that the device uses standard time (STD) all year, summer and winter! Even if the PC resets the device clock during daylight saving time, then, for instance 14:00 (DST in the PC) becomes 13:00 in the device (STD). The advantage of this is that there are no time skips in the data which would be reflected in the measured data!

Note

Even a DCF-or IRIG B002-source would only return STD in that case. Otherwise, the time is interpreted incorrectly - there is an error of one hour!

For the setting "*activated*", the PC's or synchronization source's time is interpreted and adopted according to the time zone expected.

In order for a time source's time to be adopted as the device's local time, the time source's time zone as well as its DST-state must be known.

The time supplied to the device via the RS232-interface in the form of NMEA-strings (GPS, IRIG B12X to GPS converter), is interpreted as UTC time.

With the NTP-protocol, the time zone of the clock source is also UTC. With PTP: Atomic time (TAI) + TAI-UTC-offset or also UTC. For this reason, the time statement is complete in such cases.

The time at the SYNC input terminal (DCF/IRIG B002) is expected to match the device's current time zone. The DST-state for these time signals can not always be determined correctly.

Example

- A device is to use DCF-synchronization and is set to the Central European time zone with DST-switching.
- The DCF-source returns the time 03.01.2010 04:58:13 hours
- The device interprets the time as STD, since according to the time zone setting this time is in STD.
- The DCF-source returns the time 14.07.2010 14:34:00 hours
- The device interprets the time as DST, since according to the time zone settings, this time is in DST.

If the source is not in the correct DST-state, there is a resulting error of one hour. For this reason, the device only adopts the source's DST-state upon the first synchronization procedure after activation or upon preparation of the measurement.

The "forbidden hour" resulting from the clock skipping forward is always converted to the corresponding next hour. A time during the "duplicate hour" prior to the clock skipping backward is always interpreted as DST!

After an interruption of the synchronization signal, upon re-synchronization the system determines whether the deviation between the time reference and the device time is approx. 1h (the difference between STD and DST). In that case, the system assumes that the DST-state of the time reference no longer matches the device's DST-state.

This can be the case when the device does not perform the DST-switch because a measurement is already in process or if the device did perform the DST-switch but not the DCF-source.

A device of groups A5 and higher 187 maintains its DST-state, but correctly adopts the source's time.

A device of group A4 187 always adopts the DCF-signal's DST-state upon DCF-synchronization, but upon repeated synchronization it recognizes if the source has not performed the DST-switch. Thus, the time is interpreted correctly as long as the DST-state was correct upon the first synchronization.

Note

With the setting *Switching from/to daylight saving time* activated, the switch between standard and daylight saving time happens automatically at the appropriate time. However, the following limitations apply:

- Once a measurement's start time has been determined (Start button pressed), the device clock can no longer be adjusted. The DST-state, as well, must remain intact. For this reason, the DST-switch is only performed upon the preparation of the next measurement.
- With devices of groups A4 187, the SyncRTC is adjusted directly by the DCF-synchronization. Thus it is not possible to change the DST-state as long as there is DCF-synchronization. If the DCF-signal is not valid at the time of the switch, or if it fails afterwards, the swich is performed. Upon re-synchronization to DCF, the device again adopts the DCF-signal's DST-state. If the DCF-signal fails again, there is no new switch of the DST-state!
- A measurement always uses the device's current DST-state which applied at when the measurement's start time was determined, and retains it no matter whether the SyncRTC was adjusted. Only upon the next preparation of a measurement is it possible to change the DST-state.
- Diskstart, Autostart, Suspend/Resume (imc BUSDAQ) and automatic timed start always lead to preparation of a measurement and can cause the DST-state to change.

9.2.3.6.1 Possible scenarios

Example 1

A device of group A4 187 is to use DCF-synchronization and is set to a time zone with DST-switching.
At the time of the switch from STD to DST, DCF-signal is interrupted.
The device switches to DST.
A measurement is started. The measurement uses DST.
The DCF-signal is applied again and the device is re-synchronized.
If the DCF-signal continues to return STD time, the SyncRTC adopts the time as STD. However, the running measurement retains DST.
If the DCF-signal returns DST because the DCF-source also has made the DST-switch, the SyncRTC adopts the time as DST.

Example 2

A device of group A4 187 is to use DCF-synchronization and is set to a time zone with DST-switching.
At the time of switching from STD to DST, a measurement is already running (in STD).
The connected DCF-source performs the DST-switch.
The SyncRTC adopts the DST from the DCF-signal. However, the running measurement retains STD.

Example 3

A device of group A4 187 is to use DCF-synchronization and is set to a time zone with DST-switching.
The connected DCF-source DOES NOT perform DST-switching.
The device retains its DST-state.

Example 4

A device of groups A5 to A7 187 is to use DCF-synchronization and is set to a time zone with DST-switching.
The device switches to DST at the appropriate moment, no matter whether the DCF-source switches or not.
The DCF-signal is interrupted
The DCF-signal is applied again and the device re-synchronizes.
The device adopts the correct time in any case and the SyncRTC continues to run in DST time no matter whether the DCF-source is still running in STD or in DST.

Example 5

A device of groups A5 to A7 187 is to use DCF-synchronization and is set to a time zone with DST-switching.
At the time of the switch from STD to DST, a measurement is already running (in STD).
The connected DCF-source performs the DST-switch.
The SyncRTC and the running measurement retain STD.

9.2.3.6.2 Summary

If the external source returns incorrect time, for instance because

- a DST-switch was omitted, or
- if the DST-switch happens even though it is not activated for the device, or
- if the time zone does not match that of the device, or if UTC is not supplied,

then the device's time information is faulty!

Once the start time of the first measurement has been determined after the preparation, the SyncRTC may no longer be adjusted. For this reason, after setting the start time there may not be any DST-switch. For instance, a measurement whose start time is expressed in Standard Time is carried out according to Standard Time. As well, any subsequent measurements retain the same DST-state until the next preparation procedure.

For the interface's system time, the DST-switch will still be performed. The times recorded by the file system in the folders and files will have corresponding skips!

For a Diskstart, Autostart, Suspend/Resume (imc BUSDAQ) and automatic timed start, a preparation is performed! In these cases, the DST-state can change again! If the SyncRTC was unable to adopt a synchronization source's time because a measurement was running, then if needed the synchronization is interrupted during preparation and a new synchronization process is initiated! This always happens whenever necessary for exact adoption of the time.

Devices of group A4 187:

Synchronization source: **External DCF-source (not the device's own DCF-output)**

- *Expected time zone:* Local time according to the device's settings
- The SyncRTC is set directly by the hardware and cannot be influenced. Following synchronization, the SyncRTC always has the source's exact time.
- The system assumes that the external synchronization source returns the correct local time at all times! Upon first synchronization, the DCF-source's DST-state must be correct, otherwise there is an error of one hour!!!
- As long as the first measurement's start time has not yet been determined, the system monitors whether the next DST-boundary has been passed. As long as the DCF-signal is received there is no automatic DST-switch of the SyncRTC. If the DCF-signal fails, the DST-switch is performed. Upon re-synchronization following a DCF-signal outage, the system tries to determine whether the DCF-source also experienced the DST-switch. The DST-state of the DCF-time is adopted correctly.

Synchronization source: **Device's own DCF-output (DCF-Master and DCF-Slave)**

- *Expected time zone:* Local time according to the device's settings
- As long as the first measurement's start time has not yet been determined, the system monitors whether the next DST-boundary has been passed. Then the DCF-output is deactivated (so that it is possible to write to the SyncRTC), then the SyncRTC adjusted by the DST-difference and the DCF-output re-activated. If other devices synchronize to the outputted DCF-signal, they are assigned the new time.

Synchronization source: **GPS / IRIG-B 12X converter**

- *Expected time zone:* UTC
- The system assumes that the source returns UTC time. The time is converted to the device's time zone. Before determining the first start time, a DST-switch is performed. For this purpose, the DCF-output is temporarily deactivated, if needed. If other devices are to be synchronized to the outputted DCF-signal, they are assigned the new time.

Devices of groupsA 5 and higher 187:

Synchronization source: **External DCF-source (not the device's own DCF-output) / IRIG-B 002**

- *Expected time zone:* Local time according to the device's settings
- The system assumes that the external synchronization source returns the correct local time at all times! Upon first synchronization, the SYNC-source's DST-state must be correct, otherwise there is an error of one hour!!!
- Prior to determining the first start time, a DST-switch is taken into account, independent of the SYNC-signal applied.
- Upon re-synchronization following an outage of the SYNC-signal, the system attempts to determine whether the source has performed the DST-transition. The time is adopted correctly. The device retains its DST-state.

Synchronization source: **Device's own DCF-output (DCF-Master and DCF-Slave)**

- *Expected time zone:* Local time according to the device's settings
- Prior to determining the first start time, a DST-switch is taken into account. For this purpose, the DCF-output is temporarily deactivated. If other devices synchronize to the outputted DCF-signal, they are assigned the new time.

Synchronization source: **GPS / IRIG-B 12X converter / NTP / PTP (with UTC time base)**

- *Expected time zone:* UTC
- The system assumes that the source returns UTC time. The time is converted to the device's time zone. Before determining the first start time, a DST-switch is performed. For this purpose, the SYNC-output is temporarily deactivated, if needed. If other devices are to be synchronized to the outputted SYNC-signal, they are assigned the new time.

Synchronization source: **PTP (with atomic clock time base)**

- *Expected time zone:* Atomic time (Temps Atomique International TAI)
- The source is assumed to provide Atomic Time, as well as the offset between Atomic Time and UTC of which it is aware. From this info, UTC can be calculated, making conversion to the device's time zone possible.
- If the device itself becomes the PTP-Master, it uses either the TAI-UTC-offset passed to it by the PC, or the offset which had last been communicated to it by another Master.

9.2.3.7 Application notes

9.2.3.7.1 Display of data in the curve window

If synchronized data are to be displayed correctly in a curve window, the following setting must be made for the X-axis in the curve window: *Date/time absolute*. Otherwise the trigger times of the different signals are aligned. When different triggers are used, which is always the case when multiple devices are involved, the reference to absolute time is lost.

9.2.3.7.2 What happens when?

If possible, use the ability to run multiple devices jointly in an experiment and to connect them via the SYNC connection. Set one device as the Master which synchronized all other devices' clocks. The Master can also be synchronized externally, but this is only necessary for the absolute time. If the only issue is the mutual synchronization of the various devices' data, the Master's internal clock is perfectly adequate as the time base.

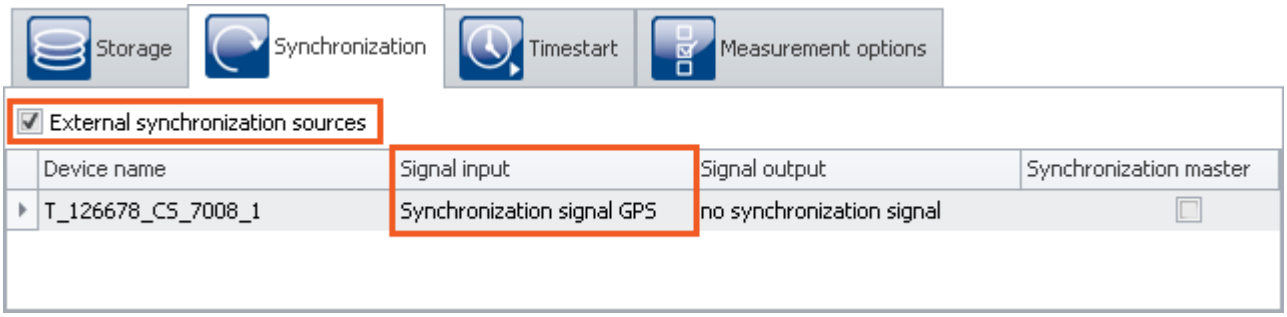
The following examples thus only pertain to external synchronization without connection of a SYNC-line between devices.

Adequate reception by the DCF77/IRIG-B or GPS receiver is not always available. How does the device behave then?

- It does not matter whether DCF77/IRIG-B or GPS is used for the time base.
- Temporary loss of synchronization can cause the device clocks to drift off by up to 1 ppm. If loss of synchronization continues, the precision may drop as low as 50 ppm in consequence of strong temperature change. Once the synchronization signal is again recognized as valid, the device clocks are synchronized again. As soon as the synchronization signal is recognized as valid, the system's behavior depends on the device type 187:
  - With devices belonging to group A4, only the frequency error 322 is corrected.
  - Devices of groups A5 and higher attempt to compensate for the phase error 322.
- Deviation directly after loss of synchronization (1 ppm):
  $t_{delta} = t_{unsync} \cdot 1 \cdot 10^{-6}$; wit $t_{delta} = deviation\ in\ seconds$
  $t_{unsync} = time\ without\ synchronization\ in\ seconds$
  $1 \cdot 10^{-6} = precision\ of\ device\ time\ (1\ ppm)$
- Deviation following long-term loss of synchronization (max. 50 ppm):
  $t_{delta} = t_{unsync} \cdot 50 \cdot 10^{-6}$

9.2.3.7.2.1 Measurement with one device

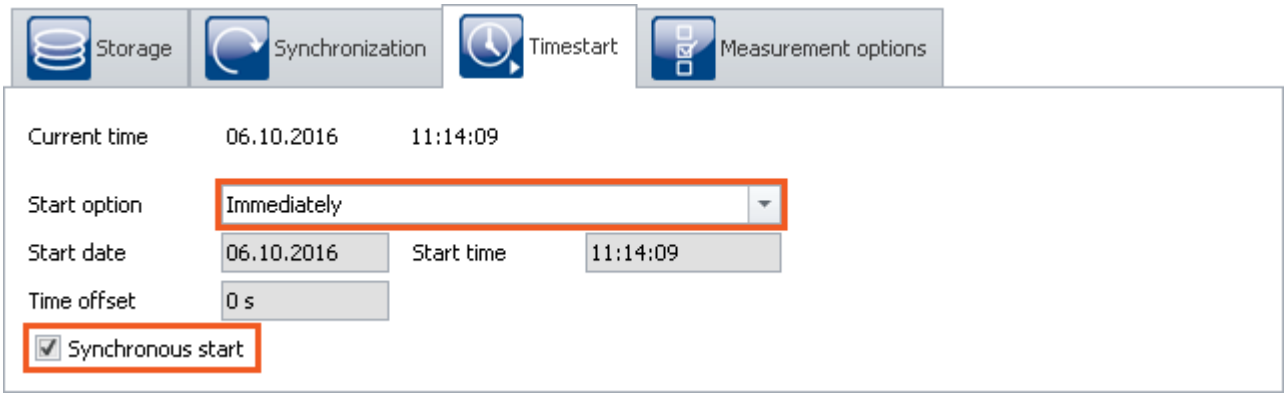
External synchronization via DCF77, IRIG-B, GPS receiver, NTP or PTP.



Settings in the Setup on the page "Devices" in the dialog "Synchronization"

Timed start option: "Immediately"

Synchronous start: activated



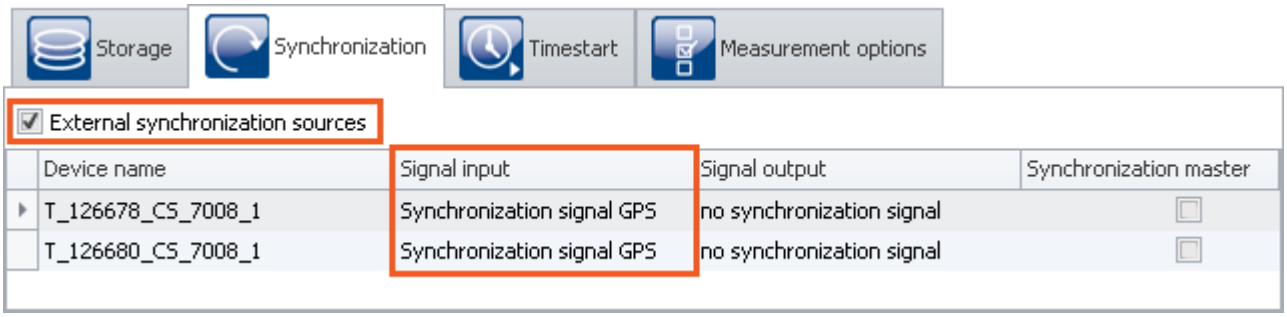
Settings in the Setup on the page "Devices" in the dialog "Timed start"

Scenario	Response	Data storage folder name
Synchronization signal always present.	Device starts and adopts time of the external clock (DCF77, IRIG-B, GPS, NTP or PTP) 298	Time stamp of the devices
Synchronization is always present. The device starts with self-activation upon activation.	Device waits until synchronization and then starts with time of the external clock.	Time stamp of the devices
Synchronization initially present. After measurement start, the clock signal is lost.	Device starts and adopts the time of the external clock. After loss of synchronization, the internal device clock takes over and drifts away*.	Time stamp of the devices
Synchronization is only present after the measurement has been prepared.	Measurement can only start once the device is synchronized.	Time stamp of the device
	Otherwise the start will not be carried out. Instead, an error message appears to notify of lacking synchronization.	or no measurement is set up
Autostart: Synchronization not present upon activation	Device waits for valid synchronization signal. Measurement starts upon synchronization see the manual: Plug-in: Setup - Diskstart 206.	Time stamp of the devices
Synchronization initially present. After measurement start the clock signal is temporarily lost (e.g. 30s) and is later permanently stable.	Device starts and adopts the time of the external clock. After loss of synchronization, the internal device clock takes over and drifts away*. As soon as the synchronization signal is recognized as valid again, the subsequent system behavior depends on the device type 187: With devices belonging to group A4, only the frequency error 322 is corrected. Devices of groups A5 and higher attempt to compensate for the phase error 322.	Time stamp of the devices

* Temporary loss of synchronization can cause the device clocks to drift off by up to 1 ppm. If loss of synchronization continues, the precision may drop as low as 50 ppm in consequence of strong temperature change.

9.2.3.7.2.2 Measurement with multiple devices

Devices are **externally** synchronized **separately**. There is **no** connection via the SYNC terminals.

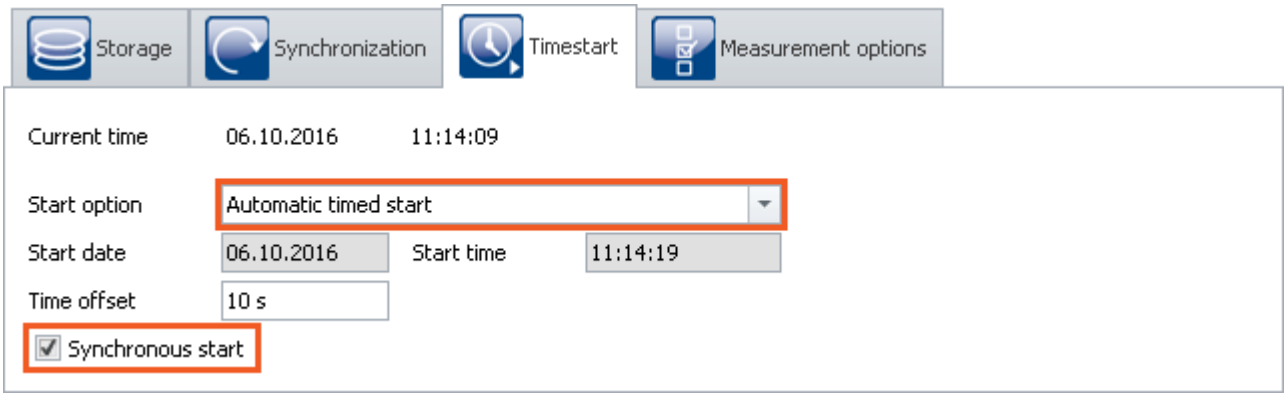


Settings in the Setup on the page "Devices" in the dialog "Timed start"

Synchronous start: activated

1) Timed start option: "Immediately": each unit behaves the same as for "*Measurement with one device* 328" mutually independently.

2) Timed start option: "Immediately" is not selected



Settings in the Setup on the page "Devices" in the dialog "Timed start"

Scenario	Behavior	Data storage folder names
Synchronization signal always present. Device is synchronized	Devices start and adopt the time of the external clock (DCF77, IRIG-B, GPS, NTP or PTP) 298	Time stamp of the devices
Synchronization always present. Devices start with self-activation upon activation.	Devices each wait independently and then start with time of the external clock.	Time stamp of the devices
Synchronization initially present. After measurement started, the clock signal is lost.	Device starts and adopts the time of the external clock. After loss of synchronization, the internal device clock takes over and drifts away*.	Time stamp of the devices
Synchronization is only present after the measurement has been prepared.	Start does not take place. Instead, an error message appears announcing missing synchronization.	No measurement set up
Autostart: Synchronization signal not present upon activation	Devices wait for a valid synchronization signal. Following synchronization, the devices each start independently. This means they don't start simultaneously, but each as soon as possible (corresponding to the Timed start option: "Immediately").	Time stamp of the devices
Synchronization initially present. After measurement start the clock signal is temporarily lost (e.g. 30s) and is later permanently stable.	Device starts and adopts the time of the external clock. After loss of synchronization, the internal device clock takes over and drifts away*. As soon as the synchronization signal is recognized as valid again, the system behavior depends on the device type 187: With devices belonging to group A4, only the frequency error 322 is corrected. Devices of groups A5 and higher attempt to compensate for the phase error 322.	Time stamp of the devices

* Temporary loss of synchronization can cause the device clocks to drift off by up to 1 ppm. If loss of synchronization continues, the precision may drop as low as 50 ppm in consequence of strong temperature change.

9.2.3.7.3 Synchronization control

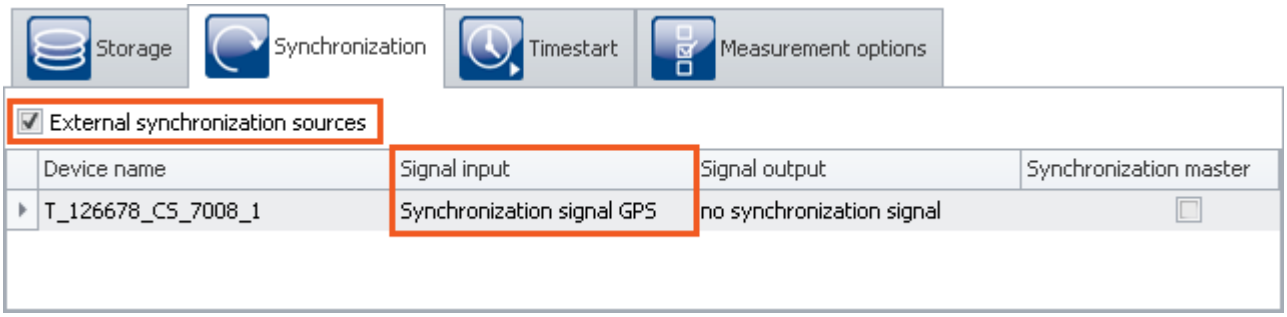
Normally, it's not necessary to stop recording of short measurements if the synchronization is lost. Use the formula presented in the section "What happens when?" 327 to calculate whether the resulting deviation is acceptable for your measurement.

The following functions and constraints must be taken into account for automated synchronization control:

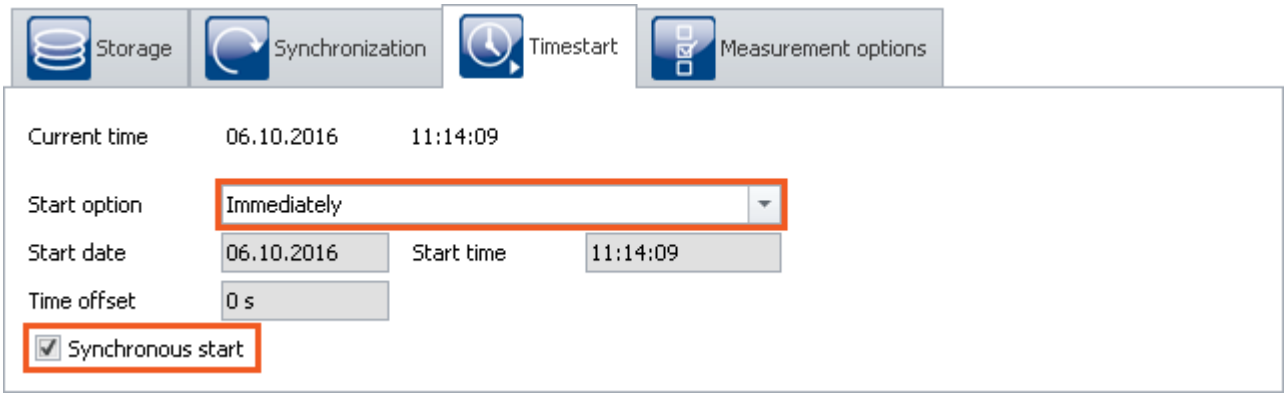
- If a device starts with a synchronized self-activated experiment, the measurement only starts once the synchronization has been established.
- **For devices of the group A4** 187**:** After synchronization has been temporarily lost, the measurement must be re-prepared so that the external pulse clock can be received again.
- The synchronization status is requested using the imc Online FAMOS function `IsSynchronized`.

9.2.3.7.3.1 Measurement controlled by PC

1. Synchronization settings



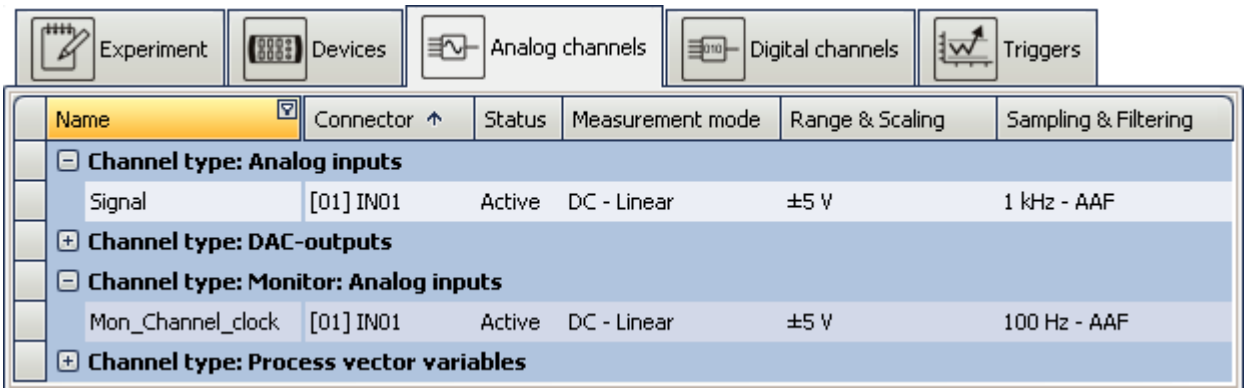
Settings in the Setup on the page "Devices" in the dialog "Synchronization"



Settings in the Setup on the page "Devices" in the dialog "Timed start"

2. Configuration

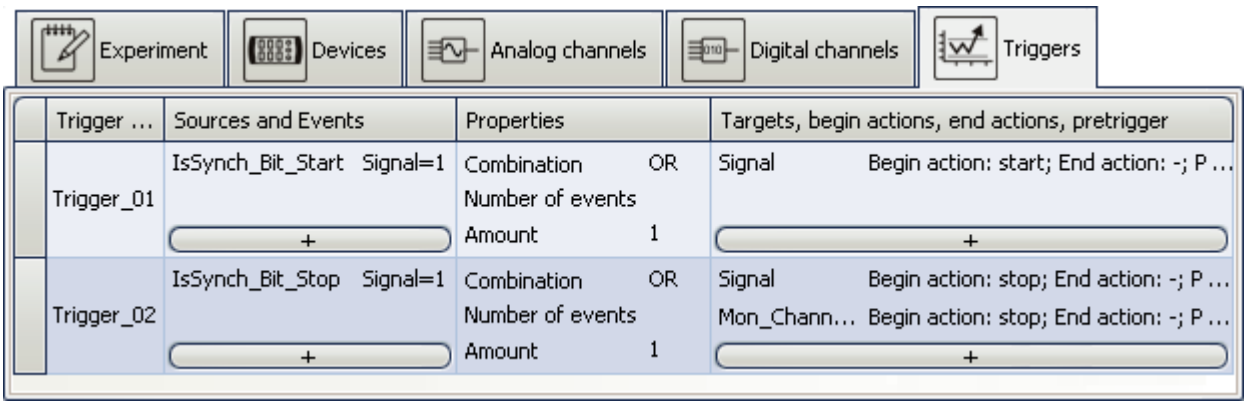
Activate along with the measurement signal to be captured a monitor channel with a sampling interval of *10ms* as the clock rate. Rename two virtual bits: *IsSync_Bit_Start* and *IsSync_Bit_Stop*.



Settings in the plug-in Setup on the page "Analog channels"

3. Trigger machine

The two virtual bits display an event at *Signal=1*. The signal to be measured is started by Bit *IsSync_Bit_Start*. The clock pulse channel is started immediately when the measurement begins. Both channels are closed once the virtual bit *IsSync_Bit_Stop* is set.



Trigger machine
Settings in the plug-in Setup on the page "Triggers"

4. imc Online FAMOS

imc Online FAMOS initially evaluates the synchronization signal. The state is displayed by *LED_01* and activates a digital output channel.

A binary channel is generated which is *1* whenever the synchronization signal is not valid. When the channel is integrated into the system, the last value indicates the sum of the unsynchronized times. When a specified time is exceeded, the stop bit is set.

```
; Create control channel for synchronization with 1Hz sampling rate
; Monitor channel:10ms sampling time
IsSync_Bit_Start= IsSyncronized( mean(Mon_Channel_clock, 100, 100) )

LED_01=IsSync_Bit_Start          ; Indicator with LED

; Create binary channel with 1Hz if not synchronized
_notSynchron= mean(Mon_Channel_clock, 100, 100)*0+1 - IsSync_Bit_Start

BEEP_01= _notSynchron     ; beeper if not synchronized
time_notSynchron= Integral(_notSynchron) ; summarize seconds

; Stop measurement, if syncsignal is not valid for more than 50s
IsSync_Bit_Stop= greater(time_notSynchron, 50)
DOut01_Bit01 = IsSync_Bit_Stop   ; Output to digital channel
```

9.2.3.7.3.2 Measurement controlled automatically (unmanned)

If the measurement is taking place on a system to which you have no access, a mechanism must be employed to re-start the device.

1. Preparation

- Set up the experiment as in the example Measurement controlled by PC 332.
- Save this experiment in the device as an autostart experiment.

2. Activating / deactivating the device via the REMOTE terminal or PowerFail.

Devices having a Remote terminal can be activated / deactivated by means of it. Devices without a Remote terminal are activated / deactivated simply by switching the power supply on / off. In both cases, the running measurement is closed properly before the device goes off.

Example: imc CRONOS-PL

- Activate: Connecting SWITCH and ON switches the device on
- Deactivation: Connecting SWITCH and OFF causes deactivation after 10s´

Example: imc CS-7008

- Put the main switch Position I
- Activate: Switch on the supply voltage
- Deactivate: Switch off the supply voltage; off after 10s

Temporarily switching off the supply ends the measurement if the duration exceeds Acquisition time (T1) after power fail 221. Upon resumption of the power supply, the device is activated, since the main switch is still in the ON-position. The autostart experiment is then automatically loaded and started once synchronization has been established.

3. Control by means of evaluating the digital output

In imc Online FAMOS, a digital output channel has already been prepared.

```
DOut01_Bit01 = IsSync_Bit_Stop   ; Output to digital channel
```

This output must be used by an external control mechanism in order to connect to the Remote terminal or to temporarily switch off the supply voltage.

9.2.3.7.4 Potential difference with synchronized devices

When using multiple devices connected via the **SYNC plug** for synchronization purposes, ensure that all devices are at the same voltage level.

Note

The yellow ring on the SYNC socket indicates that the socket is shielded from voltage differences.

9.2.3.7.5 Precision

- The maximum time difference between data from different devices is 1 sampling interval, max 3ms (as long as nothing else is set, such as for instance with NTP).
- In a synchronized start, the devices are started simultaneously. However, the moment when the first sample value arrives can differ. This is due to the system architecture and depends on various factors (e.g. averaging, sampling rate etc.). However, this doesn't affect the data's time stamp. This effect is visible in the curve window when data from different devices are displayed stacked on top of each other. In absolute time display, the samples are displayed in their correct time coordinates, but the signals of the different channels don't all begin at the same time.

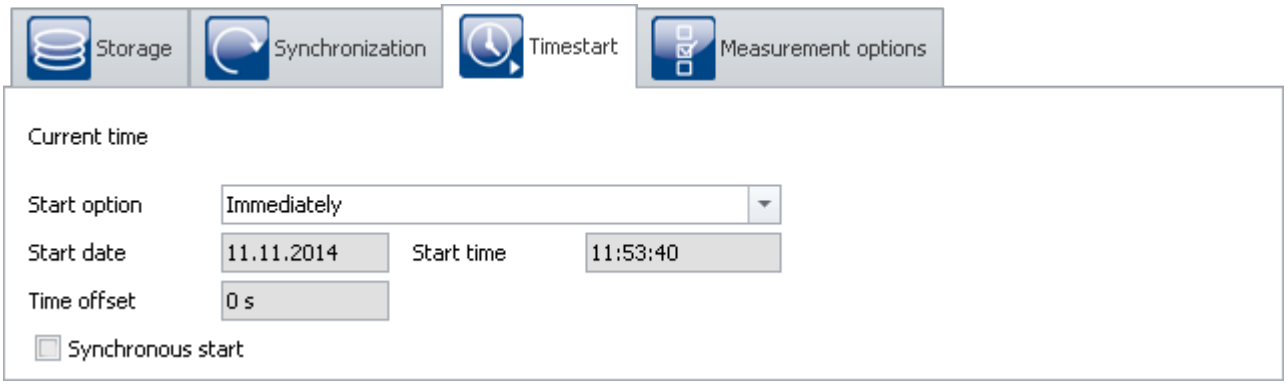
9.2.3.7.6 Constraints of the synchronization

1. **Number of devices:** Max. 20 devices via the SYNC-terminal
2. **Cable lengths:** The maximum length of a synch line (SYNC) is 200m.
3. **Start delay:** If channels are being used with very slow sampling rates, this means that the arming procedure will take a proportionally longer time. This applies not only to the first "start" of a measurement, but to each time the measurement is started or triggered. In particular, be careful to account for the start delay when choosing a starting time for an autostart measurement. See "*Breakdown of the Start/Stop and Triggering Behavior* 761"
4. **Phase:** Please note that the phase of a signal depends on the filter settings. To compare two channels in time, they have to be set by the same filter settings. See also filter settings and synchronization in the hardware manual.
   Furthermore a long RAM-buffer time and different triggers could result in a time shift. Read also the notes in section "RAM buffer duration and synchronity 429".

9.2.4 Timed start

In normal cases, the measurement starts when the Start button is pressed. If necessary, the device is prepared beforehand, which delays the measurement's beginning.

In this dialog, you can also **set other "start options"**.



Timed start

"Start options" are available both for the normal start (in response to the Start button), as well as for the Diskstart 206 (the limitations and conditions are delineated).

Parameter	Description
Start option 336	Select the point in time when a measurement is to be started once the Start-button has been pressed.
Start date / Start time	Calculated possible starting time. Depends on the start option selected.
Time offset 336	Time by which to delay the start in order to ensure that all devices can start simultaneously. Only possible with "*Automatic timed start*". (Not available for the Diskstart!)
Synchronous start	This option ensures that the start only happens once the devices have been synchronized 292. It does not ensure that the devices start simultaneously 337.

9.2.4.1 Start option

Immediately

The measurement starts as soon as possible after the Start button is pushed. Upon the first start after the preparation process, data acquisition starts exactly at the start of the next full second.

The maximum start delay at the start is 1s. The time needed for preparing 195 is not considered here. If it is necessary, this time must be added to the total. This can be preempted by clicking on the Prepare button.

Note

Using the Start option "*immediately*", simultaneous starting of all devices 337 is **not possible**.
With synchronization activated, simultaneous starting of the devices is **possible with "*Automatic timed start*"**, for example. With the option "*immediately*", the devices measure synchronously, and are started quickly, but serially.

At defined time

Not available for the Diskstart!

The measurement starts at the time set. "*At defined time*" is a setting only **appropriate to a one-time start**. For each additional start, the clock time must be re-adjusted.

Note

- Each start with the setting "*at defined time*" requires a preparation procedure for the device.
- If you start a measurement **at defined time**, for instance at 5:00 P.M., or at the next turn of the hour, there may be a delay before reading the first measurement value. This delay, caused by special exigencies of the system, depends on the sampling interval selected, and can be up to several seconds in the worst case.

Automatic timed start

Not available for the Diskstart!

Automatic timed start **enables simultaneous starting of all devices** 337 in the experiment. Here, the absolute time of the measurement's start isn't important.

To ensure that all devices are ready, a **time offset** is added to the current time. Devices connected to an RS232 interface or to a modem can have offsets of more than 30s, otherwise, at least 4 s.

Note imc Online FAMOS - OnInitAll

Every time measurement is started with "*Automatic timed start*", the control command `OnInitAll` is run in imc Online FAMOS. Even if the device itself did not need to be prepared.

9.2.4.2 Simultaneous start of all devices

The option "Synchronous Start" ensures that the start only happens once the devices have been synchronized. It does not ensure that the devices start simultaneously.

Simultaneous start of all devices is possible. For this purpose, certain conditions must be met.

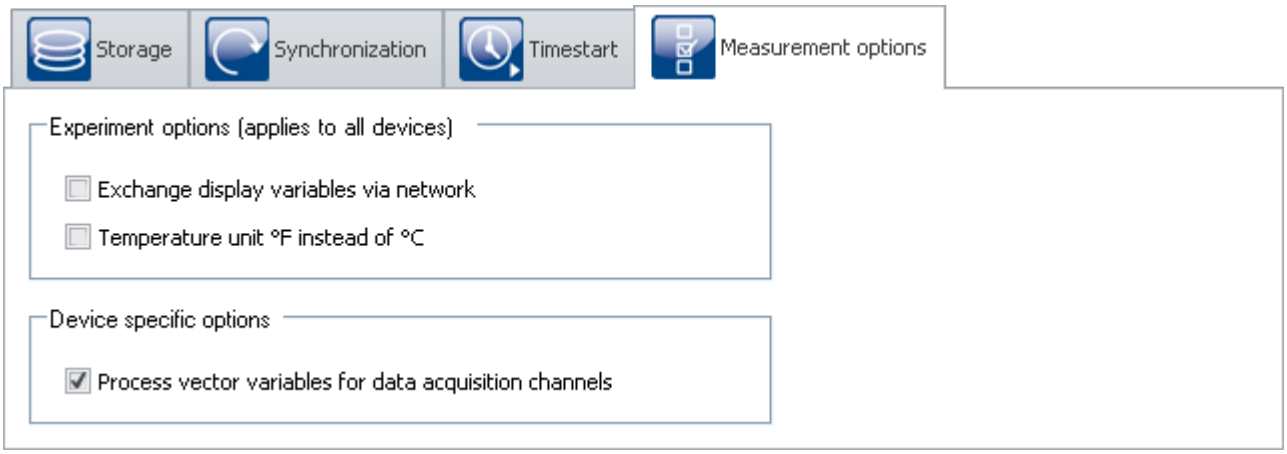
- The option: "***Synchronous start***" must be activated.
- Appropriate "***Start options*** 336" must be selected. E.g. "*Automatic timed start*".
- For all devices, a **clock signal for the signal input** 292 must be selected.

Starting is possible even without this conditions being met, but in that case, all the devices will start serially. Channels of synchronized devices then have the correct absolute time, but the triggering time is a bit different for each device. With a curve window having the X-axis denoted in seconds, the time 0s begins with the release of the trigger. Only in the display mode with the X-axis expressed in absolute time are the channels belonging to different devices all displayed in the correct times.

Reference

Regarding synchronization, observe the notes in the manual:
"*Setup pages - Configuring device*" > "*Configuring Devices*" > "*Synchronization* 292" > "*What happens when?* 327"

9.2.5 Measurement options



Measurement options

Exchange display variables via network

It is possible to exchange the display variable values via the network whenever they change. By this means, data from channels of different devices can be subjected jointly to calculation operations.

Reference See also:

Exchanging values among the devices 766

Temperature unit °F instead of °C

With this option, you set the default unit for all temperature channels.

Note

Make sure that temperature measurements performed via Fieldbusses are set separately. All temperature channels are stated in the unit set here. If, for instance, temperatures in °C are captured via CAN-Bus, but °F is set here, the °C-temperatures from the CAN channels are stated in the unit °F without conversion.

Process vector variables for data acquisition channels

For devices belonging to the firmware group A 187 (imc DEVICES)

For data acquisition channels, process vector variables 341 (pv-variables) are created. This option is activated by default.

Note Aggregate sampling rate

Even when the pv-variables are not used, they demand a portion of the device's available power. For this reason, less than the entire aggregate sampling rate 369 of your device is actually usable.

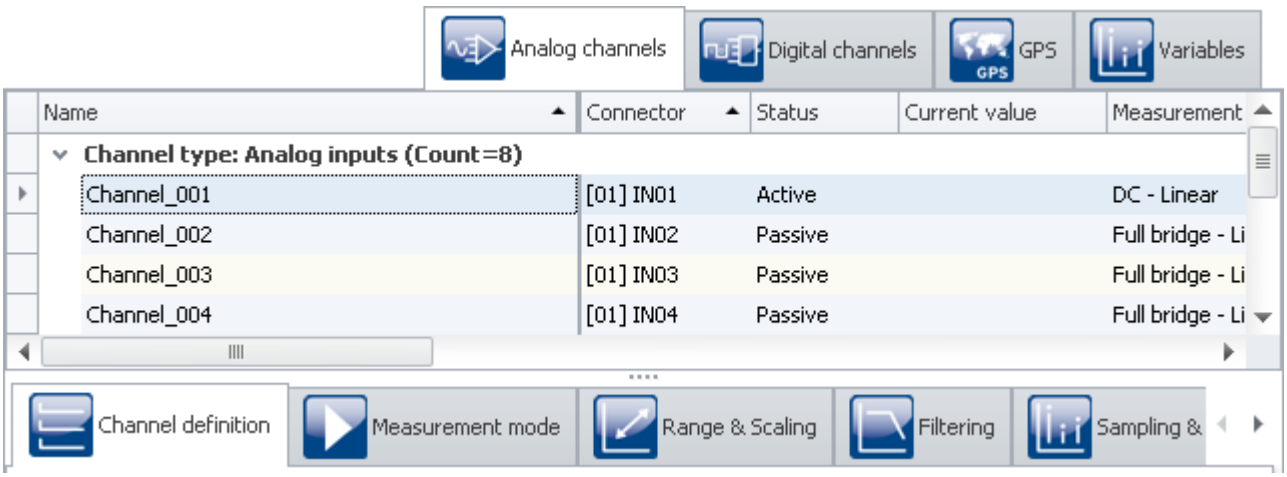
For each device, you can deactivated pv-variables for the data acquisition channels. Remove the checkmark from the selected 232 device. This makes the entire aggregate sampling rate available. Additional pv-variables (e.g. GPS data) are not affected.

For devices belonging to the firmware group B 187 (imc DEVICEcore)

Deactivation is not possible.

9.3 Configuring channels and variables

On the pages **Analog channels**, **Digital channels**, **Variables** and **GPS** you can set all channel-specific parameters.



The four pages are subdivided into two areas: the *Channel table* and the *dialogs*.

In the channel table, all channels are displayed which are present in the selected measurement devices 241. It offers quick access to many settings. In the lower region, you can select various dialogs. These offer clearly-presented access to the properties.

The availability and selection of the dialogs and of the settings depends on the channel type, i.e. on the amplifier hardware installed in the device. Also depending on the channel type, there are different applicable settings options for the parameters.

The following dialogs may be present:

Page	Description
Channel definition 344	**Name** and **Status** of the channel.
Measurement mode 349	Fundamental configuration settings such as **measurement mode**, **measurement range**, **coupling**, **power supply**, ...
Range & Scaling 351	Configuration settings such as **measurement range**, **unit**, **scaling**: Factor/Offset-scaling and 2-point scaling, e.g. for entries from the spec sheet.
Strain gauges 354	Configuration of the **strain gauge parameters**. Only for devices having strain gauge channels.
Encoder 356	Configuration of the **incremental counter channels**. Only for devices having an incremental counter modul.
Filtering 360	Configuration of the **input filters**: AAF, low-pass, high-pass, bandpass.
Sampling & Preprocessing 363	Configuration of the **sampling** and the **measurement duration**, of the **data type** and the **pre-processing**, such as mean value, RMS-value, or maximum/minimum.
Data transfer 370	Configuration of the channel-specific **data saving** on the PC or the device. **Circular buffer data storage** and **event count in the curve window** and the data storage.
Histogram / Rainflow 372	**Refresh rate** and **data saving rate** for histograms and other matrices.
Curve properties 372	Pre-defined settings for how the channel is to be displayed. Defined **channel-color** for the curve window. Specified **range for the display elements** (Widgets).

Note Additional parameters

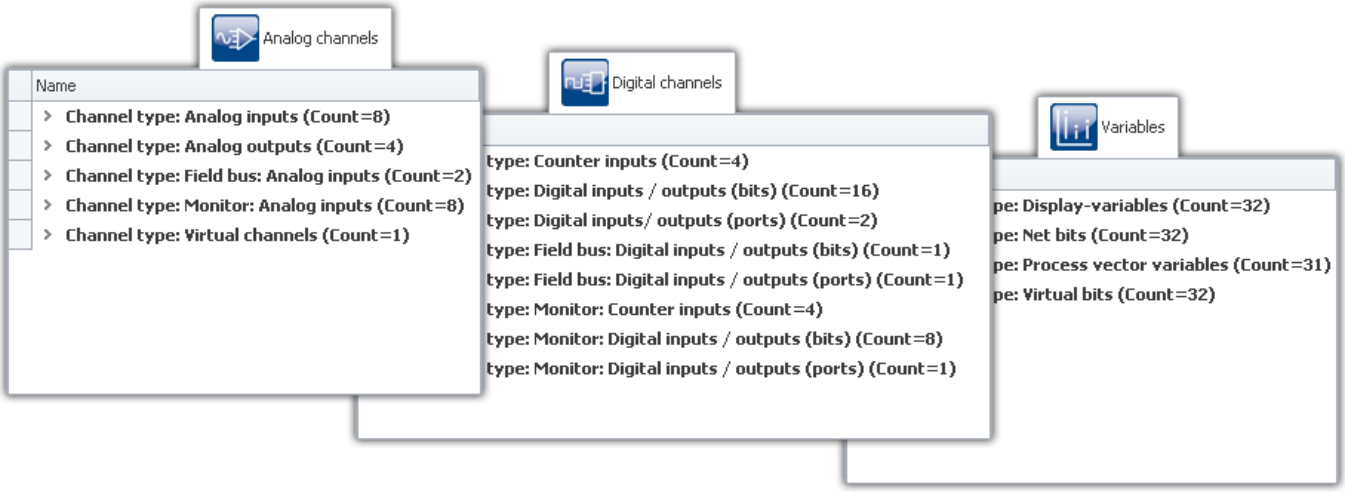
Not all parameters can be found in the dialogs. For special applications, additional settings are also required. In the chapter "*Additional Parameters* 374", you will find description s of the most important additional parameters.

Reference See also additional info on:

- Operation and configuration of the tabular display 231 and the dialogs 238
- Showing and moving columns 248
- Creating and configuring additional columns 251

9.3.1 Channel table

In this area, the ***channels*** belonging to the selected devices are displayed in a structured table, as shown below:



Channel table

In the channel table, all the channels of all devices appear in the list, along with their configurations. The list is arranged in groups, each of which is represented by a branch (or: node) in the list. The group branches can be expanded or collapsed, thus determining whether or not the entries for the channels in the branch are displayed.

To select channels, first open a branch (>). Then select one or more channels. Subsequently, you can either edit table cells 234 directly, or open the dialog 238 for the desired parameter.

All the settings controls in the dialogs take effect on the channels which are selected in the table.

The channels are grouped as follows:

Analog channels	Description
Analog outputs	Analog outputs
Analog inputs	Analog data acquisition channels
Fieldbus: Analog inputs	CAN-interface required (CAN, LIN, etc.)
Virtual channels	Channels created with imc Online FAMOS
Monitor: <xyz> 1135	Digital copies of the input channels which have the same name. With monitor channels it is possible to show slowly-sampled signals recorded before the trigger event.

Digital channels	Description
Digital inputs / outputs (bits and ports)	Depending on your device hardware configuration.
Incremental counter inputs	e.g. ENC4 or INK4 required
Fieldbus: Digital inputs / outputs (bits and ports)	CAN-interface required (CAN, LIN, etc.)
Monitor: <xyz> 1135	Digital copies of the input channels which have the same name. With monitor channels it is possible to show slowly-sampled signals recorded before the trigger event.

GPS	Description
GPS-Channels 1146 (internal "Fieldbus: Analog inputs")	GPS-data as time-domain channels.

Variables	Description
Display variables	The optionally available hand-held display unit interacts with the device via display variables. In effect, Display variables are virtual bits - they can be used for controlling the display of measurement data or for directly influencing a measurement process. These variables can also be used as temporary storage for a number in imc Online FAMOS or for the purpose of display/operation on a imc STUDIO Panel page.
Ethernet bits 1137	Up to 32 ethernet bits are available for Ethernet devices. Events can be assigned to ethernet bits and used as triggers in an Ethernet.
Process vector variables 1141	The "*Process vector*" is a collection of single-value variables (process vector variables or pv-variables). For all measurement channels of the device, these single-value variables are already pre-defined. In parallel to the already existing device-pv-variables, you can create your own pv-variables.
Virtual bits	The virtual bits behave like a 32-bit register, they can be polled and/ or set. This makes it possible to keep a record of states occurring during measurement and, for example, to use the info in the trigger machine.

Maximum channel count per device in firmware group A (imc DEVICES)

Maximum channel count per device		
Active channels within a systems...	512	Active channels of the current configuration: Total number of analog, digital, fieldbus and virtual channels, as well as monitor channels, if any.
...of which active analog channels	198 (1)	Active analog channels of the current configuration (sum of primary channels + monitor channels) (1): 128 with imc CRONOS*flex* (CRFX) and imc CRONOS-XT (CRXT), incl. output channels of type DAC-8 and DIO-Ports of type DI / DO, incl. 18 channels per CRFX/WFT-2 input
Fieldbus channels	1000	Number of defined channels (active and passive); Currently activated channels are limited by the total number of activated channels (512).
Process vector variables	800	Single value variables, each containing the last measured values. A process vector variable is automatically created for each channel.

		without monitor channels				with monitor channels			
Channel type	**determined by**	**limit (aktive+passive)**		**activated**	**total activated**	**limit (aktive+passive)**		**activated**	**total activated**
Analog channels	system-expansion	Channel	240	198	512	Channel	240	198	512
						Monitor	240		
Incremental counter	system-expansion	Channel	16	16		Channel	16	16	
						Monitor	16	16	
DIO/DAC-Ports	system-expansion	Port	16	16		Port	16	16	
						Monitor	16	16	
Fieldbus channels	flexible	Channel	1000	512		Channel	1000	512	
						Monitor			
Virtual channels (OFA)	flexible	-	-	512		-	-	512	

Occupancy for ports (examples):

- one DO module (e.g. DO-16) occupies 1 port
- one DI8-DO8-ENC4-DAC4 module occupies 3 ports
- one DAC module (e.g. DAC-8 or DAC-4) occupies 1 port

Monitor-ports: DI-ports (respectively channels) have monitor-ports, DO/DAC-ports in contrary do not have monitor-ports

Maximum channel count per device in firmware group B (imc DEVICEcore)

Maximum channel count per device		
Active channels within a systems...	1000	Active channels of the current configuration: Total number of analog, fieldbus and virtual channels, as well as monitor channels, if any.
...of which active analog channels	1000	Active analog channels of the current configuration (sum of primary channels + monitor channels)
For fieldbus log channels	any number of channels	Log channels: non-decoded CAN traffic ("dump")

9.3.1.1 imc CRONOSflex and CRONOS-XT modules

The device software lists the imc CRONOS*flex* (CRFX) and imc CRONOS-XT (CRXT) module's amplifier channels automatically in the channel table.

Information on the maximum sampling rate per channel and on the maximum aggregate sampling rate per system, as well as on other possible limitations are described in the technical specs in the manual for the respective device.

At present, there is no **Preprocessing** (e.g. Arith. Mean) possible for CRFX/CRXT amplifier channels. Channels, recorded with the modules, can be processed with imc Online FAMOS functions only (Mean(), Min(), Max(), RMS())

Connection recognition

The numbering of the module happens automatically upon first connection and remains intact. This number is displayed at the module with a 7-segment display. It appears in the channel table in the column **Connector** as slot recognition in parentheses in front of the amplifier's channel number (e.g. **[01]** IN07, **[01]** IN08, **[02]** IN01).

An overview of the modules connected is provided in the dialog "*Module properties* 226". A CRFX/CRXT module is distinguished by the address designation (*xbus...*).

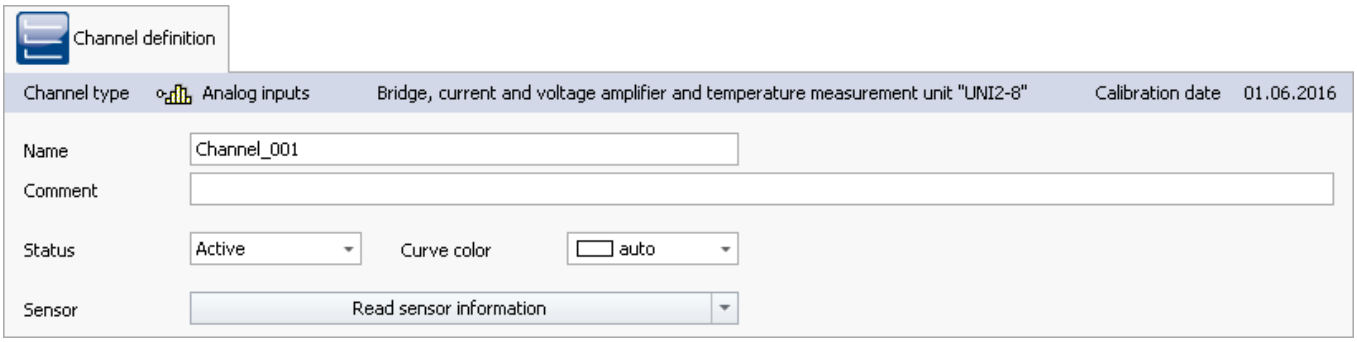
Reference

Observe the notes on **Removing and exchanging modules** and on **Re-assignment of module numbers** in the description of the dialog: "*Module properties* 226".

Note Digital CRFX modules DIO, Pulse Counter and DAC

- 32-channel DI- or DO-modules are designed as "double-modules" which behave logically as two modules and which show corresponding IDs on two 7-Segment displays.
- The maximum count of modules per device is limited. For more information on this subject, see the device's technical specs and the chapter "*Channel table* 342".
- The 7-Segment display on the front of the modules shows the assigned address in HEX-format with a prefixed "d" for "digital module". Digital Flex modules which are installed within a base unit do not have a 7-Segment display and are assigned the address "d0" at factory.
- See also the description in the user's manual imc CRONOS System Family, in the chapter "*DI, DO, DAC, HRENC and SYNTH*".

9.3.2 Channel definition



Parameter	Description		
	long name	*short name*	*column ID*
Name	*Channel name*	*Name*	*eChannelName*
	The channel's Name must be unique, i.e. no two names may have the same name. Maximum number of characters: 65. Three more characters may be added for the sake of any pv-variables. See "*Channel Name* 345".		
Comment	*Channel comment*	*Comment*	*eChannelComment*
	The Comment contains a **text for explanation concerning the channel**. The comment is saved along with the channel's settings in the experiment, and can be displayed in the Panel, for example.		
Status	*Channel status*	*Status*	*eStatus*
	The Status determines whether the particular **channel is activated for the measurement/the experiment**. • Active: Activates the channel • Passive: The channel is not recorded. However, the settings are kept.		
Color	*Color*	*Curve color*	*eCurveColor*
	Pre-defined setting for the **channel-color** in the curve window.		
Sensor	Read sensor information. The sensor information is read out and used to configure the channel. see "*TEDS - Sensors* 729"		

Calibrating the amplifiers

At delivery, the amplifiers are freshly calibrated. Routine calibration ensures reliable measurement. Along with the information on the amplifier type, the calibration date is also displayed.

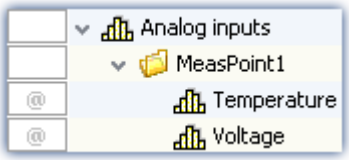
If you have any questions about your measurement amplifier's calibration, please contact our tech support.

9.3.2.1 Channel name

Note Using special characters in the name

If a channel name begins with a number or contains special characters such as "\?+!" or a space, in OFA/IFA the channel name must be set into curly braces.
Example:
Channel name = "123 My channel 100% display"
in OFA/IFA = "{123 My channel 100% display}"

Joining variables to form a group

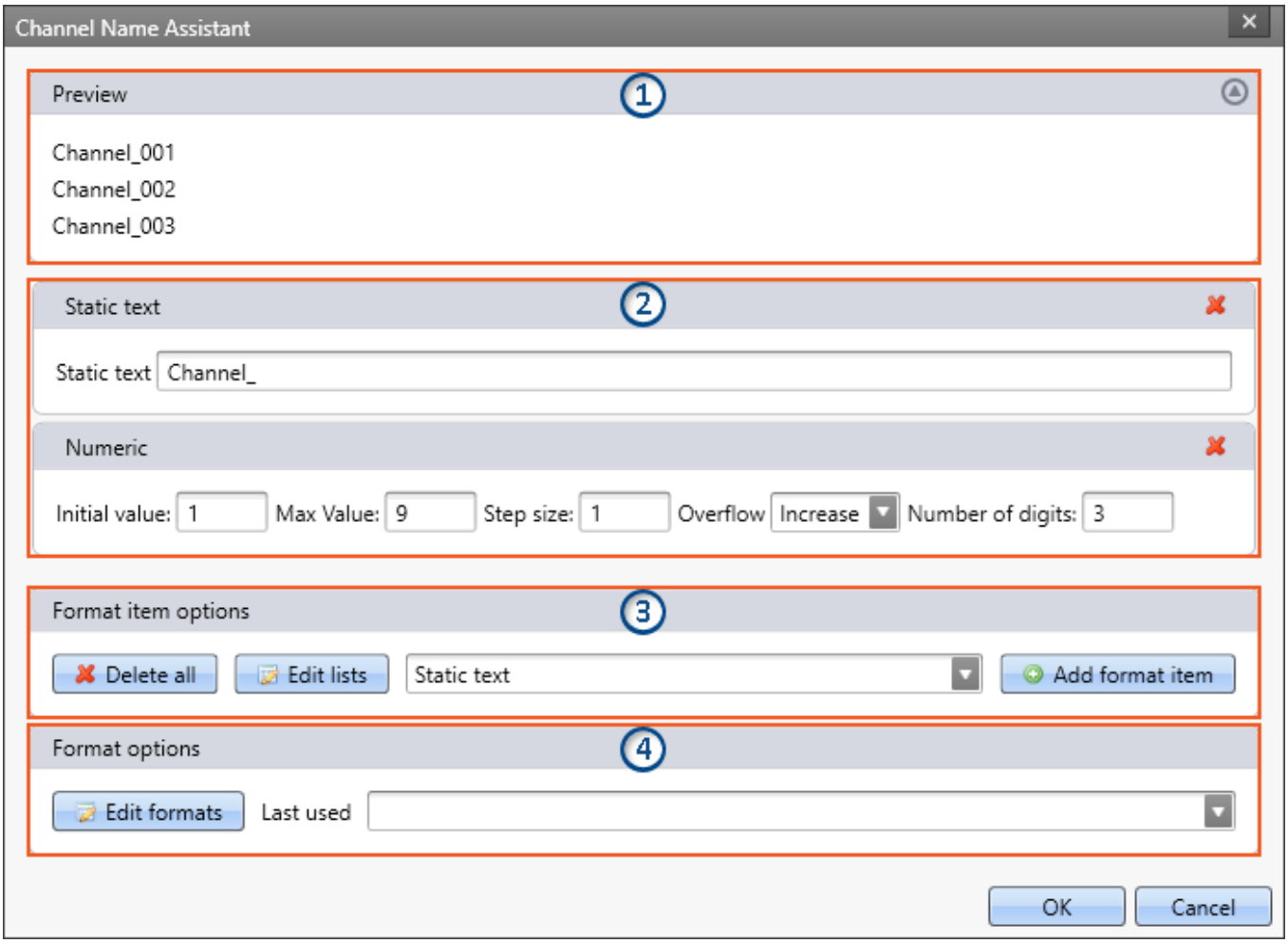


Display in the Data Browser

To join multiple variables together in a group, you can use the character '.' in the variable's name.
E.g. "*MeasPoint1.Temperature*", "*MeasPoint1.Voltage*" results in this display in the Data Browser.

Renaming multiple channels jointly (Channel Name Assistant)

By means of the "*Channel Name Assistant*", you can rename multiple channels simultaneously. For this purpose, there are a variety of naming formats available, such as numbering and text-lists.



Example with the Assistant's standard settings
As a result, the first channel is assigned the name "Channel_001", the second "Channel_002", ...

Area		Description
1	Preview	Here you find the first three results with the format items which are set.
2	Format items set	Here, all format items are defined. Configure them here also. From the top downward, the items are applied to the names. You can use Drag&Drop to modify the order. In the example above, the static text: "*Channel_*" is applied first. Next comes the (three digit) number: "*Channel_001*". Using the red "x" icon, you can delete an item.
3	New format items	**Create** here any **new format items**. Select from the list an appropriate item and add it. It will appear on the bottom. Additionally you can **edit lists** and add to them. Lists are used in the item of the same name in order to use user-defined texts for the name. Use "*Delete all*" to **delete the current configuration**. All items are deleted.
4	Saved formats	Here you can access the **most recent configurations used**. By means of "*Edit formats*", you can assign names to these configurations or delete them.

Configuring format items

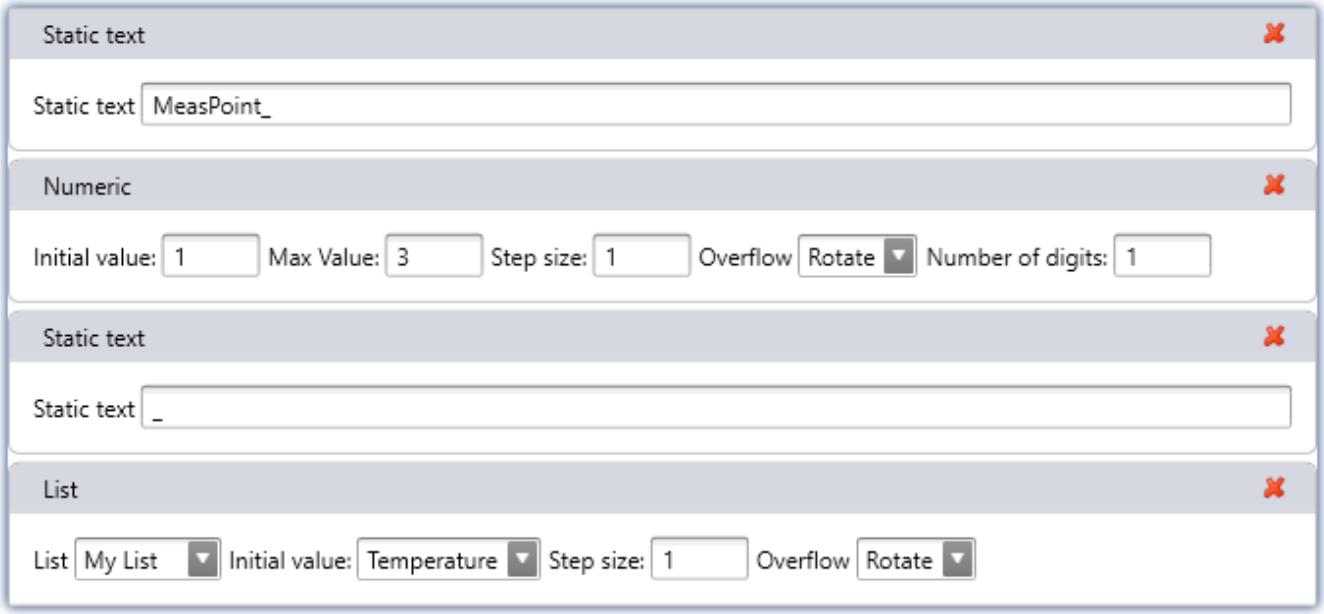
Format item	Description
Static text	The text defined is adopted as it is entered. It can also be used for special characters as separators for other items, e.g. "_".
Numeric	A consecutive number. • **Initial value:** Starting value • **Max value:** Has a function only in conjunction with rotation. See "*Overflow-Rotation*" • **Step size:** By what increments the number is to be increased. E.g. "1": 1,2,3,4,... | "2": 1,3,5,7,... • **Overflow-Rotation:** The numbers extend from the Initial value to the maximum value. After that, the number starts over at the Initial value. • **Overflow-Increase:** Applies the usual counting method. If one decimal place reaches the digit "9", the next higher decimal place is used: "10" • **Number of digits:** Minimum number of decimal places used for the number. E.g. "2": 01, 02, 03, ..., 99, 100, 101.
Alphabetical	The consecutive alphabet. • **Initial value:** Starting letter • **Max value:** Has a function only in conjunction with rotation. See "*Overflow-Rotation*" • **Step size:** By what increments the letter is to be increased. E.g. "1": A,B,C,D,... | "2": A,C,E,G,... • **Overflow-Rotation:** The letters extend from the Initial value to the maximum value. After that, the letter starts over at the Initial value. • **Overflow-Increase:** If one position reaches the letter "Z", the next position is used: "AA"

Format item	Description
List	Uses a defined list for the names. • **List:** Here, select a list to use. If applicable, create a list first ("*Edit lists*"). • **Initial value:** First list entry used. • **Step size:** The interval at which to progress through the list. E.g. "2": first entry, third entry, ... • **Overflow-Rotate:** The list entries extend from the Initial value to the last entry. Subsequently, progress continues with the first entry (not the Initial value!). • **Overflow-Increase:** Once the last entry has been used, the next position is used: "Entry1Entry1", "Entry1Entry2", ...
Column value	Uses a channel parameter as the name. Possible choices: Device name, Channel name, Channel comment, Connector, Module number, Device SN. **Example:** Information regarding the sensor connected is to be used in the name. When the sensor is read, the associated info is recorded in "*Channel comment*". This info can now be additionally used in the channel name.

Example Combination of items

The name is to be composed of a measurement point number. Each measurement point has three connectors: "Temperature", "Voltage" and "Current".

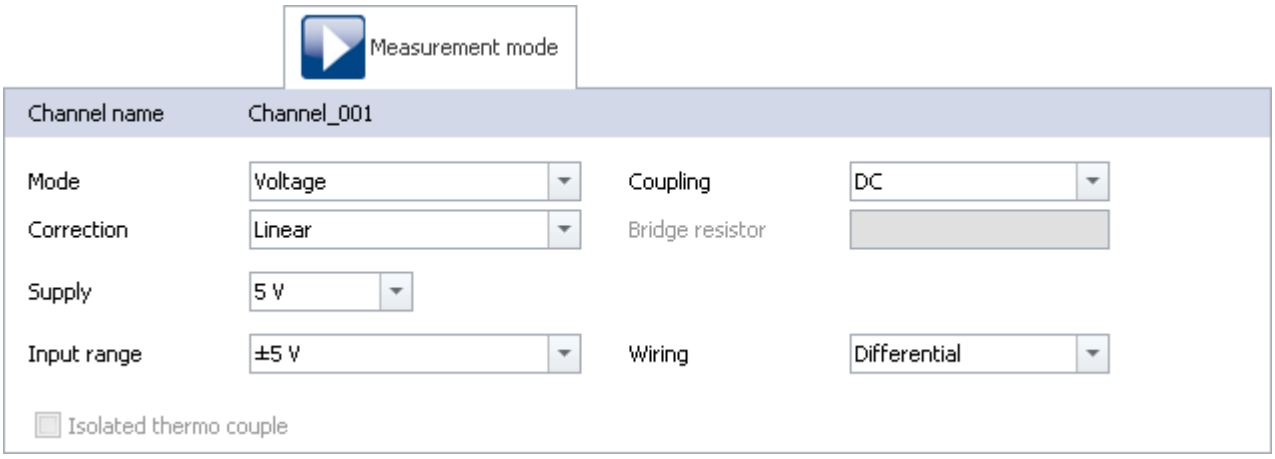
Previously, a list ("My List") had been defined, containing the three connector designations.



This configuration generates the following names of variables:

MeasPoint_1_Temperature
MeasPoint_1_Voltage
MeasPoint_1_Current
MeasPoint_2_Temperature
MeasPoint_2_Voltage
MeasPoint_2_Current
MeasPoint_3_Temperature
MeasPoint_3_Voltage
MeasPoint_3_Current

9.3.3 Measurement mode



Measurement mode

Reference

The settings available depend on the amplifier type. They are described in the pertinent device manual in conjunction with the interconnections.

Parameter	Description		
	long name	*short name*	*column ID*
Mode	*Measurement mode*	*Mode*	*eChannelMode*
	Basic settings for the measurement types: "*Voltage*", "*Current*" or "*Strain gauge*" (Strain gauge: for further settings, a new dialog appears: Strain Gauge 354).		
Coupling	*Coupling*		*eCoupling*
	Depending on the amplifier and on the mode set, DC- or bridge measurement settings are possible. If the Strain Gauge mode is selected, a variety of possible applications is listed, which lead to different input ranges 351.		
Coupling is DC	*Coupling is DC*	*DC*	*eCouplingIsDC*
	For some amplifiers, it is necessary to switch the coupling for the whole amplifier board to DC-coupling. In this case, this must be set independently of the **Coupling**.		
Bridge resistor	*Bridge resistor*	*Resistor*	*eBridgeResistor*
	The bridge resistor must be stated for all bridge types, although it is actually only absolutely necessary for the quarter bridge. For full bridges, the bridge resistor is used along with the **calibration resistor** to calculate the **indicated strain** (see: "*Cable compensation without sense line* 384"). Also, with the bridge resistor, it is possible to determine the cable resistance even if no Sense line is connected.		

Parameter	Description		
	long name	*short name*	*column ID*
Correction	*Correction*		*eCorrection*

Use of a correction curve (e.g. temperature measurement; depending on the amplifier). Requires the amplifier to operate in DC-voltage mode.

- *Linear*: No linearization of measured values
- *PT100*: Linearization according to a PT-100 thermistor characteristic curve
- *Type R, S, B, J, T, E, K, L, N, C* [1]:
  Linearization according to a thermocouple characteristic curve
- *Sensor characteristic curve*: This is displayed when a sensor 729 is used

[1] For amplifiers produced after November 2012.

The condition for **temperature measurement** is that the amplifier is working on mode: voltage (DC).

Supply	*Supply*		*eSupply*

The power supply voltage can be used either for the bridge supply or the sensor supply. Each amplifier has its own power settings globally applicable to all its channels. By changing the settings for one channel, the settings of all channels belonging to the same amplifier are changed accordingly.

Some amplifiers provide the ability to switch the power supply on or off. See parameter: Bridge supply (On/Off) 375

Input range	*Input range*		*eRange*

Lists the input ranges available.

If you use scaling for the channel or if you have carried out an adjustment, the resulting physical input range with its unit is displayed here. For temperature measurements the input ranges depend on the thermocouple type or RTD.

Input ranges rounded

Use of taring or the entering of a value for the "*Sensitivity*" easily cause the generation of input ranges having very many decimal places. The display of the input ranges is rounded off to 5 significant digits.
Parameter sets and channel parameters contain the original value.

See also: Setting the expected signal value range in addition to the measurement range 373

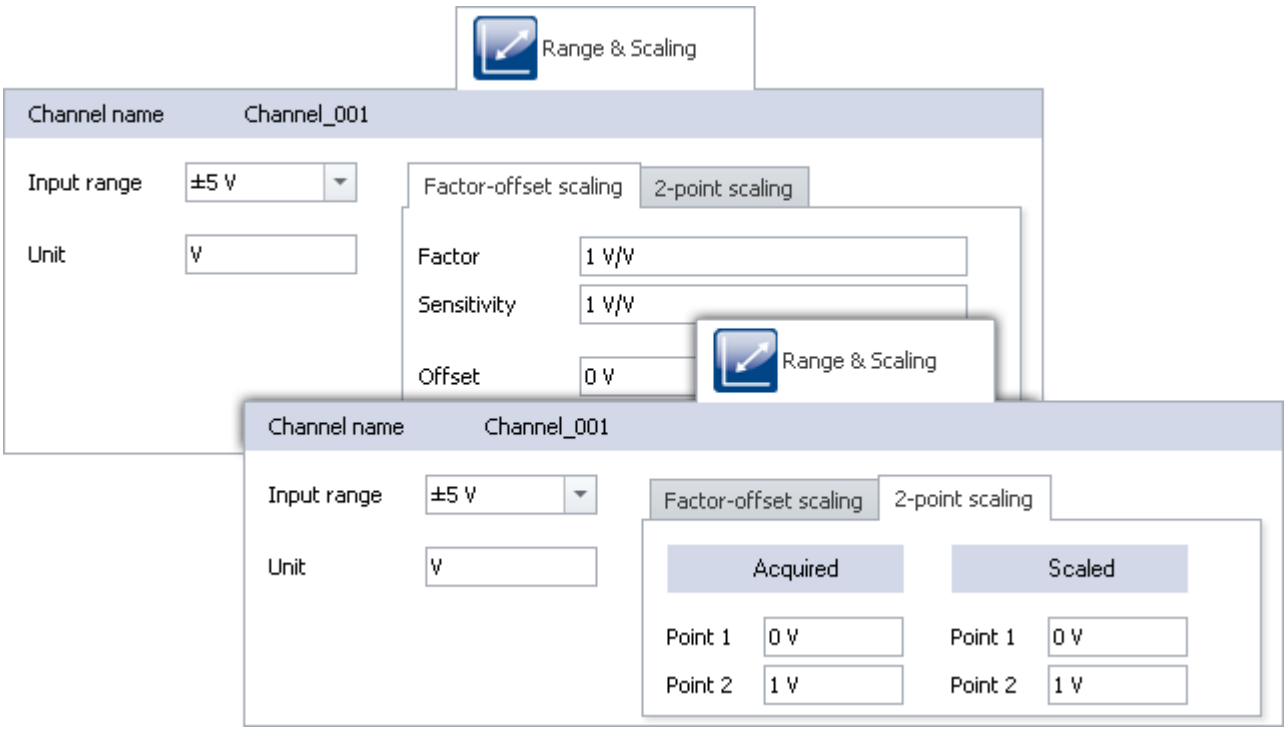
Wiring	*Channel wiring*	*Wiring*	*eWiring*

The input circuit configuration

- single ended:
  The channel signal is measured relative to ground. All channels using this setting share a common ground.
- differential:
  The channel is measured differentially.

Parameter	Description		
	long name	*short name*	*column ID*
Isolated thermo couple	*Isolated thermo couple*		*eIsolatedThermoCouple*
	Default is isolated thermocouple. When using none isolated thermocouples, that option has to be deactivated, to avoid ground loops.		
Polarization	*Polarization*		*ePolarization*
	Activates the power supply for certain microphone types. Please observe the notes on the respective amplifiers.		

9.3.4 Range & Scaling



Factor-offset scaling

Parameter	Description		
	long name	*short name*	*column ID*
Input range	*Input range*		*eRange*
	Lists the input ranges available. If you use scaling for the channel or if you have carried out an adjustment, the resulting physical input range with its unit is displayed here. For temperature measurements the input ranges depend on the thermocouple type or RTD.		

Input ranges rounded

Use of taring or the entering of a value for the "*Sensitivity*" easily cause the generation of input ranges having very many decimal places. The display of the input ranges is rounded off to 5 significant digits.
Parameter sets and channel parameters contain the original value.

Parameter	Description *long name*	*short name*	*column ID*
	See also: Setting the expected signal value range in addition to the measurement range 373		
Unit	*Unit*		*eUserUnit*
	Unit of measurement values. Select SI units without milli, micro etc., which the curve window with do automatically.		
Factor	*Scaling factor*	*Factor*	*eUserScalingFactor*
	Here, you can enter a scaling factor, in order to perform a conversion of the measured value to a physical quantity. The system will adjust the input range accordingly. Factor = 1/Sensitivity		
Sensitivity	*Sensitivity*		*UserScalingFactor_Reciproc e*
	Here you can enter a value for the sensitivity for the purpose of converting a measured value to physical quantities. The system will adjust the input range accordingly. Sensitivity = 1/Factor		
Offset	*Scaling offset*	*Offset*	*eUserOffset*
	If your signal is distorted by a DC component which cannot be compensated by the amplifier, the possibility exists to mathematically cancel this offset. In that case, the measurement signal is shifted by the amount of the offset entered. **Example:** If the signal hovers around a level of +2.5 V, an offset of -2.5 V shifts the signal towards the X-axis. Since the physical input range hasn't changed, the displayed range is also shifted by the offset amount. For the example of 2.5 V, a input range of previously ±10 V becomes a displayed input range of -12.5 V to +7.5 V.		

To scale the measured values as a physical quantity, you can perform **factor-offset scaling** or **2-point scaling**. The input range is adapted accordingly.

Option	Description
Factor-offset scaling	Here you can enter a factor or the sensitivity (=1/factor) and an offset. Physical quantity = measured value * factor + offset
2-point scaling	Here you can assign physical pond to two measured values.

Example — Example of Factor-Offset-scaling

A bridge amplifier is operated in the 100 mV/V input range. A displacement sensor is connected which unbalances the bridge by 1.6 mV/V for one millimeter of displacement. Toward this end, enter 1.6e-3 for **Factor** and for the unit m, and the input range changes to ±0.16 m. Also, if you enter for **Offset** a value of 5e-3, the input range changes accordingly to -155 mm ...165 mm.

Example — Example for 2-point scaling

A pressure sensor returns a current of 4-20 mA to correspond to inputs of 0-10 bar. The current signal is captured via a connector with a 50 Ω shunt resistor. The resulting voltage is thus:

0 V to 10 bar corresponds to 4 mA * 50 Ω to 20 mA * 50 Ω = **0.2 V to 1 V**

- For **Acquired**, enter the resulting voltage values,
- and for **Scaled**, the corresponding pressure values.

Point 1:	0.2 V	corresponding to	0 bar
Point 2:	1 V	corresponding to	10 bar

Note

Higher precision can be achieved by performing two-point scaling on the amplifier (see: Two-point scaling 382). By means of this procedure, all uncertainty tolerances are taken into account in the scaling. However, it is possible that the values scaled by the amplifier are additionally combined with the scaling described here.

Note — Notes on hardware: balancing with setted offset

Zeroing with adjusted offset leads to different results with CRFX/CRXT modules and amplifiers installed in imc CRONOS*compact*, imc C-SERIES or imc SPARTAN:

- CRFX/CRXT: Zeroing 377 adjusts the channel in the physical unit, e.g. 0 bar. This applies to all modules for the acquisition of analog voltages with 16 or 24 bits, except UNI-4.
- imc CRONOS*compact*, imc C-SERIES, imc SPARTAN, CRFX/UNI-4: The electrical value is balanced, i.e. the entered offset is retained. After the adjustment, the physical value is displayed with a negative offset. This ensures that an offset entered subsequently is not overwritten. If you wish for the balancing to determine the offset, **only enter the factor**, and then subsequently balance the offset by means of the Balancing function 377. Adaptation to the behavior of CRFX/CRXT is in preparation.

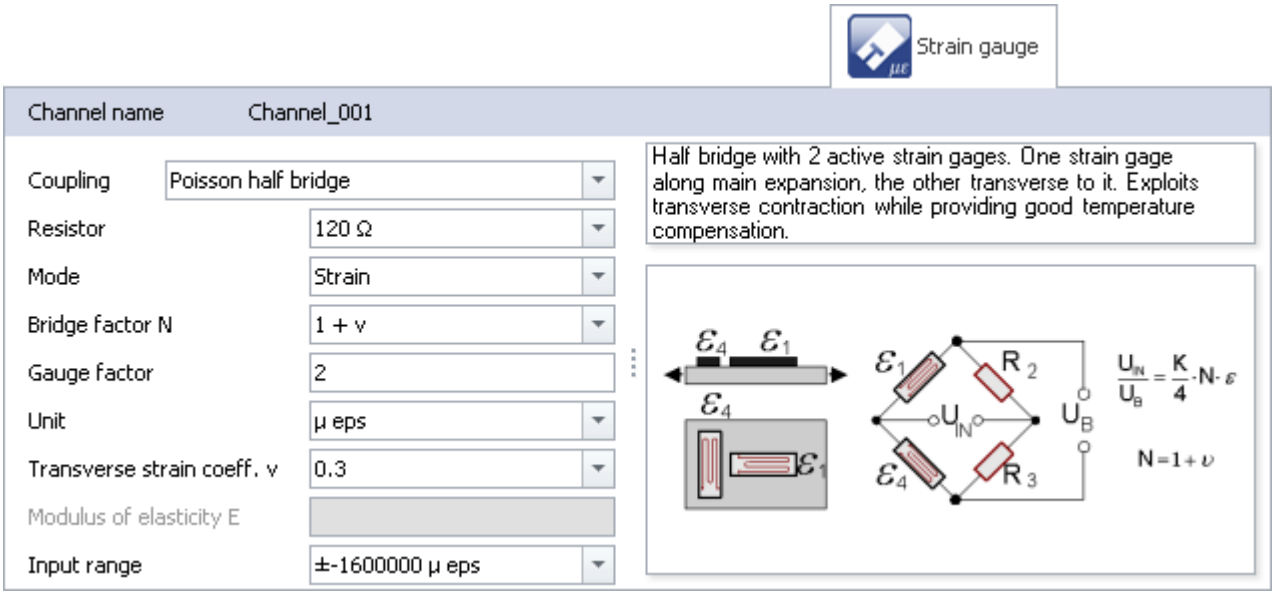
Reference — See also

Calibration of IEPE sensors 202

9.3.5 Strain gauge

Only enabled for channels in the mode: **Strain gauge**.

To configure the strain gauge, you must set the **Mode** to *Strain gauge* in the dialog **Measurement mode**. A new dialog **Strain gauge** appears, as long as the channel is selected in the channel table.



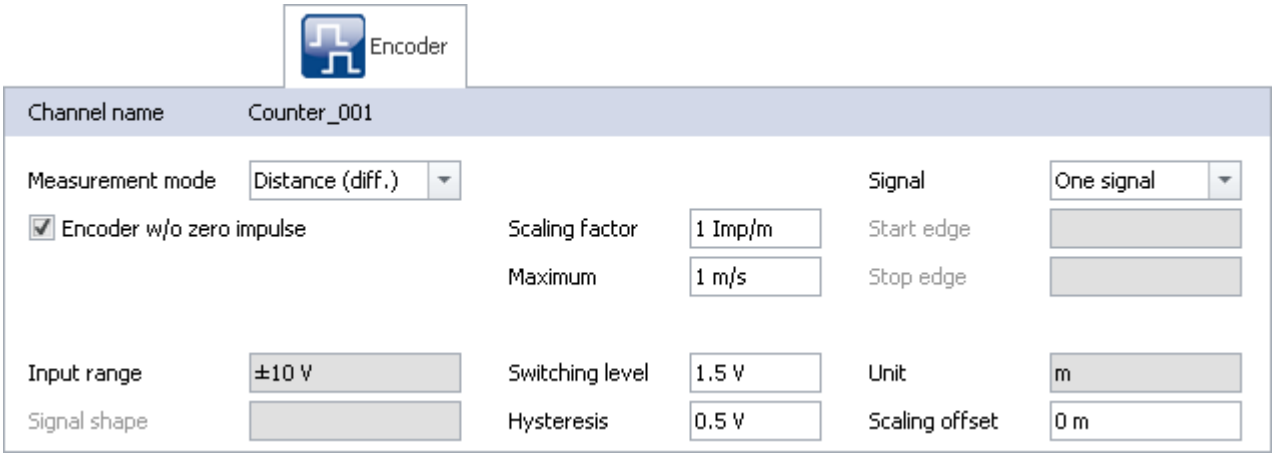
Parameter	Description		
	long name	*short name*	*column ID*
Coupling	*Coupling*		*eCoupling*
	Lists various usage possibilities. Corresponding descriptions and sketches of the strain gauge are displayed. The strain gauge is pre-set to conform to the coupling-selection. It is also still possible to make appropriate settings.		
Resistor (Bridge resistor)	*Bridge resistor*	*Resistor*	*eBridgeResistor*
	The bridge resistor must be stated for all bridge types, although it is actually only absolutely necessary for the quarter bridge. For full bridges, the bridge resistor is used along with the **calibration resistor** to calculate the **indicated strain** (see: "*Cable compensation without sense line* 384"). Also, with the bridge resistor, it is possible to determine the cable resistance even if no Sense line is connected.		
Mode	*Bridge mode*	*Mode*	*eBridgeMode*
	Strain or *Stress*.		
Bridge factor N	*Bridge factor N*		*eBridgeN*
	The available selection depends on the coupling.		
Gauge factor K	*Gauge factor*		*eBridgeFactor*
	Depends on the part's material		

Parameter	Description		
	long name	*short name*	*column ID*
Unit	*Bridge unit*	*Unit*	*eBridgeUnit*
	Unit of measurement values. For the mode **Strain**, the following units are available: *µ eps* and *µm/m*; (1 µm/m = 1 µ eps) For the mode **Stress**, the following units are available: *MPa* and *GPa* and *N/mm²* and *PSI* (pound-force per square inch) (1 MPa = 1 N/mm²; 1000 MPa = 1GPa; 1 MPa = 145,04 PSI)		
Transverse strain coeff. ν	*Transverse strain coeff. ν*		*eBridgeEps*
	Depends on the part's material		
Modulus of elasticity E	*Modulus of elasticity E*		*eBridgeEModule*
	Depends on the part's material		
Input range	*Input range*		*eRange*
	Lists the input ranges available. If you use scaling for the channel or if you have carried out an adjustment, the resulting physical input range with its unit is displayed here. For temperature measurements the input ranges depend on the thermocouple type or RTD. **Input ranges rounded** Use of taring or the entering of a value for the "*Sensitivity*" easily cause the generation of input ranges having very many decimal places. The display of the input ranges is rounded off to 5 significant digits. Parameter sets and channel parameters contain the original value. See also: Setting the expected signal value range in addition to the measurement range 373		

9.3.6 Encoder (incremental counter inputs)

Only visible for devices with incremental counter inputs.

Certain properties are dependent on the hardware and are described in the respective device's manual. There, you will also find background information on the module's technical workings.



Parameter	Description		
	long name	*short name*	*column ID*
Measurement mode	*Measurement mode*	*Mode*	*eChannelMode*
	Measurement types. See Measurement mode - Overview 358		
Encoder without zero impulse	*Encoder w/o zero impulse*		*eNullImpuls*
	The **zero impulse** starts the incremental counter inputs' logic unit. Measured values are only recorded once an event has occurred at the **index channel**. If measurement without a zero pulse is selected, measurement begins directly after preparation of the measurement is complete.		
Signal	*Counter signal*	*Signal*	*eSignal*
	Single-track encoders or Dual-track encoders • **One signal:** Single-track encoders capture a sequence of impulses without a specified direction. The value is always positive, as for example a frequency or (typically) an RPM-value. • **Two signal:** Dual-track encoders return two impulse sequences, mutually offset by 90°. In this way, the counter can distinguish in which direction the sensor is moving. This type s especially suited to measurements of velocity, distance and angle. Please observe the notes on **two-point scaling** in the section "*Single-track / Dual-track encoder* 470". Affects both the devices belonging to the imc C-SERIES, and also any devices belonging to the imc SPARTAN and imc CRONOS families which are equipped with the digital multiboard: DI16-DO8-ENC4 or the DI8-DO8-ENC4-DAC4.		
Scaling factor	*Counter scaling factor*	*Scaling factor*	*eUserScalingFactorENC*
	How many pulses represent one physical unit depends on the measurement mode set.		

Parameter	Description		
	long name	*short name*	*column ID*
Maximum	*Maximum*		*eMaximum*
	Maximum value or input range end value The measured value should lie within the upper third of the range. The resolution results from the maximum divided by 2^{16}, or 2^{15} for dual-track encoders.		
Start edge	*Counter start edge*	*Start edge*	*eStartEdge*
	Start edge on time and pulse measurement. Depends on the measurement mode 358.		
Stop edge	*Counter stop edge*	*Stop edge*	*eStopEdge*
	Stop edge on time and pulse measurement. Depends on the measurement mode 358.		
Input range	*Counter input range*	*Input range*	*eInputRange*
	This setting is available only for certain incremental counters. For information on this subject, refer to the material on the respective incremental counters in the device manual.		
Signal shape	*Counter signal shape*	*Signal shape*	*eSignalform*
	This setting is available only for certain incremental counters. For information on this subject, refer to the material on the respective incremental counters in the device manual.		
Switching level	*Counter switching level*	*Switching level*	*eLevel*
	Here you set the lower voltage limit interpreted as the HIGH level.		
Hysteresis	*Counter hysteresis*	*Hysteresis*	*eHysteresis*
	The hysteresis ensures that duplicate impulses are not counted in the case of noisy signals. These two settings are used to derive the signal levels as follows: HIGH = Signal > Switching level LOW = Signal < (Switching level - Hysteresis)		
Unit	*Unit*		*eUserUnit*
	Measurement value unit. Depends on the measurement mode set.		
Scaling offset	*Scaling offset*		*eUserOffset*
	Here you can enter an offset.		

Note

Low-pass filter

Set the low-pass filter accordingly (see "*Filtering* 360"). The Low-pass ensures that peaks in the signal do not lead to duplicate counts.

Make sure that the frequency of the useful signal is not below the low-pass frequency set. Otherwise, your useful signal will be filtered smooth.

Sampling times/rates

For all active incremental counter channels (and associated active Monitor channels) of one module (e.g. CRFX) or a device (e.g. C-SERIES), only one sampling interval/-rate may be set.

9.3.6.1 Measurement mode - overview

Reference

More information about the measurement modes are described in the section: "*Measurement types*" > "*Incremental Counters Channels* 465" or in the respective device manual.

As a rule, what type of measurement is performed is determined by your setting for the **Measurement mode**:

Event counting

Measurement mode	Description
Events	Number of events within a sampling interval
Distance (differential)	Path traveled within one sampling interval. For this purpose, the number of pulses per meter must be entered. The absolute distance is not directly measurable but must be calculated in imc Online FAMOS.
Angle (differential)	Angle traveled within one sampling interval. For this purpose, the number of pulses per revolution must be entered. The absolute angle can be calculated in imc Online FAMOS or determined by the mode Angle (abs).
Angle (absolute)	Absolute angle. The differential angle measurement is converted to the absolute angle. By taking the zero impulse (the counter with no zero impulse should not be selected) into account, the absolute angle position is determined and indicated. Otherwise, the angle value is assumed to be 0° when the measurement begins.

Time measurement

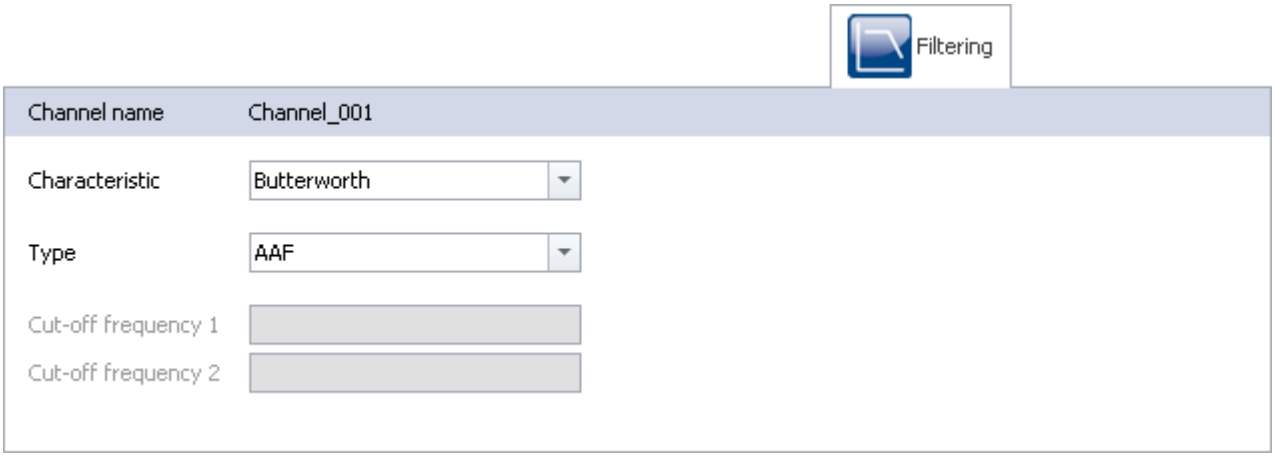
Measurement mode	Description
Time	The time between two signal edges is determined. For this purpose, the settings options for the **Start** and **Stop** of the measurement appear. There are several possibilities: positive edge > negative edge: ↑ > ↓ negative edge > positive edge: ↓ > ↑ positive edge > positive edge: ↑ > ↑ The combination negative edge > negative edge: ↓ > ↓ is not allowed
Pulse time	The point in time of a signal edge within the sampling interval is returned. Certain imc Online FAMOS functions used this information, e.g. when determining a pulse signal's plot of angular speed: OtrEncoderPulsesToRpm. The mode **Pulse time** depends on the sampling interval. The entry only appears for some amplifiers if the appropriate sampling interval is set (≤ 1 ms or ≤ 100 µs; see **device manual**).

Combined measurement

Measurement mode	Description
Frequency	The time of two consecutive pulses is converted to frequency. If the frequency was previously multiplied or divided, this can be reflected in the scaling value. The frequency is always unsigned, for which reason there is no dual-track encoder for it.
Speed	The sequence of pulses is converted to m/s. Toward this end, the amount of pulses per meter must be entered.
RPM	The sequence of pulses is converted to revolutions per minute. Toward this end, the amount of pulses per revolution must be entered.

9.3.7 Filtering

The amplifiers are able to filter the signal before it's subject to further processing in the system. The 6th-order filter with Butterworth or Bessel characteristics performs high-, low- and band-pass filtering.



Filter settings (example: Bandpass)

Here you can select the **filter characteristics** and the **filter type** for the selected channel. The cutoff frequency 2 can only be set for bandpass filters.

Along with filter types to select by their cutoff frequencies, there is also an **anti-aliasing filter (AAF).** This is a low-pass filter whose cutoff frequency is set to comply with the requirements of the Sampling Theorem. Thus, frequency components above ½ of the sampling frequency are suppressed. This refers exclusively to the input channel, not to its **monitor channel** 1135. **Aliasing effects** are possible with a monitor channel by resampling.

The default setting is AAF, where a cauer filter is calculated internally. The displayed characteristic (in the picture Butterworth) is only used at a given cut-off frequency.

9.3.7.1 Filter settings

Theoretical background

The filter setting is especially important in a signal-sampling measurement system: the theory of digital signal processing and especially the **sampling theorem** (Shannon, Nyquist) state that for such a system, the signal must be restricted to a limited frequency band to ensure that the signal has only negligible frequency components beyond one-half of the sampling frequency ("Nyquist-frequency"). Otherwise, "aliasing" can result – distortions which cannot be removed even by subsequent filtering.

The imc device is a sampling system in which the sampling time (or sampling rate) to be set is subject to this condition. The low pass filter frequency selected thus hinges on how band-limited the signal to be sampled at that rate is.

The control AAF for the filter setting stands for "Automatic Anti-aliasing Filter", and automatically selects the filter frequency in adaptation to the sampling rate selected. The rule this is based on is given by:

AAF-Filter frequency (-80 dB) = sampling frequency · 0.6 = Nyquist frequency · 1.2

AAF-Filter frequency (-0.1 dB) = sampling frequency · 0.4 = Nyquist frequency · 0.8

General filter concept

The imc system architecture is actually a two-step system in which the analog signals are sampled at a fixed "primary" sampling rate (analog-digital conversion with Sigma-Delta ADCs). Therefore a fixed-frequency analog low pass filter prevents aliasing errors to this primary rate. The value of this primary rate is not visible from the outside, depends on the channel type and is generally greater than or equal to the sampling rate which is selected in the settings interface.

The filter to be set is realized as a digital filter, which offers the advantage of precise characteristic and matching with respect to magnitude and phase. This is especially important for the sake of matching of channels which are jointly subjected to math operations.

For any data rate to be set in the system configuration (f_sample), then digital anti-aliasing filters (low pass filters) ensure compliance with the conditions for the Sampling Theorem. Three cases can be distinguished.

Implemented filters

Filter-setting "Filter-Type: without":

Only the (analog) anti-aliasing filter, matched to the primary data rate is in effect.
This setting can be useful if maximum bandwidth reserves are to be used and there are known limitations on the measured signal's spectral distribution, which justify not performing consistent filtering.

Filter-setting "Filter-Type: AAF":

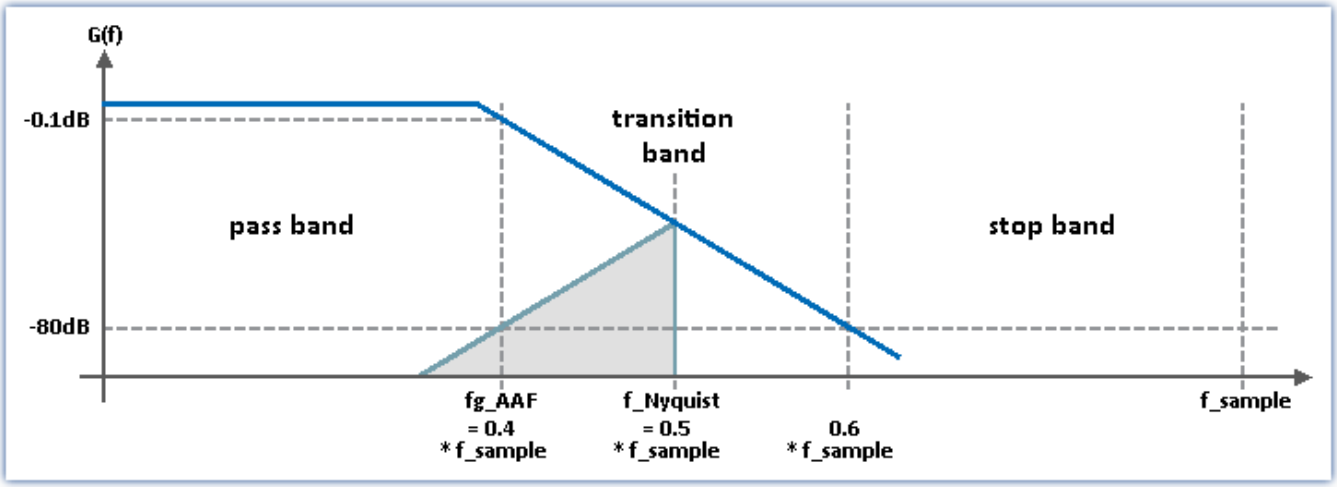
The (digital) anti-aliasing filters are elliptical Cauer filters. Their "tight" characteristic curve in the frequency range makes it possible to have the cutoff frequencies approach the sampling and Nyquist frequencies much closer without having to make a compromise between the bandwidth and freedom from aliasing.

The automatic selection of the cutoff frequency in the setting "AAF" is based on the following criteria:

- In the pass band, a maximum (AC-) gain uncertainty of 0.06% = -0.005 dB is permitted. The pass band is defined by the cutoff frequency at which this value is exceeded.
- The stop band is characterized by attenuation of at least -80 dB.
  This damping is considered sufficient since discrete disturbance frequencies can never reach 100% amplitude: the input range is mostly filled by the useful signal.
- The transition band is typically situated symmetrically around the Nyquist-frequency. This ensures that the aliasing components reflected from the stop band back into the pass band are adequately suppressed, by at least -80 dB. Remnant components from the frequency range between Nyquist-frequency and stop band limit only reflect back into the range beyond the pass band (pass band to Nyquist), whose signal content is defined as not relevant.

The criteria stated are fulfilled with the Cauer-filters by the following configuration rule:

- fg_AAF (-0.1 dB) = 0.4 · f_sample
- Characteristics: Cauer; Filter-order: 8th order



Filter-setting "Filter-type: Low pass" (band pass and high pass):

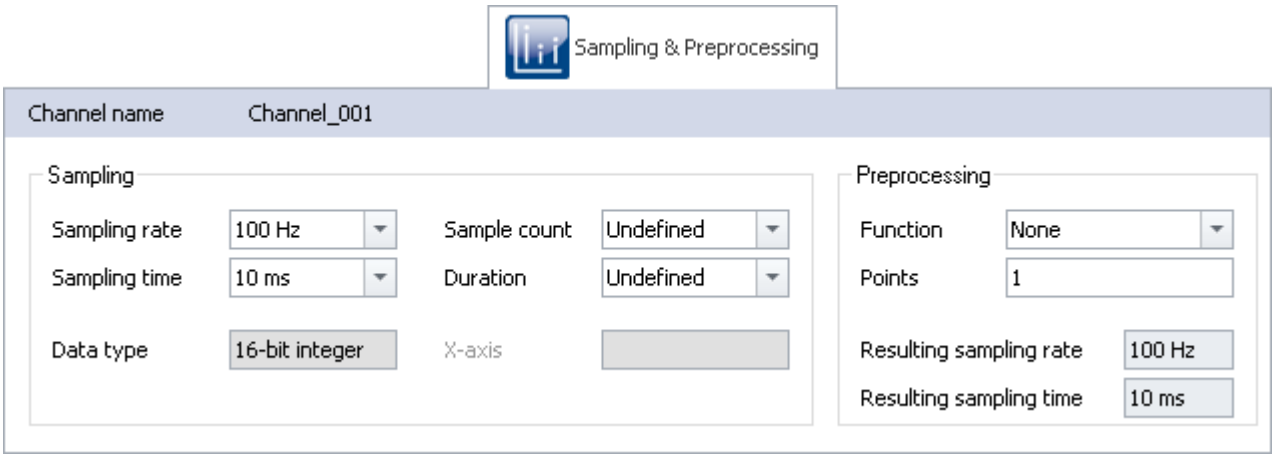
A low pass frequency can be set manually, which satisfies the application's requirements. In particular, a cutoff frequency significantly below the Nyquist frequency can be set which guarantees eliminating aliasing in any case, though consequently "sacrificing" the corresponding bandwidth reserves.

with fg_AAF (3 dB)	= f_sample / 4	attenuation at Nyquist-freq.:	1/64	= -36 dB
with fg_AAF (3 dB)	= f_sample / 5	attenuation at Nyquist-freq.:	1/244	= -48 dB
with fg_AAF (3 dB)	= f_sample / 10	attenuation at Nyquist-freq.:	1/15630	= -84 dB

- Characteristics: Butterworth, 8th order (48 dB/octave)

For devices of firmware group A 187: Other possible filter settings are "*band pass*" and "*high pass*" - both 4th order.

9.3.8 Sampling & Preprocessing



Sampling & Preprocessing

The availability and selection of settings depends on the channel type.

Parameter	Description		
	long name	*short name*	*column ID*
Sampling rate	*Sampling rate*		*SampleRate*
	Setting for the sampling rate. Sampling rate = 1/Sampling time **Active channels** can use a maximum of two different sampling times/rates per device. For all active **incremental counter channels** (and associated active Monitor channels) of one module (e.g. CRFX) or a device (e.g. C-SERIES), only one sampling interval/-rate may be set. See also: Notes on the "Sampling rate" and the "Aggregate sampling rate" 369		
Sampling time	*Sampling time*		*eSampleTime*
	Setting for the sampling time. Sampling time = 1/Sampling rate **Active channels** can use a maximum of two different sampling times/rates per device. For all active **incremental counter channels** (and associated active Monitor channels) of one module (e.g. CRFX) or a device (e.g. C-SERIES), only one sampling interval/-rate may be set. See also: Notes on the "Sampling rate" and the "Aggregate sampling rate" 369		
Sample count	*Sample count*		*SampleCount*
	The amount of measured values per measurement. Select *undefined* (or enter "0") for unlimited measurement duration. When the Sample count is entered, the resulting measurement duration is corrected accordingly. The measurement duration is given by: Sample count * Sampling time.		
Duration	*Duration*		*eDuration*
	The duration of the measurement. Select *undefined* (or enter "0") for unlimited measurement duration. When the measurement duration is entered, the resulting sample count is corrected accordingly. The sample count is given by Duration/Sampling time.		

Parameter	Description		
	long name	*short name*	*column ID*
Measurement mode	*Measurement mode*	*Mode*	*eChannelMode*
	Basic setting for the measurement type **Digital Inputs**: • **Bit Input:** the port indicates the current state • **Sampling:** the port continually captures data. These digital channels can be displayed the same way as analog channels. Changes apply to the port and the corresponding bits.		
Data type	*Channel data type*	*Data type*	*eDataType*
	Resolution of measurement results: • **16-bit integer** • **Float (24-bit mode)** (*32-bit float (24-Bit mantissa)*) The availability of this option depends on the device. Samples with 24-bit resolution generate twice as much data volume. For 24-bit resolution, please be aware of the increased memory requirement 1155 and please observe the technical specifications stated in the user's manual for the respective device. In 16-bit mode, for devices belonging to Group A7 187 and higher, the maximum aggregate sampling rate achievable via EtherCAT is 2000 kHz.		

Settings for fieldbus channels

The data of these channels are captured via a special Fieldbus interface. The necessary definitions are made by use of the Fieldbus Assistant. Fieldbus channels are also included in the trigger machine.

Channel type	Description
Fieldbus: Analog inputs	The Fieldbus channels carry data acquired by sensors which are connected to a fieldbus such as a CAN-Bus.
Fieldbus: Digital inputs / outputs	The Fieldbus channels carry digital data acquired by sensors which are connected to a fieldbus.

If Fieldbus channels are selected in the Channel table (assuming the device is configured correspondingly), there are additional parameters to set:

Parameter	Description		
	long name	*short name*	*column ID*
X-axis	*X-axis*		*eXFormatVariable*
	• *Sampling rate*: The setting Sampling time is related to the settings **Duration** or **Sample count** and **Sampling rate/time** as previously discussed. • *Time stamp*: Data acquisition is handled by external, asynchronous sampling within the fieldbus. The corresponding time coordinate for the data point is passed as a time stamp along with the data for subsequent processing. In this case, the parameters **Sample count** and **Sampling rate/time** are irrelevant.		
Duration	*Duration*		*eDuration*
	If the parameter X-axis is set to *Time stamp*, the duration of the measurement depends only on the setting in **Duration**.		
Assignment	*Assignment*		*eAllocation*
	The data can additionally be assigned to a Display variable (only for analog Fieldbus inputs).		

Preprocessing

Simple processing of measured values directly in the device. By means of preprocessing, the data rate is reduced. The setting "*Processing points*" specifies how many data points are to be processed. The returned result is a new value, and the data rate is thus reduced by a factor of this number of points. In contrast to the analogous functions in imc Online FAMOS, this technique does not allow you to set processing for a moving window of data.

Functions

The following functions are available:

Preprocessing	**Group A**		**Group B**		
	CRC, CRSL, C-SERIES	**CRFX, CRXT**	**EOS**	**ARGFT**	
	Original & monitor channel	**Original & monitor channel**	**Original & monitor channel**	**Original channel**	**Monitor channel**
Arithmetic mean	●	---	---	---	●
Minimum	●	---	---	---	●
Maximum	●	---	---	---	●
RMS	●	---	---	---	●
AAF	---	---	---	---	●
Min-Max	●	---	---	---	---
Envelope	---	---	---[2]	---	●
Reduction	---[1]	---[1]	---	---	●

1 The reduction for the monitor channels is possible by changing the sampling rate.

2 imc EOS monitor channels are always envelope reduced channels.

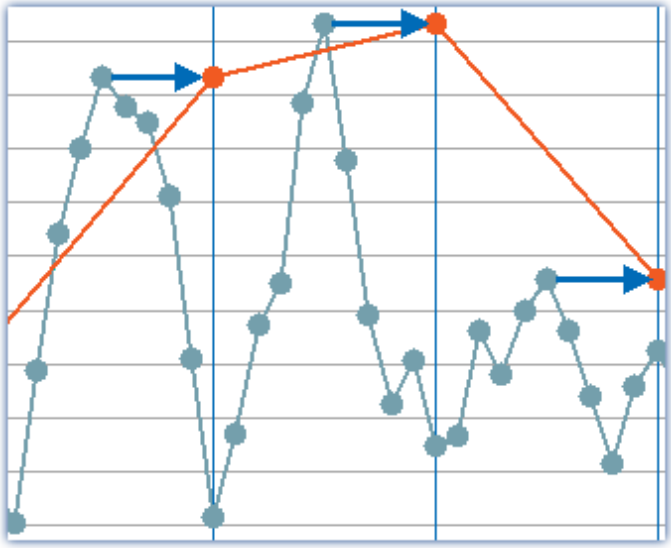
The data reduction process

Every x points, a new value appears at the **point in time of the xth value**.

The following example illustrates how data reduction is proportional to the point count.

- Function: Maximum
- Points: 10

In the example, the system finds the maximum value among the last 10 values and sets it at the position of the last of the 10 values.



Example: Maximum across 10 values. The result (orange) is assigned the time stamp of the 10th value.

Due to the averaging duration, the first measured value is displayed as well as sorted with a delay. See "*Breakdown of the Start/Stop and Triggering Behavior* 761"

Note System resource demands

For devices belonging to the firmware group A 187 (imc DEVICES):

The preprocessing is performed by the device's base board and **thus makes demands** on the device **processor**.

For devices belonging to the firmware group B 187 (imc DEVICEcore):

The preprocessing is performed by the respective amplifier boards in the device and **thus does not make any demands** on the device's **processor**.

Monitor channel as envelope type

A data-reduced envelope type monitor channel (min-/max-channel) is generated.

This process is somewhat different than the other data reduction methods - See "*Envelope type monitor channel (min-/max-channel)* 1132"

Parameters for the preprocessing

Parameter	Description		
	long name	*short name*	*column ID*
Function	*Processing function*	*Function*	*eProcessing*
	Simple processing of the measured values. Different functions are available depending on the channel type/device. In contrast to the corresponding functions in imc Online FAMOS, it is not possible here to set processing of moving segments of data. Due to the averaging duration, the first measured value is displayed as well as sorted with a delay. See "*Breakdown of the Start/Stop and Triggering Behavior* 761"		
Points	*Processing points*	*Points*	*eProcessingPoints*
	The number of data points (measured values) on which the function is to be carried out. The **number of points** is also the **reduction factor**. This means that the resulting and stored sampling rate results from the set sampling rate/points. **Monitor channels** 340 receive the data **without preprocessing**. Thus, the maximum sampling rate of a monitor channel is the set sampling rate.		
Resulting sampling rate	*Resulting sampling rate*		*PreprocessedSampleRate*
	Resulting count of values from the sampling rate/time set and the amount of points. Resulting Sampling rate = Sampling rate/Points		
Resulting sampling time	*Resulting sampling time*		*PreprocessedSampleTime*
	Resulting count of values from the sampling rate/time set and the amount of points. Resulting Sampling time = Sampling time*Points		

Special function: Data reduction for digital inputs (reduction)

Only once the signal applied changes is a new measured value recorded. The respective time of data capture is recorded as a time stamp along with the measured data, for the purpose of subsequent data processing.

Therefore the port has to be set to Measurement mode: "*Sampling*".

Only whole ports can be captured with data reduction. Individual bits always are dependent on their port. As soon as an individual bit changes its state, the port is sampled and given a time stamp.

Note

Transfer to PC is not possible: Results of the `TransRec` function and DI-port compressed channels are blocked from transfer to the PC.

9.3.8.1 Sampling rate and the Aggregate sampling rate

Tip: Avoid large sampling times

Say, for instance, a temperature is to be measured, and that 1 sampling per 10 s is sufficient resolution. A large sampling time has disadvantages:

There is no averaging. If the signal is not subjected to analog smoothing, there will be no noise suppression. High frequency noise will cause aliasing errors.

Tip: Mean-function

- Set a higher sampling rate in the Configuration dialog. Especially for temperature measurement, sampling rates of at least 200 Hz should be set, if no analog anti-aliasing filter is used. Use averaging to reduce the data to the sampling rate desired.
- To the extent possible, use the imc Online FAMOS function `Mean` instead of the "*Arith. Mean*" on the page: "*Analog/Digital channels*" in the dialog: "*Sampling & Preprocessing*". Only the `Mean` function avoids the Start/Stop-delay.

For devices belonging to the firmware group B 187 (imc DEVICEcore):

You can set a separate sampling rate for each physical measurement channel. The aggregate sampling rate of the system is the sum of the **sampling rates of all active channels**. Active **monitor channels** are included in the calculation of the aggregate sampling rate at the reduced rate.

The aggregate sampling rate can take a maximum value of 4 MHz or 5 MHz with devices of the firmware group B 187.

Please see the pertinent technical data sheet to learn whether the aggregate sampling rate depends on the use of pv-variables or on the trigger count.

The sampling rates of the **virtual channels** computed by **imc FAMOS** do not contribute to the aggregate sampling rate.

Reference

Please consider the specifications in chapter "*Technical Specs*" of the device manual and the description of the channels configuration.

For devices belonging to the firmware group A 187 (imc DEVICES):

Among the measurement device's physical measurement channels, up to two different sampling times can be in use. The smallest possible sampling time is 10 µs, corresponding to a channel sampling rate of 100 kHz (sampling frequency). The aggregate sampling rate of the system is the sum of the sampling rates of all active channels.

The aggregate sampling rate can take a maximum value of 400 kHz with devices of group A4 to A5 187.

The aggregate sampling rate can take a maximum value of 2 MHz via EtherCAT else 400 kHz with devices of group A7 187. This maximum aggregate sampling rate (2 MHz via EtherCAT) can only be achieved when not using the process vector variables, without any trigger, and only in 16-bit mode! A CRFX / DAC-8 module contributes 5 kHz and 16 bits per channel, regardless whether it is used in the experiment.

The sampling rates of the **virtual channels** computed by **imc FAMOS** do not contribute to the aggregate sampling rate. Along with the (maximum of) two "primary" sampling rates, the system can contain additional "sampling rates" resulting from the effects of certain data-reducing imc Online FAMOS functions (ReductionFactor RF).

Note

There is one constraint when selecting **two different sampling rates**: **Two sampling rates** having the ratio **2:5** and lower than 1 ms are not permitted (e.g. 200 µs and 500 µs). Any attempt to set sampling rates which do not comply with this rule will cause an error message to be posted:

"The two active sampling intervals may not be in a ratio of 2:5. Error number: 365"

Note Note for imc CRONOS*flex*/CRONOS-XT

For CRFX/CRXT modules, the following limitations apply due to the CRFX systembus's 5 kHz bus rate:

- A CRFX/CRXT-channel sampling at a slower rate than 5 kHz produces the same data load on the systembus as a 5 kHz channel. Thus for instance, 20 channels sampled at 100 Hz produce a load of 100 kHz on the systembus instead of 2 kHz.
- With 2 kHz channels, the resulting data load per channel is 10 kHz due to the 2:5 ratio.
- DIO- and DAC modules support a maximum sampling rate of 5 kHz, even though 100 kHz can be set in the software.

Note Note for GPS channels

Due to system limitations, **GPS channels** for determining the fastest sampling rate in the system are not taken into account. For an working configuration, at least one other channel (fieldbus, digital or analog) must be sampled at the same or faster sampling rate.

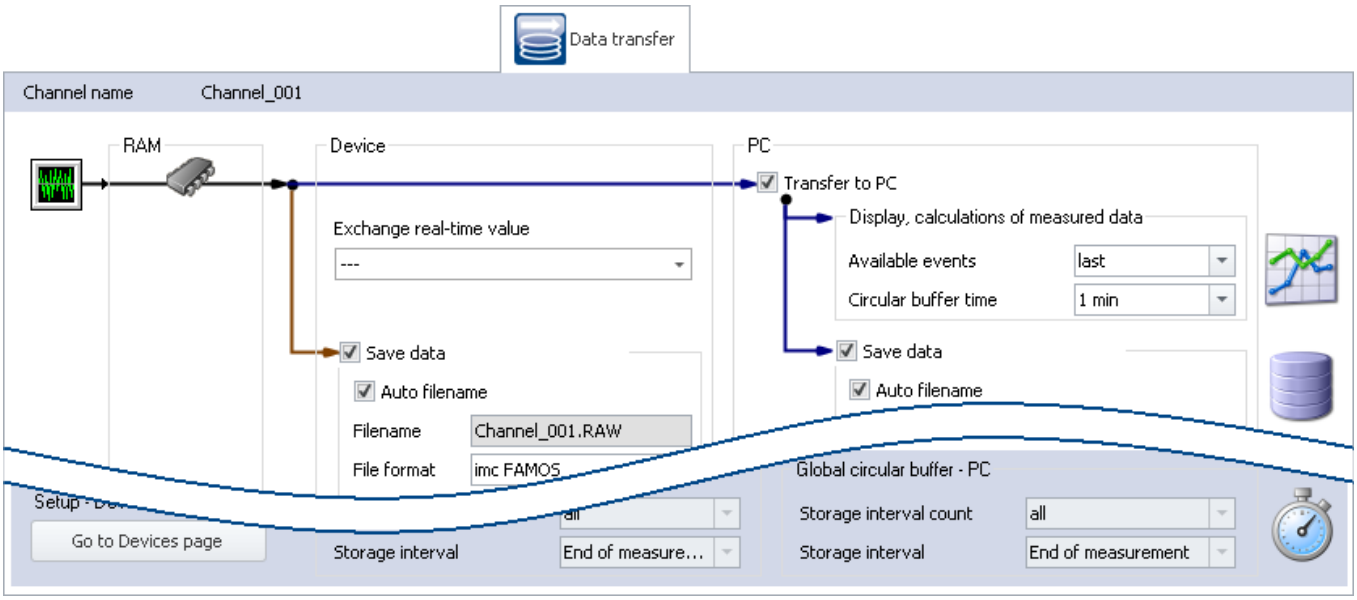
Reference

Please consider the specifications in chapter "Technical Specs" of the device manual and the description of the channels configuration.

9.3.9 Data transfer

Reference

This chapter contains a brief introduction to saving channel signal data. More detailed descriptions of all functions are presented in the section: "*Storage options and directory structure* 411".



Transfer to PC

For transfer of the measured data to the PC, there are two regions of the dialog:

Region	Description
Display, calculations of measured data	Temporary storage of measured data. E.g. for display of a Panel page and for subsequent processing.
Save data	Saving of measured data in the experiment folder. The data structure is set in the device settings under Storage 292. Per interval or measurement, one file is created per channel, which contains the respective measurement files

Each of the regions is set separately.

Reference See also

- Storage options: Settings for the channels 413
- Available events and Circular buffer time: see "*Events and circular buffer* 422"
- "*Exchange real-time value*": See "*Exchanging values among the devices* 767"

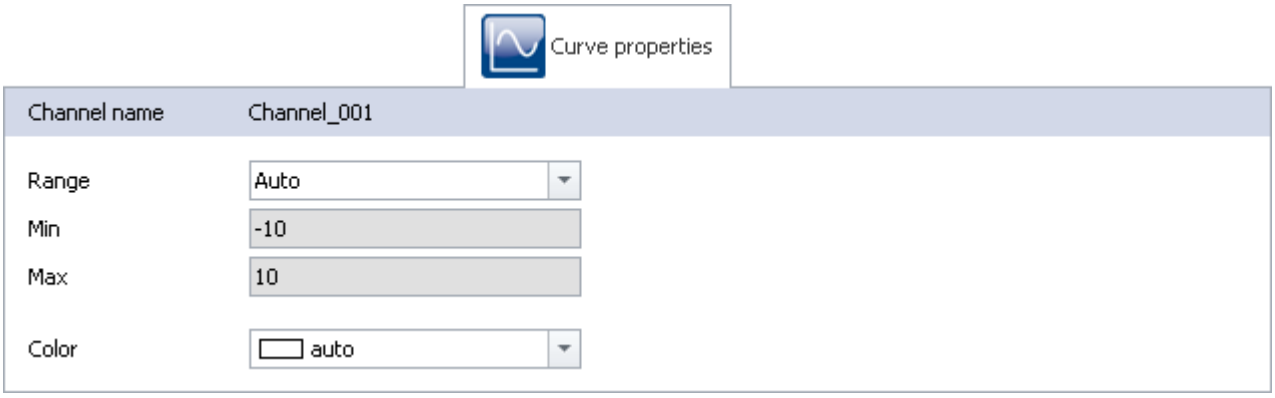
9.3.10 Histogram / Rainflow

Only visible for certain virtual channels generated in imc Online FAMOS using appropriate functions.

Here you can set defaults. These will affect the display and storage of the data of the respective virtual channel: e.g. Histogram or Rainflow.

Parameter	Description		
	long name	*short name*	*column ID*
Histogram display update interval	*Histogram display update interval*		*eHistogramUpdateInterval*
	Time interval for updating data which are derived from counting functions		
Histogram storage interval	*Histogram storage interval*		*eHistogramSaveInterval*
	Time interval for saving data which are derived from counting functions		

9.3.11 Curve properties



Parameter	Description		
	long name	*short name*	*column ID*
Range	*y-axis range*	*Range*	*eCurveYAxisOption*
	Pre-defined setting for the **range of display elements** (Widgets). The setting is applied to the (display) range if: • "auto" is set in the curve window or • the "*Range*" for the other Widgets is set as "*From the variable*". The following possibilities are available:		

Selection	Function	Wirkt auf Widgets
Auto	Max and Min values match the measurement range set for the channel	• scale-Widgets
Input range	Max- and Min-values match the measurement range set for the channel	• scale-Widgets • curve window (rezoom)
User-defined (min, max)	Max- and Min-values can be set in the parameters further below	• scale-Widgets • curve window (rezoom)

Min	*y-axis min*	*Min*	*eCurveYAxisMin*
	If "*y-axis range*" is set to "*User-defined*": The transmitted Minimum-value for the display range.		
Max	*y-axis max*	*Max*	*eCurveYAxisMax*
	If "*Y-Axis option*" is set to "*User-defined*": The transmitted Maximum-value for the display range.		
Color	*Curve color*	*Color*	*eCurveColor*
	Pre-defined setting for the **channel-color** in the curve window.		

Setting the expected signal value range in addition to the measurement range

For some sensors, it is sometimes desirable not only to define the measurement range, for example ± 10 kN or - 270..500 °C, but also an expected value range, e.g 0..9 kN or 20..80 °C. The expected value range is usually only a subset of the measurement range. The intention would be to display only values within that range in order to be able to directly concentrate attention on the relevant portion.

Whenever values outside of this range are measured, they are to be considered overmodulation. This provides a quick indication that something is not right at the moment, even if there is no actual overmodulation from an electrical standpoint.

Specify whether the measurement range ("*Auto*"/"*Input range*") or other Min- and Max-values ("*User-defined*") transfer to the channel. The Widgets apply these presets if no other settings ("*From Variable*") have been made.

In many cases, no modification is necessary any longer, or the range can be governed by means of the parameters for all Widgets which display the channel.

Instead of that resulting range, it is also possible to specify a separate, different range for all Widgets.

Level indicators

"*Current value*"-column in the Setup, as well as the Widget "*Level indicator*"

The level indicator displays also use the ranged defined here. When values outside of this range are measured, the level indicators indicate overmodulation even when there is none from an electrical standpoint.

9.3.12 Additional parameters

Additional parameters are available, which are either already displayed in the channel table, or which you can add manually using the "*Column chooser* 248".

Category: Channel

Parameter	Description		
	long name	*short name*	*column ID*
Balance at device startup	*Balance at device startup*	*Balance at startup*	*eBalanceAtDeviceStart*
	This parameter has two purposes: • Before a Diskstart/Autostart measurement starts, **balancing** of the selected channels is performed. • The channels selected can be balanced using the **imc Online FAMOS function** `RunAutoBalance` 1008. This balancing operation starts automatically before every Diskstart-measurement. The type of balancing performed is set on the following page: Setup page: "*Channel balance*" > "*Balance* 377". Available options: Taring or bridge balancing (depending on the hardware).		
Connector	*Connector*		*ePlugInName*
	The number returns a fixed identification of the variable. E.g. "[03] IN04". It is composed of: • [i]: the module's address (mainly only for physical inputs and outputs) • xyz: abbreviation for the channel type (IN = analog channel, MIN: monitor: analog channel, ...) • j: consecutive numbering within the respective module or entire channel type, if no address available Examples: [03] IN04: analog input, fourth input at the address "03" V005: Virtual channel with the number 5 [00] DO01Bit05: Digital outputs (bit) fifth bit at Port: "01", at address: "00" [02] MCIN02: Monitor: incremental counter-inputs, second input at the address "02" PV007: Process vector variable with the number 7		
Enumerated channel number	See "Channel number"		

Additional parameters are available, which are either already displayed in the channel table, or which you can add manually using the "*Column chooser* 248".

Parameter	Description		
	long name	*short name*	*column ID*
Channel number	*Enumerated channel number*	*Channel number*	*eEnumeratedChannelNumber*

Corresponds to the column "*Connector*", with the exception of the module address. The module address is not included in the display and so the count is performed across all modules, rather than each module separately.

The column: "Connector" does not correspond to the consecutive numbering on the front panel for some devices (e.g. imc SPARTAN and imc CRC).

Example:

Above: "Connector"; below: "Enumerated channel number"

[01] IN01	[01] IN02	[01] IN03	...	[01] IN07	[01] IN08	**[02] IN01**	[02] IN02	[02] IN03	...	[02] IN07	[02] IN08	**[03] IN01**	[03] IN02
IN001	IN002	IN003	...	IN007	IN008	**IN009**	IN010	IN011	....	IN015	IN016	**IN017**	IN018

Category: Module

Parameter	Description		
	long name	*short name*	*column ID*
Bridge supply	*Bridge supply (On/Off)*	*Bridge supply*	*eBridgeSupplyOnOffAction*

Some amplifiers provide the ability to switch the power supply on or off. The power supply voltage can be used either for the bridge supply or the sensor supply.

Additional parameters are available, which are either already displayed in the channel table, or which you can add manually using the "*Column chooser* 248".

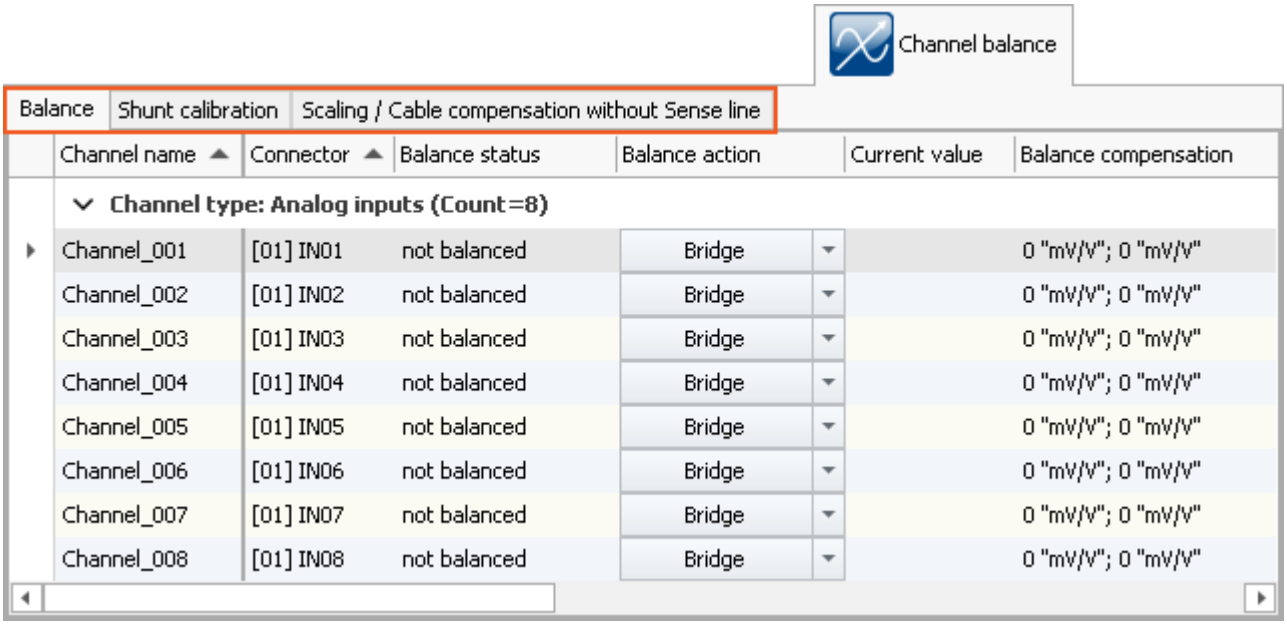
Category: Other

Parameter	Description		
	long name	*short name*	*column ID*
Current value	*Current value*		*CurrentValueDisplay*
	Shows the current reading of the following devices/channel types: • analog inputs, incremental counter inputs, field bus channels, FunctionSimulator channels, Application module channels, GPS channels • pv-variables, display variables • ethernet bits, virtual bits If no value is displayed, the measurement has not been prepared/started. For channels having an associated pv-variable (e.g. analog inputs or incremental counter inputs), the following applies: • The current value is displayed as soon as the action "*Prepare*" is performed (so for instance, also before and after the measurement). • If the value is displayed in gray font, channel settings have been changed since the "*Prepare*" procedure. The other channel types display the current reading as long as the channel is returning measurement data. If the measurement is not running, the last value is retained and displayed in gray font. This is to clearly indicate which value is still being updated and which might be obsolete. **Level indicator for channels having a measurement range (except incremental counter inputs)** Color coding indicates in which portion of the measurement range the value is located (green/red shading).		

9.4 Channel balance

On the page **Channel balance**, you can have various adjustment and calibration types performed, and obtain information on the configuration.

To get to the desired page, there is a row of tabs at the lower left.

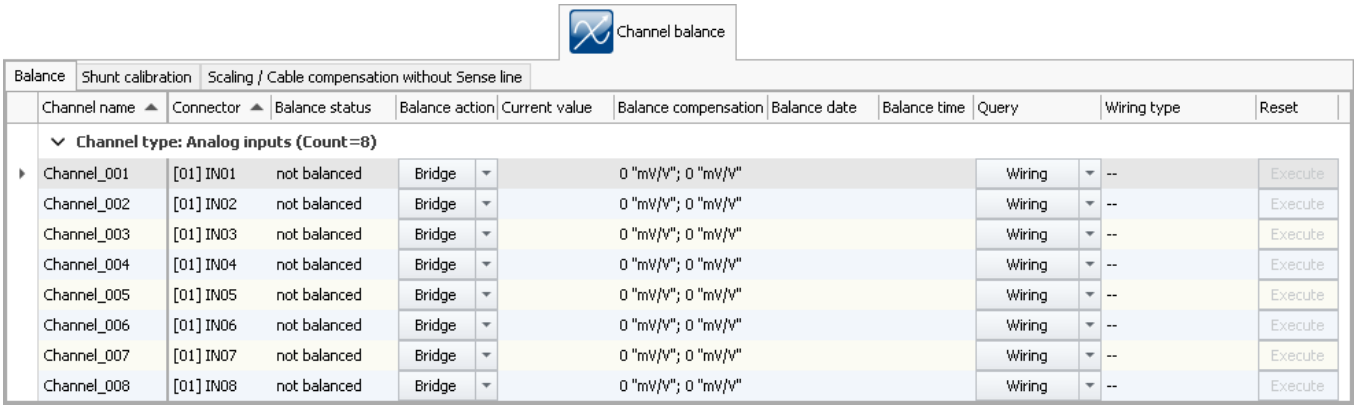


Balance 377	• Tare • Bridge • Query wiring • Reset charge
Shunt calibration 381	• Shunt calibration
Scaling / Cable compensation without Sense line 382	• Two-point scaling • Cable compensation

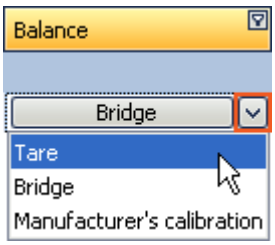
9.4.1 Balance - tare and bridge

Balance

This page enables, among other things, **taring** and **bridge balancing** of amplifier channels to be performed.



To perform an adjustment, first select the channel and in the column *Balance* using the button a right, the desired action.



Adjustment only starts once you click on the button after making your selection.

Balance	Description
Bridge	Performs physical bridge balancing for all selected bridge channels. Balancing can eliminate any offset distorting the measurement signal. The offset can be a multiple of the input range. With bridge balancing, the input range selected is retained. With multiple channels, the bridge balancing is performed in parallel amplifier-by-amplifier, but sequentially within an amplifier.
Tare	With taring, the distorting offset is canceled mathematically from the signals of the measured channels, which leads to a shift of the input ranges. The measurement chain including sensors is taken into account. Observe the notes on hardware 378.
Manufacturer's calibration	Manufacturer's calibration resets all selected channels to the original imc-factory calibration setting

Note — Notes on hardware: balancing with setted offset

Zeroing with adjusted offset leads to different results with CRFX/CRXT modules and amplifiers installed in imc CRONOS*compact*, imc C-SERIES or imc SPARTAN:

- CRFX/CRXT: Zeroing 377 adjusts the channel in the physical unit, e.g. 0 bar. This applies to all modules for the acquisition of analog voltages with 16 or 24 bits, except UNI-4.
- imc CRONOS*compact*, imc C-SERIES, imc SPARTAN, CRFX/UNI-4: The electrical value is balanced, i.e. the entered offset is retained. After the adjustment, the physical value is displayed with a negative offset. This ensures that an offset entered subsequently is not overwritten. If you wish for the balancing to determine the offset, **only enter the factor**, and then subsequently balance the offset by means of the Balancing function 377. Adaptation to the behavior of CRFX/CRXT is in preparation.

Response of the balancing to changing the measurement range and measurement mode

The balancing process depends on the measurement range and measurement mode (Voltage – Current – Strain Gauge). This means that the balancing only applies for the currently valid setting. Whenever the measurement range or mode is changed, a new balancing process is necessary. The balance values are saved internally and are re-activated once the system returns to the associated setting.

For instance, if the measurement range is changed, the balance value applicable for the current range setting is saved internally, but cleared from the settings for the sake of the new measurement range. If there is already a balance value pertaining to the new measurement range, that value is loaded upon performing the preparation procedure

Example

Automatic restoration of the balance value upon changing the measurement range

Measurement range: ±5V
Value applied: 1V
Balancing performed -> shifted measurement range is -6 .. 4V

Measurement range changed to ±2.5V -> This clears the balance value
Value applied: 2V
Balancing performed -> shifted measurement range is -4.5 .. 0.5V

Measurement range changed to ±5V -> This clears the balance value
Prepare -> shifted measurement range is "-6 .. 4V", which matches the case of the first balance value.

Note

The results are saved with the experiment. Previous results are only deleted once a new adjustment or manufacturer's calibration has been performed.

FAQ

Question: Can I perform balancing also during a running measurement?

Answer: Yes. It is possible to perform taring or bridge balancing which a measurement is running. Stopping the measurement is not necessary.

Upon starting the next measurement after a bridge balancing action, the "*Current value*" and the "*Level indicator*" automatically adopt the channel's value range boundaries. If the balancing is performed during the measurement, the element does not display the correct value level since it only knows the previous range boundaries. The indicated measurement range in the file also retains the previous state.

Question: When is the balance OK?

Answer: Balance OK means that the balancing or taring has been performed successfully. With taring, the input range has been adapted accordingly and the procedure is complete. Successful balance is achieved once the remaining offset lies within the guaranteed tolerances.

Correspondingly, there are various error message which can be returned.

- The connected bridge is unstable, i.e. the unbalance initially found is replaced by another, not eliminated.
- The connected bridge's initial unbalance is too high. How much of an initial unbalance can be compensated depends on the amplifier type (UNI-8, BR-4, etc.) and input range and is stated in the corresponding spec sheets.

Balancing is performed independently by the amplifier card, which receives the command to perform the balance on the signals which are connected to it. A balancing value is calculated on the basis of the measured value (DAC-settings with BR4 or DCB4; or 24Bit ADC value with Uni8 and similar systems). Subsequently, a new measurement is made using this value, which either confirms the previously calculated value or, with DAC-controlled balancing, causes an additional iteration of the process.

Question: Are the balancing values saved with an experiment transferred when it is opened on a different device 281?

Answer: When opening an experiment which is transferred to a different device, the balancing values saved with the experiment are also retained. Be aware that the balancing values are subject to some imprecision because they result from the balancing of the bridge and the tolerance of the bridge completion in the amplifier. This means that in the case of half or quarter bridges, an additional balancing is required to achieve high precision.

Question: Can balance values be exported as a parameter set and then imported again later?

Answer: If the exported channel's amplifier type and all relevant amplifier settings (bridge mode, measurement range, 16- or 24-bit resolution...) match those of the imported channel, then export and import are categorically possible. With **taring**, the balance value corrects the measured vale and is thus not critical. With **bridge balancing**, there is an additional issue to consider since the balance values result both from the bridge balance adjustment process and from the tolerance of the bridge completion in the amplifier. For more on this topic, you can read the section: "*Import parameters* 1175".

Automatic balancing upon Diskstart/Autostart or using imc Online FAMOS

Activate the property "*Balance at device startup*" (Setup page: "*Channel balance*" or "*Analog channels*") for the desired channels. If you wish, you can also display the parameter of the same name as an additional column by using the "*Column chooser* 248".

Parameter Category: Channel	Description *long name*	*short name*	*column ID*
Balance at device startup	*Balance at device startup*	*Balance at startup*	*eBalanceAtDeviceStart*
	This parameter has two purposes: • Before a Diskstart/Autostart measurement starts, **balancing** of the selected channels is performed. • The channels selected can be balanced using the **imc Online FAMOS function** `RunAutoBalance` 1008. This balancing operation starts automatically before every Diskstart-measurement. The type of balancing performed is set on the following page: Setup page: "*Channel balance*" > "*Balance* 377". Available options: Taring or bridge balancing (depending on the hardware).		

Query wiring

With this function, measurements of the Sense lines are taken and the circuit configuration is derived from the resulting data. In such cases, the hardware is unable to distinguish whether a circuit without Sense lines is involved or whether a bridge was accidentally not connected or connected incorrectly. Thus, this function is not intended for detecting the circuit but for verifying whether the desired circuit was correctly recognized by the hardware.

Note

With the quarter bridge, the hardware is unable to distinguish between 5- and 6-wire measurement

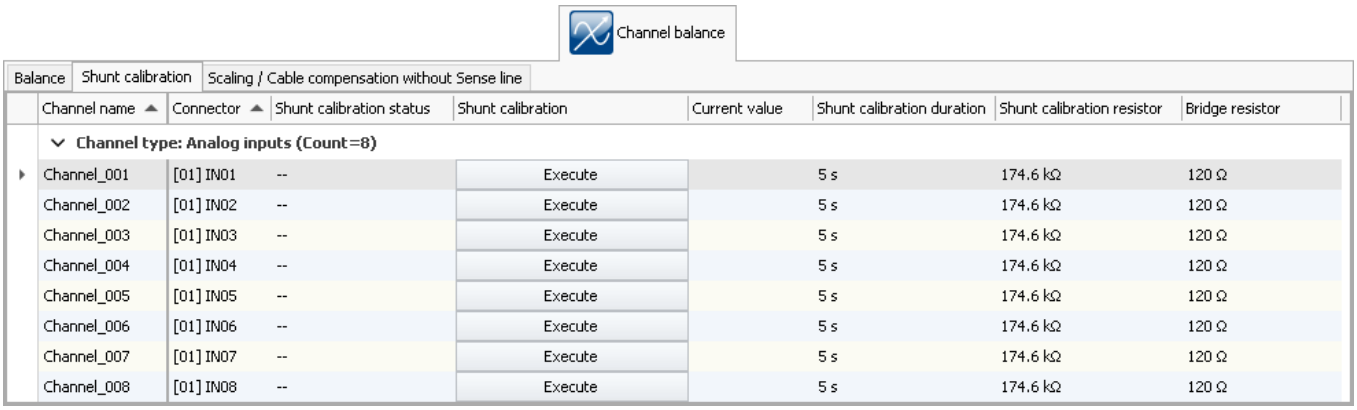
Resetting the charge

For charge amplifiers (coupling = "Charge/DC"), as well as for audio amplifiers with peak evaluation (filter type = "Peak"), it is periodically necessary to carry out a reset.

In the column Reset, the button *Execute* becomes active if this function is supported by the selected amplifier channel. This button manually resets the channel.

9.4.2 Shunt calibration

This page enables **shunt calibration** of amplifier channels.



When a shunt calibration is carried out, the bridge is unbalanced for a specified time by means of a shunt, a resistor connected in parallel. The shunts available depend on the amplifier type involved.

A shunt calibration can be carried out at any time during a running measurement.

The duration to be set is the minimum duration. It is much longer and depends on the system's load.

Note

The shunt calibration is to be considered as only a **function test**. Since the shunt is located within the device, the results are falsified by the resistance of the leads used. Precisely this fact is exploited, conversely, in cable compensation without a Sense line. The deviation measured is used to calculate the cable resistance.

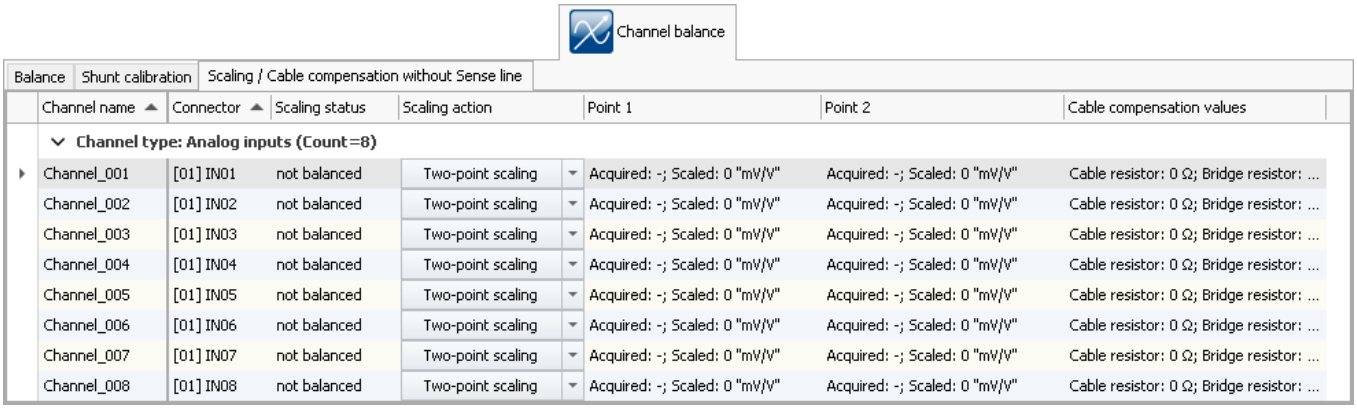
9.4.3 Run balance or shunt calibration

Besides operation by means of the device software, it is also possible to balance or shunt calibration as follows:

- By means of the commando: "*Execute device action* 1741"
- Following a device's Diskstart/Autostart 207 (only balance)
- imc Display: By means of a button on the "imc Display" which is linked to the Adjustment/Shunt calibration 780.
- imc Display: By means of the imc Online FAMOS function `DisplaySetButton` 931, which carries out an balance/shunt calibration if the button is linked with this function 780.

9.4.4 Scaling / Cable compensation without sense line

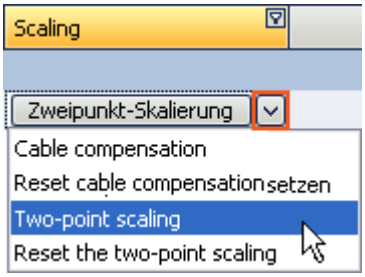
This page enables **Two-point scaling** 382 and **Cable compensation without Sense line** 384 to be performed.



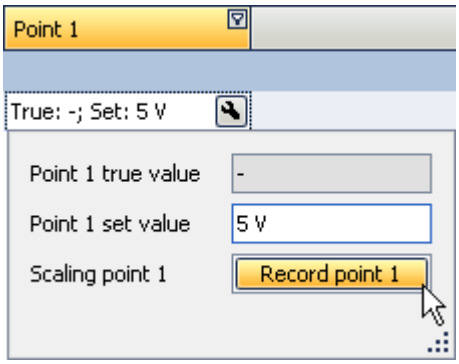
Two-point scaling

Two-point scaling allows linearization of the input range between two defined points.

To perform two-point scaling, first select the channel and in the column *scaling*, using the button at right, select *Two-point scaling*.

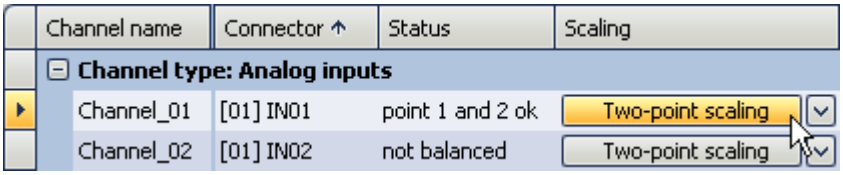


In the columns *Point 1/2*, set the respective target values. Record Point 1 and Point 2 in succession.



The status of the procedure is indicated in the column *Status*.

Once Points 1 and 2 have been successfully specified, the actual scaling must be performed. To do this, click on the button in the column *scaling*.



Note

- If bridge balancing or bridge taring is carried out after two-point calibration, any offset present is canceled, but the slope between the two points remains intact.
- The points must be separated by at least 1% of the input range. If it is necessary for technical reasons to perform the adjustment over a small input range, there is a problem since upon return to larger input range the prevalent two-point scaling will be rejected. In this case, it will be necessary to resort to the two-point scaling on the dialog Range & Scaling 351's page *Channels (Analog or Digital)*. A description is provided in "*Information and Tips* 760".
- The two-point scaling described here is performed by the amplifier module by measuring the signals applied. These signals are passed on to the base system. The base system itself is also able to perform an additional Two-point scaling by manual input 351 by means of manual input. If both procedures are used, they are connected in series, meaning that the scaled valued from the amplifier become the input values of the base system's subsequent two-point scaling.

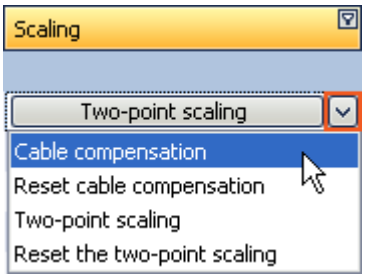
Resetting two-point scaling

To reset a two-point scale, in the column *scaling* use the button at right, *Reset the two-point scaling*. The results in the column *Point 1/2* are then no longer displayed. The two-point scaling is only deleted upon clicking on the button.

Cable compensation without sense line

If the bridge resistor is know, it's possible to determine the cable resistance, even or in fact especially if the Sense line isn't connected. Toward this end, a shunt calibration is performed. The deviation from the expected unbalance is used to determine the cable resistance.

In order to perform cable compensation, in the column *Scaling* use the button at right to select *Cable compensation*.



Set the bridge resistance if still needed.

To start the compensation, click on the button in the column *Scaling*.

After performing the cable compensation, the lie resistance values found will be displayed in the column *Cable compensation values*.

Note

- A bridge circuit with a SENSE-line is always preferable to cable compensation without a SENSE-line, in terms of the precision of the measurement results!
- For the cable compensation, the general rule of thumb in measurement engineering applies: select an input range so that the maximum expected input value reaches the top third of the input range. Otherwise, the cable compensation will not work reliably.

Resetting the cable compensation

To reset cable compensation, in the column Scaling select the button at right, Reset Cable Compensation. Only once the button has been pressed is the cable compensation deleted.

9.5 Trigger and events

A complex trigger machine is implemented in the measurement device. This chapter describes the detailed handling of the trigger options of imc measurement devices.

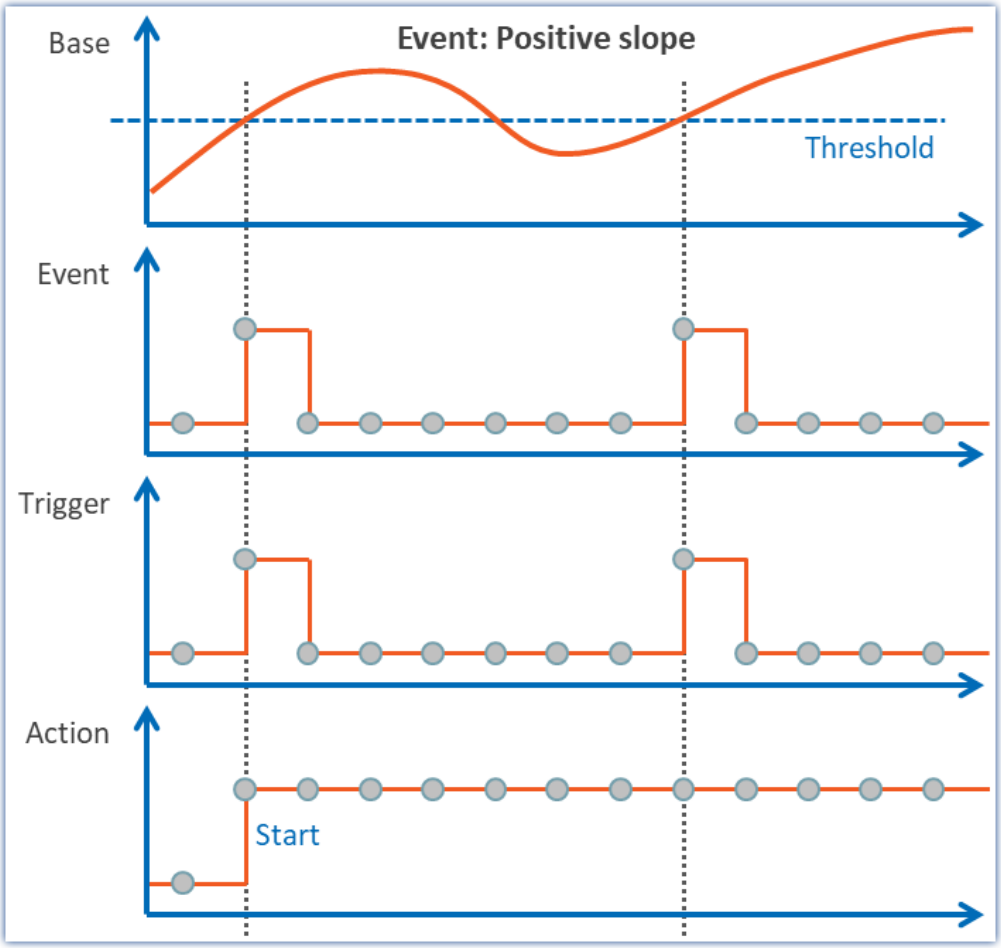
Many kinds of measurements are intended to start only once certain events occur. This applies, for instance, to measurements for which it is known in advance when measurement is to take place.

An example would be measurements of a bridge's vibration when heavy trucks drive over it. In such a case the vibration amplitude is greater than in response to a passing car, so the start of the measurement can be triggered by a specific vibration amplitude caused by a heavy truck.

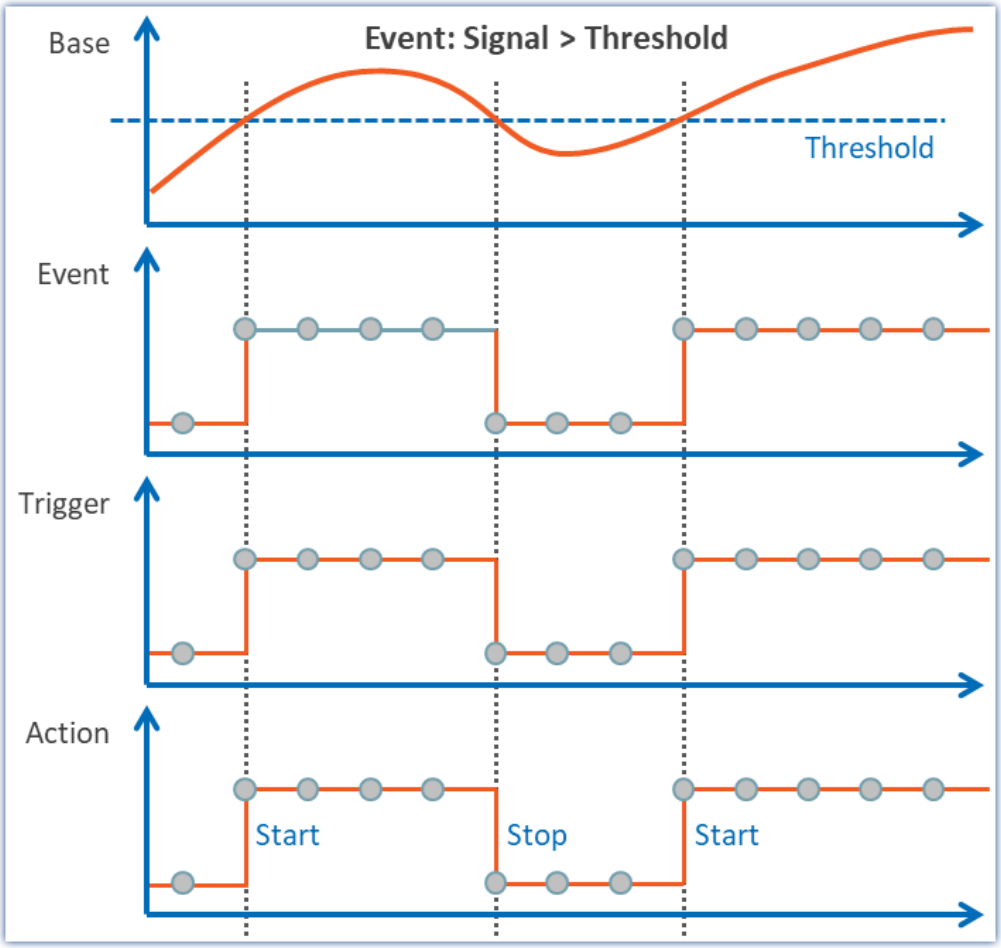
It is possible to continually monitor measurement channels and only to capture data in response to certain events. For this purpose triggers are used which start and stop channels based on the occurrence of certain events.

As soon as a certain event occurs in a signal (e.g. a particular level is exceeded > positive edge), the associated event channel's value becomes 1. This state remains in force for an increment of time. The trigger linked to the event is simultaneously triggered and the associated action is carried out.

Observe the two examples below:



An event releases a trigger and initiates an action
Example: Positive slope



An event releases a trigger and initiates an action
Example: Signal > Threshold

9.5.1 Overview

To begin, the "untriggered" measurement are explained. Afterwards, a description of the imc devices' Trigger-Machine 387 is presented:

- Starting and stopping by trigger. What to make note of (see: Sources and events 389)
- Special aspects of repeat measurements (see: Multiple triggering (Multi-Shot operation) 400)
- Triggering with virtual channels (see: Virtual Channels from imc Online FAMOS 404)
- Use of triggers in imc Online FAMOS (see: Online-Trigger 405)
- Slow background measurement with rapid triggering in the foreground (see: Example: Background Measurements 406)
- Event-driven setting of digital outputs (see: Example: Setting event-driven digital outputs 406)

Untriggered measurements

Untriggered measurement is started using the Start-button ( ). No settings need to be made in order to do this. Recording of the data begins immediately. When measurement concludes is determined by the channel having the longest measurement duration. If this measurement is "*undefined*", close the measurement by means of the Stop-button ( ).

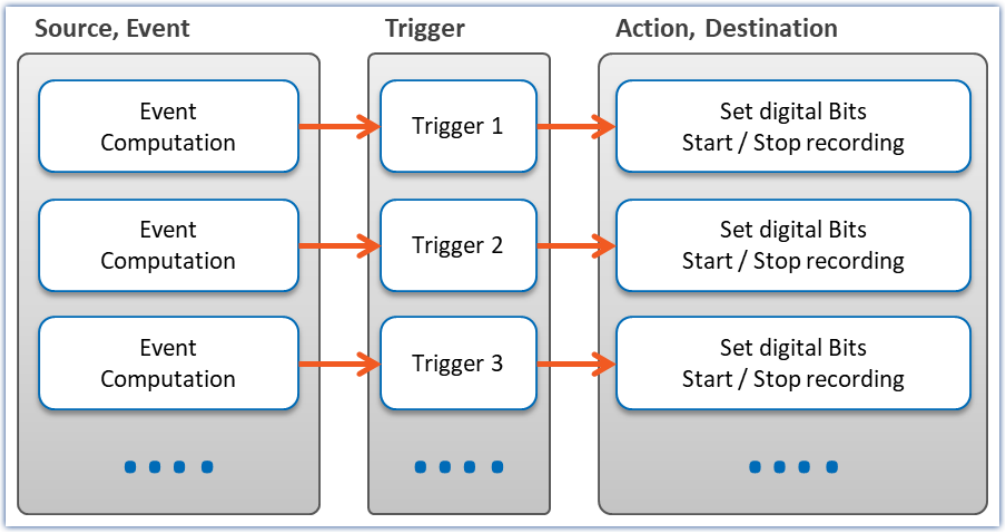
Note

Direct starting of the measurement is performed internally by an *"immediate trigger"* 397.
All channels which are not assigned to any other trigger start with the internal "*BaseTrigger*" (an "*immediate trigger*"). For more on this topic, see the description under "*Trigger Machine* 387".

9.5.2 Trigger machine

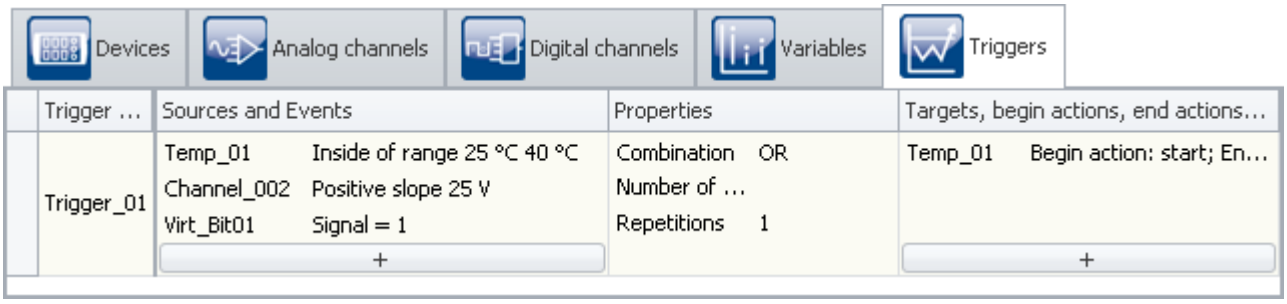
The measurement device implements a complex trigger machine which is capable of accomplishing difficult assignments. It has the following particular features:

- With the trigger machine, triggers can each be released independently (the **count** of triggers **depends on the device type**; refer to the "*Technical Specs*")
- Individual channels can be started and stopped in response to events
- Digital inputs can govern triggers
- Events can set digital outputs
- Virtual bits can be used for internal linkages
- Events can be combined with AND or OR operations
- The trigger machine works in real-time



Structure of the trigger machine

The relevant settings are made on the Triggers page of the dialog.



The Trigger dialog page

Note Limitation - CRFX/CRXT

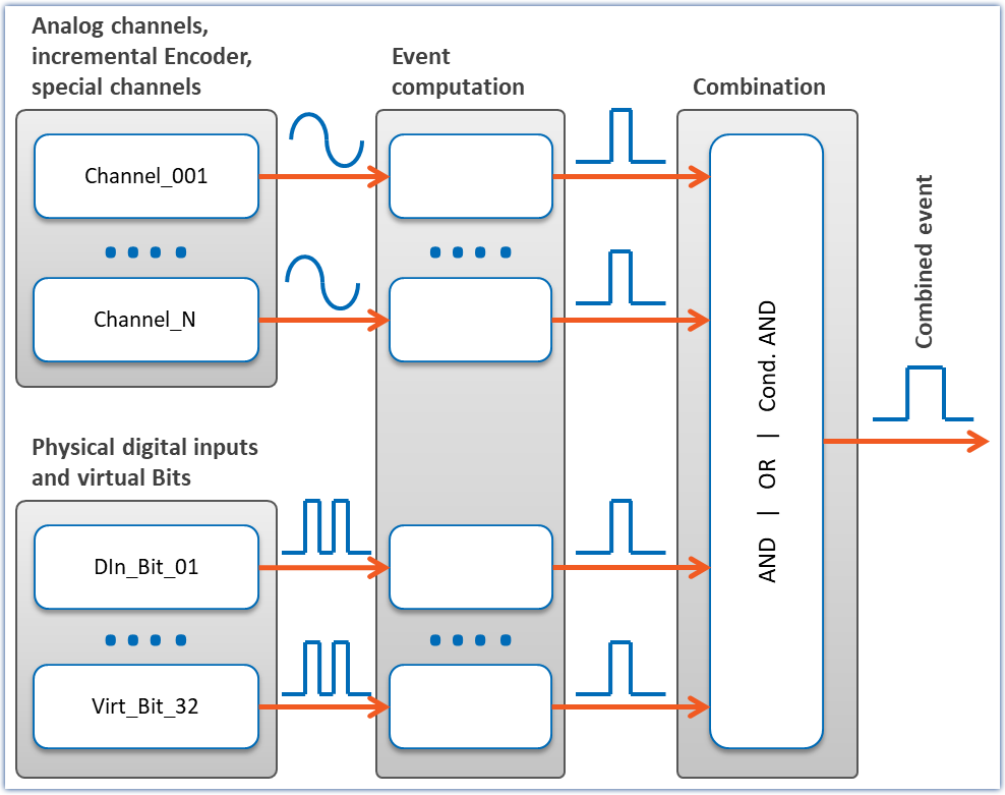
In CRFX/CRXT modules up to 8 trigger events per conditioner can be calculated.

9.5.2.1 Definitions

Term	Definitions
Event 389	An Event is a defined course or state of a measured signal. It is a binary unit of information, so it can only be either "*true*" or "*false*". Analog or digital variables can generate events.
Action 397	An Action can be the start or stop of data capture, or setting of a digital output.
Trigger	The Trigger is a defined combination of events in response to which an action is performed.
Hysteresis 395	If an channel triggers the start and end of the measurement by exceeding the threshold, the hysteresis ensures that the noise component on the signal does not end the measurement again immediately.
Dwell time 394	A Dwell time is a technique for artificially prolonging events.
Pretigger 402	A Pretigger serves to record the signal course leading up to an event.

9.5.2.2 Sources and events

On the basis of each input channel (analog, incremental counter, Fieldbus) and each digital input (bit), it is possible to calculate an event. Multiple events can be joined to form a combined event. Calculation of the combined event is illustrated in the following outline:

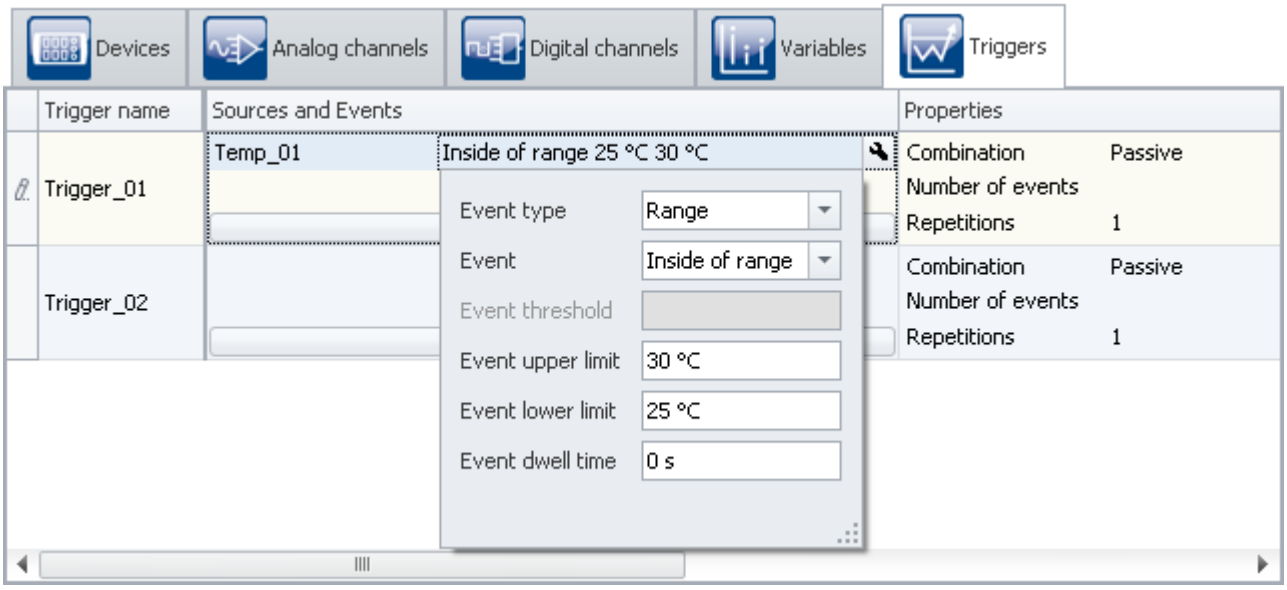


Event computation

Calculation of events from analog variables

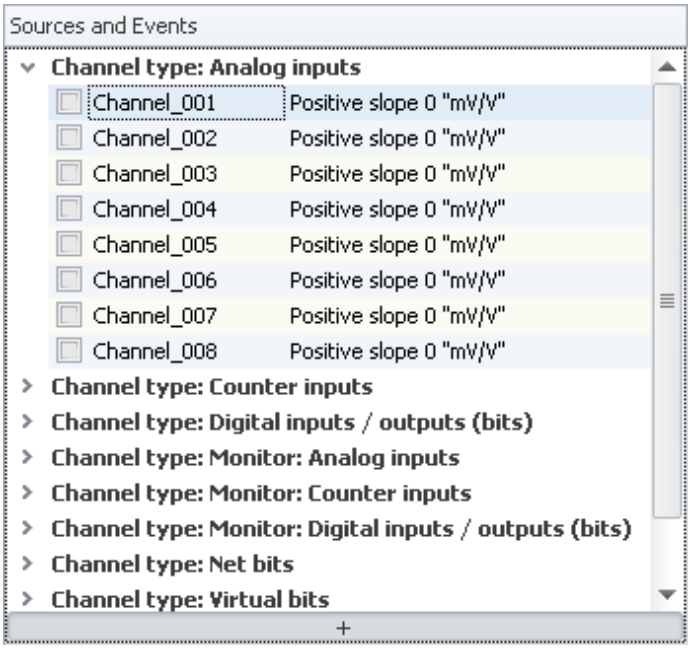
After pre-processing (correction by characteristic curve, scaling, averaging or formation of the RMS), the analog signal's physical value is entered into the event computation, which determines whether an event has occurred, according to the event type set.

The definition is made on the Setup page: "*Triggers*":



Example: The event occurs when the signal is within the boundaries 25 and 30 °C

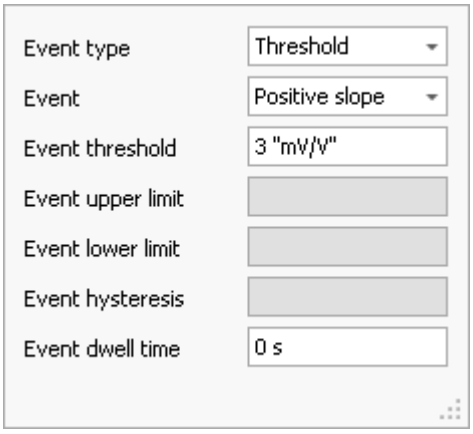
Here, the channels (sources) and conditions (events) for releasing the triggers are set. Clicking the button + takes you to a selection dialog for the sources.



Selecting a channel

Channels having the preprocessing "*Envelope*" can not serve as a trigger source.

At left are shown the available channels and at right the events set. To select a channel, click in the checkbox in front of the channel name. To configure the events, click on the event's entry, which opens a menu for setting the event.



Setting an event

The "*Event hysteresis* 395" can only be set for a device belonging to the firmware group B 187 (imc DEVICEcore).

The available choices for "*Event type* 392" are "*Threshold*" and "*Range*". The specific event is selected next.

9.5.2.3 Event types

Event type: Threshold

The simplest and most common case is the event type "**threshold value**". In the box "Dwell time 394", you specify how long the event remains in effect once the condition is no longer met.

You can select the following events and their thresholds:

Event	Description
Positive slope	If the monitored channel's signal crosses the threshold from below, the event has occurred. This can theoretically happen very briefly, but in practical terms, it lasts for the space of one sampling interval. Note that in spite of the brevity of the event, a trigger may be released! This means it is even possible to set a digital output this way! In this case, the use of a "Dwell time 394" is frequently helpful.
Negative slope	The event has occurred if the monitored channel's signal crosses the threshold from above. Like "*Positive slope*", only in the reverse direction.
Signal>Level	As soon as the monitored channel's signal level is higher than the threshold, the event has occurred. This can be for a brief moment or a longer period, depending on the course of the signal.
Signal<Level	As soon as the monitored channel's signal level is higher than the threshold, the event has occurred. Like "*Signal > Threshold*", only in the reverse direction.

Event type: Range

The user sets an amplitude range in which the event is defined as operative. The range is determined by a lower and an upper boundary. In the box "Dwell time 394", you specify how long the event remains in effect once the condition is no longer met.

The following events and boundaries are available for selection:

Event	Description
Entering range	If the monitored channel enters the range, whether from above or below, the event is defined to have occurred. This can theoretically happen for only a brief instant, but as a practical matter, lasts for at least one sampling interval duration. Note that in spite of the brevity of the event, a trigger may be released! This means it is even possible to set a digital output this way! In this case, the use of a "Dwell time 394" is frequently helpful.
Exiting range	If the monitored channel's signal exits the range, whether upwards or downwards, the event is triggered. Like with "*Entering range*", only in the reverse direction.
Inside of range	As long as the monitored channel's signal level is within the range, the event is in effect. This can be for a brief moment or a longer period, depending on the course of the signal.
Outside of range	As long as the monitored channel's signal level is outside of the range, the event is in effect. Like "*Inside of range*", only in the reverse direction.

Events from digital signals and virtual bits

Digital signals come either from the device's digital inputs or the virtual bits. The 32 virtual bits (which amount to a 32-bit register), can also be read like digital inputs. For bits, there is no event type. In the box "Dwell time 394", you specify how long the event remains in effect once the condition is no longer met.

The following events are available for selection:

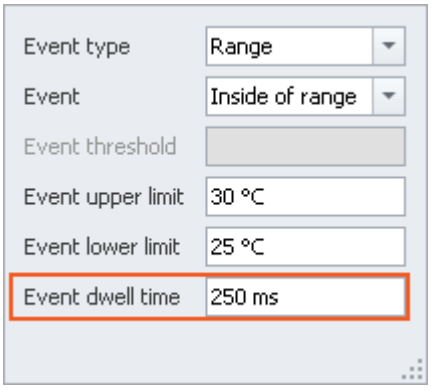
Event	Description
Signal = 1	If the monitored bit's value is *1*, the event is defined to have occurred. This can happen for either a brief instant or for a very long period, depending on the course of the measured signal.
Signal = 0	If the monitored bit's value is *0*, the event is defined to have occurred. The opposite of "*Signal=1*".
Signal change: 1 -> 0	If the monitored bit's value changes from *1* to *0*, the event is defined to have occurred. This can theoretically happen for only a brief instant, but as a practical matter, lasts for at least one sampling interval duration. Note that in spite of the brevity of the event, a trigger may be released! This means it is even possible to set a digital output this way! In this case, the use of a "Dwell time 394" is frequently helpful.
Signal change: 0 -> 1	If the monitored bit's value changes from *0* to *1*, the event is defined to have occurred. The opposite of "*Signal change 1 -> 0*".

Note

For technical reasons, calculation of the trigger occurs only after 2 sampling intervals of the slowest channel. For instance, if the slowest channel is sampled at a rate of 1Hz, it means that any change of bit values is taken into account only after 2 seconds.

9.5.2.4 Dwell time: Artificially prolonging events

With events which remain in effect only for brief periods (signal edges, level-crossing), it is often necessary to prolong the event artificially by defining a "*Dwell time*" > 0.



Einstellung der Haltezeit

maximum dwell time	
@ sampling rate	**dwell time in hours**
100 kHz	5
50 kHz	10
20 kHz	25
10 kHz	50
5 kHz	100
2 kHz	200
1 kHz and lower	500

Every event is prolonged by the duration of the Dwell time. By this means, even a very brief event can be retained for a defined duration.

Example

Example 1

A digital output is activated for an appreciable amount of time, even though the signal transition is very brief. A connected LED stays shining for a certain period instead of flashing for an instant.

Example 2

A measurement could be programmed to start not immediately upon detection of a signal edge, but rather following a pre-defined delay. This can be achieved, for instance, by making the signal edge first set a bit after a Dwell time and then set it back. The virtual bit's later transition from "*1*" to "*0*" is another event which is used for triggering the measurement's start.

9.5.2.5 Hysteresis

The event hysteresis can only be set for a device belonging to the firmware group B 187 (imc DEVICEcore).

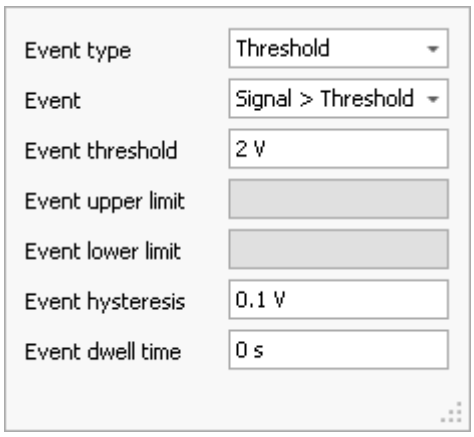
If an channel triggers the **start and end** of the measurement by exceeding the threshold, the hysteresis ensures that the noise component on the signal does not end the measurement again immediately.

"*Signal > threshold*": The recordings starts when the threshold is exceeded, but is stopped only at `threshold - hysteresis`. With "*Signal < threshold*", the recording starts accordingly when the signal falls below the threshold and stops when the signal exceeds `threshold + hysteresis`.

The default setting of the hysteresis (auto) is 0.5% of the measuring range (maximum value-minimum value).

Example Hysteresis

The settings shown start recording when the signal exceeds 2 V. The recording is stopped when the signal falls below 1.9 V (2 V- 0.1 V).



Setting an event

Automatic adjustment of the hysteresis to the measurement range

When the hysteresis is set to "*auto*", the following definition applies:

- "Auto" means: 0.5 % of the measurement range. Thus, changes to the measurement range affect the hysteresis.
- Exception: Temperature mode: Here, the hysteresis is 2 °C (3.6 °F) regardless of the measurement range.

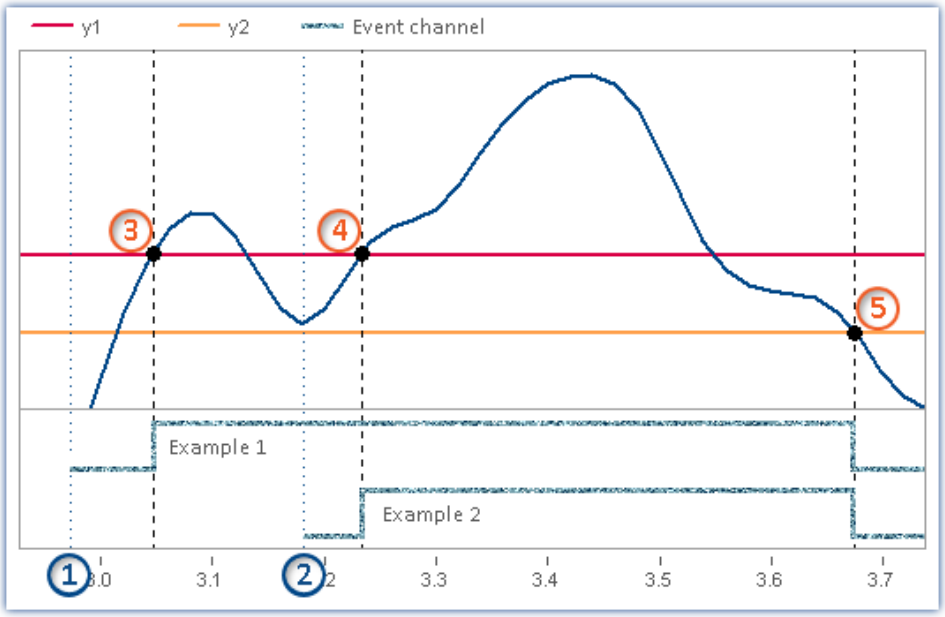
Influence of the hysteresis on the measurement's start

The test of whether an event occurs begins immediately upon conclusion of the action "*Prepare*", irrespective of whether the measurement is running. After the measurement, the testing procedure continues to run for the benefit of any subsequent measurement.

In conjunction with the hysteresis, the following may happen: An event occurring **before start of the measurement** can cause data recording on the channel to start, even though the reading is below the threshold at the measurement's starting time. Reason: before the start, a threshold violation was detected, and the reading is located within the hysteresis band without having passed below it.

The smaller the hysteresis, the less likely that this will happen.

Example — Influence of the hysteresis on the measurement's start



y1: Threshold value
y2: End of the hysteresis band

Example 1:

At the point in time ①, the action "*Prepare*" is concluded. At this moment, the system begins testing for occurrence of an event.

At the point in time ③, the threshold violation is detected (internal event channel → "true"). If the measurement would start now, data recording on the channel would begin immediately.

If the measurement starts at the point in time ②, the reading is below the threshold value. But data recording on the channel begins immediately anyway. Reason: After violating the threshold at the point in time ③, the signal has not yet passed back below the hysteresis band (internal event channel value remains "true"). Data recording only stops once the signal has passed below the hysteresis band at the point in time ⑤.

Example 2:

The action "*Prepare*" is concluded at the point in time ②. The measurement is started before time ④.

In this case, as in the previous example, the reading lies within the hysteresis band. However, data recording does not begin yet. Reason: the preceding threshold violation was not detected because it occurred before the Prepare action. Data recording only starts at the point in time ④ (internal event channel → "true") and then stops at the point in time ⑤.

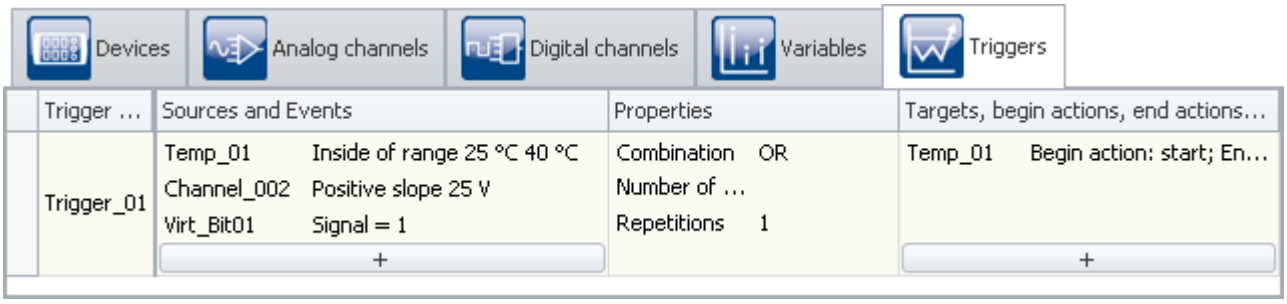
9.5.2.6 Combinations of events

It is possible to join multiple triggers in a logical expression. Five operators are available for the purpose:

Combinations	Description
AND	If all events are "*true*" at the same time, the combined event is "*true*" and the specified actions are carried out. For this reason it may not be sensible to combine two signal edge events with AND, since it is unlikely that two signals would have edges at the same time, unless the events were artificially prolonged by means of appropriate "*dwell time* 394".
OR	As soon as one of the events is "*true*", the combined event is "*true*" and the associated actions are carried out.
Conditional AND	If a certain number of events ("*Repetitions* 400") are "*true*" at the same time, the combined event is "*true*" and the associated actions are carried out.
Immediate	At the start of the measurement, the actions are performed. For this reason, no triggering event can be assigned. "*BaseTrigger*" is by default a "*Immediate trigger*", meaning it automatically starts all channels on which measurement is not subject to other triggers.
Passive	A passive (deactivated) trigger is no longer in use and automatically deletes all associated actions so that the channels can be entered for other purposes.

Setting the logical operation to combine events is done on the Setup page: "*Triggers*" in the column "*Properties*".

To change the combination, open the selection list in the column "*Properties*" and select the desired combination.



Combining events

9.5.2.7 Actions and targets

A variety of actions, triggered in a variety of ways, can be set as the target.

A trigger with the action: "**Start data recording**" can be triggered in a variety of ways:

- The combined event is "*true*"
- By means of the user interface of imc STUDIO
- By means of imc Online FAMOS

There is only one way to release a trigger with the action: "**Set bit**" and "**Stop recording of data**":

- The combined event is "*true*"/"*false*"

Action: Starting data recording

Recording of the data **can only start of the trigger is "*armed*"** and the maximum count of trigger repetitions has not yet been reached.

A trigger is armed upon starting the measurement. This means it is ready to be released.

During data recording, the trigger is in the state "*released*".

If subsequently the recording of data on all of the trigger's target channels is concluded, for instance in consequence of the measurement duration set, the trigger is re-armed. If the maximum number of trigger repetitions has not yet been reached, the trigger can be released again.

Action: Setting bits

Each transition of the state of the combined event can set a bit to "1" or "0"; the combined event can include digital outputs and virtual bits. Virtual bits are in essence a 32-bit register which can be read from and written to at random. they have no physical terminal and only serve the purpose of exchanging information as well as halting states.

Note — Independence of count and measurement

- "*Set bits*" is **independent of the setting for "*Repetition*"**. It can be actuated any number of times.
- "*Set bits*" is **independent of whether a trigger is armed**. It is actuated independently of the measurement (after Preparation).

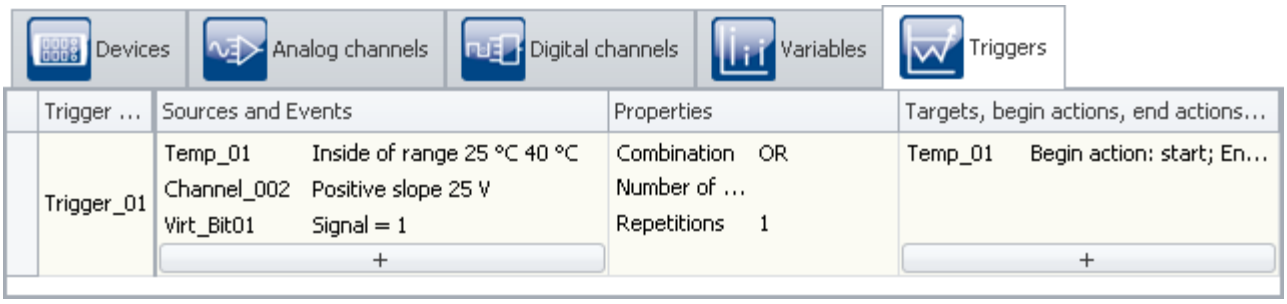
Note — Changing the value retroactively

Setting of a bit is performed one single time until the trigger is fired again. While the trigger is active, the bit's value can be changed without the trigger always resetting it, for example, by means of the PC or imc Online FAMOS. This also applies to the virtual bits and the digital output bits.

Defining actions and targets

Making settings for the targets and actions is done on the Setup page: "*Triggers*" in the column "*Targets, begin actions, end actions, pretrigger*".

Here, the targets are specified. For instance a channel which is to measured or stopped upon release of the associated trigger.

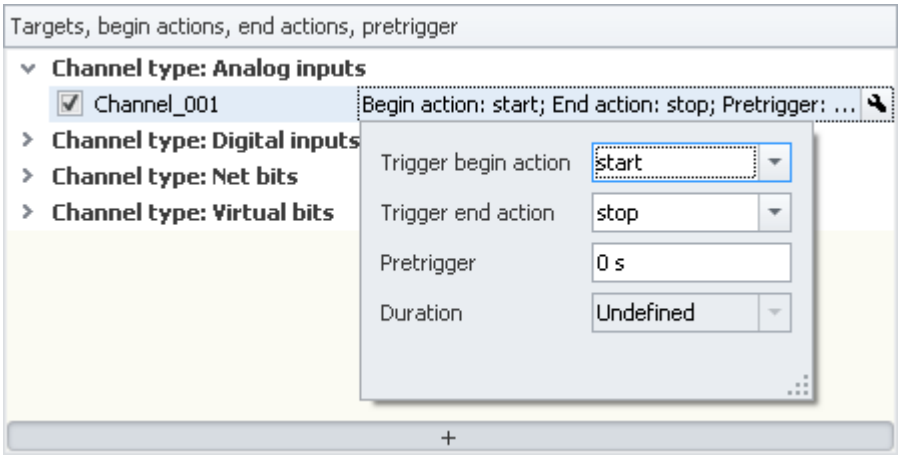


Defining targets and actions

The button + calls a dialog for selecting targets.

To select a target, click on the selection box.

To define the actions, click the left mouse button over the region at right.



Setting actions

Here, you can define the actions to be performed.

Parameter	Description
Trigger-start action	Action performed when the combined event becomes "*true*".
Trigger-end action	Action performed when the combined event becomes "*false*".
Pretrigger	A pretrigger determines the duration of data recording before the triggering time. See "*Pretrigger* 402".

Possible targets for data acquisition channels

Targets	Description
Begin action: start End action: -	If the trigger is "*armed*" and the combined event is "*true*", data recording is started.
Begin action: stop End action: -	If the combined event is "*true*", data recording is stopped.
Begin action: start End action: stop	Like "*Begin action: start*". Additionally, data recording stops is the combined event is "*false*". Thus the channel's signal is recorded as long as the combined event is "*true*".

A channel can be started and/or stopped by exactly one trigger. The two actions can be either in response to two different triggers or to the same one.

Possible targets for digital bits

Targets	Description
Begin action: Begin = 1 End action: -	If the combined event is "*true*", the bit is set to "*1*".
Begin action: - End action: End = 1	If the combined event is "*false*", the bit is set to "*0*".
Begin action: Begin = 0 End action: -	If the combined event is "*true*", the bit is set to "*0*".

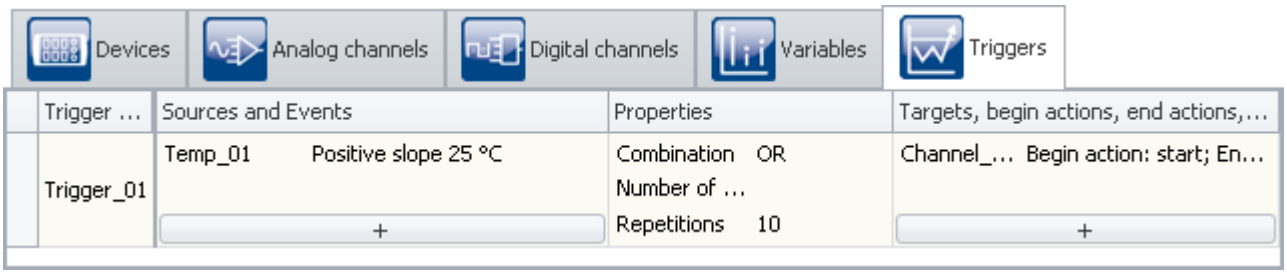
Targets	Description
Begin action: - End action: End = 0	If the combined event is "*false*", the bit is set to "*1*".
Begin action: Begin = 1 End action: End = 0	Like "*Begin action: Begin = 1*". Additionally, the bit is set to "*0*" if the combined event is "*false*". Thus, the bit is set to "*1*" as long as the combined event is "*true*".
Begin action: Begin = 0 End action: End = 1	Like "*Begin action: Begin = 0*". Additionally, the bit is set to "*1*" if the combined event is "*false*". Thus, the bit is set to *0* as long as the combined event is "*true*".

A bit can be set to 1/0 by exactly one trigger. The two actions can be either in response to two different triggers or to the same one.

9.5.2.8 Multiple triggering (multi-shot operation)

Within a measurement, it is possible for individual triggers to release multiple times.

Setting the "Repetitions" on the Setup page: "*Trigger*" in the column "*Properties*".



Setting the amount

Using the property: "***Repetitions***", you can specify the number of trigger releases. Normally, this repetitions is set to "*unlimited*". You can specify any repetitions desired.

In consequence, the trigger is rearmed after each episode of data recording until the specified number of trigger releases has been reached.

If you do not already know exactly how often you wish the trigger to release, set the number of trigger releases to "*unlimited*".

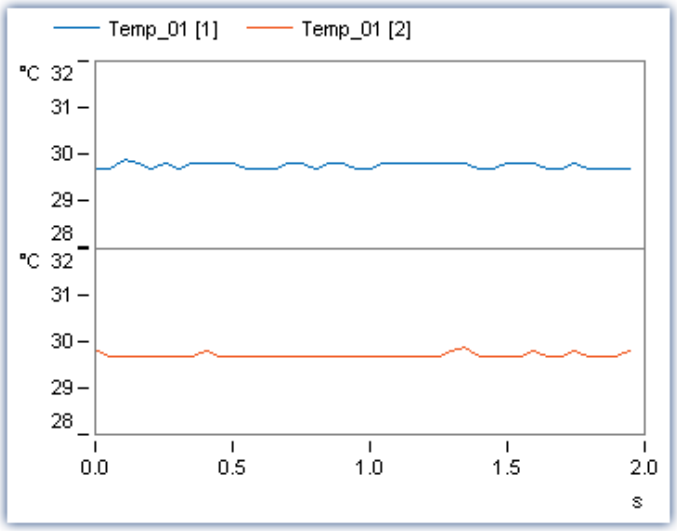
Note

- If you have set an unlimited number of trigger releases, the measurement never comes to an end by itself. Only after clicking on the Stop-button will the measurement conclude.
- The "*Repetitions*" can be set to **up to 32767 triggers**. Any larger number is automatically altered to "unlimited".
- "*Set bits*" is **independent of the setting for "*Repetition*"**. It can be actuated any number of times.
- Please note the remarks for pretriggers 402.

All data belonging to a measurement are written to a common folder when the data are saved; this also applies to channels with multiple trigger releases. These channels can be saved in a file having multiple "*Events*", or in individual files (see the documentation for the device configuration > "*Storage* 292").

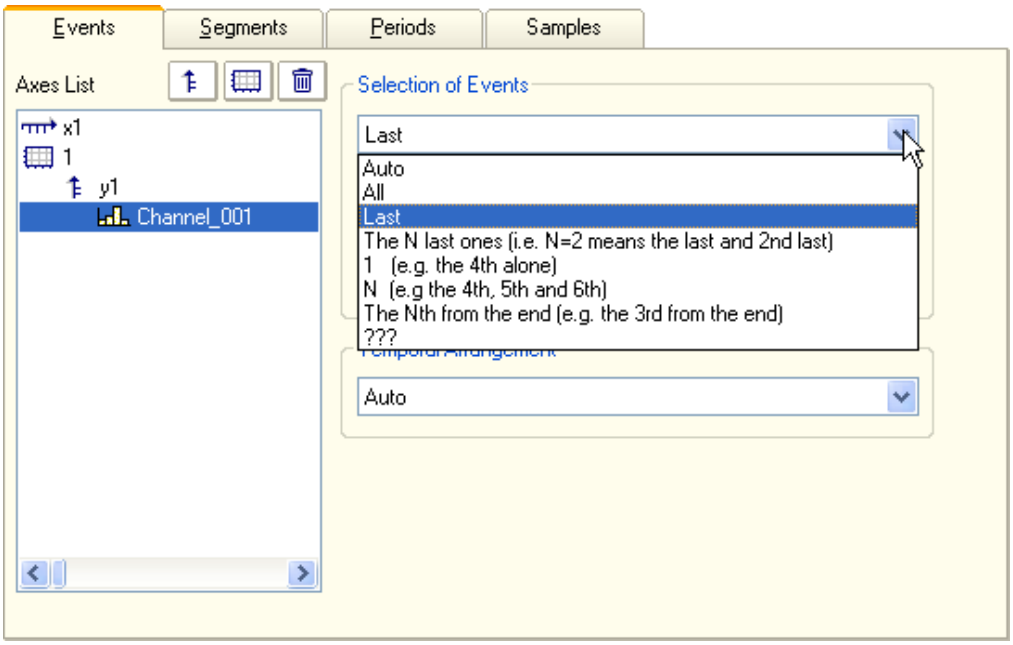
Among other things, multiple triggering enables easy, visual comparison of data from different trigger releases.

In the curve windows, you can compare the individual events:



Curve window with multiple events of the same channel

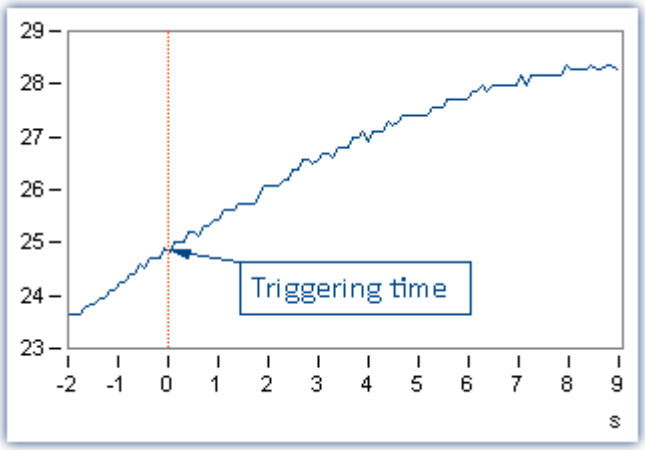
Toward this end, the curve window must be configured appropriately. Select the menu item "*Configuration*" > "*Events, segments, periods...*". On the page *Events*, you can select how many events you wish to have displayed.



9.5.2.9 Pretrigger

For every channel' signal recorded in response to a trigger, it is possible to also record the course of the signal leading up to the trigger event. A pretrigger determines the duration of recording before the trigger moment. The trigger moment's relative time is defined as 0. Pretrigger values have times < 0.

Setting the pretrigger is performed on the Setup page: "*Triggers*" in the column "*Target, begin actions, end actions, pretrigger*".



Note

- The pretrigger is defined for channels which are assigned to a defined trigger. **If one deleted a trigger assignment for a channel**, which is the same as assigning the channel to "*BaseTrigger*", **the pretrigger is deleted automatically**.
- To be able to view a signal's plot including the pretrigger region, the first measurement (after prepare) needs to have been started before the trigger event by at least as much as the pretrigger time. For the second measurement filling the pretrigger will be started after stopping the first measurement.
  This is important to keep in mind, particularly with repeated trigger processes, since in such cases, the pretrigger only fills up after the preceding trigger event is complete. The pretrigger time counts toward the measurement duration set. **If the pretrigger is not completely filled, the overall measurement duration is that much shorter.**

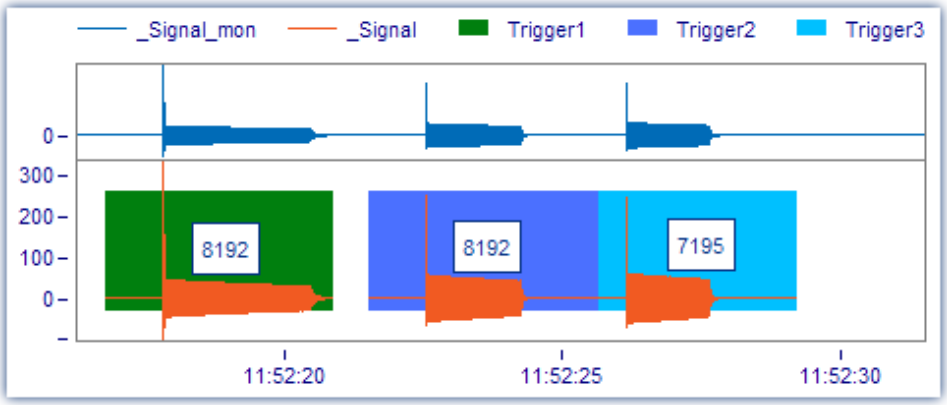
Example

Setting for the measurement duration: 10 s, Pretrigger time: 2 s

Measurement duration after trigger = (10-2)s = 8 s

If the trigger releases after 1s: Period of data capture = 1s +8s = 9s

- *Example: FFT with 8192 samples of which 1024 samples are for the pretrigger:*
  At 500 Hz sampling rate -> Measurement duration = 4.1 s, pretrigger time: 512 ms

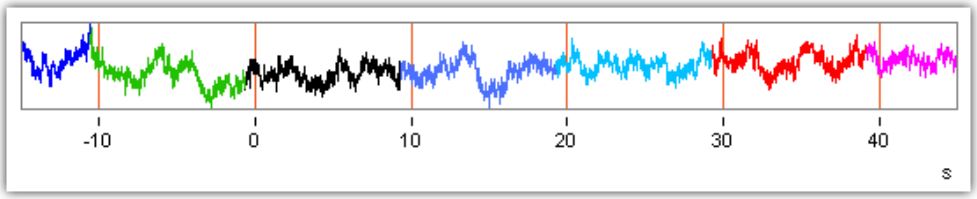


For the third event, the time was not sufficient for filling the pretrigger completely, and only 7195 samples were captured. Since the FFT needs 8192 for the calculation, it will not be possible to calculate the spectrum for this event.

Pretriggers in conjunction with intervals

If the pretrigger's measured data are located in time across interval boundaries, it is possible to treat the measured data as follows:

- The pretrigger's data can be saved together with the first interval (interval of the trigger firing) (*default setting*).
- The pretrigger's data can be cut correctly. In this way, interval folders can be created subsequently.



Example

Each color denotes one interval-folder. At "0", the trigger fired. The two colors preceding "0" are the pretrigger data. They can be saved either together with the interval of the trigger-start (black), or in their own separate interval folders.

The associated setting is made in the Options: "*Options* 128" > Area: "*Variables*" > "*Datapool*" > "*Split up pretrigger data into intervals* 136".

Note — Notes on the pretrigger preceding the measurement start

This function does not apply to measured data which were recorded in the pretrigger before the measurement's start. The pretrigger is already filled up for the second measurement, for example, before the measurement had been started. These measured data are always in the first interval-folder.

Reference See also

Saving Interval 423

9.5.3 The process of triggered measurement

With imc STUDIO, a measurement begins when the Start-button is pressed (or when the device is activated, if it is in Autostart mode). Next, all triggers are armed. Untriggered channels which are assigned to the "*BaseTrigger*", immediately start recording data.

imc STUDIO concludes measurement by itself once all component measurements specified have been performed. The measurement is over once no trigger has any outstanding repetitions to perform and all partial measurements are finished.

By clicking on the Stop-button, you can conclude a measurement before the appointed time.

Reference See also

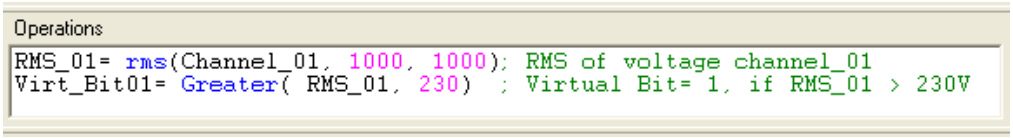
Breakdown of the Start/Stop and Triggering Behavior 761

9.5.4 Information and tips

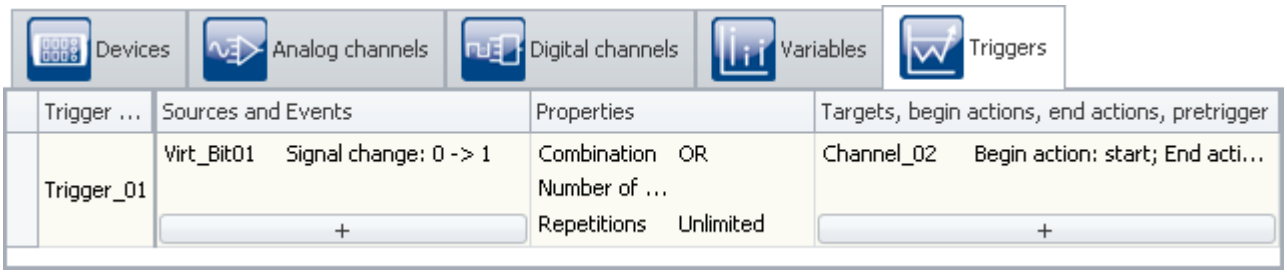
9.5.4.1 Trigger machine and imc Online FAMOS

Virtual channels from imc Online FAMOS

Virtual channels generated by imc Online FAMOS can not be used directly as events for the trigger machine. Instead a virtual channel can be checked for events using imc Online FAMOS functions. The result switches one of the virtual bits which in turn can be evaluated by the trigger machine.



imc Online FAMOS: Exceeding the RMS value of 230V sets Virtual Bit 01



Triggering of Channel_02 when the RMS values is exceeded, accomplished indirectly using Virt_Bit01

Note

It is possible to set a virtual bit both in the trigger machine and in imc Online FAMOS. To avoid this conflict, any bits set in imc Online FAMOS should never appear on the target side of the trigger machine.

Online-Trigger

imc Online FAMOS with Control Commands 855 enables targeted actions in response to particular states of the trigger: `OnTriggerStart`, `OnTriggerMeasure`, `OnTriggerEnd`.

Note

This only applies to triggers having a Start action: start.

The following applies to a trigger's control commands:

- When the trigger releases, the content of the control command `OnTriggerStart` is performed once.
- As long as the trigger is released, the content of the control command `OnTriggerMeasure` is performed continuously.
- When the triggering is finished, the content of the control command `OnTriggerEnd` is performed once.

If channels having different measurement durations are assigned to the same trigger, the triggering is completed at the end of the longest measurement.

- It is generally to be recommended that all of a trigger's physical channels are set with the same measurement duration.

9.5.4.2 Display before trigger events

Triggered channels are not displayed in a curve window before the trigger has been released. The data runs through the device as soon as the measurement has been started, but they can't be seen. If the sensor is disconnected or defective, it is not obvious why the trigger does not release.

For that reason we have the monitor channels. Those channels are sampled from the active channels. They can't be sampled faster than their original channels. But it is possible to define different or no trigger conditions to those monitor channels. So these channels can be seen before the original channels have been triggered.

Fieldbus channels

There are no monitor channels for most fieldbus channels. But they can be created with a trick. Use the Assistant to create within a channel's message an additional channel having a different name and the exact same settings. This channel can then be captured without triggering.

9.5.4.3 Trigger-variables in the Data Browser

The Data Browser contains not only a trigger's results channels but also **variables** for every **trigger** configured. These can represent, for example, **information on the trigger's status**. The trigger variables convey a variety of kinds of information, which can each be displayed separately and can easily be moved to the Panel by means of Drag&Drop.

Trigger		Description
Trigger Name e.g. Trigger_01 or BaseTrigger	Event number	With multi-triggering: count of previous trigger releases
	State	Current state of the trigger: armed, released, stopped
	Trigger time	Time when the state last changed

Note The Start-action must include a channel

The variables **only return values** if a channel is set for the trigger's **Start-action**.

9.5.5 Examples

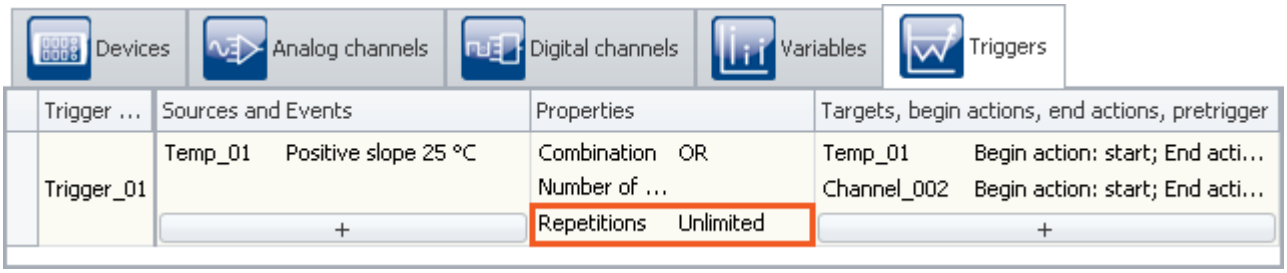
9.5.5.1 Background measurements

Two measurement tasks are to be performed in parallel by the same device. For instance, if you wished to record each activation and each braking procedure of some equipment, at high resolution. Between the recording of such measurements, there are often long pauses. Conversely, there are some signals which must continually be measured without any pause or gap in the data, e.g. temperatures and an engines RPMs.

This situation is referred to as a triggered measurement in the foreground at a high sampling rate and an untriggered background measurement at a low sampling rate.

Set the foreground measurement channel's trigger settings on the Setup page "*Trigger*" accordingly.

Also set "***Repetitions***" to "*Unlimited*".



Trigger settings for the measurement in the foreground

On the Setup page: "*Analog/Digital channels*", set the measurement duration of the background measurement channels to "*Undefined*". In this case, you must end measurement manually using the Stop-button.

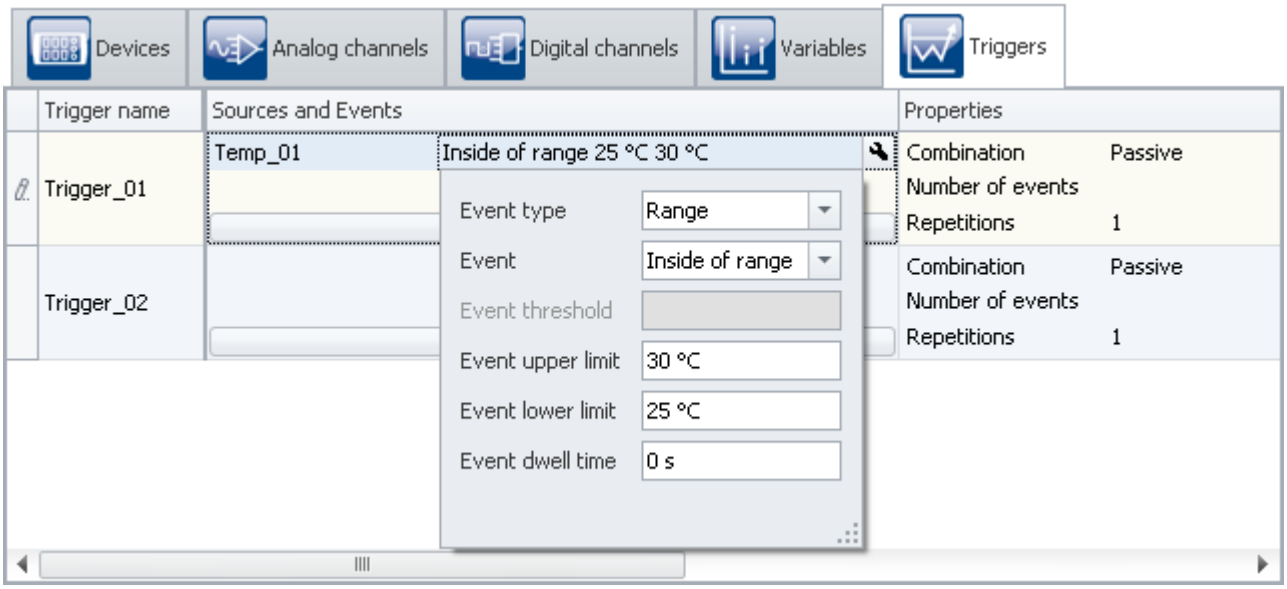
With these settings, it is possible for the trigger to release an infinite amount of times in succession during a measurement. After each concluded measurement, the trigger rearms automatically.

The measurement itself last until the background measurement is stopped by the Stop-button.

9.5.5.2 Event-controlled digital outputs

A digital output is to be set to 1 for the duration that a signal is within a specified tolerance band.

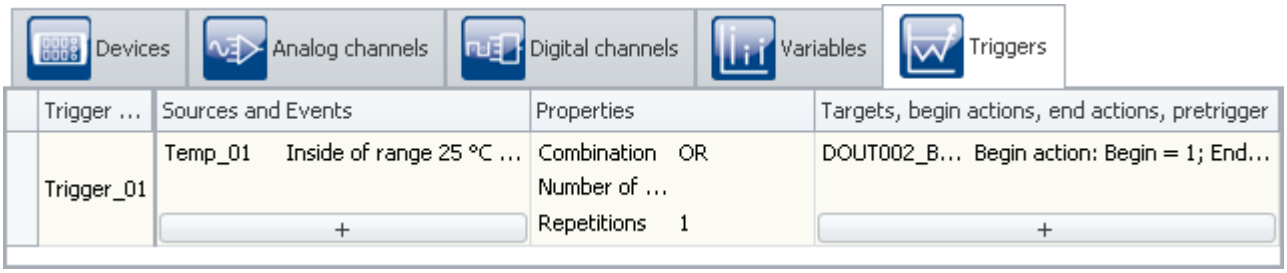
Set the channel's trigger event on the Setup page: "*Triggers*" accordingly:



Settings for the event

The event is defined to occur while the signal is within the specified range of 25 °C to 30 °C.

Next, the trigger is set.



Trigger settings for a bit output

Select as the "***source***" the channel from which the event is derived (here: "*Temp_01*").

Select as the "***target***" the desired digital output which you wish to set (here: "*DOUT002_Bit01*"). Set as the "***Begin action***": "*Begin=1*" and as the "***End action***": "*End=0*".

If the signal of the channel "*Temp_01*" is within the boundaries set while the measurement is running, the digital output bit's value is "*high*".

9.5.5.3 Start measurement by a virtual bit.

Problem: My measurement is started by the transition of a virtual bit from *0* to *1*. This only works at the first start. If I start the measurement again without re-doing the preparation, it doesn't work.

Reason: The virtual bit is set the first time and isn't set back afterwards. At the next start, the bit is already *1*, and no more transition from *0* to *1* takes place.

Solution: Once the trigger has been released, the bit must be set back again, which can also be handled at the end.

9.5.5.4 Logging triggers

- Virtual bit is the Start/Stop trigger for a channel;
- Using `RecordEvent(..)`, the trigger's starting and stopping times are to be recorded:
  RecordEvent(vrtB1_Start_K2, "Trigger K2 ON")
  RecordEvent(Not vrtB1_Start_K2, "Trigger K2 OFF")

→ the first notification to arrive is "Trigger K2 OFF"

- The bit was set = 0 upon starting.

Solution: The *RecordEvent* function only indicates transitions from 0 to 1, but the VirtBit_01 in the `OnTriggerMeasure` of Trigger_01 is 1. If one wanted to make a record of this first event, the calls of the *RecordEvent* function must be made in "*BaseTrigger*".

9.6 Additional pages

By default, only a selection of the available Setup pages is displayed. It is possible to have additional pre-configured pages shown, or to create your own*.

Among others, the following pages are available:

- TEDS 729
- Application module 409
- ...

* The availability of this function depends on your particular product license. See the technical data sheet

Displaying a page

To add a page to the user interface,

- open the context menu over a page-tab or over the empty space to the right of one.
- Click on the corresponding page entry in the list under "*Insert complete layout*".

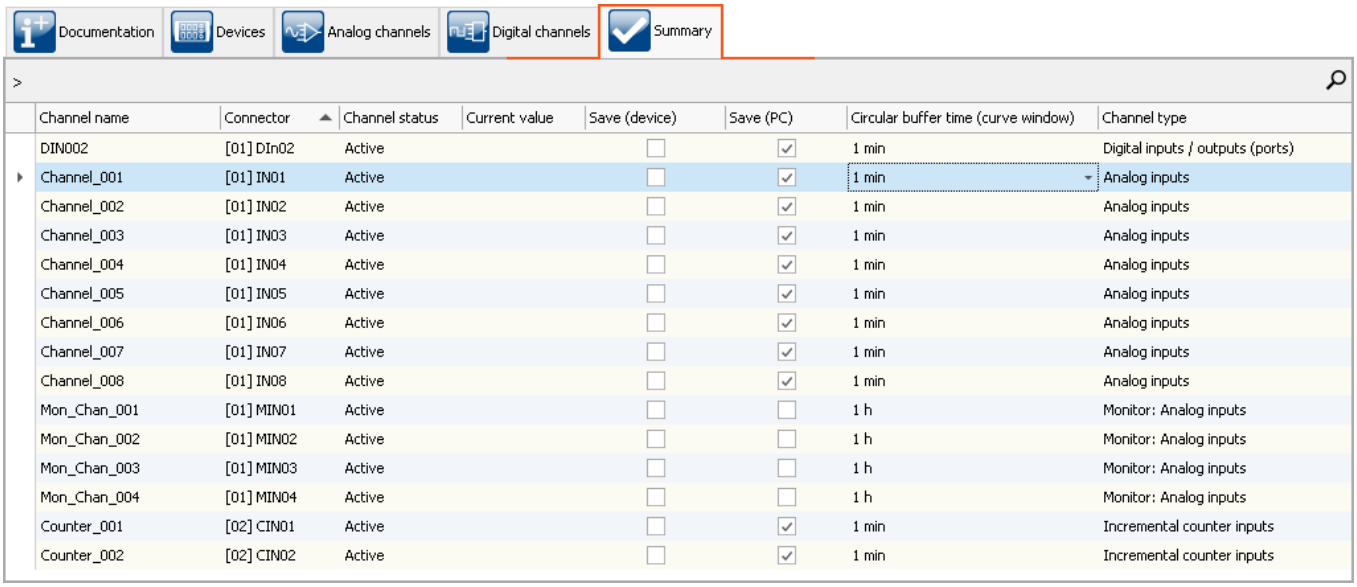
In consequence, the page is inserted to the right of the selected locations.

Note Saving views

Changes to the view, such as: "Insert Setup page", must be saved with the view in order to be visible the next time the program is started.
See the manual "*imc STUDIO (general)*" > "*Views* 155".

9.6.1 Summary

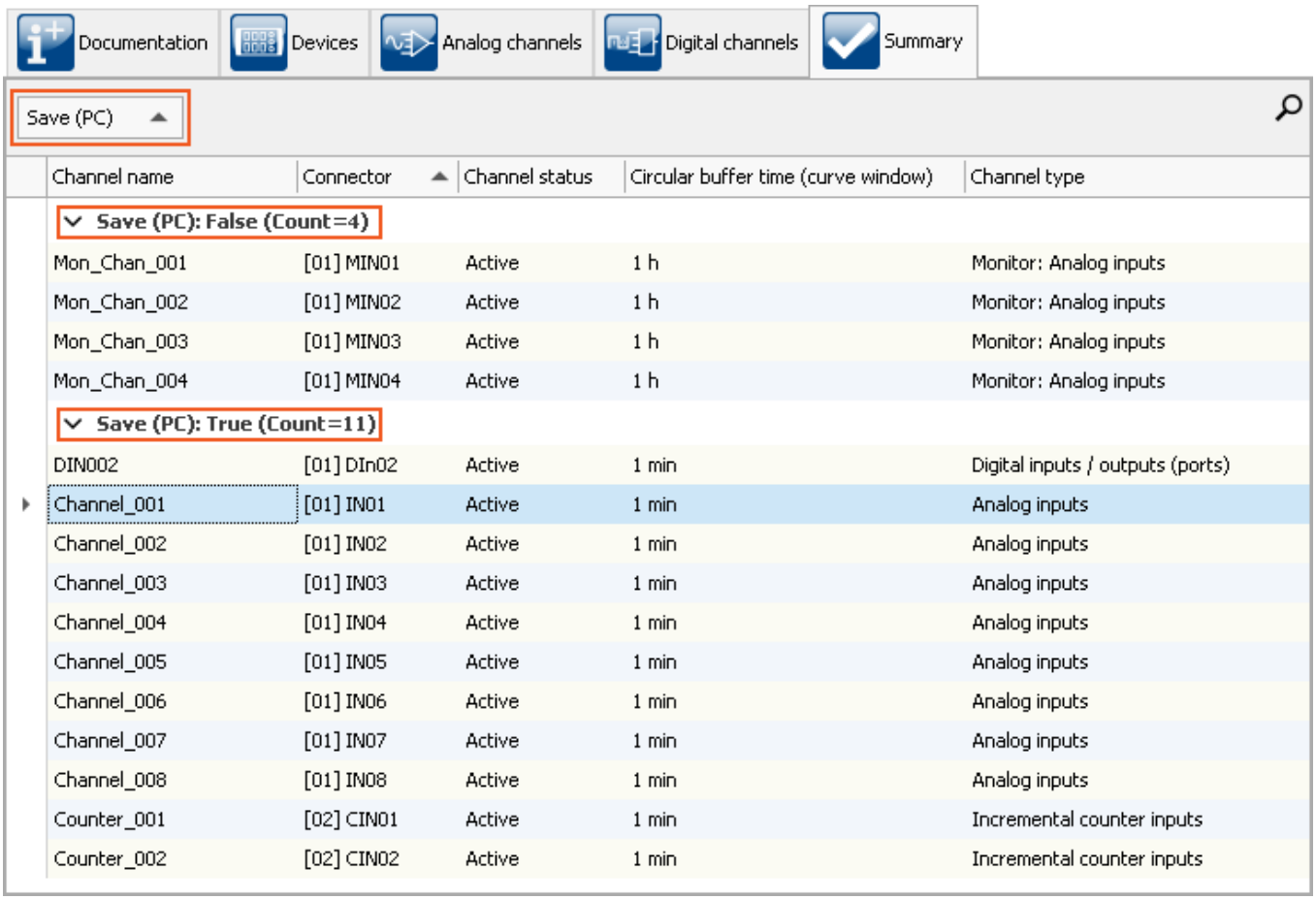
This page provides a quick overview of all active channels, the data storage status, and of the circular buffer time. Additionally, the page can be filled with any arbitrary choice of channel parameters, as accustomed.



Channel name	Connector	Channel status	Current value	Save (device)	Save (PC)	Circular buffer time (curve window)	Channel type
DIN002	[01] DIn02	Active		☐	☑	1 min	Digital inputs / outputs (ports)
Channel_001	[01] IN01	Active		☐	☑	1 min	Analog inputs
Channel_002	[01] IN02	Active		☐	☑	1 min	Analog inputs
Channel_003	[01] IN03	Active		☐	☑	1 min	Analog inputs
Channel_004	[01] IN04	Active		☐	☑	1 min	Analog inputs
Channel_005	[01] IN05	Active		☐	☑	1 min	Analog inputs
Channel_006	[01] IN06	Active		☐	☑	1 min	Analog inputs
Channel_007	[01] IN07	Active		☐	☑	1 min	Analog inputs
Channel_008	[01] IN08	Active		☐	☑	1 min	Analog inputs
Mon_Chan_001	[01] MIN01	Active		☐	☐	1 h	Monitor: Analog inputs
Mon_Chan_002	[01] MIN02	Active		☐	☐	1 h	Monitor: Analog inputs
Mon_Chan_003	[01] MIN03	Active		☐	☐	1 h	Monitor: Analog inputs
Mon_Chan_004	[01] MIN04	Active		☐	☐	1 h	Monitor: Analog inputs
Counter_001	[02] CIN01	Active		☐	☑	1 min	Incremental counter inputs
Counter_002	[02] CIN02	Active		☐	☑	1 min	Incremental counter inputs

Summary - all active channels

By means of the grouping box at the top left, it is possible to use Drag&Drop to apply grouping to individual columns. Thus for instance, it would be possible to group channels according to whether or not their data are to be saved.



Channel name	Connector	Channel status	Circular buffer time (curve window)	Channel type
Save (PC): False (Count=4)				
Mon_Chan_001	[01] MIN01	Active	1 h	Monitor: Analog inputs
Mon_Chan_002	[01] MIN02	Active	1 h	Monitor: Analog inputs
Mon_Chan_003	[01] MIN03	Active	1 h	Monitor: Analog inputs
Mon_Chan_004	[01] MIN04	Active	1 h	Monitor: Analog inputs
Save (PC): True (Count=11)				
DIN002	[01] DIn02	Active	1 min	Digital inputs / outputs (ports)
Channel_001	[01] IN01	Active	1 min	Analog inputs
Channel_002	[01] IN02	Active	1 min	Analog inputs
Channel_003	[01] IN03	Active	1 min	Analog inputs
Channel_004	[01] IN04	Active	1 min	Analog inputs
Channel_005	[01] IN05	Active	1 min	Analog inputs
Channel_006	[01] IN06	Active	1 min	Analog inputs
Channel_007	[01] IN07	Active	1 min	Analog inputs
Channel_008	[01] IN08	Active	1 min	Analog inputs
Counter_001	[02] CIN01	Active	1 min	Incremental counter inputs
Counter_002	[02] CIN02	Active	1 min	Incremental counter inputs

Summary - grouped according to data saving

9.6.2 Application module

Note The page is not available

By default, this page is not displayed. To display it, please follow the instructions in the section: "*Additional Pages* 407".

The imc Application Module serves to **integrate measurement channels from "third party" devices or systems** into an imc CRONOS*compact* respectively imc CRONOS*flex* system via standard hardware interfaces.

Examples of possible channel sources include:

- Specialized complex sensors
- "third party" devices
- Bus systems (e.g. fieldbusses)

The standard interfaces supported include, in particular:

- Ethernet
- serial interfaces (RS-232, RS-485, RS-422)

The systems to be integrated are typically user-customized or dedicated devices by third-party manufacturers. The integration is achieved by means of a standard hardware module (APPMOD), which comes with a dedicated processor for which a custom application is programmed. This program is either created by imc on commission or can also be created and implemented by qualified partners or trained users provided with specialized development tools.

This user-specific hardware and software expansion is supported by the device software (imc STUDIO). A special version of the device software is not necessary.

Characteristics:

- encapsulated, custom hardware + software solution, embedded in a standard system
- Standard system with complete software support
- Flexible support by unaltered standard operating software
- Standard hardware component
- Stand-alone, autonomous system environment

Reference Manual

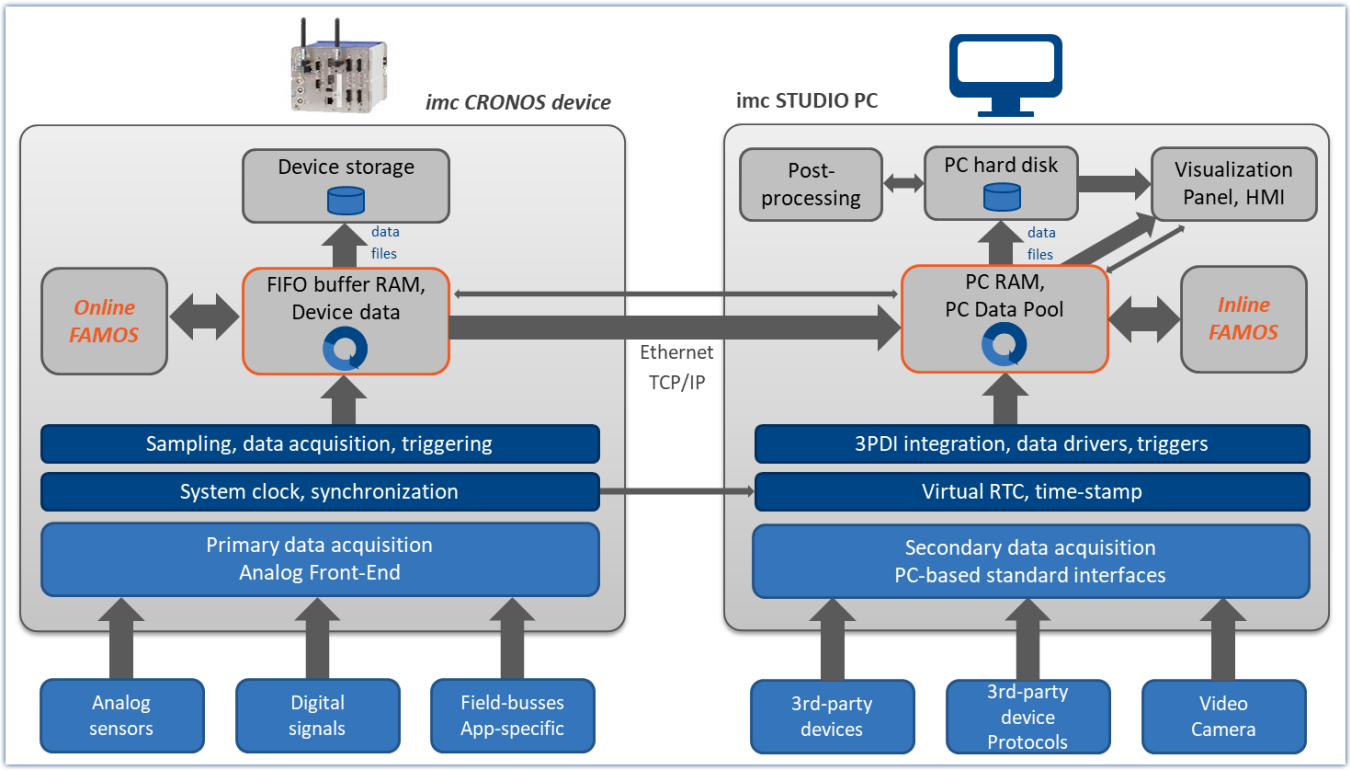
The imc Application Module describe the use and initiation of the module, such as the RS-232, RS-485, RS-422.

9.7 Storage options and directory structure

This chapter describes the different options for saving measurement data with imc STUDIO.

The data stream

Once the measurement starts, the measurement device generates a data stream. The goal in every measurement is to process this data stream and to observe and or save it.



Simplified representation of the data stream - illustrated by the example of an imc CRONOS device

The data stream is first written to FIFO RAM buffer memory 427 in the device.

This RAM buffer is read by a signal processor ("DSP").

- If you use an imc Online FAMOS 851 program in the experiment, the data stream is read and processed at this point of the data stream.
- For closed-loop control applications, it can make sense to interrupt the data stream at this point already (deselect "*Transfer to the PC* 413").

After this step, the data can either be saved by the device itself, or transmitted to the control-PC and saved if desired. Whether or not data are transferred to the PC and whether they can be saved can be set for each channel separately.

Note — Display without Data Storage

Data storage isn't necessary in such settings as system setup or trial readings.

Data saving		Description
Saving by **the device** 415	By default, data saving is **deactivated**	The device can save the measured data to the internal storage medium 415 or to a network computer 419 (network sharing). What options are available depends on the particular device model and configuration. Data storage by the device enables autonomous operation without connection with any controlling PC, e.g. in disconnected mode or in Autostart 206.
Saving and displaying data on **the PC** 421	By default, data saving is **activated**	Here, the data stream is transferred to the controlling PC via the **connection** (mostly LAN) ("*Transfer to the PC* 413 "). The data can be **displayed** using the Panel pages. Additionally, the transferred measurement data can be **saved** on the PC. • By default, the **data for the display** are saved on the PC in a **circular buffer**. By using the circular buffer memory, it is possible to temporarily store the data in the higher speed ("rapid access") working memory (RAM). • For **data storage** purposes, circular **buffer memory is not used**.

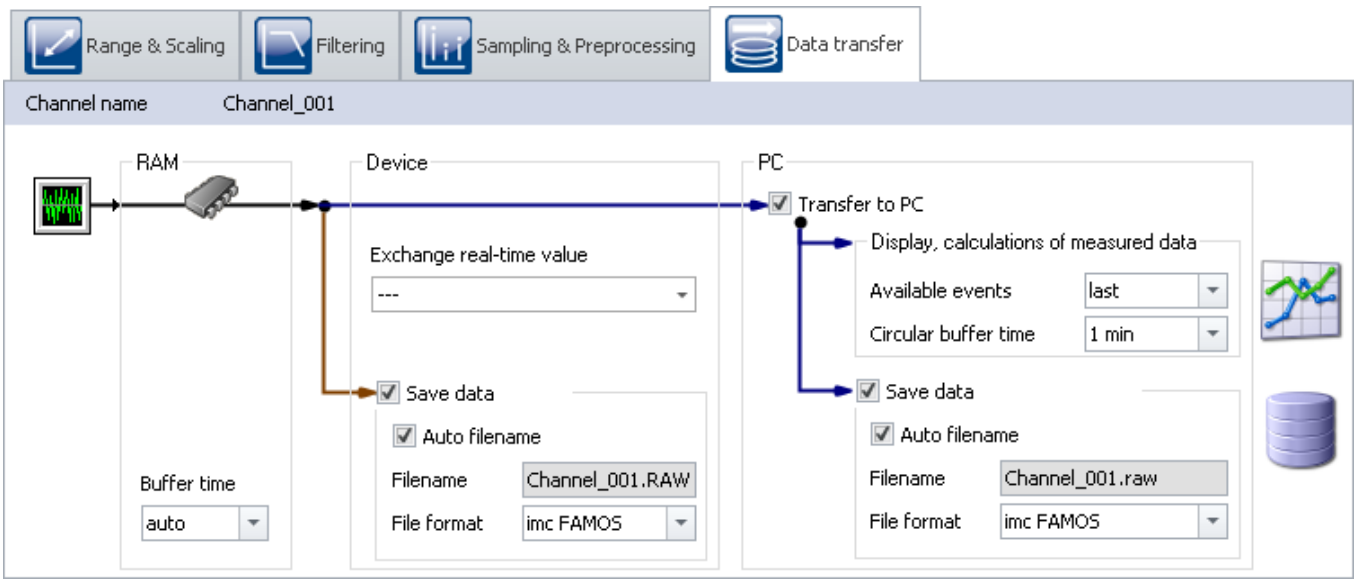
Data saving by the PC and by the device

There are a variety of data storage settings. Appearing below are links to descriptions of the most important parameters:

Channel-specific parameters	Device-parameters
File-Name 413 and File-Format 414	Interval storage 423
Circular buffer memory 414	Save trigger events in individual files 434
Available events 414	Storage location on the device 415
Transfer to PC 413	

9.7.1 Storage options - Settings for the channels

Setting the storage options for the channels is performed on the Setup page: "*Analog/digital channels*" in the dialog: "*Data transfer*".



Accumulating measured data are temporarily stored in the device until they are transferred to the PC for purposes of display, analysis or storage.

imc STUDIO offers the following options located under the dialog: "*Data transfer*":

1. Only online processing (no transfer to or display on the PC)
2. Saving on internal storage medium (device drive)
3. Display and calculation of the data on the PC (with or without Circular buffer[1])
4. Saving on the PC's hard drive

Items 2 through 4 above can be combined in any way desired. All storage options can be set for each channel separately.

[1] Circular buffer: the data are saved on the PC's memory for the specified amount of time and then deleted (FIFO).

Parameter	Description
Transfer to PC	When the box is checked, the data are transmitted to the PC. This is a prerequisite for displaying , calculating and storing data on the PC. • **Only online processing:** When the box is **not** checked, the data are **not** transferred to the PC. The data are only available for calculations in imc Online FAMOS or for storage in the device. No display of the measured data in the curve window or in the Data Browser.
Save data	• **Saving on the PC** 421: Saving of the measured data on the PC hard drive according to the folder structure 433 set. • **Saving on the device hard drive** 415: Saving of the measured data device hard drive according to the folder structure 433 set. Channels using this option are recorded under the current experiment name on the device's internal drive.
Auto filename	If the box is checkmarked, imc STUDIO will create a file name: • **PC** and **Device:** The channel name will be used as file name

Parameter	Description
Circular buffer time on the PC (for display/calculation)	Here you set how long the maximum time segment is. When using this option, the required memory is known from the beginning of the measurement. **Deactivating circular buffering:** If you enter "0", the circular buffer is deactivated and all data are saved. • Recommendation: For the measured data (display and calculations), use an activated circular memory buffer. This transfers the data to the faster-working RAM memory. • The circular buffer duration set is also valid for segmented channels such as the FFT. After the time period set, the count of segments no longer increases, since the first segments are deleted instead. See "*Events and circular buffer* 422"
Available events (Display)	With this setting, you determine whether all events (triggered events) can be displayed, or only the last one. • **all:** As soon as the setting is set to "all", the channel is an **event-based** data set. E.g., it is possible in this way to display all events in one curve window. • **last:** Only the last event is available for display or calculations (recommended). • It is only possible to set the amount if the respective channel is not set for circular buffer memory, which means it is set to "*Unlimited*". • Note on analysis using imc FAMOS: If possible, use the measured data saved on the hard drive for analysis purposes. See "*Events and circular buffer* 422"
File format	Format of the saved measured data. By default, the measured data are saved in the file format imc FAMOS. The desired file format can be set in the file format selection list: **Format** / **Description** imc FAMOS 1153 — Default format for all device variables. imc FAMOS ZIP 1153 — Compressed format for many fieldbus variables. CANAlyse-MDF 1154 — CANAlzser-MDF format for CAN-Log data. "*imc FAMOS ZIP*" and "*CANAlyse-MDF*": Applicable with devices of the firmware group A 187 (imc DEVICES).
Exchange real-time value	See "*Exchanging values among the devices* 767"

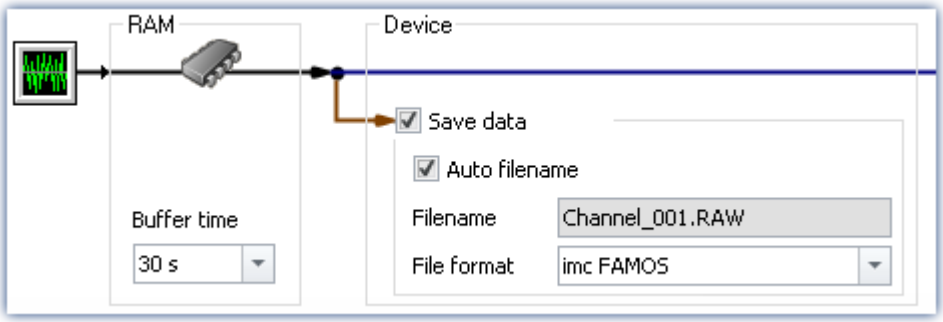
9.7.2 Storage in the device

Setting the storage options for the channels is performed on the Setup page: "*Analog/digital channels*" in the dialog: "*Data transfer*".

Saving to the internal storage medium

You can save the measured data directly in the device. For this purpose, an appropriate data storage medium is necessary (see "*Device hard disk, removable drive* 443").

Data saving by means of the device enables autonomous operation, without needing connection with a controlling PC, e.g. in disconnected mode or in Autostart 206.



By default, data saving by the device is deactivated. To activate the data saving, set a check in the box for **Save data**.

The channels are saved to the internal device drive under the current experiment name according to the folder structure 433 set.

Storage location

There are several options for saving data by the device:

- Saving on a **hard drive**, which is built into the device housing.
- Saving on a **removable data carrier** which can only be exchanged while the device is deactivated (**not Hotplug-capable**)
- Saving on a **removable data carrier** which can be exchanged during operation (**Hotplug**)
- Saving on a **network computer** 419 (network enabled)

Which possibilities are available depends on the device hardware configuration and type.

To set the storage location, open the dialog: "*Storage*" on the Setup page: "*Devices*".

Note The maximum file size: 2 GB

Per interval or measurement, one file is created per channel, which contains the respective measurement files.

If this file reaches **2 GB** in size, then a new measurement folder with a new file is created automatically. This folder has the same name as the first one, however it is appended with a suffix 002, e.g. "*2023-12-12 14-30-00.002*".

If this new file, too, reaches 2 GB in size, then a new file is created again, this time with the suffix 002.

Data transfer - Access via Windows Explorer

For devices belonging to the firmware group B 187 (imc DEVICEcore):

The menu item "*Data (Device)*" ( ) causes the Windows-Explorer to start while indicating the device selected.

Ribbon	View
Extra > Data (Device) ( )	Complete
Home > Data (Device) ( )	Standard

Using this menu item, a separate Windows-Explorer is opened for each device, providing direct access to the internal memory.

For the purpose of accessing the tool, it is necessary to log in. The factory default settings are:

- User: "*imc*"
- Password: device serial number

Note Manual entry

The device is called by reference to either its IP address or its internal name. The device's name is constructed as follows: "*imcDev*" + SerialNumber; e.g.

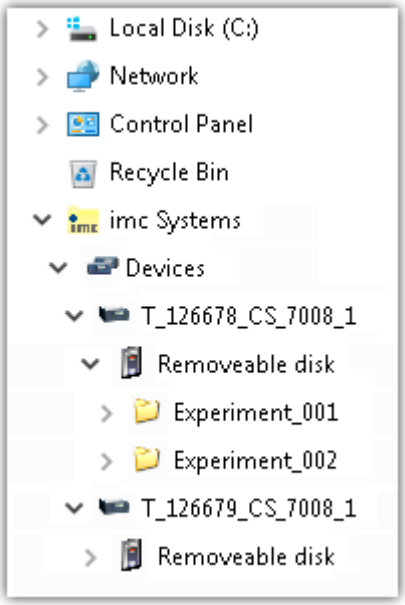
\\imcDev4120110
\\10.0.12.70

For devices belonging to the firmware group A 187 (imc DEVICES):

The menu item "*Data (Device)*" ( ) causes the Windows-Explorer to start while indicating the device selected.

Ribbon	View
Extra > Data (Device) ( )	Complete
Home > Data (Device) ( )	Standard

Access via "*imc Systems*" - an Explorer shell extension



While installing the operating software, if the option "*Extension for Windows-Explorer* 26" is activated, you are able to copy, display or delete the files of measured data saved within the device (e.g. on the removable storage medium). The method of doing this is the familiar one under Windows.

This function is independent of the device software. As well, selection of the devices in the tree diagram is independent of the device list in the operating software.

In order to get access to your device's storage medium, you must first add it to the tree diagram (see "*imc Systems - Adding a device* 449"). Subsequently, you are able to navigate to the desired data in the storage medium and thus to work with them.

Reference

Data transfer

Additional information and notes on transferring the data from the intern hard drive to the PC is presented in the section:
"*Device hard disk, removable drive*" > "*Data transfer - firmware group B (imc DEVICEcore)* 444"
"*Device hard disk, removable drive*" > "*Data transfer - firmware group A (imc DEVICES)* 448"

Storage Options

Information on the options for storage of the measured data is presented in the section: "*Storage Options - Settings for the Channels* 413".

9.7.2.1 Supplemental files: DirClosed and imcSyslog

In addition to files of measured data, the measurement system also creates two system files.

DirClosed

On the device hard drive, the file "*DirClosed*" is created as soon as a measurement data folder is completely concluded. The file has no content. If this file exists, it is possible to access the measured data. For imc LINK or any personally created transfer programs, reliable copying and deleting are ensured.

imcSyslog

Note

The following description applies to devices belonging to the firmware group A 187 (imc DEVICES). Devices belonging to the firmware group B 187 (imc DEVICEcore) do not use the "*imcSyslog*" file.

While a device is operating without being connected to any PC, the device is not able to report any errors. So if for instance a data overflow 755 occurs during online processing or when saving the data, this information must be saved in the device.

Such information is saved in the file "*Syslog*", which is located in the respective measurement data folder. At this time, only important information regarding cases of data overflow are recorded. By this means, the measured data can be reconstructed except for the lost samples.

Name of the file

DeviceXXXXXX.syslog, "XXXXXX" standing for the device's serial number, e.g. Device120345.syslog.

Structure of the file when functioning normally

```
imcSyslog Version 1.0
imc DEVICES 2.13R8 (19.8.2020)
Nothing to report!
```

Structure of the file in case of data overflow

```
imcSyslog Version 1.0
2020-09-21 11:08:06.929190 M#:Data overflow on disk! F#:V006_G01.raw E#:1 FE#:0 O#:206 L#:1253
2020-09-21 11:08:08.147645 M#:Data overflow on disk! F#:V006_G01.raw E#:1 FE#:0 O#:1459 L#:1800
2020-09-21 11:08:11.029787 M#:Data overflow on disk! F#:V006_G01.raw E#:1 FE#:0 O#:3259 L#:1113
2020-09-21 11:08:16.071192 M#:Data overflow on disk! F#:V006_G01.raw E#:1 FE#:0 O#:6918 L#:1304
2020-09-21 11:08:22.906673 M#:Data overflow on disk! F#:V006_G01.raw E#:1 FE#:0 O#:10111 L#:1054
2020-09-21 11:08:24.630697 M#:Data overflow on disk! F#:V006_G01.raw E#:1 FE#:0 O#:11186 L#:1100
Log closed!
```

The structure is designed so that the information can be both read and also processed by a repair program.

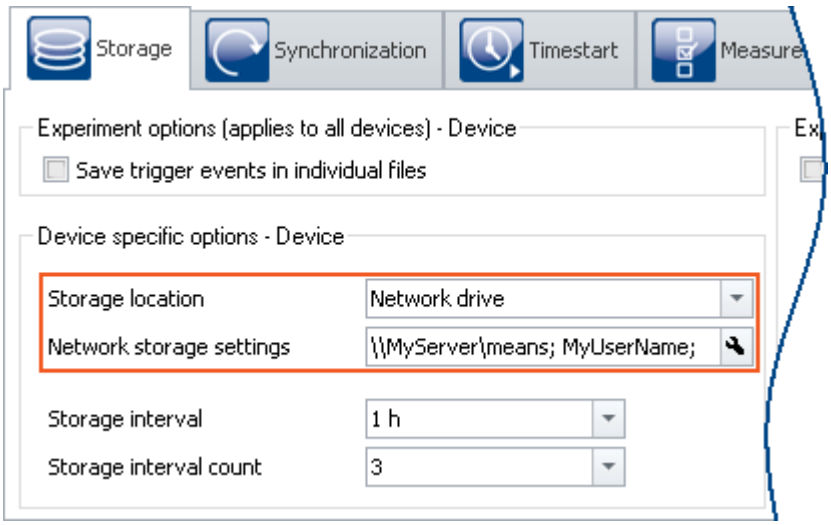
2020-09-21 11:08:06.929190	M#:	Data overflow on disk!	F#:	V006_G01.raw	E#:	1	FE#:	0	O#:	206	L#:	1253
1	2	3	4	5	6	7	8	9	10	11	12	13

	Description	Example
1	Date and time of the event	e.g. 2020-09-21 11:08:06.929190
2	Identifier for the message text	M#
3	Report as an English-language text	e.g.: Data overflow on disk!
4	Identifier of the filename	F#
5	Name of the file in which the overflow occurred	e.g. V006_G01.raw
6	Identifier of the absolute event number	E#
7	Absolute number of the event in which the data overflow occurred	e.g. 1 for the first trigger release on this channel since the measurement start
8	Identifier of the relative event number	FE#
9	Relative number of the event, in reference to the events in the file affected	e.g. 0 for first event in this file; 1 for second event in the file...
10	Identifier for the offset	O#
11	Offset of the overflow within the file in 16-bit words	e.g.: 206. -1 -> The overflow is located between the file for the last interval and this one
12	Identifier for the length of the data overflow	L#
13	Length of the data overflow in 16-bit words	e.g.: 1253

9.7.3 Storage on a network drive

With the storage with the device, you can also specify a network drive as the target. I.e. the device can independently save the measured data to the network without that the program is running on the PC.

However, the device must support network data storage, otherwise the option is not available (see "*Device overview* 187").



"Storage" dialog
Network storage

The data are not saved in the device, but on a centralized data collector.

Systems which use the Server Message Block (SMB) can be addressed as the target location (see "*Prerequisites* 420").

In order to set data saving on a network drive, open the dialog: "*Storage*" on the Setup page: "*Devices*".

Fundamental differences from data storage on the PC:

- In contrast to storage on the PC, it is not necessary for the measured data to be saved in the imc STUDIO database with the experiment.
- The measurement results are not displayed as a saved measurement in the Data Browser.
- Technically, there is an additional difference between data storage on the PC and data storage in the device/network drive:
  - With data storage on the PC, the operating program handles the data saving.
  - Data storage on the network drive is administered by the measurement device.

This means it is no longer necessary for the operating software to be running during measurement.

Advantages:

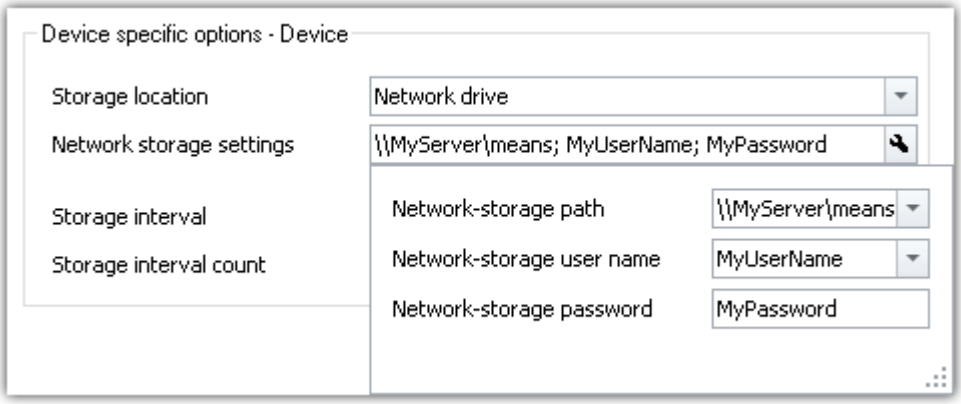
- The operating program does not need to be running during the measurement.
- Data storage in a PC via the network thus now also works in Autostart and in Sleep/Resume modes (e.g. imc BUSDAQ 187).
- Centralized data storage, independent of the experiment's storage location.
- Copying of the internal drives to the central data storage area following a measurement is omitted.

Prerequisites:

- A device that supports network storage (see "*Device overview* 187")
- Sharing of a network drive with writing permissions
- Secure network
- Target system using the Server Message Block (SMB) (formerly known as: Common Internet File System (CIFS)): LINUX, WIN2000 and Windows XP/Vista/7/8/10

Setting

The network drive is specified by means of the target PC's IP address and the name of the folder shared. The network storage path must be stated in UNC notation (\\server\share\ - DNS or only IP address possible).



Dialog "Storage"
Network storage settings

Ensure that write access to the network storage path (clearance, "share") is possible with the specified user name and password. If you did not protect the target system with a password, leave the field blank (WIN2000 and Windows XP/Vista/7/8/10 do not support this at present).

The measurement instrument creates a subfolder on the network drive for the data, which is composed of the device's name and the experiment's name.

The "*Waiting time [s]*" refers to the maximum time available for mounting the network drive when preparing the measurement.

Example

The following network path is selected (**\\MyServer\means**)
The device has the name: **imcCronosPL_123368**
The experiment has been saved under the name: **Test1**
When you start the measurement, the measurement results are saved to the following path:
\\MyServer\means\imcCronosPL_123368\Test1.

Note

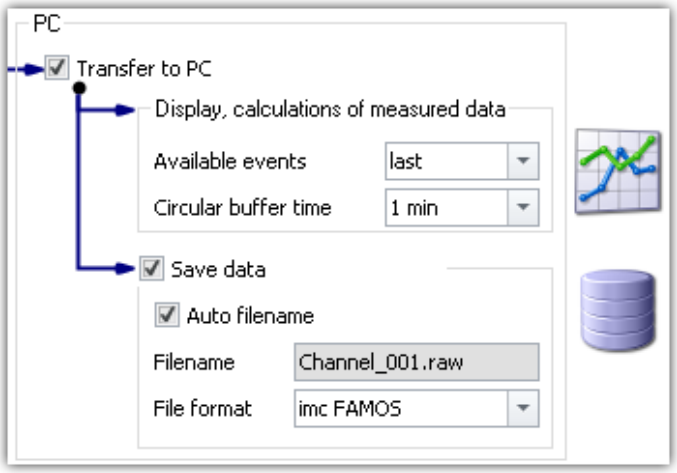
If the target PC is not available, the system reports a WINSOCK error. This disables the device for approx. 5 minutes and can only be prematurely canceled by rebooting.

9.7.4 Transfer and storage on the PC

Setting the storage options for the channels is performed on the Setup page: "*Analog/digital channels*" in the dialog: "*Data transfer*".

Transfer to PC

The measured data are transferred to the PC if the option "*Transfer to PC*" is activated. During measurement, you can also have the data displayed on the PC while being saved on the hard drive.



Display and calculations of measured data

The measured data available for display on Panel pages and for calculations.

Save data

To activate data storage on the PC, set a check in the box for "*Save data*".

Storage location

The **measurement files** are saved in a defined folder structure **together** with the **experiment**. There are a variety of available options for modifying the folder structure 433. Additionally, it is possible to **personally adapt how and where data are saved** using the options 435.

After conclusion of the measurement, display the data thus saved e.g on Panel pages, or with imc FAMOS.

Note

Make sure, that there is enough free memory on the selected drive for the expected measurement data.

Access via Windows Explorer

The menu item "*Data (PC)*" ( ) causes the Windows-Explorer to start while indicating the memory location of the measured data.

Ribbon	View
Extra > Data (PC) ( )	Complete
Home > Data (PC) ( )	Standard

Reference See also

- Information on the options for display and storage of the measured data is presented in the channel: "*Storage Options - Settings for the Channels* 413".
- Information on display of the stored measurement data is presented in the chapter: "*Panel* 1205".
- Device hard disk, removable drive 443

9.7.5 Events and circular buffer

Events

Events result from interruptions in the time plot of measured data, during which the measurement actually continues to run, but no measurement data are recorded. This can occur either intentionally, due to multiple triggering for example, or unintentionally due to a data overflow.

In order to be able to display multiple events together in one data set, a data set containing multiple Events is generated, a so-called "*Event-based data set* 1131".

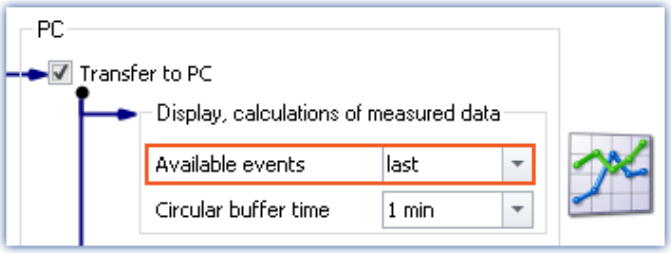
Events in data storage

All events are **always** saved.

- As soon as multiple events can occur (activated multi-triggering), the channel becomes an **event-based** data set.
- If multi-triggering is not activated or no triggering at all (e.g. connected with the BaseTrigger), then an event-based data set will usually not be generated (the exception is a data overflow).

Special case: For storage of the measured data on the PC, each event can be saved to a separate file. (see Device settings > "*Storage* 292" > "*Save trigger events in individual files* 434")

Events in visualization and for calculations



- It is only possible to set the amount if the respective channel is not set for circular buffer memory, which means it is set to "*Unlimited*".
- Note on analysis using imc FAMOS: If possible, use the measured data saved on the hard drive for analysis purposes.

Parameter: Available events

Parameter	Description
Available events (Display)	With this setting, you determine whether all events (triggered events) can be displayed, or only the last one. • **all:** As soon as the setting is set to "all", the channel is an **event-based** data set. E.g., it is possible in this way to display all events in one curve window. • **last:** Only the last event is available for display or calculations (recommended).

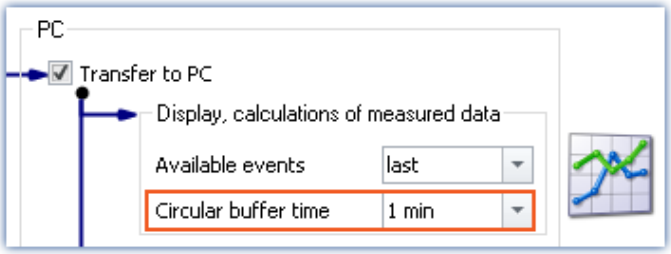
Circular buffer

Circular buffering in data storage

For data saving purposes, there is **no circular buffer**.

Circular buffering in visualization and for calculations

By default, the **data for the display** are saved on the PC in a **circular buffer**. By using the circular buffer memory, it is possible to temporarily store the data in the higher speed ("rapid access") working memory (RAM).



- Recommendation: For the measured data (display and calculations), use an activated circular memory buffer. This transfers the data to the faster-working RAM memory.
- The circular buffer duration set is also valid for segmented channels such as the FFT. After the time period set, the count of segments no longer increases, since the first segments are deleted instead.

Parameter: Circular buffer time

Parameter	Description
Circular buffer time on the PC (for display/calculation)	Here you set how long the maximum time segment is. When using this option, the required memory is known from the beginning of the measurement. **Deactivating circular buffering:** If you enter "0", the circular buffer is deactivated and all data are saved.

Warning Deactivated circular buffer

This setting is absolutely not suitable for long-term measurements

Recommendation: For the measured data (display and calculations), use an activated circular memory buffer. This transfers the data to the faster-working RAM memory.

9.7.6 Saving interval

Saving interval - What is it (for)?

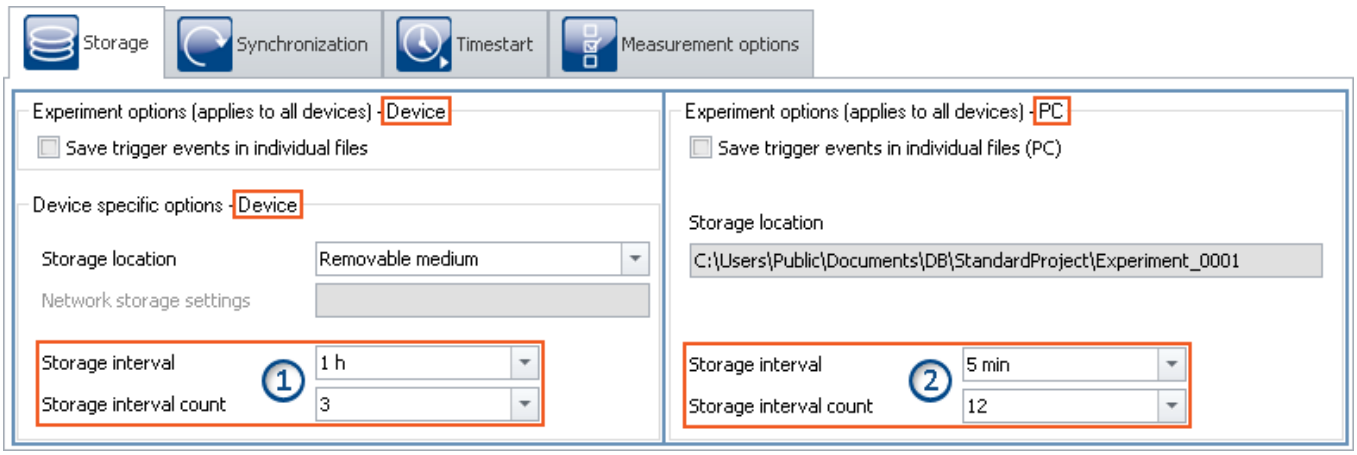
Upon the elapse of a certain time interval, a running data set is closed and a new file created.

- A number of saving intervals is specified. This number times the duration of the interval equals the maximum memory depth. When this number of intervals has been exceeded, the recordings of the oldest respective intervals are deleted.
- The setting is valid for all of a device's channels.

Suppose that there is a power outage during a running measurement; there is no guarantee that the files are closed properly. Therefore the last recorded experiment can not be evaluated. By setting a saving interval, you determine the maximum possible data loss due to a system failure.

Activating interval data saving

To activate interval data saving, open the dialog: "*Storage*" on the Setup page: "*Devices*".



Interval saving can be set for data storage on both the devices and on the PC separately.

- On the left side, you see the settings for the data storage on the devices 426: The interval saving can be set **for each device** separately ①.
- On the right side, you see the settings for storage on the PC: the storage interval set **applies to all the devices** ②.
- The interval storage setting **not for each channel separately**!

Note

- Activating the interval storage does not activate the storage for each channel separately.
- Activate additional data storage 413 on the PC or in the device for the channels desired!

Storage interval / Storage interval count

Parameter	Description
Storage interval / Storage interval count	With these two parameters, you can limit the amount of data and number of files. In this way, it is possible to prevent complete exhaustion of available memory even during long-term measurements. **Storage interval:** After this much time elapses, a new file is created for the duration of the next saving interval. **Storage interval count:** Save the last x intervals sets the desired number of intervals. Once this number is reached, the oldest interval is deleted. Multiplying the count and the interval determines which measurement duration is available after the measurement has been run. **Example:** Storage interval = 5 min, Storage interval count = 12. The measurement duration is set to 24 h. This ensures that after one day (24 h) **at least** the last 60 min of the measurement are available in intervals of **a maximum of** 5 min.

Note Amount of data saving intervals

Once the measurement is concluded, you usually obtain **one data saving interval more** than your settings specified. This ensures that at minimum the desired amount of data are available at the end of the measurement.

- Example:
  Measurement duration: 60 s; with **intervals** of 10 s; and an **interval count** of 3 -> result in 30 seconds of expected data.
- Measurement start: 00:05
- The intervals are allocated as follows:
  00:05 to 00:10 <- deleted
  00:10 to 00:20 <- deleted
  00:20 to 00:30 <- deleted
  00:30 to 00:40
  00:40 to 00:50
  00:50 to 01:00
  01:00 to 01:05

In order for at least 30 seconds to be available after the measurement ends, the last 4 intervals must remain saved. In this example, that means 35 seconds of measured data. If the oldest interval were deleted, then there would only be 25 seconds.

- Incomplete intervals can be avoided by means of a **Timed start**. By this means, start measurement exactly at an **interval boundary**.
  For technical reasons, with some data types, one extra interval is created. This mostly affects such data types as matrices, histograms or TimeStampASCII (TSA) data.

During the measurement the PC usually contains **two extra saving intervals**, since it is not yet possible to access the current interval. As a result, in this case at least the desired data are always available for display/analysis.

Warning Measured data will be deleted

- When interval storage is active, all **folders are deleted except for the specified number** of intervals!
- If you **prepare or start** a measurement, **the number of existing measurement folders is reduced** to the number of storage intervals.
  This also applies if the number of intervals is decreased.
- Only measurements which are not loaded are automatically deleted in this way. Any measurements which are loaded are ignored. Thus, any measurements which are currently being viewed or analyzed on the Panel are not automatically deleted.
  In consequence, it may happen that more saved measurements remain in existence than were specified. The measurements will only be automatically deleted if the particular measurement has been closed.
- The measurements can be deleted manually at any time via the context menu of the Data Browser.

FAQ

Why is the first interval shorter?

Answer: To achieve a standardization of interval folders on the PC and on the device, the first interval is cut at the next interval.

Example:

- The measurement starts at 8:55:03, the saving interval has been set to 10 minutes.
- The first interval will be closed at the next whole-number 10 minute interval, that is 9:00:00. From here it will be in steps of 10 minutes: 13:10:00, 13:20:00 etc.

Reference See also

See also:

- Pretriggers in conjunction with intervals 403

9.7.6.1 Saving interval on the internal device storage medium

By means of the settings "*Storage interval*" and "*Storage interval count*", you specify a defined storage space. A **limited count** produces the same data storage effect as **circular buffer storage**. Once the count specified has been reached, the **oldest data are deleted**.

Special feature of circular buffer operation (limited count of data saving intervals)

There is a special feature with **Storage in the device** if a defined interval count is set (not applicable if "*all*" is set). If the **free memory is already exhausted** before the required folder count has been reached, the **circular buffer saving is performed with fewer intervals**! However, this only happens if there are at least four intervals.

Implementing the circular buffer operation

- A loop is used to test whether there are at least four measurement-folders in the current experiment folder.
- The oldest folder is deleted and a check is made of whether sufficient memory is now free.
- If this is not the case, the process is repeated.
- As soon as sufficient memory is available, data recording continues.
- If only three or fewer folders are left, data recording to the storage medium stops and the error message "Disk full" is posted.

FAQ

Under what circumstances will circular buffer operation not work?

Answer:

1. If the storage medium is already full and a new experiment is to be created.
2. If a measurement is of indefinite length and no saving interval was set.
3. If the storage medium memory allotted to the current experiment is so little that it is exhausted before at least four folders are saved. This is true even if the folders require different amounts of memory.

Why must there be at least four folders?

Answer:

1. For ring-operation, only the storage medium memory is used which was occupied by old folders belonging to the same experiment, or which is still free.
   As a rule: no data belonging to another experiment are deleted.
2. At all times, at least two closed folder containing readable data must exist.
3. In consequence, the current folder doesn't count, because it may not contain data. Then, if at least two valid folder must exist when deletion occurs, three folders must be present before deletion. Together with the current folder, that makes four folders necessary before deletion can take place!

Note

- The number of samples per saving interval may vary, especially for heavy data throughput. However no sample will be lost.
- A concluded interval is denoted by the file "*DirClosed* 417".

9.7.7 RAM time duration - Outage of the network connection

Loss of the network connection between the PC and the device often causes measured data to be lost.

Note — Saving measured data in the device

Remedy: Save an extra copy of the measured data in the device. In this way, the measured data can be compared and reconstructed.

The RAM buffer duration is a time interval reserved for each channel, for the duration of which the data are stored in the measurement device's RAM. The longer this time interval is, the less likely data overflow is to occur if the PC's connection to the measurement device is interrupted.

For example, if a faulty connection to the PC prevents the data from being transferred to the PC, data overflow will occur after elapse of the RAM buffer duration set. A long duration is also necessary if you intend to switch the internal data storage medium during the course of a measurement (Hot-Plug).

Example

Suppose the active channels have a RAM buffer duration of 10 s. If the PC doesn't retrieve any data from the Ethernet-interface for 10 s, the buffer overflows, i.e. data to be transferred to the PC are lost. The message "*Data overflow*" is posted.

Behavior upon outage of the network connection

At regular intervals, imc STUDIO attempts to restore the connection with the device. If this is not possible, a message is posted accordingly. Once the connection has been re-established, data acquisition on the individual channels resumes. This happens on each channel separately and can thus lead to differing results, for instance, if the RAM-buffer sizes are different.

Result	Possible causes
Data storage resumes seamlessly from the previously saved data. The data are retrieved from the RAM; there is **no loss of data**.	For example, if the RAM buffer is sufficiently large that the data can be retained in the device.
A **new "event"** is generated (event-based data sets). The RAM-buffer data are not retrieved. The **measurement data** acquired during interruption of the network connection **are missing** and are not saved. In accordance with the applicable settings, either all events or only the last event is displayed 414 (the option does not apply to data storage).	For instance, if a gap in the data is detected. A data overflow, because the RAM-buffer duration was exceeded.
A new measurement folder is created and has the extension ".002". The **measurement data** acquired during interruption of the network connection **are missing** and are not saved.	E.g if the saved data can not be opened or if the data type does not permit events.

Reference See also

- "*Behavior upon Data Overflow* 755"
- "*Avoidance of Data Overflow* 755"

9.7.7.1 RAM time duration for imc DEVICEcore devices

The RAM memory available is distributed among the active channels. Thus, in an emergency the maximum available memory is at the system's disposal.

How much memory space is available for the buffer depends on a variety of factors, such as the size of the imc Online FAMOS source texts and other internal files. Thus it is not possible to specify exactly.

The possible amount of RAM buffer per channel depends on the channel count, the respective sampling rates, the aggregate sampling rate and the data types, among other things.

Note RAM-buffer time and synchronization

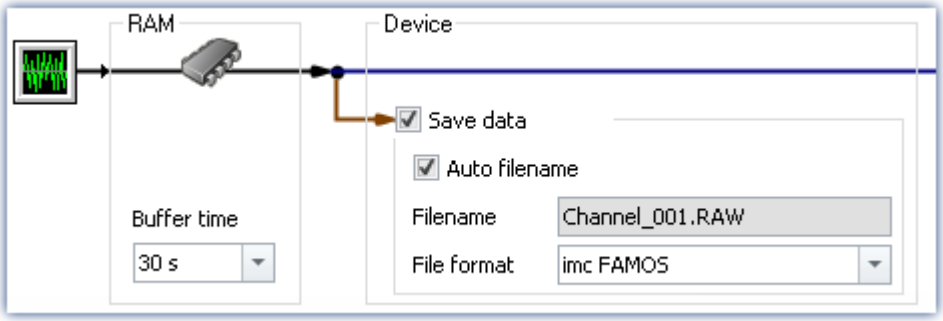
RAM-buffer time settings have **no effect** on synchronization of imc DEVICEcore-devices. These settings only need to be considered when using imc DEVICES-devices 429.

Reference Time-stamped data

Also observe the notes on the RAM buffer duration for channels with time-stamped (e.g. some fieldbus channels), in the section: "*Time-stamped channels - Setting the RAM-buffer duration* 1132".

9.7.7.2 RAM time duration for imc DEVICES devices

Setting the RAM options for the channels is performed on the Setup page: "*Analog/digital channels*" in the dialog: "*Data transfer*".



A channel's RAM buffer volume is computed by multiplying sampling rate x buffer duration. For instance, a channel sampled at 5 kHz for the recommended RAM buffer duration requires 10 s 5000 x 10 = 50,000 Samples. When preparing for measurement, the system checks among other things whether there is enough free RAM available for the requirements of all channels.

If **error messages are posted during the preparation process**, then reduce the number of active channels are the buffer duration of individual channels, such as those which are not saved to the PC.

Reference Time-stamped data

Also observe the notes on the RAM buffer duration for channels with time-stamped (e.g. some fieldbus channels), in the section: "*Time-stamped channels - Setting the RAM-buffer duration* 1132".

Automatic adjustment of the RAM-buffer

When **multiple high-sampling-rate channels** are used, the **RAM-buffer** in the device may eventually become **insufficient**. When the buffer size is fixed, the following error will often occur: "*Insufficient memory in device! Please observe the RAM buffer time of the channels and the number of triggers.*".

By means of the **setting "*auto*"**, the RAM buffer time is **dynamically adjusted** between 2 seconds and 10 seconds.

If one of the calculated **buffer times falls below 2 s**, the **error message** stated above also appears.

"*auto*" is the default setting. If the buffer is set to a **fixed value**, it is **no longer adjusted automatically**.

RAM-buffer time and synchronization

Absolutely synchronized recording of multiple channels, e.g. for computing power from a current signal and its associated voltage signal, is ensured in one of the three following ways:

1. Select for the channels to be mathematically combined the **same sampling interval** and the **same RAM buffer time**. Assign these channels to the same trigger.
2. Avoid multiple triggers and prepare the measurement before starting it, even if the configuration has not been changed. To force a repeat preparation, select the Ribbon *Setup-Control > Reconfigure*.
3. Select a **RAM buffer time** resulting in a **buffer volume less than 65,536**. Example: 20 kHz sampling rate: 65,536 / 20,000 = 3.3 s. You can use the keyboard to enter this value in the input box next to the suggested values.

If none of these options was implemented, there may be a time offset between the channels. The cause for this is as follows:

There are 2^16 = 65,536 addresses available for a channel's RAM buffer. This means that for a RAM buffer volume of up to 65,536 samples, each sample can be addressed individually. If a greater value results from sampling rate x buffer time, the RAM buffer is organized in blocks.

Example

A sampling rate of 10 kHz x 10 s buffer time results in 100,000 > 65,536 samples. Here, the RAM buffer must be organized in 50,000 blocks of 2 samples apiece. This results in a tolerance in the addressing of magnitude block size-1, in this example, 1 sample.

In rare cases, this tolerance can cause a time offset between two channels. The possible time offset is then in the range between 0 and (block size-1) samples.

9.7.8 Overview storage and display options

Analog, digital, or other equidistantly sampled or time-stamped channels (e.g data-reduced channels)

0/1 = deactivated / activated | --- = Option is ignored

For display and calculation (with or without storage of measured data)

Multiple triggering?	Circular buffer	Circular buffer Time	Available events
No	unlimited		---
	activated	any (3)	---
Yes	unlimited		Last / All (2)
	activated	any (3, 4)	Last (1)

Storage of measured data on the PC and on the device

Multiple triggering?			Saved events	Storage interval	Storage interval count
No			---	any (3)	1..n or all
Yes			All	any (3)	1..n or all

(1) For channels with multi-triggering, circular buffering pertains to the last trigger-event. The circular buffer memory does NOT extend across multiple trigger-events! Only the last event is displayed.

(2) The default setting is: Display of the "last" event (also for channels with an unlimited trigger-event count). By this means, overflow of memory during long measurements is to be avoided. Regardless of this setting, for the display in the curve window, you can set the displayed events to "All/last/some".

(3) The circular buffer length and the number of saving intervals can take 'any' value in the framework of the storage options for the PC used. The minimum storage size is 2 data samples.

(4) The buffer memory size is limited by the channel's trigger length. Since the amount of data on a trigger is limited, more memory space would be useless.

FFT or similar virtual channels

0/1 = deactivated / activated | --- = Option is ignored

SL = Segment length (length of the FFT or similar)

For display and calculation (with or without storage of measured data)

Multiple triggering?	Circular buffer	Circular buffer Time	Available events
No	unlimited		---
	activated	n*SL	---
Yes	unlimited		Last / All (2)
	activated	n*SL	Last (1)

Storage of measured data on the PC and on the device

Multiple triggering?			Saved events	Storage interval	Storage interval count
No			---	any (3)	1..n or all
Yes			All	any (3)	1..n or all

(1) For channels with multi-triggering, circular buffering pertains to the last trigger-event. The circular buffer memory does NOT extend across multiple trigger-events! Only the last event is displayed.

(2) The default setting is: Display of the "last" event (also for channels with an unlimited trigger-event count). By this means, overflow of memory during long measurements is to be avoided. Regardless of this setting, for the display in the curve window, you can set the displayed events to "All/last/some".

(3) The circular buffer length and the number of saving intervals can take 'any' value in the framework of the storage options for the PC used. The minimum storage size is 2 data samples.

Matrix or histograms

0/1 = deactivated / activated | --- = Option is ignored

For display and calculation (With or without storage of measured data)

Multiple triggering?	Circular buffer	Circular buffer Time	Available events
No	unlimited		---
	activated	---	---
Yes	unlimited		Last / All (2, 5)
	activated	---	Last (1, 5)

Storage of measured data on the PC and on the device

Multiple triggering?			Saved events	Storage interval	Storage interval count
No			---	any (3)	1..n or all
Yes			Alle (5)	any (3)	1..n or all

(1) For channels with multi-triggering, circular buffering pertains to the last trigger-event. The circular buffer memory does NOT extend across multiple trigger-events! Only the last event is displayed.

(2) The default setting is: Display of the "last" event (also for channels with an unlimited trigger-event count). By this means, overflow of memory during long measurements is to be avoided. Regardless of this setting, for the display in the curve window, you can set the displayed events to "All/last/some".

(3) The circular buffer length and the number of saving intervals can take 'any' value in the framework of the storage options for the PC used. The minimum storage size is 2 data samples.

(5) The calculations of matrices and histograms are based only on the data measured for the current trigger event.

© 2023 imc Test & Measurement GmbH

9.7.9 Data storage and directory structure

By default, the **folders** are named along with their **date and time**.

Timestamp - date time
\2021-06-12 17-01-30
\Channel_001.raw
\Channel_002.raw
\Channel_003.raw
\---.raw
\2021-06-16 14-01-30
\---.raw
\Starttime
\Filename.raw

The folder name is formed from:

- the start time, or
- the new interval's start time, or
- the point in time when data saving resumes after an interruption.

By means of the Data Browser, you are able to change a measurement's name at a later time. The folder name automatically changes accordingly.

Modifying the data storage and folder structure

The folder structure can be modified using the following options:

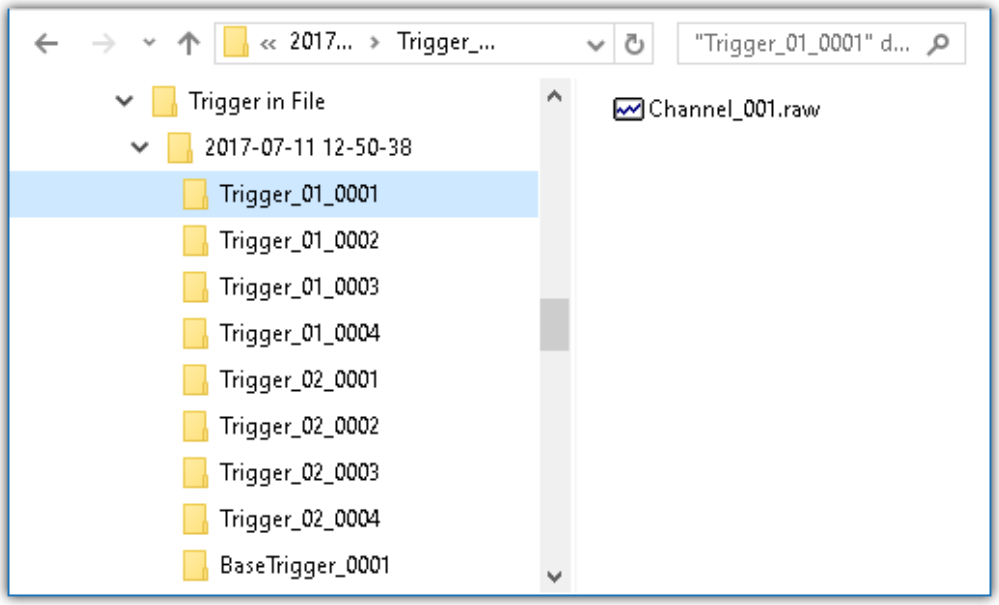
Options		Description
Save trigger events in individual files 434	Setup-page: "*Devices*", on the dialog: "*Data transfer*"	Each trigger firing can be saved to its own separate file. For this purpose, additional subfolders are created for each trigger firing.
Customized storage location for measurement data 435	Options: "*General options*" > "*Storage location*"	On the PC-side, you can personally define your own **storage location** and **folder structure**. For the folder designations, you can also use **metadata**.

9.7.9.1 Save trigger events in individual files

The option: "*Save trigger events in individual files* 292" lets you set the system's data saving behavior with triggered Events:

Not activated	Activated
Every triggered event is saved to the same file.	Each occurring trigger event is saved to a new file.
All events are always saved. As soon as multiple events can occur (activated multi-triggering), the channel becomes an **event-based** data set (see "*Data transfer > Available events* 422").	• For each trigger event, a separate folder is generated, in which the associated triggered channel data are saved. • Already while the measurement is still running, it is possible to access data on events which have already been completed.
\2021-06-12 17-01-30 \Channel_001.raw \Channel_002.raw \Channel_003.raw \2021-06-16 14-01-30 \---.raw \Starttime (measurement number) \Filename.raw	\2021-06-12 17-01-30 \BaseTrigger_0001\Channel_001.raw \Trigger_01_0001\Channel_002.raw \Trigger_01_0002\Channel_002.raw --- \Trigger_02_0001\Channel_003.raw \2021-06-16 14-01-30 \BaseTrigger_0001\---.raw \Starttime (measurement number) \TriggerName_TriggerNumber\Filename.raw

With this setting active, the folder structure for the measured data may appear as shown below:



Note Selection in Data Browser

In order to display the measured data associated with the events, add the column "*Event time* 1215" manually in the Data Browser.

Warning

- It is necessary to be very careful when using this option. Inconvenient selection of trigger conditions may cause the hard drive to quickly be filled with trigger folders.
- The time which the system needs to access data increases along with the number of folders.
  - For this reason, regarding data storage in the device, when **1000 trigger folders** accumulate, a **new measurement data folder** is created. This does not apply to storage on the PC.

9.7.9.2 Customized storage location for measurement data

When data storage is activated, the measurement results are normally saved to the database in the Experiment-folder.

According to the particular data storage settings, a Directory Structure 433 is established in the Experiment-folder.

It is possible for you to define your own data storage structure. The settings for this purpose are available at options: "*General options*" > "*Storage location measurement data* 129".

Even if the measurement folder is changed within an experiment (e.g. by means of placeholders), all measurements saved are listed in the Data Browser. On the hard drive, the measurements are located at a variety of positions. In the Data Browser, they are all displayed together.

Warning

- When you change these settings, some data storage settings may no longer be used. The settings can be used in conjunction with placeholders, but don't necessarily need to be.
- You define the structure's appearance. This may mean that the measurement results are always saved to the same folder. In order to avoid data being overwritten, a post-fix is automatically appended.
- Ensure that you possess reading and writing privileges for the data storage drive.

Definition of terms

Term	Description
Base path for measurement data storage	The base directory path for storage of measured data represents the default filepath for experiments. In this folder, all measurements are saved (according to the structure specified in "*Folder naming*").
Folder naming for measurement data storage	This specifies the structure below the base path. For instance, folders are set up here according to the settings for the directory structure 433.
Preview	The preview provides an example of where the measurement would be saved according to the current settings. Be aware that the preview is for demonstration purposes only and may not reflect certain settings for the Verzeichnisstruktur.

Custom base path for measurement data storage

Default setting: `<EXPERIMENT.PATH>`

- Returns the Experiment-path. All measurement results are saved with the experiment.

If you specify a different folder, all measurement results are saved to this new folder and no longer according to the Experiment-path.

Examples - Custom base path

Example 1

`C:\Measurement\<EXPERIMENT.NAME>\Results`

- Measurement results for the experiment: *Experiment_0001* are saved as follows, for example:
  C:\Measurement\Experiment_0001\Results2013-01-01 08-00-00
  C:\Measurement\Experiment_0001\Results2013-01-01 09-00-00
  C:\Measurement\Experiment_0001\Results2013-01-01 10-00-00

Example 2

`C:\Measurement\<EXPERIMENT.NAME>`

- Measurement results for the experiment: *Experiment_0002* are saved as follows, for example:
  C:\Measurement\Experiment_00022013-01-02 08-00-0
  C:\Measurement\Experiment_00022013-01-02 09-00-00
  C:\Measurement\Experiment_00022013-01-02 10-00-00

Example 3

`<EXPERIMENT.PATH>\Results`

In a case where the database was moved to the following path:
c:\DB

- Measurement results for the experiment: *Experiment_0003* are saved as follows, for example:
  C:\DB\StandardProject\Experiment_0003\Results2022-01-03 08-00-00
  C:\DB\StandardProject\Experiment_0003\Results2022-01-03 09-00-00
  C:\DB\StandardProject\Experiment_0003\Results2022-01-03 10-00-00

Custom folder naming for measurement data storage

Default setting: `\<STORAGE.FOLDERNAME>\`

- Returns the experiment's data saving settings

If you specify a different folder naming, all measurement results will be save according to the specified folder structure.

Useful placeholders for the folder naming:

- `<STORAGE.FOLDERNAME>`
  Generates a folder from the data storage settings (e.g. *2022-01-01 08-00-00*)
  The result can change during a measurement, for instance when interval saving is activated. This placeholders ensures that each measurement result is assigned its own folder.
- `<STORAGE.MEASUREMENT>`
  Returns the data and time of the measurement start (e.g. *2022-01-01 08-00-00*). The result remains unchanged until the end of the measurement. By means of this placeholder, it is possible for instance to assign each measurement its own permanent folder.
- `<VARS.VALUE>`
  Returns a variable's value. The placeholder may serve as its own counter or as a metadatum representing the module serial number for the path.
- `<SETUP.SQL>`
  Returns the value of a cell in a Setup-page. This could be uses as, for instance, a metadatum representing the module serial number for the path.

Examples - Custom folder naming

In all examples, no custom folder naming for measurement data storage is used. This means that the Experiment-folder is used.

Additionally, the database for all examples was moved to the following path: *c:\DB*

Example 1

\Results\<STORAGE.FOLDERNAME>

- Measurement results for the experiment: *Experiment_0004* are saved as follows, for example:
  C:\DB\StandardProject\Experiment_0004**\Results\2022-01-01 08-00-00**
  C:\DB\StandardProject\Experiment_0004**\Results\2022-01-01 09-00-00**
  C:\DB\StandardProject\Experiment_0004**\Results\2022-01-01 10-00-00**
- In the Data Browser, the results are displayed as follows:
  Results.2022-01-01 08-00-00
  Results.2022-01-01 09-00-00
  Results.2022-01-01 10-00-00

Example 2

<STORAGE.MEASUREMENT>\<STORAGE.FOLDERNAME>

Interval saving (1 min) is activated. After every three intervals the measurement is stopped.

- Measurement results for the experiment: *Experiment_0005* are saved as follows, for example:
  C:\DB\StandardProject\Experiment_0005**\2022-01-02 08-00-00\2022-01-02 08-00-00**
  C:\DB\StandardProject\Experiment_0005**\2022-01-02 08-00-00\2022-01-02 08-01-00**
  C:\DB\StandardProject\Experiment_0005**\2022-01-02 08-00-00\2022-01-02 08-02-00**
  C:\DB\StandardProject\Experiment_0005**\2022-01-05 08-00-00\2022-01-05 08-00-00**
  C:\DB\StandardProject\Experiment_0005**\2022-01-05 08-00-00\2022-01-05 08-01-00**
  C:\DB\StandardProject\Experiment_0005**\2022-01-05 08-00-00\2022-01-05 08-02-00**
  C:\DB\StandardProject\Experiment_0005**\2022-01-10 08-00-00\2022-01-10 08-00-00**
  C:\DB\StandardProject\Experiment_0005**\2022-01-10 08-00-00\2022-01-10 08-01-00**
  C:\DB\StandardProject\Experiment_0005**\2022-01-10 08-00-00\2022-01-10 08-02-00**
- In the Data Browser the results are displayed as follows:
  2022-01-02 08-00-00.2022-01-02 08-00-00
  2022-01-02 08-00-00.2022-01-02 08-01-00
  2022-01-02 08-00-00.2022-01-02 08-02-00
  2022-01-05 08-00-00.2022-01-05 08-00-00
  2022-01-05 08-00-00.2022-01-05 08-01-00
  2022-01-05 08-00-00.2022-01-05 08-02-00
  2022-01-10 08-00-00.2022-01-10 08-00-00
  2022-01-10 08-00-00.2022-01-10 08-01-00
  2022-01-10 08-00-00.2022-01-10 08-02-00

Example 3

```
\Object <SETUP.SQL("SELECT TestPartNo FROM Project")>\Measurement
<VARS["Measurement_No"].VALUE("0")>\
```

The *SETUP*-placeholder returns the content of the column *Test part number*.
The *VARS*-placeholder returns the content of the variable *Measurement_No*.

Attention: The user must ensure that the Setup column is filled and that the variable has a value, which is raised automatically once per measurement.

- Measurement results for the experiment: *Experiment_0006* are saved as follows, for example:
  C:\DB\StandardProject\Experiment_0006**\Object 12\Measurement 1**
  C:\DB\StandardProject\Experiment_0006**\Object 12\Measurement 2**
  C:\DB\StandardProject\Experiment_0006**\Object 12\Measurement 3**
  C:\DB\StandardProject\Experiment_0006**\Object 13\Measurement 1**
  C:\DB\StandardProject\Experiment_0006**\Object 14\Measurement 1**
- In the Data Browser, the results are displayed as follows:
  Object 12.Measurement 1
  Object 12.Measurement 2
  Object 12.Measurement 3
  Object 13.Measurement 1
  Object 14.Measurement 1

Note — Point in time for determining target folder

The target folder is only determined once data which are to be saved arrive at the PC. Up until this point in time, it is possible to control the folder path. E.g. by means of the Setup pages. When columns of this Setup page are used to set the path (via `<SETUP.SQL>`), the new values are observed.

Using the menu action: "*Suspend/Resume data storage*", data storage is interrupted/resumed. Upon the start of the resumed storage, the target folder is also determined again from the beginning.

This means you can change the target folder during the measurement.

FAQ

Why not use a <SYSTEM.*> placeholder?

Answer: The `<STORAGE.*>`-placeholders should be used here instead of the `<SYSTEM.*>`-placeholder. The `<STORAGE.*>`-placeholders are resolved once and then apply to all devices. The `<SYSTEM.*>`-placeholders are resolved anew each time. This means that they are resolved again for each device and in consequence that may lead to a different folder for each device.

The results of previous measurements are deleted although their origin is a different experiment

Answer: If you have activated interval-saving, it doesn't matter where the measurement results located in the target folder come from.

If the specified amount has been reached, older measurement results are deleted.

This can also happen when different PCs use the same measurement data folder and the same experiment name.

9.7.10 Controlling data storage

9.7.10.1 Targeted data saving, or saving subsequent to measurement

In addition to the optional automatic data saving, you can also perform targeted saving of the data currently in the PC (Current measurement) either during or subsequent to the measurement.

Targeted saving of the current measurement is only possible until a new measurement has been prepared or started!

In order to save the measured data in this way, select the ribbon item "*Save current data / Export current data*".

Ribbon	View
Project > Save current Data ( )	Complete
Home > Save current Data ( )	Standard

Ribbon	View
Project > Export current Data ( )	Complete

Note

No single values can be saved in this way. These include, for instance, Display variables.
Be aware that with this function, only such variables are saved, which are found in the Data Browser under **Current measurement**. For these variables, a **circular buffer memory** is usually activated, so that not all results since the measurement start are available.
For device-variables, therefore, the Data Transfer Settings 370 are applicable. The storage type: "*Display, calculation of measured data*" is used.

Save current data

Using "*Save current data*", the measured data are saved in the database. If additionally the saving of measured data was activated, the results of data saving and of targeted saving are located in parallel next to each other.

When the button is clicked, a dialog appears in which the folder name can be entered. The folder is created and whatever measurement results are already available at the time are saved.

The **file format** used for saving is "*.dat*". Conversion of the measured data to other formats is possible by means of "*Export current data*" or by means of the imc Format Converter.

Saved measurement data are secure

Any measured data which are already saved cannot be overwritten by the menu action "*Save current data*". If a repeat attempt is made to perform "Save" to an existing file, the system refuses to perform this action. It is also not possible to perform in conjunction with measured data belonging to a continually saved measurement.

Export current data

When the button is clicked, a folder selection dialog appears, in which the target folder can be selected. The measurement results available up to that point in time are saved there.

The files are exported in the configured file formats: "*Options*" > "Save current data" > "*Variable export* 137".

Options

In the Options 136, it is possible to configure how the data saving works.

- **Folder name / Path**
  Here, a folder can be specified or a path entered, in which to save the variables. It is a preferable choice to use placeholders such as `SYSTEM.TIME`, so that the path can vary. Otherwise, the results will always be overwritten. In this way, you can also integrate the metadata into the folder path.
  Path: You can pre-define the data storage path in which the data storage dialog is to launch. If the folder doesn't exist, it is created temporarily. But if you don't save the files in it, the folder is deleted once the dialog is closed. However this only happens when it had been created by this method and if it is really empty.
- **Show dialog**
  The dialog for entering the folder name / path can be deactivated. The condition for this is: the option "*Folder name / Path*" has been filled in.
- **Save to one file per file format (Save all in a single file)**
  If this option is activated, all variables are saved to a single (multi-channel) file. The file name can be specified. The file format for the file does not need to be specified. However, if no file name is specified, all variables are automatically saved to the file *data.dat*. Furthermore, for each file format a separate file is created.
  If the option is not activated, each variable is saved in a separate file.
- **Overwrite existing files without confirmation prompt**
  (For export only) - If this option is activated, files having the same name in the destination folder will be overwritten without a confirmation prompt. This option is useful for automated routines.
- **Mapping instruction**
  If desired, you can provide you own personal mapping instruction according to which to determine the channel type selection.

9.7.10.2 Suspend and resume data storage

If the saving of measured data is activated 370, then you are able to govern the data storage **during the running measurement**. You can **interrupt (suspend) data storage** and **resume it again** at a later time.

Ribbon	View
Home > Suspend data storage ( )	all
Setup-Control > Suspend data storage ( )	Complete
Home > Resume data storage ( )	all
Setup-Control > Resume data storage ( )	Complete

Effect

If you **suspend data storage**, the **files are closed** and appear in the Data Browser (as if the measurement had been concluded). Any **more data** which arrive or are calculated subsequently **are not saved**.

When you **resume data storage**, then **new files are created** in a **new measurement folder** (as if the measurement had just been started). Any data arriving before "resumption of data saving" are not saved.

This function is designed for **interrupting data saving for longer periods of time**. Please do **not use it to make "cuts" of data records** (concluding one measurement folder and immediately beginning a new one).

Note

- Data storage **con only be controlled** if it was previously **activated**.
- This **suspension/resumption** of data storage **is not synchronized** across the devices! Thus, the respective **data excerpt boundary points** of different devices/channels **may differ by a few measurement points**.

Scope

These actions **apply to all devices and to the PC**. No separation is possible. Any additional components which return data to be saved (such as 3rd-party devices) are also affected.

Note Video-files are not affected

The **saving of Video files** is **not affected** by this function. Video files are always recorded.

Reference Data storage state after preparation

Using the option "Data storage state after preparation", you can control the function's state. E.g. it is possible to make data storage generally active following "Preparation".
See the option: "*Setup*" > "*General*" > "*Data storage state after preparation* 134".

9.7.10.3 Close folder - imc Online FAMOS function CloseSaveInterval

If imc Online FAMOS is set to control commands, you'll find the function `CloseSaveInterval` in group **system**. That command is closing the current folder of measurement data and create the next folder. The function is executed with a delay, that depend on the load of the device.

9.7.11 Information and tips

9.7.11.1 The last measurement

In many cases, various reports are composed and analysis procedures as well as other actions are performed based on the data from the last measurement conducted. Many of these actions are performed automatically. In order for this to be possible, these actions must be able to recognize the "Last measurement".

Toward this end, there are various approaches.

1. You can select the measurement in the Data Browser, which assigns a measurement number to it (this happens automatically). There are many places in which you can access a measurement by this measurement number. This makes automated accessing of all measurements possible, including of the last measurement.
2. In various commands and actions, you are able to select whether you wish to access the "Last measurement". Internally, a record is stored of which measurement was last (recommended).

What data becomes the "last measurement"?

- A measurement folder closed by, for instance:
  - conclusion of a measurement
  - conclusion of an interval
  - saving of measurement data concluded due to interruption.
- Data saving by means of the menu item "*Save current Data*": The system considers this a measurement and it can become the "Last Measurement".

Notes

Each measurement folder represents a measurement

The last measurement is always only "**one**" folder. Thus for example, when interval data saving is activated, each interval is a separate measurement. It is always the last interval concluded which is considered the last measurement.

How long is the info stored?

The path to the measurement remains known to the system even when the **experiment is re-loaded**. When the **last measurement is deleted**, the measurement performed right before it becomes the "Last Measurement".

There are a variety of ways to access the last measurement:

- Placeholders 174: e.g. <MEASUREMENTS["MEASUREMENT#LAST"].PATH>
- "imc FAMOS-command 1708" (data source)
- "Set Measurement Number 1729"-command

Does imc STUDIO contain information once a measurement has been concluded?

Upon conclusion of a measurement (or, if applicable, of an interval), the "*MeasurementFolder_Closed*"-event is triggered. This event is found among the Sequencer-events 1694. After this event, you are able to run various commands.

Special cases:

- Measurement data folders created by different methods: In the Data Browser, only such folders are considered the "Last Measurement" if they were genuinely created by running a measurement. Whenever folders are created in a different way, e.g. imc FAMOS-events or by means of Scripting, etc., they are not treated as the "Last Measurement".
- Subsequent adding of files: When imc FAMOS-results are saved to an already existing measurement, this does not count as a "new" "Last Measurement".

9.7.11.2 Incomplete measurement files

Measurement files being written during the measurement process are not complete and can neither be deleted or copied. If either of the options "*Saving interval*" or "*Save trigger events in individual files*" is used, it is possible to access the data even during a running measurement. However, this only applies to data from completed events. The current interval or trigger event is subject to the same limitation.

For storage in device: Once all files have been concluded, the system creates an additional file "*DirClosed* 417". As soon as it is present, you have reliable access to the measured data.

9.8 Device hard disk, removable drive

This section describes how to handle the storage media of the imc measurement devices and how to use them with imc STUDIO.

For saving measured data, some devices support an internal storage medium. Depending on the device type, the storage medium could be, for example, a removable drive or a hard drive (see "*Device overview* 187"). Card-Bus interface are not recommended.

The storage media are exclusively for data acquisition with imc STUDIO.

Storage media with verified performance can be purchased as accessories from imc. Hard drives are ordered with the device and can only be installed subsequently by imc.

For devices of firmware groups A and B (see "*Device overview* 187"), a distinction is made between how the storage media are to be handled.

- For devices of the firmware group B (imc DEVICEcore) 443
- For devices of the firmware group A (imc DEVICES) 446

9.8.1 For devices of the firmware group B (imc DEVICEcore)

Swapping the storage medium

Pressing the button signalizes to the system that you intend to remove the storage medium. Once this is done, the device stops access to the storage medium. If you were to remove the drive without prior announcement, it could produce defective clusters. If the storage medium is removed while a measurement is in progress, the data records are not completed. Therefore, always proceed as follows when swapping the storage medium:

1. **Important!** Before removing the storage medium from the measurement device, first announce the procedure to the system by pushing the button, in order to **avoid damage** to the storage medium.
2. Once the LED blinks, remove the storage medium.
3. Insert the new storage medium. Devices indicate by a short flash that the new drive has been successfully recognized.

Hot-Plug (exchanging the storage medium during a measurement)

It's possible to exchange the storage medium during a running measurement. This makes it possible to carry out a measurement without a PC practically without any limitations.

While swapping the storage medium during a running measurement, the data are stored in the measurement device's internal memory. If you complete the process within the specified RAM buffer duration, this is certain to work without any loss of data (see section "*RAM buffer time* 427"). Note that not only the time for the swap must be buffered, but that the buffered data must also be transferred to the new disk once the swap has been completed.

Swapping the storage medium

1. **Important!** Before removing the storage medium from the measurement device, first announce the procedure to the system by pushing the button, in order to **avoid data loss and damage** to the storage medium. The LED will **shine continuously** in green.
2. Once the device is ready for removal of the storage medium, the LED **blinks**.
3. Remove the already full storage medium.
4. No announcement is necessary for inserting a storage medium.

Errors in accessing the storage medium

Errors can have the following causes, among others:

- The data rate is too high, the storage medium can't keep up and data overflow results.
- The storage medium is full.

If the PC is connected to the measuring device, imc STUDIO documents the error with a **message in the Logbook**.

9.8.1.1 Storage media

Storage media	Description
SSD (Internal hard disk)	Applicable to devices having a hard drive (see "*Device overview* 187"). • SSD hard drives can only be formatted in the device. • The specified operating temperature range of the media is relevant.
microSD	Concerning devices with microSD slot (see "*Device overview* 187"). • Only microSD storage media verified by imc should be used, as these have been specially qualified by us for high data rate of 5 MS/s. • The specified operating temperature range of the media is relevant.

9.8.1.2 Data transfer

The storage medium can be accessed **directly via Windows Explorer**. Alternatively, the storage medium can be inserted into a **card reader** on the PC (suitable for large amounts of data due to faster transfer).

Warnung

Do **not use force** to insert or remove the device storage medium.

Access via Windows Explorer

The menu item "*Data (Device)*" ( ) causes the Windows-Explorer to start while indicating the device selected.

Ribbon	View
Extra > Data (Device) ( )	Complete
Home > Data (Device) ( )	Standard

Using this menu item, a separate Windows-Explorer is opened for each device, providing direct access to the internal memory.

For the purpose of accessing the tool, it is necessary to log in. The factory default settings are:

- User: "*imc*"
- Password: device serial number

Note Manual entry

The device is called by reference to either its IP address or its internal name. The device's name is constructed as follows: "*imcDev*" + SerialNumber; e.g.

\\imcDev4120110
\\10.0.12.70

9.8.1.3 Formatting

The data volume (microSD) **can not be formatted in the device**.

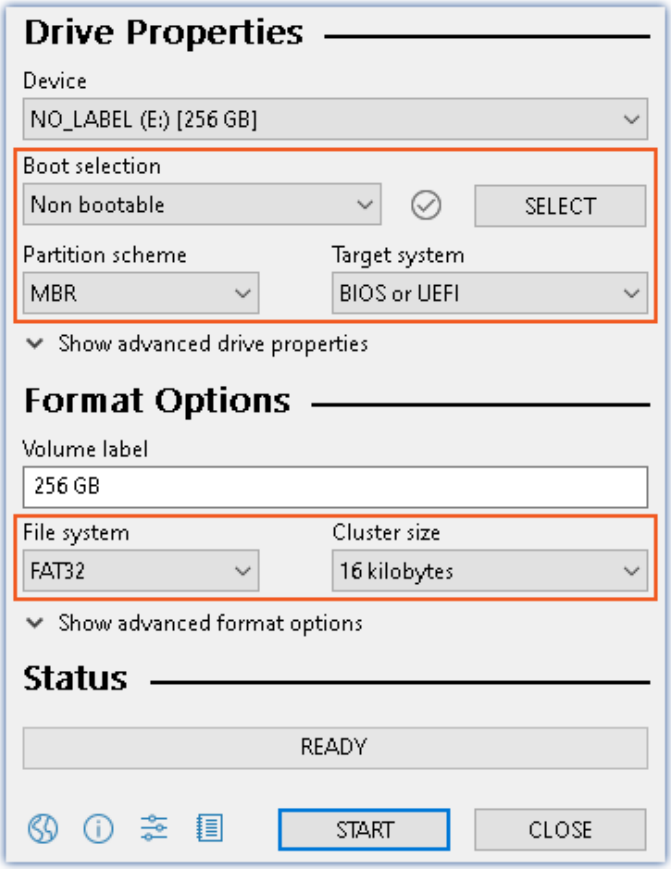
For this purpose, please use an appropriate tool. Such a tool is provided on the installation medium for imc STUDIO in the folder "*Tools\RemoveableDriveFormatter*":

- "*rufus-4.1p.exe*"

This program can be used for the purpose of formatting the data volume.

Make the **following settings** for the data volume:

- Boot selection: "*Not bootable*"
- Partition scheme: "*MBR*"
- File system: "*FAT32*"
- Cluster size: "*16 kilobytes*"



Settings required for formatting by means of Rufus 4.1

9.8.1.4 Known issues and limitations

The file system becomes slower and slower.

Creating more than 1000 folders is to be avoided, since this significantly slows down the file system.

9.8.2 For devices of the firmware group A (imc DEVICES)

Swapping the storage medium

Pressing the button signalizes to the system that you intend to remove the storage medium. Once this is done, the device stops access to the storage medium. If you were to remove the drive without prior announcement, it could produce defective clusters. If the storage medium is removed while a measurement is in progress, the data records are not completed. Therefore, always proceed as follows when swapping the storage medium:

1. **Important!** Before removing the storage medium from the measurement device, first announce the procedure to the system by pushing the button, in order to **avoid damage** to the storage medium.
2. Once the LED blinks, remove the storage medium.
3. Insert the new storage medium. Devices indicate by a short flash that the new drive has been successfully recognized.

Hot-Plug (exchanging the storage medium during a measurement)

It's possible to exchange the storage medium during a running measurement. This makes it possible to carry out a measurement without a PC practically without any limitations. It is only necessary to check the amount of memory available using imc Online FAMOS. To do this, use the function `DiskFreeSpace` belonging to the group "*System*". You can set an LED, for instance, or a digital output or a beeper to be activated when less than the minimum amount remains. One convenient solution would be to have a readout of the remaining space outputted by a display variable, which would indicate by a display on the device how the remaining memory decreases.

While swapping the storage medium during a running measurement, the data are stored in the measurement device's internal memory. If you complete the process within the specified RAM buffer duration, this is certain to work without any loss of data (see section "*RAM buffer time* 427"). Note that not only the time for the swap must be buffered, but that the buffered data must also be transferred to the new disk once the swap has been completed.

Swapping the storage medium

1. **Important!** Before removing the storage medium from the measurement device, first announce the procedure to the system by pushing the button, in order to **avoid data loss and damage** to the storage medium. The LED will **shine continuously** in green.
2. Once the device is ready for removal of the storage medium, the LED **blinks**.
3. Remove the already full storage medium.
4. No announcement is necessary for inserting a storage medium.

Errors in accessing the storage medium

Errors can have the following causes, among others:

- The data rate is too high, the storage medium can't keep up and data overflow results.
- The storage medium is full.

The device signalizes any error by flashing this LED. Its further responses depend on whether or not the device is connected to the PC.

- If no PC is connected, for instance in cases involving automatic self-start capability, the **button lights continuously**. At the end of an experiment, always check for this if measurements are taken without PC aid.
- If the PC is connected to the measuring device, imc STUDIO documents the error with an **message in the Logbook** and switches the LED off. Any one-time data overflow only shows up in the Logbook, since the LED is reset afterwards. If data overflow occurs repeatedly, The LED is activated again, the PC records the message again, and as a result the LED **blinks intermittently**.

9.8.2.1 Memory volumes

Storage media larger than 16 GB are possible (see "*Device Overview* 187"), but see the notes on the use of the storage media 755. When the formatting is FAT32, the maximum address space is 8.8 TB.

9.8.2.2 Storage media

Storage media	Description
CF Cards (Compact Flash)	For devices of group A4 and A5 187: The device group exclusively uses CF cards for storage medium.
USB Storage Medium	Concerning devices with USB (see "*Device overview* 187"). Memory sticks or external hard drives can be connected at this terminal. • **Always set only one storage medium unit!** Devices belonging to group A6 187 have two USB terminals and a slot for the ExpressCard. However, the device can use only one data storage medium. The system determines which one it is upon activation, and there is no fixed order of precedence among the media. For this reason, delete any which you do not wish to use for the measurement before switching the device on. • With USB, the **Hot-Plug** 446 functionality is **available**. Make sure that sufficient time is available for swapping the storage medium. How much time is required for de-registering and re-registering with the system depends on the particular storage medium and on the number of channels. As an orientation value, we recommend at least 30 s, even for simple configurations! **Do not use USB hard drives with external power supply** Please do **not use** any USB hard drive **which has an external power supply**. Such a drive may not be connected at the imc USB-port. Otherwise, when the measurement device is powered down, the imc USB-port's current limiting mechanism may be destroyed.
ExpressCard	Applicable to devices having ExpressCard slot (see "*Device overview* 187"). • **Remove any connected USB storage medium!** Devices belonging to group A6 187 have two USB terminals and a slot for the ExpressCard. However, the device can use only one data storage medium. The system determines which one it is upon activation, and there is no fixed order of precedence among the media. For this reason, delete any which you do not wish to use for the measurement before switching the device on. • With ExpressCards, the **Hot-Plug** 446 functionality is **available**.
CFast	Concerning devices with CFast slot (see "*Device overview* 187"). • **Remove a connected USB storage medium!** The device can use only one data storage medium. The system determines which one it is upon activation, and there is no fixed order of precedence among the media. For this reason, delete any which you do not wish to use for the measurement before switching the device on. • With CFast cards, the **Hotplug** 446 functionality is **available**.
SSD	Applicable to devices having a hard drive (see "*Device overview* 187"). • With SSD hard drives, **Hot-Plug** 446 **is not possible**! If the SSD is used in a **frame for removable data carrier**, it can be exchanged while the **device is deactivated**. • SSD hard drives appear in the device software as a hard drive and an be read out via the Explorer-shell 448. • Due to the formatting, the content of the SSD in the PC is **not displayed** when the SSD is connected directly in the PC. SSD hard drives can only be formatted in the device 452. • **In addition to** the SSD, a CF/CFAST-card can be inserted in the measurement device and used **alternatively**.

9.8.2.3 Data transfer

The internal storage medium can be accessed **directly via Windows Explorer**. Alternatively, the storage medium can be inserted into a **card reader** on the PC (suitable for large amounts of data due to faster transfer).

Warning

- Do **not use force** to insert or remove the device storage medium.
- During a **running measurement** having a high sampling rate, you should **never** try to **access the storage medium in the device** using the Windows Explorer shell. Otherwise, this additional burden could cause a data overflow.

Note

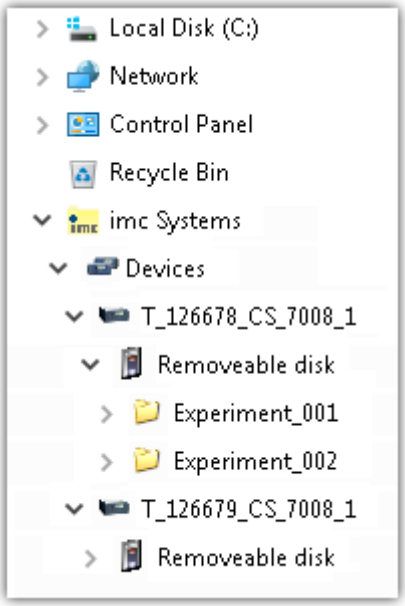
A tip on **interval saving**: Supposing the system's power supply suffers an outage during a measurement, it would not be possible to guarantee that the data file on the storage medium is terminated properly. This may lead to a failure to record the last measurement taken. Saving at intervals is a way to reduce this risk (see section "*Saving Interval* 423").

Access via Windows Explorer

The menu item "*Data (Device)*" ( ) causes the Windows-Explorer to start while indicating the device selected.

Ribbon	View
Extra > Data (Device) ( )	Complete
Home > Data (Device) ( )	Standard

Access via "*imc Systems*" - an Explorer shell extension



While installing the operating software, if the option "*Extension for Windows-Explorer* 26" is activated, you are able to copy, display or delete the files of measured data saved within the device (e.g. on the removable storage medium). The method of doing this is the familiar one under Windows.

This function is independent of the device software. As well, selection of the devices in the tree diagram is independent of the device list in the operating software.

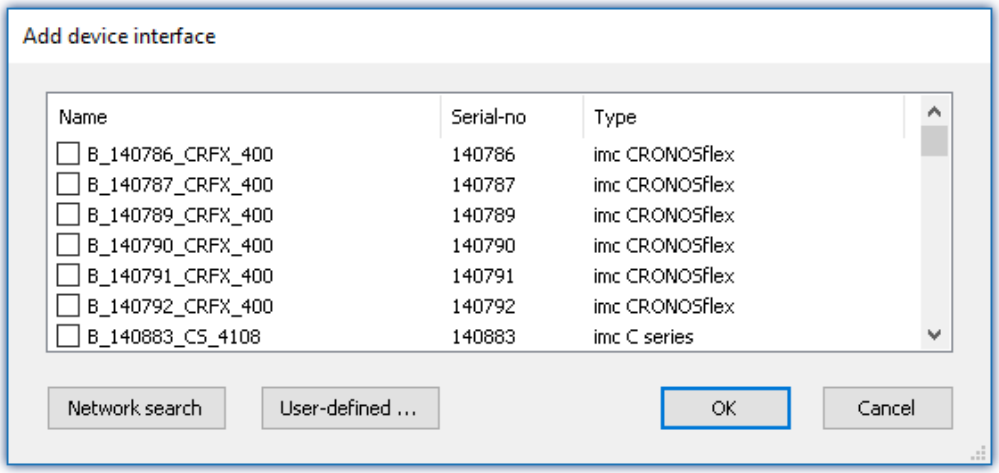
In order to get access to your device's storage medium, you must first add it to the tree diagram (see "*imc Systems - Adding a device* 449"). Subsequently, you are able to navigate to the desired data in the storage medium and thus to work with them.

9.8.2.3.1 imc Systems - Adding a device (Add)

Even if you have already been connected with the device by means of the imc STUDIO software, it is still not listed in the Explorer. It's possible to measure with one device while copying data from another.

- Click on "*Devices*" under "*imc Systems*" to highlight that entry.
- Open the context menu over the "*Devices*"-area and select "*Add*".

The "*Add device interface*" dialog appears:



Add Device interface

Search for devices	Description
Network search	"*Network search*" causes the system to search the network for any suitable devices. How long this will take depends on how many devices are connected and on the network type. Ultimately, the devices found are listed. Select your measurement device and confirm your selection with "*OK*". The measurement device is then available.
User-defined	In a structured network (network with routers, Internet, ...), imc devices could not be integrated by means of a network search. With the knowledge of the IP address or of the domain name (DNS name), it is now possible to integrate a device into the list.

Reference

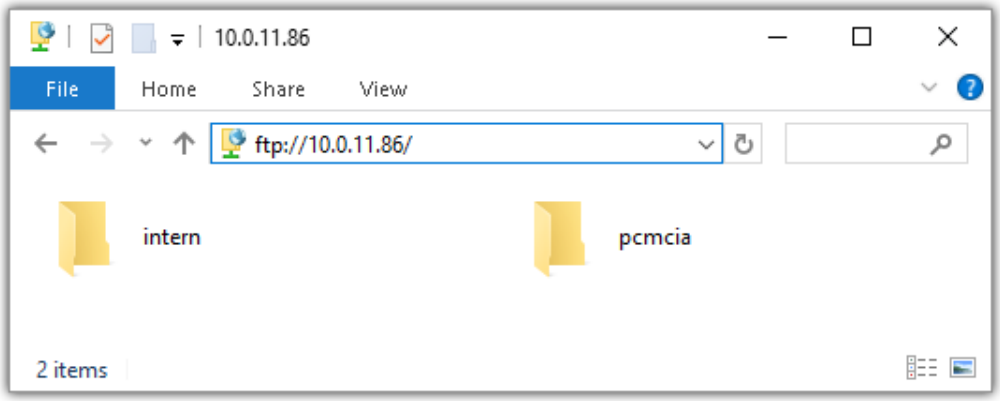
More information is presented in the section: "*Setting Up - Connect the device*"

- General: "*Device connection via LAN* 55"
- "*Connection via a direct address* 58"

9.8.2.3.2 FTP access

It is also possible to access the device's internal data via FTP, as well as to transfer data. Other goals are to change the configuration of devices via FTP and to restart the device for measurement with the altered configuration. Application areas include test drives, where there is no way to connect the devices directly with the device software. The Diskstart/Autostart capabilities are applied and enhanced. For this purpose, the device is usually configured for autostart. Upon activation, the configuration is loaded and the measurement, as well as data transmission, starts automatically.

Start the explorer and enter "*ftp://*" and the IP address of the device:



Note

- In general it is a read only operation. If you intend to delete files via FTP, you have to add "*imc@*" between "*ftp://*" and the IP address:
  Example: ftp://imc@10.0.10.219
- Furthermore, a password can be assigned to protect access via FTP. This password is entered into the device properties 222.

Warning

The following limitations apply when accessing the storage media in a device via an FTP-client:

- The device can't delete folders, accessed by a FTP-client.
- It is not possible to replace the storage medium during measurement (Hot-Plug).

9.8.2.4 Partition and file system

Fundamentals of FAT16/FAT32

Before using a data storage medium, it should be formatted and maybe also partitioned. The devices support both FAT32 and FAT16 (maximum 2 GB) as file systems.

Note

Regular formatting is recommended

Take every opportunity to format the storage medium. In this way, any damaged media are recognized in time and repaired if necessary. **A damaged file system may cause data loss or to failure of the measurement system to start up correctly.**

To avoid data loss, begin by saving all necessary data!

Using a storage medium in different kinds of device

There are no known limitations. However, **formatting is always recommended** when changing, as a precaution against data loss.

Additional notes

- To select the appropriate file system for the respective application, observe the notes on the data rate and on "*Avoidance of Data Overflow* 452".
- No limitations regarding the currently available storage medium volumes are known.
- The maximum filesize is 2 GB. In case a signal would exceed that limit, use interval saving 423.

Reference **General limitations of the file systems**

Please be aware of the general restrictions pertaining to the respective file system. Information on this topic is presented, among other places, at the Help and Supportarea on the Microsoft homepage.

9.8.2.4.1 Formatting

The formatting can be performed directly in the PC's hard drive by the Windows operating system, or in the device using the Explorer shell.

Note — Recommendation

imc recommends formatting in the device: In comparison to formatting by Windows, this provides higher data writing rates for high-speed channels.

Only **one(!)** partition may be created. Multiple partitions may cause the measuring device not to recognize the storage medium.

Warning — Please back up the data first

Formatting causes all data on the storage medium to be deleted. Before performing the formatting, ensure that all data have been saved on a different storage medium.

Note — Cluster size - Avoidance of Data Overflow

The size and number of assignment units (clusters) and thus also the file system 452 used have a substantial effect on the storage medium's speed! Small clusters can dramatically reduce the speed! If high data rates are required, it is normally recommended to have a size of 8 kB/cluster.

The optimum size of the clusters must be determined for each storage medium separately. For all of them, the following applies:

- **Few channels having a high data rate**

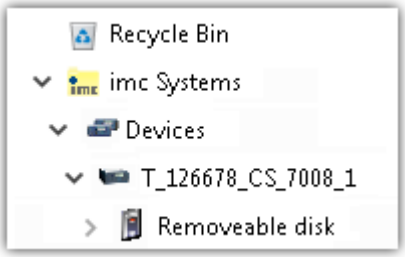
  If a few channels having a high data rate are being recorded, then **large clusters** on the data carrier provide better advantage. Formatting with FAT32 on the PC and drive sizes < 8 GB creates disadvantageously small clusters, which in conjunction with the full aggregate sampling rate can lead to a data overflow.

  Whenever using cards of up to 8 GB, always use the formatting by the device.
  In the device, cards larger than 512 MB are formatted with 8 kByte clusters and cards larger than 4 GB are formatted with 16 kByte clusters. Cards of up to 1 GB can alternatively be formatted by the PC with FAT16. With cards of 16 GB onward, there is no difference whether the formatting is done in the PC or in the device.

- **Very many channels with a low data rate**

  If hundreds of channels having a low data rate (e.g. CAN channels) are saved, the exact opposite is rue. Here, **small clusters** are an advantage. This means that drives with up to 8 GB should in such cases be formatted in the PC with FAT32.

Formatting in the device (Recommended)



For **formatting in the device**, navigate via the Explorer shell "*imc Systems* 448 " to the desired device.

There, open the properties of the drive: context menu > "*Properties*" (not via the Navigation pane in the Explorer).

Go to the Property dialog under the tab: "*Tools*".

Start the formatting by clicking on "*Format now!*".

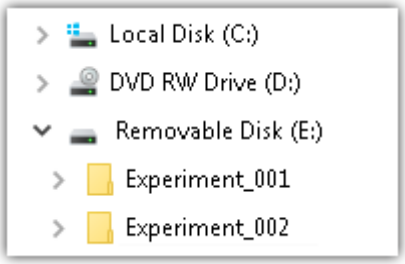
The device performs the formatting according to the following rule:

Drive size	Cluster size	File system
<= 512 MB	2 kB	FAT16
<= 4 GB	8 kB	FAT32
> 4 GB	16 kB	FAT32

Note

Formatting the storage medium is not allowed if an experiment whose data are to be saved internally has just been prepared.

Formatting using the Windows-Explorer



To perform **formatting of a storage medium via the Windows-Explorer** 449, navigate to the desired storage medium. Run the formatting by means of the context menu, for example.

Select one of the following two file systems: "FAT32" or "FAT" ("FAT16").

The file system "FAT32" is designed for media **larger** than 32 MB. Under no circumstances can smaller media can be formatted to "FAT32". With "FAT32", Windows generates 4 kByte clusters when the drive size is up to 8 GB, which is inconvenient for high-speed writing rates.

SSD hard drives are inherently formatted with Ext2 and for that reason **can not be formatted directly in the PC** , but only in the device 452.

However the Ext2 format offers these advantages:

- Mistaken duplicate occupation of individual clusters is not possible.
- Integration into the operating system is accomplished much more quickly than with FAT32.
- Higher writing output than with FAT32.

9.8.2.4.2 Possible problems

Due to the limitations discussed previously, the following problems can occur:

- It is possible that the increasing number of folders 757 slows down the saving process in a way that a loss of data can't be avoided.
- With a storage medium formatted by Windows to FAT32 452, data overflow can occur if a high aggregate sampling rate is generated by a few high-speed channels.
- With a storage medium formatted in the device 452, data overflow can occur if a high aggregate sampling rate is generated by very many low-speed channels.

Other limitations are presented here 454.

9.8.2.5 Known issues and limitations

If the memory card can not be read under Windows:

The memory cards must first be partitioned (formatted) under Windows. Windows automatically generates the correct partitioning information. Subsequently, the memory card should be formatted again in the device.

When in doubt, please contact our tech support 8.

The system won't recognize the data storage drive.

Answer 1: Devices support FAT32 450.

Answer 2: If there are two storage media simultaneously plugged into your device, only one media will be detected (e.g. USB and CFast). Only the first plugged media will be detected.

The file system becomes slower and slower.

Creating more than 1000 folders is to be avoided, since this significantly slows down the file system.

9.9 Measurement types

9.9.1 Temperature measurement

Two methods are available for measuring temperature. Measurement using a **PT100** requires a constant current, e.g. of 250 µA to flow through the sensor. The temperature-dependent resistance causes a voltage drop which is correlated to a temperature according to a characteristic curve.

When measuring with **thermocouples**, the temperature is determined via the series of voltages of different alloys. The sensor generates a temperature-dependent voltage which is relative to the terminal point on the plug. To find the absolute temperature, the temperature of the terminal point must be known. This is determined with a **PT1000** directly in the terminal plug and requires a special plug type.

The measured voltage is converted into the displayed temperature value according to the characteristics of the temperature scale IPTS-68.

Note Making settings with imc software

A temperature measurement is a voltage measurement whose measured values are converted to physical temperature values by reference to a characteristic curve. The characteristic curve is selected using the "Correction" parameter on the " Measurement mode" tab. Amplifiers which enable bridge measurement, must first be set to the "*Voltage*" measurement mode in order for the temperature characteristics curves to be available for selection.

9.9.1.1 Thermocouples as per DIN and IEC

The following standards apply for thermocouples, in terms of their thermoelectric voltage and tolerances:

Thermocouple	Symbol	max. temp.	defined up to	(+)	(-)
	DIN IEC 584-1 (2014-07)				
Iron-constantan (Fe-CuNi)	**J**	750°C	1200°C	black	white
Copper-constantan (Cu-CuNi)	**T**	350°C	400°C	brown	white
Nickel-chromium-Nickel (NiCr-Ni)	**K**	1200°C	1370°C	green	white
Nickel-chromium-constantan (NiCr-CuNi)	**E**	900°C	1000°C	violet	white
Nicrosil-Nisil (NiCrSi-NiSi)	**N**	1200°C	1300°C	red	orange
Platinum-Rhodium-platinum (Pt10Rh-Pt)	**S**	1600°C	1760°C	orange	white
Platinum-Rhodium-platinum (Pt13Rh-Pt)	**R**	1600°C	1760°C	orange	white
Platinum-Rhodium-platinum (Pt30Rh-Pt6Rh)	**B**	1700°C	1820°C	n.a.	n.a.
	DIN 43710				
Iron-constantan (Fe-CuNi)	**L**	600°C	900°C	red	blue
Copper-constantan (Cu-CuNi)	**U**	900°C	600°C	red	brown

If the thermo-wires have no identifying markings, the following ***distinguishing characteristics*** can help:

- Fe-CuNi: Plus-pole is magnetic
- Cu-CuNi: Plus-pole is copper-colored
- NiCr-Ni: Minus-pole is magnetic
- PtRh-Pt: Minus-pole is softer

The color-coding of compensating leads is stipulated by DIN 43713. For components conforming to IEC 60584: **The plus-pole is the same color as the shell; the minus-pole is white**.

9.9.1.2 PT100 (RTD) - measurement

RTD (PT100) sensors can be directly connected in 4-wire-configuration. An additional reference current source feeds a chain of up to 4 sensors in series.

With the imc Thermo plug, the connection terminals are already wired in such a way that this reference current loop is closed.

Note

If fewer than 4 PT100 units are connected, the current-loop must be completed by a wire jumper from the "last" RTD to -I4.

If you dispense with the "support terminals" (±I1 to ±I4) provided in the imc Thermo plug for 4-wire connection, a standard terminal plug or any DSUB-15 plug can be used. The "current loop" must then be formed between +I1 (DSUB Pin 9) and -I4 (DSUB Pin 6).

9.9.1.3 imc Thermo plug (T4)

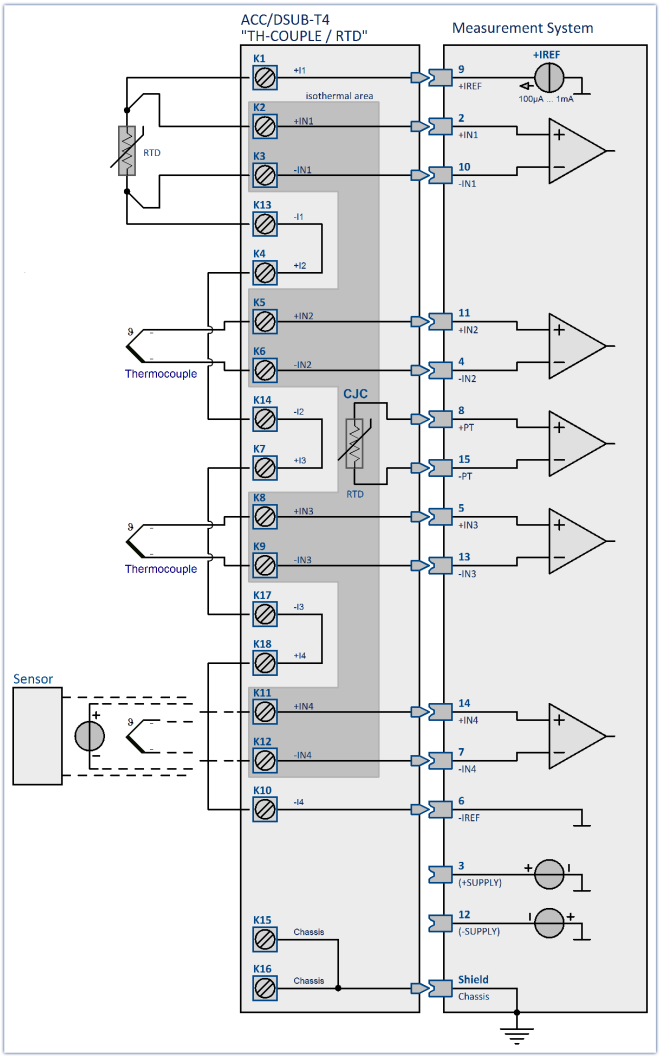
The imc Thermo plug contains a screw terminal block in a DSUB-15 plug housing with a built-in temperature sensor (PT1000) for **cold junction compensation**. This provides for direct connection of thermocouples of any type, directly to the differential inputs (+IN and -IN) without external compensation leads. That plug can also be used for **voltage** measurement.

The difficulty with thermocouple measurements are the "parasitic" thermocouples which inevitably form where parts of the contacts made of different materials meet. The temperature sensor measures the temperature at the connection terminal and compensates the corresponding "error"-voltage. Normally, the connection to this compensation point (inside the device) is made by special compensation leads or plugs made of material identical to the respective thermocouple type, in order not to create additional (uncontrolled) parasitic thermocouples.

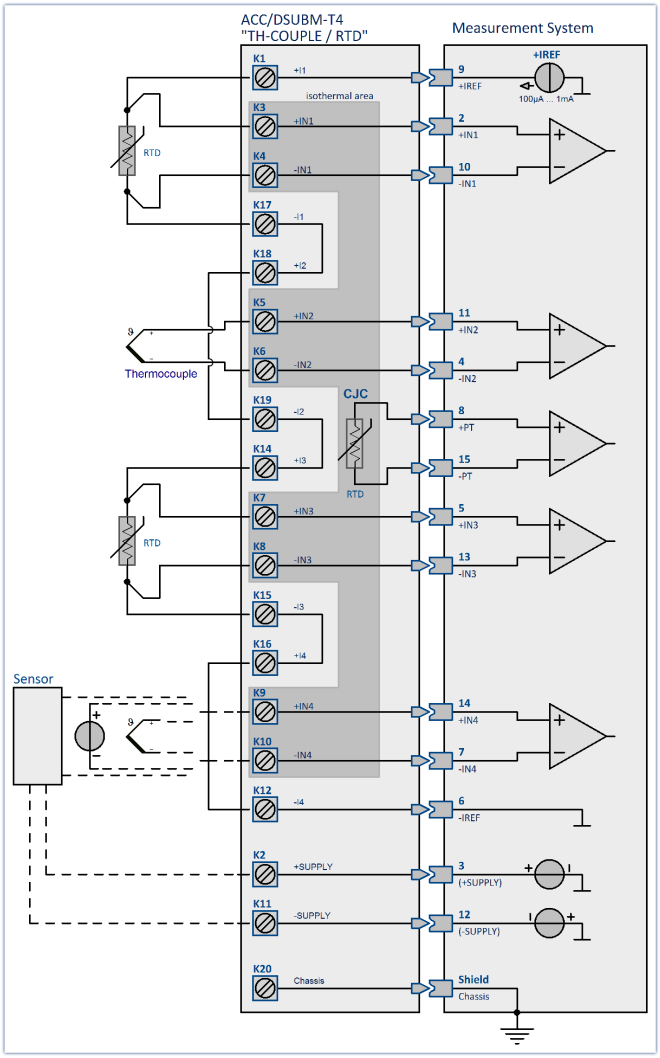
imc's system avoids the problem through the use of individual compensation sensors directly inside the plug, thus offering an especially simple, flexible and cost-effective connection solution.

9.9.1.3.1 Schematic: T4 plug

Plastic plug (ACC/DSUB-T4, discontinued)



Metal plug (ACC/DSUBM-T4)



9.9.2 Bridge measurement

Bridge channels are for taking readings from **measurement bridges** such as resistor bridges or strain gauges. The channels are equipped as **non-isolated differential** amplifiers and can alternatively be used for direct **measurement of voltages**.

There is a distinction among the following operating modes:

➢ **Sensor**
- Full bridge
- Half bridge
- Quarter bridge (120 Ω)

➢ **Strain gauge**
- Full bridge with 4 active strain gauges in uniaxial direction
- Full bridge with Poisson strain gauge in adjacent bridge arms
- Full bridge with Poisson strain gauge in opposing bridge arms
- Half bridge with one active and one passive strain gauge
- Half bridge with 2 active strain gauges in uniaxial direction
- Poisson half bridge
- Quarter bridge with 120 Ω strain gauge

9.9.3 Bridge measurements with wire strain gauges

Strain in this sense refers to the ratio of a body's original length to the change in length due to a force exerted upon it.

$$\varepsilon = \frac{dL}{L}$$

By selecting "*Strain gauge*" as the measurement mode, common bridge circuits and configurations for wire strain gauges are offered for selection. The scaling can be adjusted in terms of typical parameters for strain measurements such as the gauge factor or Poisson's ratio, the transversal expansion coefficient.

If a strain gauge adheres to a test object, the strain on the object is transmitted to the bridge circuit. The changes in the lengths of the bridge arms cause their impedances to change. There is a correlation between the changes in length and the changes in resistance:

$$\varepsilon = \frac{dL}{L} = \frac{dR/R}{k}$$

Legend:	
ε	strain
dL	change in length
L	original length
dR	change in resistance
R	resistance of strain gauge
k	Gauge factor, describing the ratio of relative length change to change in resistance

The changes in resistance caused by the strain are very small. For this reason, a bridge circuit is used to translate these changes into voltage changes. Depending on the circuit, from one to four strain gauges can be employed as bridge resistors.

Assuming that all bridge resistors have the same value, we have:

$$V_a = V_e \cdot \frac{dR}{4 \cdot R} = \frac{V_e}{4 \cdot R} \cdot k \cdot \varepsilon$$

Legend:	
Va	measurement voltage
Ve	excitation voltage

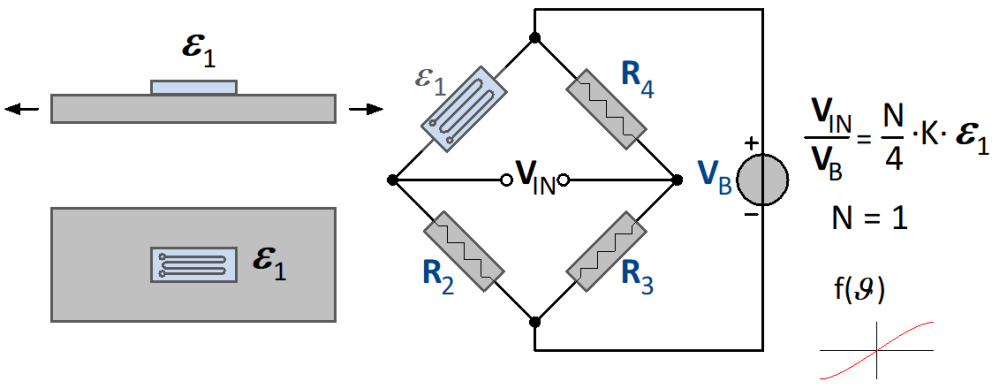
$$\varepsilon = \frac{V_a \cdot 4}{V_e \cdot k}$$

For concrete measurement tasks, the arrangement of the strain gauges on the test object is important, as well as the circuitry of the bridge. On the "Bridge circuit", you can select from among typical arrangements. A graphic below shows the position on the test object and the bridge circuitry. Notes on the selected arrangement are displayed in the text box.

Note

For an easier operation, ranges that are unsuitable are hidden in the operating software.

9.9.3.1 Quarter bridge for 120 Ohm strain gauge



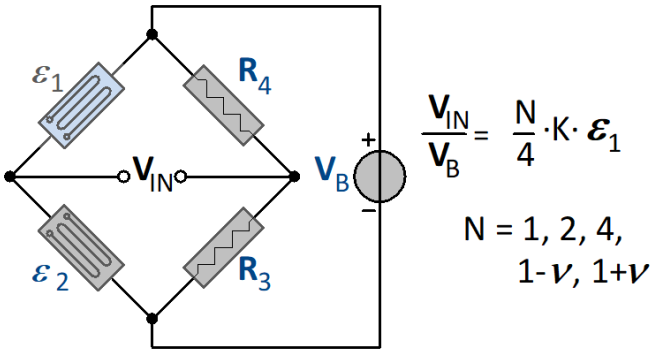
This strain gauge arrangement uses an **active** strain gauge which is positioned on the test object in a uniaxial stress field. This strain gauge is joined by three passive resistors within the module to form a full bridge. The strain gauge can have a resistance value of 120 Ω.

This arrangement does not come with temperature compensation. The strain is computed as:

$$\varepsilon\left[\frac{\mu\mathrm{m}}{\mathrm{m}}\right] = \frac{4 \cdot 1000}{k} \cdot \frac{V_a}{V_e}\left[\frac{\mathrm{mV}}{\mathrm{V}}\right]$$

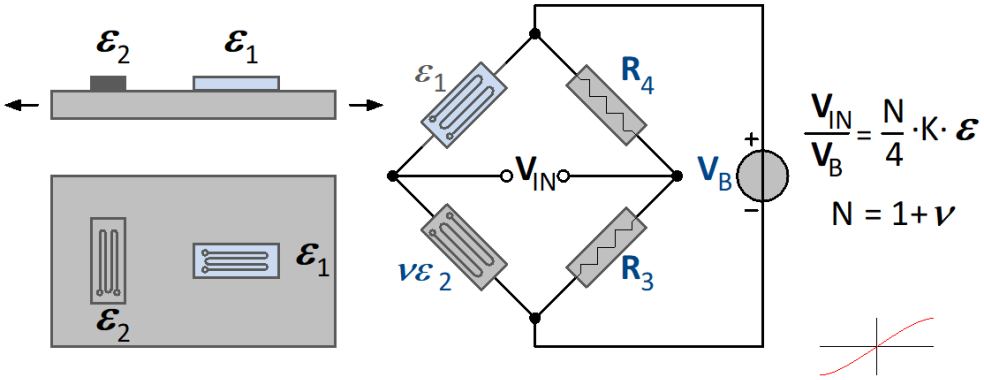
Legend:	
k	gauge factor

9.9.3.2 General half bridge



General half bridge with bridge completion in measurement device. N has to be set from a list.

9.9.3.3 Poisson half bridge

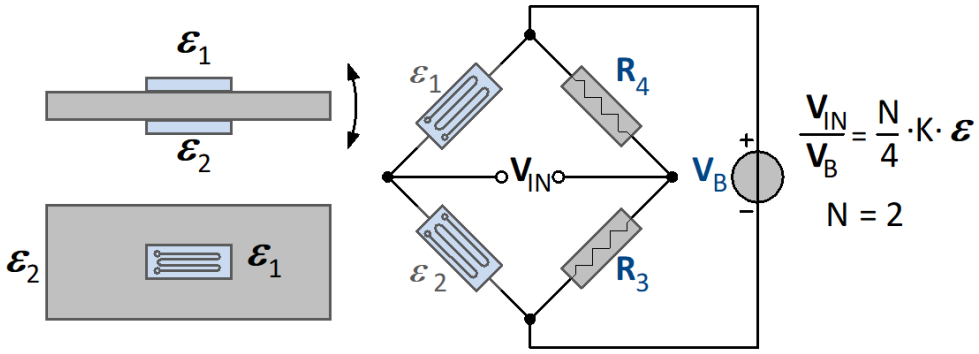


In this circuit, two active strain gauges are used. The strain gauge is positioned transverse to the main direction of strain. The transversal contraction is exploited. For this reason, the Poisson's ratio for the material, which is its transversal expansion coefficient, must be supplied along with the gauge factor. This circuit offers good temperature compensation. The strain is computed as:

$$\varepsilon\left[\frac{\mu\mathrm{m}}{\mathrm{m}}\right] = \frac{4 \cdot 1000}{k \cdot (1+\nu)} \cdot \frac{V_a}{V_e}\left[\frac{\mathrm{mV}}{\mathrm{V}}\right]$$

Legend:	
k	gauge factor
ν	Poisson's ratio of test object material

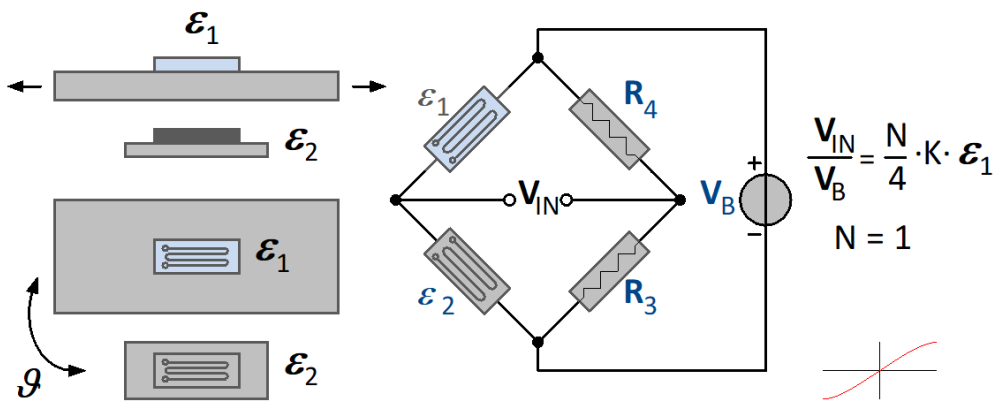
9.9.3.4 Half bridge with two active strain gauges in uniaxial direction



Two active strain gauges are placed under stress in opposite directions but equal magnitude, i.e. one strain gauge is under compression and another under equal tension. (bending beam circuit). This arrangement doubles the measurement's sensitivity to a bending moment. On the other hand, longitudinal force, torque and temperature are all compensated for. The strain is computed as:

$$\varepsilon\left[\frac{\mu\mathrm{m}}{\mathrm{m}}\right] = \frac{4 \cdot 1000}{2 \cdot k} \cdot \frac{V_a}{V_e}\left[\frac{\mathrm{mV}}{\mathrm{V}}\right]$$

9.9.3.5 Half bridges with one active and one passive strain gauge

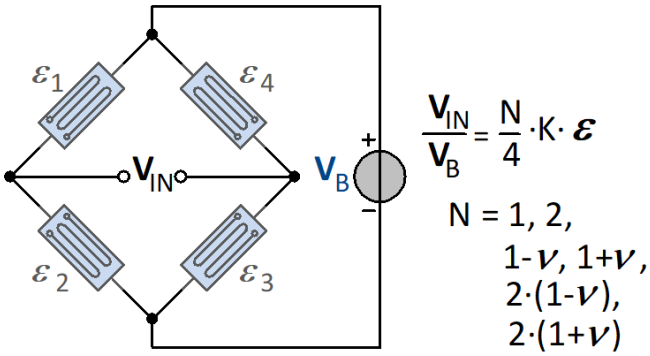


This circuit involves strain gauges. The first one is positioned on the test object, the second on a sample of the same material under the same ambient temperature and serves the purpose of temperature compensation. The strain is computed as:

$$\varepsilon\left[\frac{\mu\mathrm{m}}{\mathrm{m}}\right] = \frac{4 \cdot 1000}{k} \cdot \frac{V_a}{V_e}\left[\frac{\mathrm{mV}}{\mathrm{V}}\right]$$

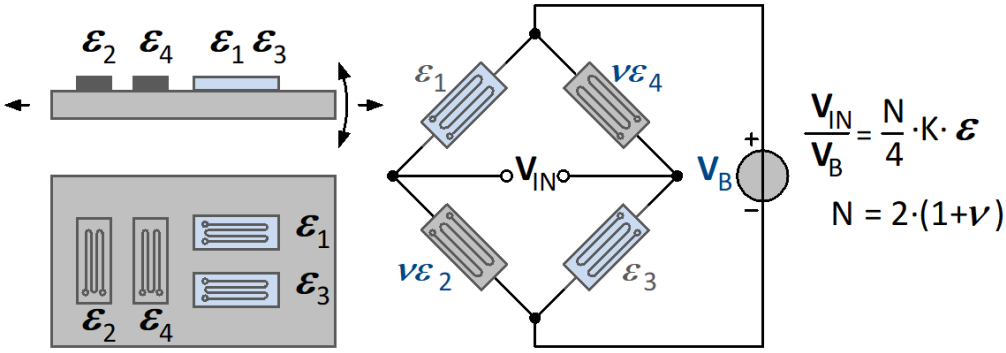
Legend:	
k	gauge factor

9.9.3.6 General Full bridge



General full bridge. N has to be set from a list.

9.9.3.7 Full bridge with Poisson strain gauges in opposed branches

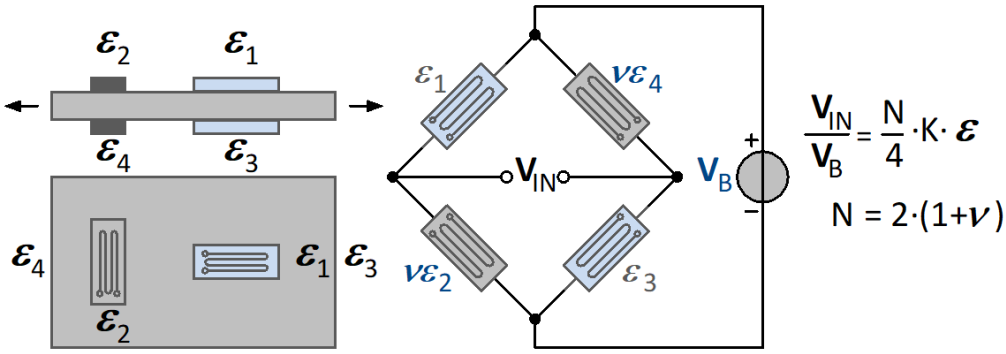


Two active strain gauges are positioned along the longitudinal strain and are joined by two transversally positioned strain gauges to complete the bridge (torsion bar arrangement). In the bridge, the longitudinal strain gauges are located in opposite branches. This circuit provides better exploitation of transversal contraction and longitudinal force as well as good temperature compensation. In this arrangement, the transversal expansion coefficient must be specified. The strain is computed as:

$$\varepsilon\left[\frac{\mu\mathrm{m}}{\mathrm{m}}\right] = \frac{4 \cdot 1000}{2 \cdot k \cdot (1+\nu)} \cdot \frac{V_a}{V_e}\left[\frac{\mathrm{mV}}{\mathrm{V}}\right]$$

Legend:	
k	gauge factor
v	Poisson's ratio of test object material

9.9.3.8 Full bridge with Poisson strain gauges in adjacent branches

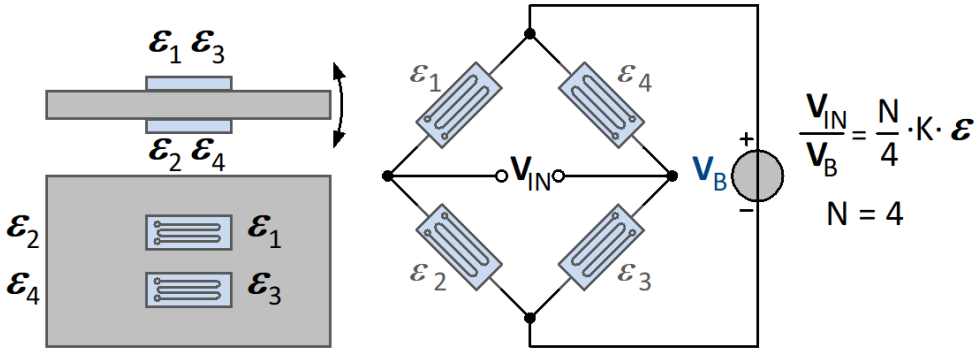


Full bridge with 4 active strain gauges. 2 active strain gauges complemented by 2 transverse Poisson strain gauges. They are located in opposed bridge arms. Higher exploitation of transverse contraction longitudinal expansion while providing good temperature compensation.

$$\varepsilon\left[\frac{\mu m}{m}\right] = \frac{4 \cdot 1000}{2 \cdot k \cdot (1+\nu)} \cdot \frac{V_a}{V_e}\left[\frac{mV}{V}\right]$$

Legend:	
k	gauge factor
ν	Poisson's ratio of test object material

9.9.3.9 Full bridge with 4 active strain gauges in uniaxial direction



The circuit consists of 4 active strain gauges. Two are under compression and the others under equal tension. The strain gauges under tension are positioned in opposite bridge arms. The sensitivity to the moment of bending is increased. At the same time, longitudinal force, torque and temperature are compensated. The strain is computed as:

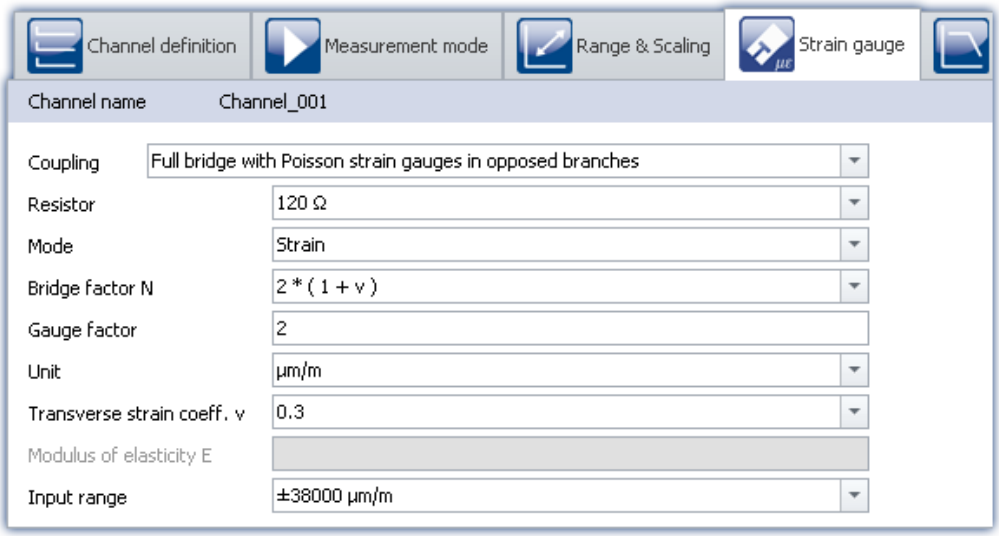
$$\varepsilon\left[\frac{\mu m}{m}\right] = \frac{4 \cdot 1000}{4 \cdot k} \cdot \frac{V_a}{V_e}\left[\frac{mV}{V}\right]$$

Legend:	
k	gauge factor

9.9.3.10 Scaling for the strain analysis

It is possible to choose whether to determine the strain or the mechanical stress suffered by the part. In the range of elastic deformation, the axial stress (force / cross section) is proportional to the strain. The proportionality factor is the modulus of elasticity.

Mechanical stress = modulus of elasticity * strain (Hooke's law)



Gauge factor (K-factor)

The K-factor is the ratio by which the mechanical quantity (elongation) is transformed to the electrical quantity (change in resistance). The typical range is between 1.9 and 4.7. The exact value can be found in the spec sheet for the strain gauge used. If the value entered for this parameter is outside of this range, a warning message will appear but the module can still be configured.

Transverse strain coeff.

(poisson's ratio): If a body suffers compression or tension and is able to be freely deformed, then not only its length but also its thickness changes. This phenomenon is known as transversal contraction. It can be shown that for each kind of material, the relative change in length is proportional to the relative change in thickness D. The transversal elongation coefficient (Poisson's ratio) is the material-dependent proportionality factor. The material constant is in the range 0.2 to 0.5.

In bridge circuits where the strain gauges are positioned transversally to the main direction of strain, this constant must be supplied by the user. The ratios for various materials are available in the list box. These values are only for orientation and may need to be adjusted.

Elastic modulus

The elastic modulus E, is a material parameter characterizing how a body is deformed under the action of pressure or tension in the direction of the force. The unit for E is N/mm^2. This value must be entered for the mechanical stress to be determined The e-moduli for various materials are available in the list box. These values are only for orientation and may need to be adjusted.

Unit

When the strain is determined, the readings appear with the unit µm/m.

For the mechanical stress one can toggle between GPa and N/mm^2 .

1 GPa = 10^3 N/mm^2

Note that the elastic modulus is always in GPa.

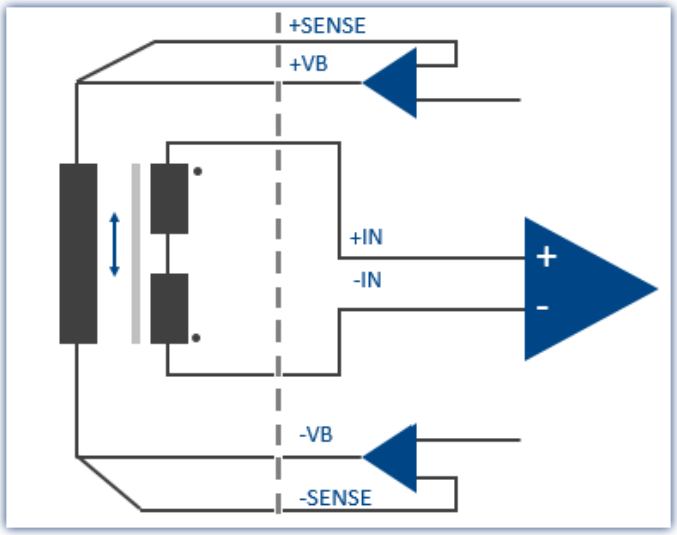
9.9.4 LVDT

LVDT and inductive displacement transducers are operated with carrier frequency mode bridge amplifiers such as C-60xx or C-60xx-1 or module BR(2)-4.

LDVT (Linear Variable Differential Transformer) and RVDT (Rotary Variable Differential Transformer) are based on the transformer principle with an actively driven primary coil. They are configured as full bridge circuit where the two inversely phased secondary windings are compensating each other in the neutral position. Depending on type and winding ratios, the typical output signals are in the range of 200 mV/V but can also exceed 1000 mV/V.

In contrast, inductive displacement sensors do not possess any galvanically isolated primary but consist of two differential coils, wound in phase, that form an inductive half bridge circuit. Their signal range is typical 80 mV/V.

Full-Bridge

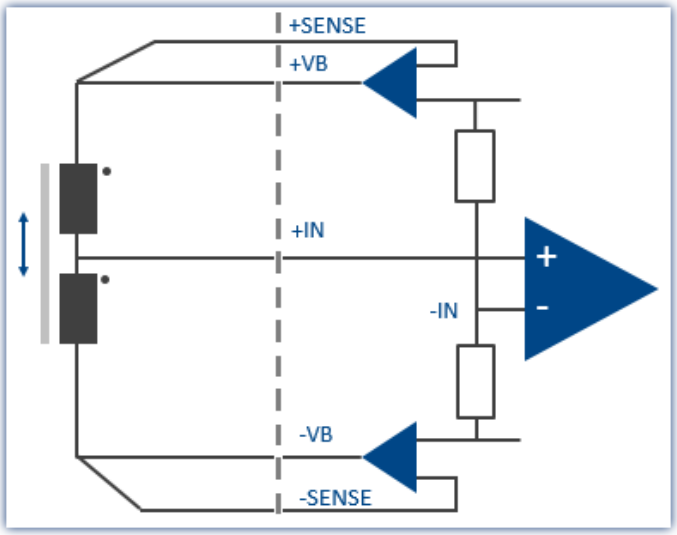


Typically 200 mV/V signal or even < 1000 mV/V.

Opposite winding directions of the two secondaries

Galvanically isolated primary / secondary

Half-Bridge



Typically 80 mV/V signal.

Uniform winding directions of the two coils.

9.9.5 Incremental Counters Channels

The incremental counter channels are for measuring **time** or **frequency**-based signals. In contrast to the analog channels as well as to the digital inputs, the channels are not sampled at a selected, fixed rate, but instead time intervals between slopes (transitions) or number of pulses of the digital signal are measured.

The **counters** used (set individually for each of the 4 channels) achieve time resolutions of up to 31 ns (32 MHz); which is far beyond the abilities of **sampling procedures** (under comparable conditions). The *sampling rate* which the user must set is actually the rate at which the system evaluates the results of the digital counter or the values of the quantities derived from the counters.

Note Sampling rate for incremental counter channels

Only one sampling rate can be set per module.

9.9.5.1 Signals and conditioning

9.9.5.1.1 Mode

The various modes comprise the following measurement types:

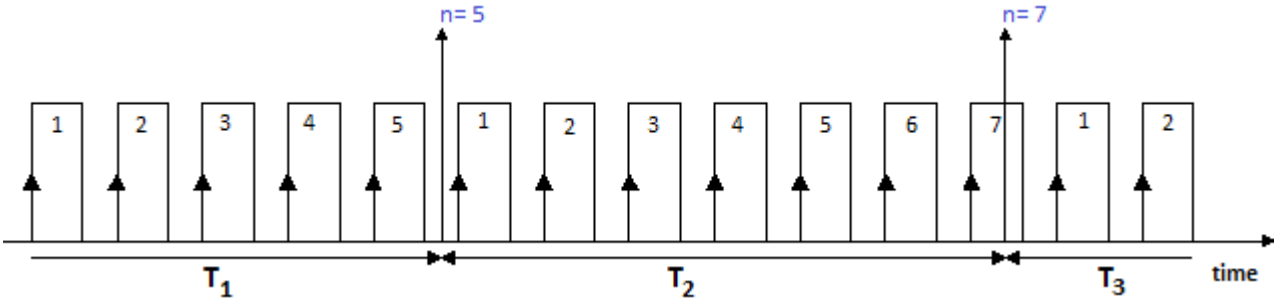
Event-counting	Time	Combined measurements
• events • distance(differential) • angle (differential) • angle (sum) • angle (abs 0-360°) • distance (sum)	• time • pulse time	• frequency • speed • RPM

Event-Counting

The following variables are derived from **Event counting**:

- events 471
- distance(differential) 471
- angle (differential) 471
- distance (abs.) 471
- angle (abs.) 471

The amount of events occurring within one sampling interval is counted. The event counter counts the sensor pulses within the sampling interval. **An event is a positive edge in the measurement signal which exceeds a user-determined threshold value**.

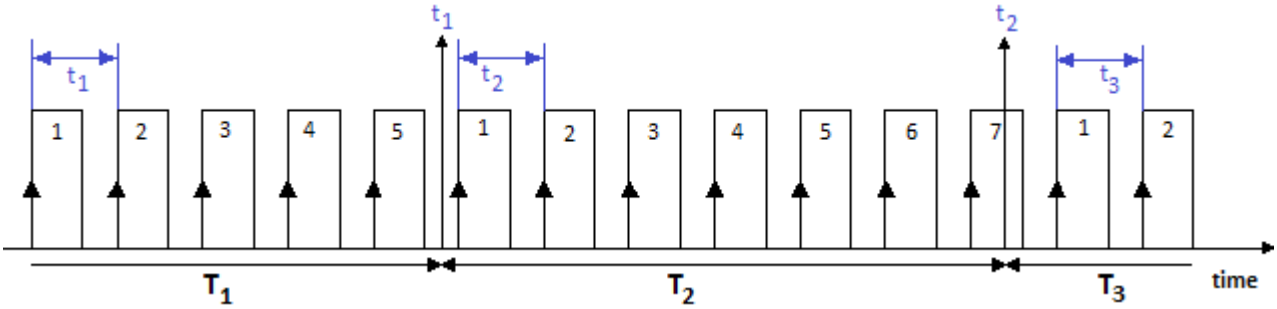


Time Measurements

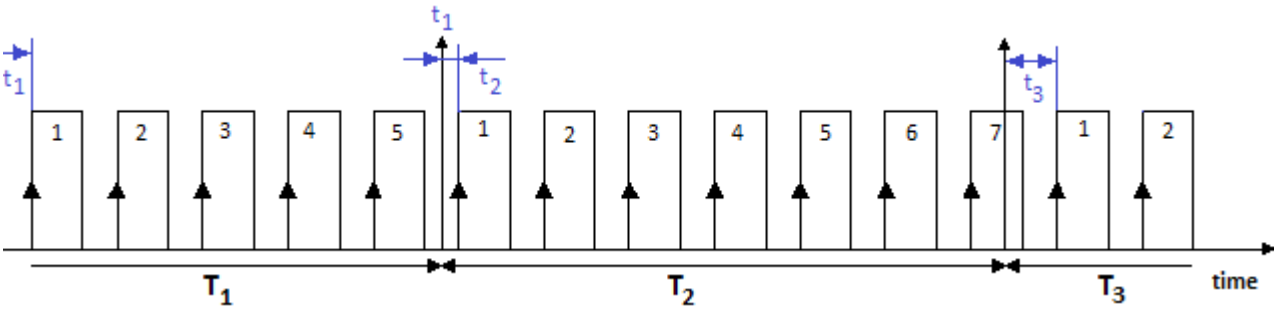
Exclusive measurement of **time** is performed as:

- time 472 (of two successive signal edges)
- pulse time 473 (time from the beginning of one sampling interval until the next signal edge)

Any other pulses occurring within the sampling interval are not evaluated for these measurement types.



time

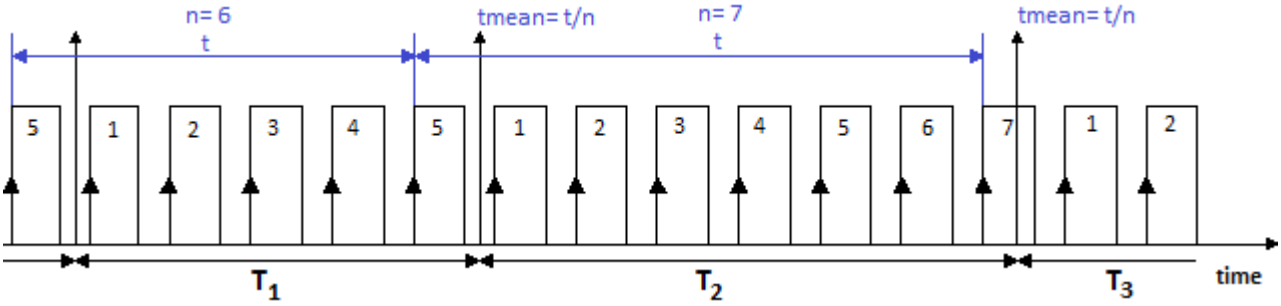


pulse time

Combination Mode

Determining a frequency and the derivative quantities RPM and velocity is based on the **combination of event counting and time measurement**. In other words, during a sampling interval, the number of events occurring as well as the time interval between the first and last event are measured:

- frequency 474
- speed 474
- RPM 474



The frequency is determined as the number of events counted divided by the time between the first and the last "complete" event in the interval. An event is complete when a positive edge is succeeded by a subsequent positive edge.

The frequencies must lie within the bandwidth of the module used. If the maximum frequency is exceeded during a measurement, the system returns the input range end value instead of the true measured values.

The derivative quantities displacement and angle measurement have the following settings:

- Choice of single-track and dual-track encoder 470
- Start of measurement with or without "Zero impulse" 470
- Number of pulses (per unit)

The input ranges and resolutions for the RPM or velocity also depend on the number of encoder pulses set. If the number of pulses is known, the RPM and velocity values can easily be computed:

Parameter	Description
RPM	Input range = ([Frequency input range in Hz] * 60 / [Encoder pulses per revolution]) in RPM Resolution = ([Frequency resolution in Hz] * 60 / [Encoder pulses per revolution]) in RPM

Behavior in response to missing signal pulses

If a sequence of signal pulses is slowing down and then one sampling interval elapses without any pulse, no calculation can be performed for that sampling interval. In that case, the system assumes that the rotation speed is simply decreasing and an attenuating signal course is extrapolated. This "estimated" measurement value is then closer to the true value than the value determined from the preceding sampling interval. This technique has demonstrated its validity in practice.

Note

In extreme cases, the sensor does not return any more pulses, e.g. in case of a sudden outage. Then the algorithm generates an attenuation curve, meaning values > 0, even if the measurement object is actually no longer moving.

9.9.5.1.2 Measurement procedures

Measurement procedures	Description
Differential measurement procedures	The quantities derived from *event-counting*, **Events, Distance** and **Angle** denoted by the annotation **(diff.)** are "differential" measurements. The quantity measured is the respective change of displacement or angle within the last sampling interval. (positive or, for dual-track encoders, negative also) or the newly occurred events (always positive). If, for instance, the total displacement is desired, it must be calculated by **integration** of the differential measurements using imc Online FAMOS functions.
Cumulative measurements	The quantities derived from *event-counting*, **Distance** and **Angle** appearing with the annotation **(abs.)** are "**cumulative**" measurements. In cumulative measurements, the return value is the **sum** of all displacement or angle changes, or of all event which occurred.

9.9.5.1.3 Scaling

A **maximum** value must be entered under ***Input range*** (max. frequency etc, depend on mode). This **Maximum** determines the scaling factor of the computational processing and amounts to the range which is represented by the available numerical format of 16bits. Depending on the measurement mode (quantity to be measured), it is to be declared as an input range's unit or in terms of a corresponding max. pulse rate.

In the interest of maximizing the **measurement resolution** it is recommended to set this value accordingly.

The ***Scaling*** is a sensor specification which states the relation between the pulse rate of the sensor and it's corresponding physical units (sensitivity). This is also the place to enter a conversion factor for the sensor along with any physical quantity desired, for instance, to translate the revolutions of a flow gauge to a corresponding volume.

The table below summarizes the various ***measurement types' units***; the **bold**/*cursive* letters denote the (fixed) primary quantity, followed by its (editable) default physical unit:

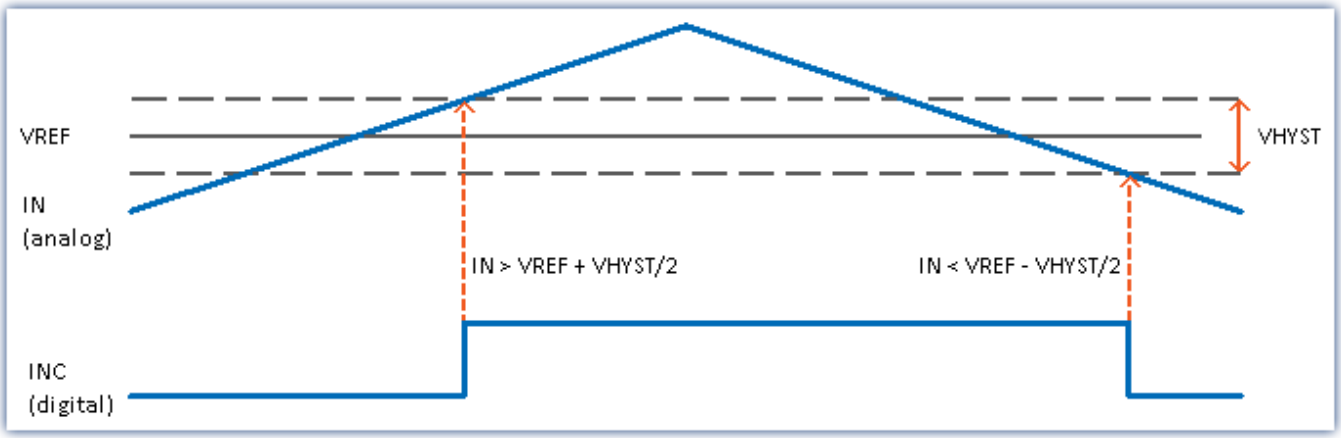
Measurement quantity	(Sensor-) scaling	Range	Maximum
Linear motion	**pulse**/*m*	m	m/s
Angle	**pulse**/*U*	U	U/min
Velocity	**pulse**/*m*	m/s	m/s
RPM	**pulse**/*U*	U/min	U/min
Event	**pulse**/*pulse*	1 pulse	Hz
Frequency	**Hz**/*Hz*	Hz	Hz
Time	**s**/*s*	s	s
Pulse time	**Hz**/*code*	Hz	Hz

9.9.5.1.4 Comparator conditioning

The incremental counters' special properties make **special demands for signal quality**: the very high resolution offered by the detector or counter means that even very short impulses can be captured and evaluated, which sampling-based measurement methods (such as for the digital inputs of the DI16 module) would not (or almost never) be able to detect. Therefore, the digital signals must have clear edges in order not to produce disturbed readings. Spurious impulses or contact bouncing can lead to artifacts such as enormous peaks in RPM-signals etc..

Simple sensors working on the principles of induction or photoelectric relays often emit unconditioned analog signals which must be evaluated according to a threshold condition. Aside from that, problems can occur even with conditioned encoder signals (e.g. TTL-levels) due to long cables, bad reference voltages, ground loops or interference. imc incremental counter channels are able to counteract these problems thanks to a special 3-stage conditioning unit.

First comes a high-impedance **differential amplifier** (± 10 V range, 100 kΩ) which enables reliable acquisition from a sensor even over a long cable as well as effective suppression of common mode interference and ground loops. Next, a (configurable) **smoothing filter** offers additional interference suppression adapted to the measurement situation. Lastly, a **comparator** with adjustable threshold and hysteresis serves as a digital detector. The (adjustable) **hysteresis** also serves to suppress interference.



The *digital signal* changes from **0 to 1** when the *analog signal* exceeds the **VREF + VHYST/2** threshold.

The *digital signal* changes from **1 to 0** when the *analog signal* falls below the **VREF - VHYST/2** threshold.

The size of the hysteresis represents the width of a range-band inside of which the signal can fluctuate (due to signal noise and interference) without an impulse being recorded.

Ranges:

- VREF (Threshold) = -10 V .. +10V
- VHYST (Hysteresis) = +100 mV...+4V
- Low pass filter: None, 20 kHz, 2 kHz, 200 Hz

9.9.5.1.5 Single-track / Dual-track encoder

The **single-track encoder** returns a simple pulse sequence. This means that the pulse count and the time between pulses can be determined, but not the rotation direction of the incremental counter.

A **dual-track encoder** returns two pulse sequences with a 90° offset. Along with the pulse frequency, the rotation direction can also be indicated as positive or negative. To configure a measurement with a dual-track encoder, set the parameter "***Counter signal*** 356" which is on the Setup page "*Digital channels*" under the tab "*Encoder*", along with the desired "*Mode*".

Note Problems with two-point scaling of analog inputs

Affects both the devices belonging to the imc C-SERIES, and also any devices belonging to the imc SPARTAN and imc CRONOS families which are equipped with the digital multiboard: DI16-DO8-ENC4 or the DI8-DO8-ENC4-DAC4.

When an input is set to **dual-track encoder**, it is not possible to **calibrate** the scale with **two-point scaling** for any **analog inputs**. When you click "*Record*" to take a measurement, the following message appears: "*The device is not prepared to allow necessary initialization! Please execute menu action "Prepare" (device control)! imcDevices V2.x Adapter*"

However, the "*Prepare*" procedure does not solve the problem. Instead, temporarily set the incremental counter inputs of the modules affected to "*Single-track encoder*" in order to be able to measure the two data points used for two-point scaling.

9.9.5.1.6 Zero pulse (index)

The **zero pulse** starts the incremental counter channels' counter mechanism. This means the measured values are only recorded, if an event occurs at the **index-channel**. If measurement without a zero pulse is selected, the measurement starts directly upon starting the measurement.

The **index signal** is differential and set by the comparator settings of the **first** incremental counter channel of the module, even for modules that have several index tracks. The bandwidth is limited to 20kHz.

Note

- By default, the option "*Encoder w/o zero impulse*" is activated in imc STUDIO. If this option is deactivated and the zero pulse fails to appear, the encoder module does not start the measurement at all! In that case, the channels only return zero.

9.9.5.2 Mode (events-counting)

Mode - Events	Description
Events	The event counter counts the sensor pulses which occur during a single time interval (differential event counting). The interval corresponds to the sampling time set by the user. The maximum event frequency is about 500 kHz. **An event is a positive edge in the measurement signal which exceeds the user-set threshold value.** The derivative quantities displacement and angle measurement have the following settings: • Choice of single-track and dual-track encoder 470 • Start of measurement with or without "Zero impulse" 470 • Number of pulses (per unit)

Mode - Distance	Description
Distance (differential)	Path traveled within one sampling interval. For this purpose, the number of pulses per meter must be entered.
Distance (absolute)	Absolute distance. The differential distance measurement is converted to the absolute distance. By taking the zero impulse (the counter with no zero impulse should not be selected) into account, the absolute distance position is determined and indicated. Otherwise, the distance value is assumed to be 0° when the measurement begins.

Mode - Angle	Description
Angle (differential)	Angle traveled within one sampling interval. For this purpose, the number of pulses per revolution must be entered. The absolute angle can be calculated in imc Online FAMOS or determined by the mode Angle(abs).
Angle (absolute)	The differential angle measurement is converted to the **absolute** angle. By taking the zero impulse (the counter with no zero impulse should not be selected) into account, the absolute angle position is determined and indicated. Otherwise, the angle value is assumed to be 0° when the measurement begins.
Angle (sum)	The differential angle measurement is converted to the **cumulative** angle. In the process, any zero pulse is evaluated only one time. For this reason, angles which are > 360° are possible.

Note

When using incremental counter modules that work internally with a 16-bit counter, encoders with high pulse rates can lead to overflows. The count is always carried out with sign: 2^{16}= 65536, i. e. ±32767. With dual-track encoders the pulse number is quadrupled internally and leads to a maximum number of pulses per revolution of 8192. For encoders with more pulses per revolution, the hardware must have a 32 bit counter, e. g. imc CANSAS*fit*-ENC6, otherwise an event count must be carried out instead and converted with imc Online FAMOS.

9.9.5.3 Mode (time measurement)

Time measurement

The time measurement mode allows the definition of **edge conditions** between which the time interval is to be measured.

The following combinations are possible:

positive edge	>	negative edge:	(↑ > ↓)
negative edge	>	positive edge:	(↓ > ↑)
positive edge	>	positive edge:	(↑ > ↑)
The combination negative edge	>	negative edge:	(↓ > ↓) is not allowed

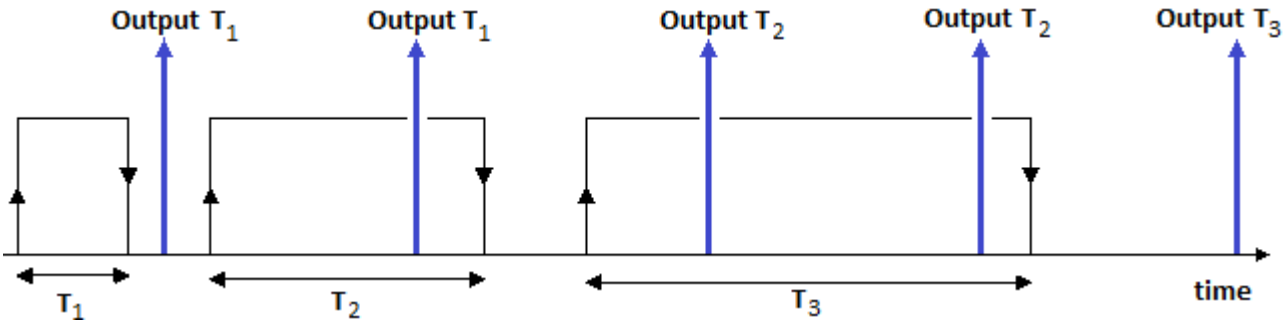
To ensure a high time resolution for the measurement results, suitable scaling must be set for the measurement. An **input range (INC4)** or **Max. time (s) (ENC-6)** specifies the maximum time interval which can be measured between the selected starting and stopping edge. **The time between the signal edges may not be greater than the selected input range.** If the maximum time interval is exceeded during measurement, the system returns the input value range end instead of the true measured value.

Input range	Time resolution	Input range	Time resolution
1 ms	31,25 ns	250 ms	8 µs
2 ms	62,50 ns	500 ms	16 µs
4 ms	125 ns	1 s	32 µs
8 ms	250 ns	2 s	64 µs
16 ms	500 ns	4 s	128 µs
30 ms	1 µs	8 s	256 µs
60 ms	2 µs	16 s	512 µs
120 ms	4 µs	30 s	1024 ms

Time resolution of INC4

The time resolution corresponds to the value of an LSB (Least Significant Bit).

During sampling intervals when no time measurement was possible (because either a starting or stopping edge was missing), the last valid return value continues to be returned until a time measurement is completed. If there is no valid return value, zero is returned. If more than one time measurement is completed during a single sampling interval (due to multiple starting and stopping edges), the last time measured is returned.



Above is illustrated a measured signal from which time readings are taken. Each reading starts at a positive edge in the signal and is stopped at a negative edge. The "up" arrows indicate the times at which the system returns a result. The returned values in this case are T1 –twice; T2 –twice; and T3.

Pulse Time

The point in time at which the edge is located within the sampling interval is determined. This information is needed by some functions in imc Online FAMOS, e.g. for determining the course of the RPMs from a pulse signal: OtrEncoderPulsesToRpm.

The measurement variable ***Pulse Time*** refers to phase-based data which is only relevant to special applications (particularly order-tracking analysis). It is required for subsequent online calculations. It represents the time between the last detected (asynchronous) pulse and the (synchronous) sampling time at which the counter readings were sampled and evaluated. The unit associated with this variable is called *Code*.

Note

The mode *Pulse Time* depends on the sampling rate. For all ENC-4 types, the entry is visible only if the sampling rate is equal or smaller 1ms. For HRENC-4 the sampling rate must be equal or less 100µs.

PWM

Pulse width modulation (PWM) is a type of modulation in which a technical variable (e.g. electrical current) switches between two values. In the process, the **duty cycle ratio is modulated at constant frequency**. PWM is also known as pulse duration modulation (PDM).

A good illustration of this modulation type would be a switch used to continually switch a heater on and off. The higher the ratio of the on-time to the off-time, the higher the average heating power is.

Measurement of PWM can not be performed directly with the device software. However, if the frequency is known, it is possible to perform it indirectly by time measurement with the following settings:

The **ratio** is the *Duration of HIGH (signal) level* over the *Period duration*.

The *Duration of HIGH (signal) level* is obtained by means of a **time measurement** from *positive to negative (signal) edge*.

The *Period duration* is the **inverse of the frequency**, which must be known.

$PWM = t_{pulse}/t_{Period\ duration} * 100\%$ or $t_{pulse} * f * 100\%$

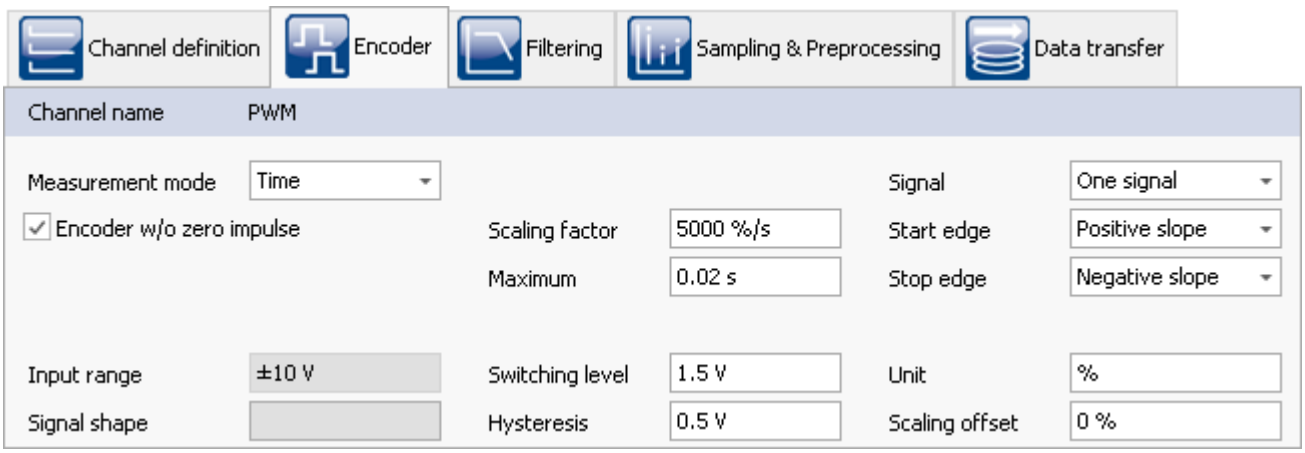
Example:

f= 50Hz, Pulse duration = 10ms

Scaling: $t_{pulse} * f * 100\%/s = 5000\%/s$

at 10ms: 0.01s*5000%/s= 50%

This can be entered directly via the scaling:



Settings for PWM measurement in time mode

9.9.5.4 Mode (combined measurement)

Mode	Beschreibung
Frequency	Frequency is determined by means of a combination measurement 467. If the frequency was previously multiplied or divided, this can be reflected in the scaling value. The frequency is always unsigned, for which reason there is no dual-track encoder for it.
Speed	The sequence of pulses is converted to m/s by means of a combination measurement 467. Toward this end, the number of pulses per meter must be entered.
RPM	The sequence of pulses is converted to revolutions per minute by means of a combination measurement 467. Toward this end, the number of pulses per revolution must be entered.

9.9.5.5 Information and tips

Problem: Measured value does not settle at zero

Answer: This problem is seen where RPM, velocity or frequency signals trail off to zero.

Whenever individual pulses are missing, the system measures the elapsed time and interpolates an RPM- (or velocity, etc.)-value on this basis. Physically speaking, this is the most realistic assumption to make about a rotating machine, but using this technique means that some results are only estimates.

If no pulses arrive for a long time, there is no definitive information on which to base calculations. In this case, the last value calculated may remain >0.

9.10 Fieldbusses

This chapter describes the different fieldbusses and their operation with imc STUDIO.

9.10.1 General notes on fieldbusses

Fieldbus channels: Operating methodology

Analog channels are configured using the Setup page.

There is a dialog corresponding to each fieldbus, the fieldbus assistant. In a sense, these dialogs allows you to configure the "digital signal conditioning" for the fieldbus channels. Just like with the amplifier channels, the channels defined in the fieldbus assistant are also available on the Setup page for the digital channels. For the user, these appear as additional analog inputs.

Therefore, you first need to define how to extract the channels from the messages using the fieldbus assistant. Once the fieldbus channels have been created, the storage and recording settings can be specified on the Setup page for the digital channels. Fieldbus channels can also be used in imc Online FAMOS and imc Inline FAMOS.

Fieldbus channels: Synchronization with analog channels

Parallel recording of analog data and fieldbus channels is possible.

With fieldbusses, it is generally not possible to determine exactly when the AD-conversion at the sending subscriber has taken place. The resulting time offset is not taken into account.

The time of a message's reception is registered in the device with a minimum uncertainty of 100 µs. This is the time at which the message reception is complete. As well, due to the various pipelines (AD-converter, time-offset correction, trigger-calculation), there is generally a time uncertainty of one sampling period in determining times. In most fieldbusses, it is even two sampling periods.

If the clocks of an analog data recording channel and a fieldbus channel are to be compared, then the possible maximum time offset Δt_{max} is

$$\Delta t_{max} = max(100\ \mu s\ , 2 \cdot max(t_{samp-analog}, t_{samp-FB})) + t_{FB-delay}$$

$max(a, b)$	function that returns the larger of two values
$t_{samp-analog}$	the sampling period used on the analog channels
$t_{samp-FB}$	the sampling period used on the fieldbus channels
$t_{FB-delay}$	the maximum delay between A/D conversion in fieldbus sensor and actual transmission on the fieldbus

Example

A fieldbus channel is sampled every 2 ms and an analog channel every 1 ms. The maximum delay on the fieldbus is 3 ms.

$$\Delta t_{max} = 2 \cdot 2\ \text{ms} + 3\ \text{ms} = 7\ \text{ms}$$

Treatment of channel names

When importing channel names, it is possible that the naming rules valid for imc devices are not observed. In general, the names are adopted and when used in imc Online FAMOS or imc FAMOS etc. are enclosed with **{...}**.

Only exception: In order to avoid mistaking channels' names with those of process vector variables, any channel names beginning with "***pv.***" are renamed to "***pv_***".

General notes on sampling and time stamping

Each channel which is recorded with the fieldbus, can be stored in two different ways.

- Sampled with a fixed rare
- Time stamp for each sample

Reference See also

- Data recording with sampling or with time stamp 752
- Capturing fieldbus input data 752
- Time-stamped channels - Setting the RAM-buffer duration, circular buffer duration and pretrigger duration 1132

9.10.2 ARINC-Bus Interface

Introduction - What is the ARINC Bus?

The ARINC bus is a fieldbus used primarily in civil aviation. All data sent on the ARINC Bus consist of 32-bit words. Below, the typical data diagram for the ARINC 429 is displayed.

		MSB	LSB		LSB	MSB
32	31 30	29 11		10 9	8 1	
P	SSM	Data		SDI	Label	

The label is transferred beginning with the MSB (most significant bit). In contrast, transfer of the data bits begins with the LSB (least significant bit). The label is a value between 1 and 255, of which most are defined in the specs for **A**ERONAUTICAL **R**ADIO, **INC**.; some few are reserved for future purposes.

Upon request, imc devices can be equipped with the ARINC 429 fieldbus interface (see data sheet). This makes it possible to record and process data on the ARINC bus in parallel with the analog measurement data. With the help of the ARINC-Assistant, all channels can quickly and easily be configured for the ARINC Bus.

The ARINC interface offers the following performance characteristics:

- 8 receiver channels
- both binary- (BNR) and decimal-encoded data (BCD) and discrete data (DSC) can be captured. In fact, these three data types can appear together in a single ARINC word.
- 4 Transmitter channels (via process vector variables)

This chapter is geared for users who are already familiar with the configuration of analog and digital channels in imc STUDIO.

Note

Also observe the general notes on file buses in regard to control philosophy 474 and the sampling rate either as fixed sampling rate 752 or time stamp 752.

9.10.2.1 ARINC-bus assistant

Before the imc device can record an ARINC data channel, the ARINC data word must be defined. This task is accomplished by the ARINC-Bus Assistant. After this step has been taken and the Assistant closed, an additional measurement channel is present in imc STUDIO, which can be used in a manner very similar to that of an analog or digital channel. This means that the channel can, for instance, be subjected to mathematical operations in imc Online FAMOS, or be recorded in response to a trigger working in conjunction with other analog channels. In the Ethernet Assistant, only settings specific to the Ethernet are made.. All other measurement specifications, such as measurement duration or triggering links, are handled just as if they applied to ordinary analog or digital measurement channels.

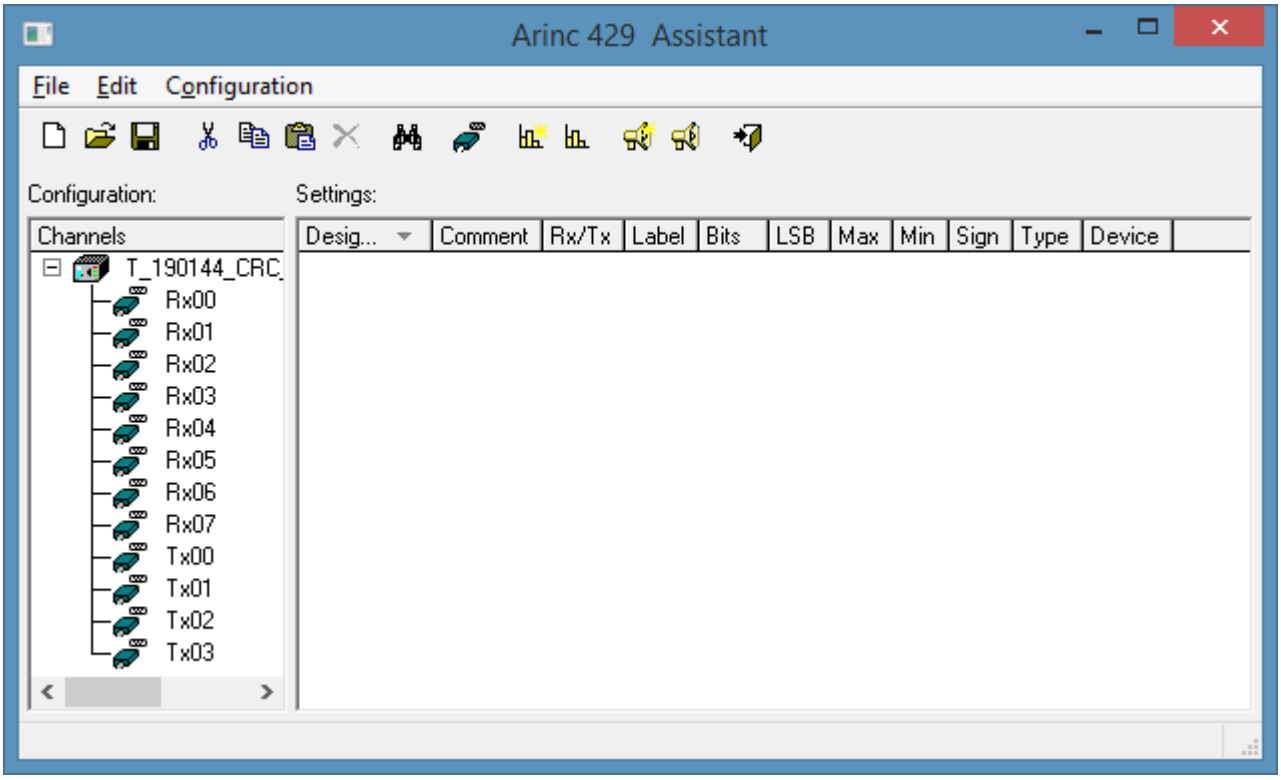
Philosophy of the ARINC-bus

In order to better understand how to work the Assistant, it is useful to review the underlying philosophy of the ARINC-bus. In contrast to the CAN-bus, where several senders can be connected to a bus and the user has broad powers to specify the formulation of the CAN-messages, the arrangement of ARINC signal channels is set by a central authority (AERONAUTICAL RADIO, INC.). This means that the format and interpretation of an ARINC signal normally is unequivocally determined according to its label-number and equipment-ID. For this reason, the format of an ARINC signal in a measurement should be defined only once, and then recorded in a list of available options for measurement procedures.

9.10.2.2 Starting the assistant

Start the Assistant from within the imc STUDIO software via the Ribbon *Setup-Configuration > ARINC-Assistant*.

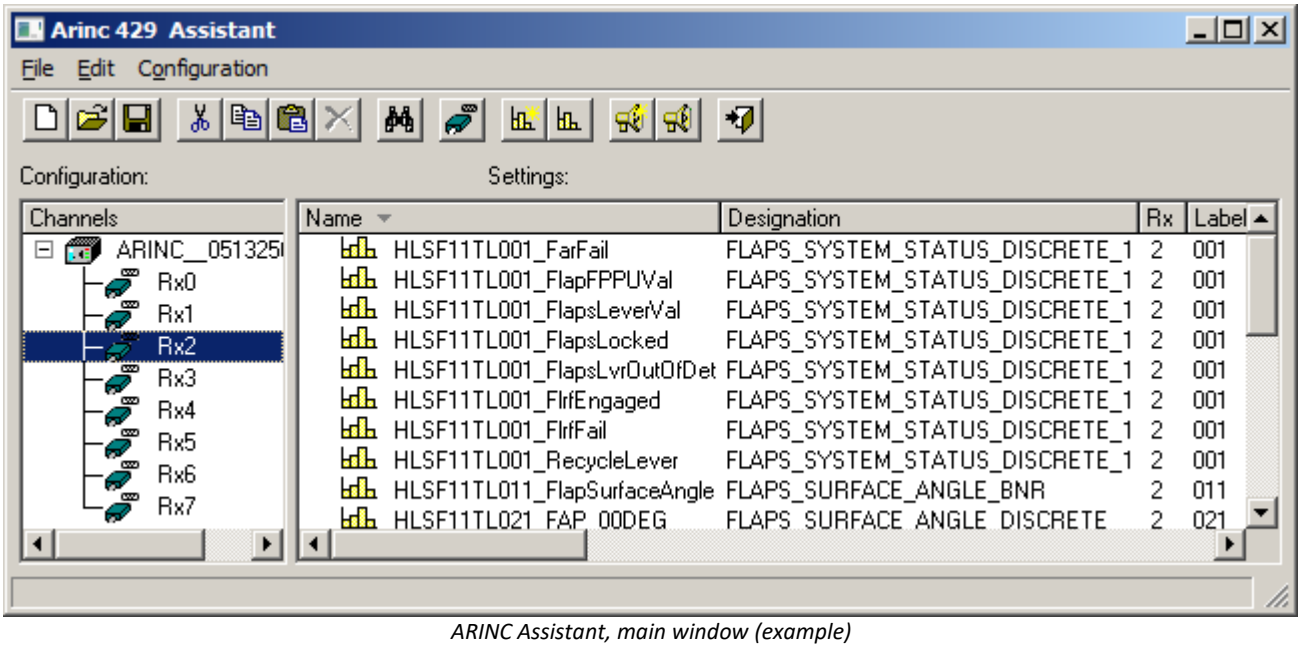
The Assistant begins by displaying the following dialog:



ARINC Assistant

The ARINC-Assistant's main window is divided in halves:

- The left side, designated **Configuration** 496, lists the device and constituent channels.
- The right side with the title **Settings** lists the ARINC-signals which are defined and are assigned to the currently selected Rx-channel.



ARINC Assistant, main window (example)

The definitions made here under **Settings** for the respective ARINC-signals are saved in a separate configuration file (.idb-format) and thus independently of the associated experiment. This configuration file can be considered a sort of database, in which all defined ARINC-signals are saved together with the assignment to the respective channel. Accordingly, in order to configure an experiment with the ARINC-bus, it is enough to load just one ARINC-configuration file and to close the Assistant. All defined ARINC-signals then appear in the imc STUDIO operating software as channels and can be displayed or subjected to further processing in imc Online FAMOS.

9.10.2.3 ARINC configuration file

All ARINC signal definitions are saved to a configuration file (.idb-format) . This configuration file can be used as a central database, into which you can load every experiment and where you can delete, copy, or move ARINC-signals at will.

Thus, the signal definitions are independent of the experiment. As a matter of principle, it is possible to create a configuration file, tailored to the respective experiment, from an exiting configuration file, quickly and at any time.

The format *.idb2: With this format, the Parity-Bit and SSM are additionally saved.

9.10.2.3.1 Creating a new configuration file

To create a new configuration file, it's possible to either click the button in the toolbar or to select the menu item *File > New*.

9.10.2.3.2 Loading a configuration file

To open an already existing configuration file, click on either the corresponding button in the toolbar or the menu item *File > Open*. The default extension for the file is *.idb. It is also possible to load *.csv-files, in order to import already existing ARINC signal databases of one's own.

Note

- If log channels have been activated (Settings > log channel), those channels are listed in the channel list.
- If a new IDB file will be loaded, the log channels disappear from the channel list. However they are still active for the bus settings.
- After reloading a ARINC configuration, those log channels must be deactivated and reactivated. Settings like saving will be lost and must be redone.

9.10.2.3.3 Save configuration file (as)

To save a configuration file, click on either the corresponding button in the toolbar or the menu item *File > Save*.

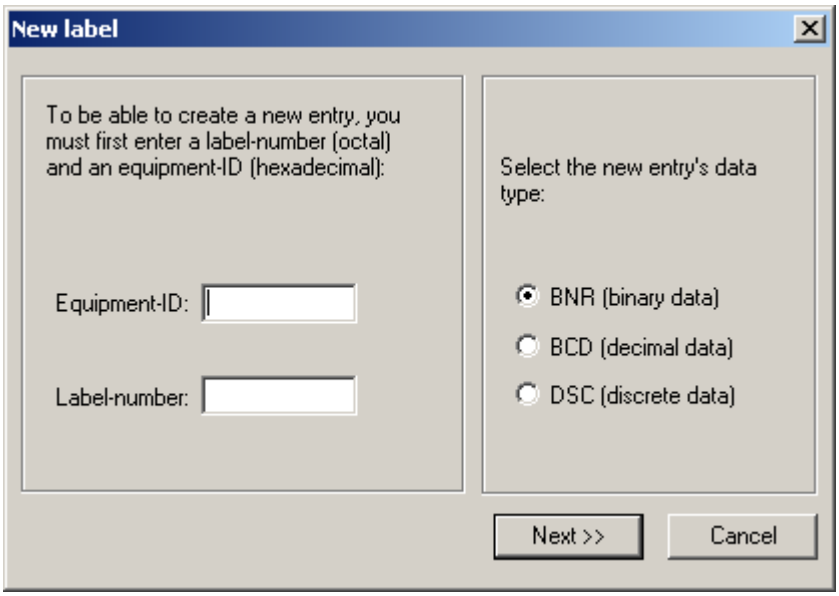
If the open configuration file is to be saved under a different name or in a different directory, menu item *File > Save as* is used. In the Windows file dialog which then appears, the name, without its extension, must be entered. The extension *.idb is appended automatically.

9.10.2.3.4 Receiver channel (Rx) - creating a ARINC label

A new ARINC-signal (referred to in the following as an ARINC-label) can only be created in the configuration file and never directly in the imc STUDIO: Setup.

- To create an ARINC-label, first either open an existing configuration file or create a new one.
- Then, in the left portion, Configurations, select the Rx-channel which is to receive the ARINC-signal.
- Afterwards, either click on the button in the title bar, or select the menu item *Configuration* > *New Label*.

A dialog appears:



New label

On the left side of this dialog, enter the **Equipment-ID** (hexadecimal) and the **Label-number** (octal). The correct entries for these can be found in the ARINC spec sheet.

The following data are presented as an example:

```
LABEL 103
EQPT. ID (HEX) 003
PARAMETER NAME Selected Airspeed
DATA TYP BNR
UNITS Knots
RANGE (SCALE) 512
SIG DIG 11
RESOL 0. 25
MINIMUM TRANSIT INTERVAL (ms) 100
MAXIMUM TRANSIT INTERVAL (ms) 200
MAXIMUM TRANSPORT DELAY (ms)
```

As seen above, the entries to be made in this case would be 103 for the label number and 003 for the equipment-ID. The entries once made, click on the "More >>" button to proceed to the configuration dialog for the ARINC-label.

In the right-hand portion of the dialog, the data type for the encoding of the data on the ARINC-bus is selected. The available options are the data types **BNR** for binary encoding, **BCD** for BCD encoded data and **DSC** for discrete data. Clicking on *Next* >> brings up the next dialog, whose particular setup depends on the respective data type selected. How each of the particular data types are defined is described in a separate section.

9.10.2.3.4.1 Configuring a BNR data word

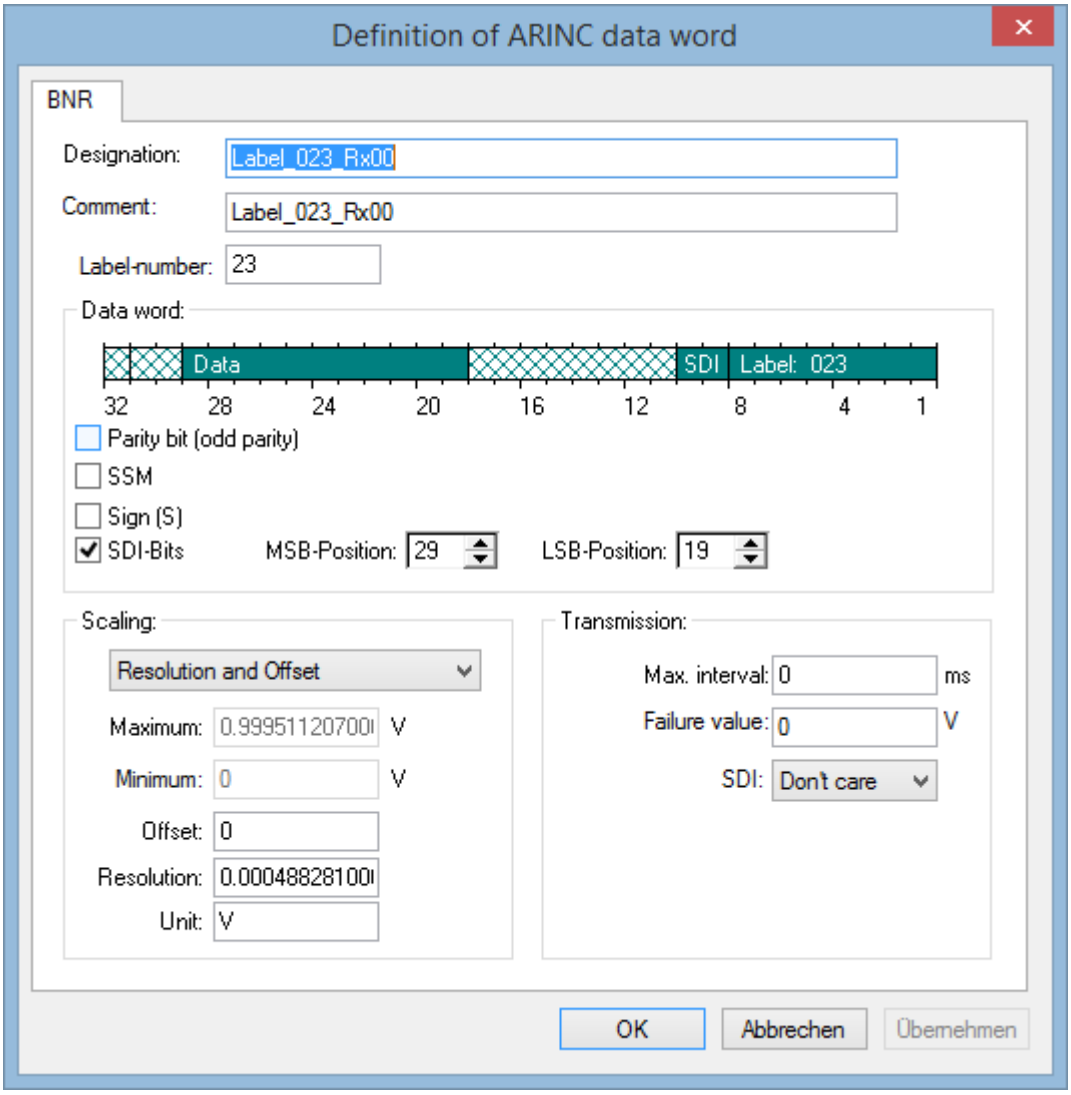
The dialog is thematically arranged in groups each denoted by a gray frame.

The group **Data word** comprises all settings concerning the structure of the ARINC data word. The currently set word structure is visually represented by a green horizontal bar.

In the group **Scaling** the settings are grouped, which are needed for computing the ultimate measured value from the raw data.

In the group **Transfer**, settings are made concerning the timing conventions valid for the transfer of the ARINC data word.

The following is an explanation of the controls:



BNR data word

Settings	Description
Designation	Is an arbitrary, 64 character long string (may not contain quotation marks). It is assigned to the label and displayed in the Assistant-interface. It is also preserved in imc STUDIO as a channel comment. It is analogous to the *PARAMETER NAME* on the ARINC spec sheet, Attachment 2, and should always be specified.

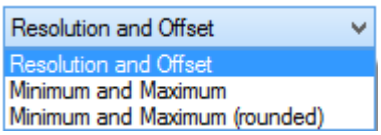
Data word

The checkboxes "*Paritybit*", "*SSM*" and "*Sign*" expand or limit the maximum MSB-position.

Settings - Data word	Description
Paritybit	• If the box is checked, then an maximum MSB-position of 31 can be selected. • If the box is not checked, and the SSM and Sign boxes are also not, then a maximum MSB-position of 32 can be selected for the data word.
SSM	• If this box is not checked, then depending on whether the **Paritybit** box is checked, a maximum MSB-position of 31 or 32 can be set. • If the box is checked, the maximum MSB-position which can be set is 28.
Sign(S)	• If this is active, the sign is evaluated. Otherwise it is ignored. • In general, it is good to leave this option "off" if only a positive value range was specified.
SDI Bits	With this toggle you can determine whether or not SDI-bits are present in the label. This is not where the SDI address is set; that is done in the group *Transfer*. Instead, you can set whether or not the SDI is present at all.
MSB Position	The bit position of the most significant bit. In a BNR data word, the MSB position is usually Bit 28.
LSB Position	Bit-position of the least significant bit. The number of bits between the positions of the MSB and LSB is the number of significant bits. This characteristic number can be found in the *SIG DIG* column in the ARINC spec sheet, Attachment 2.

Scaling

The scaling can be determined by "*Resolution*" and "*Offset*" or by "*Minimum*" and "*Maximum*".



Calculation rule per selection list

Settings - Scaling	Description
Maximum	Largest measured data value. The highest raw data value is mapped to this value.
Minimum	Lowest measured data value. The lowest value from the raw data maps to this value. Typically, with signed values, the input range is symmetric, i.e. Minimum = -Maximum. If the box for value sign is checked, you must verify whether the minimum is correct, since it will not be adjusted automatically.
Offset	This box is computed automatically. • The offset is computed with sign (S) as: *OFFSET = (Maximum+Minimum) / 2*. • The offset is computed without sign (S) as: *OFFSET = Minimum*.
Resolution	This box's value is computed automatically. You may wish to verify that it matches the column *RESOL* of the ARINC specs, Attachment 2. • The resolution is computed as: *RESOL = (Maximum-Minimum) / 2^(SIGDIG + S)*.
Unit	The signal's unit. Corresponds to the column UNITS in the ARINC spec sheet, Attachment 2.

Transmission

Settings - Transmission	Description
Max. interval	The maximum time which may elapse between the transfer of two measurement values. • If this time is exceeded, a zero is transferred to denote a missed measurement value. • If this interval is not known or not important, enter 0. Then no misses will be recorded.
Failure value	The value which is transferred if the maximum permitted interval between two measured values is exceeded.
SDI	If necessary, you can change the SDI-bits for each channel in the ARINC-Assistant. To do this, click on the combobox from which the SDI address is selected.

9.10.2.3.4.2 Configuring a BCD data word

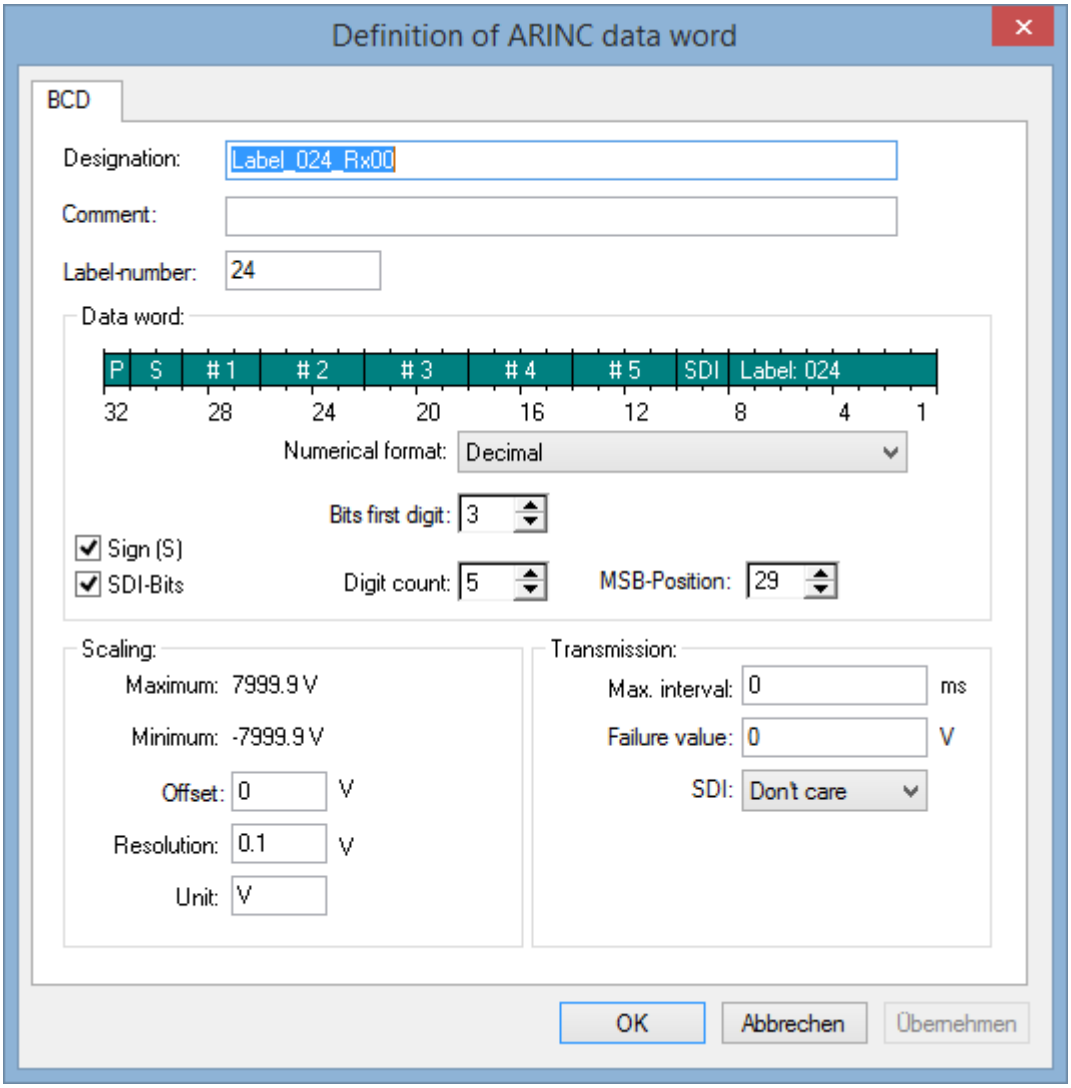
The dialog is thematically arranged in groups each denoted by a gray frame.

The group **Data word** comprises all settings concerning the structure of the ARINC data word. The currently set word structure is visually represented by a green horizontal bar.

In the group **Scaling** the settings are grouped, which are needed for computing the ultimate measured value from the raw data.

In the group **Transfer**, settings are made concerning the timing conventions valid for the transfer of the ARINC data word.

The following is an explanation of the controls:

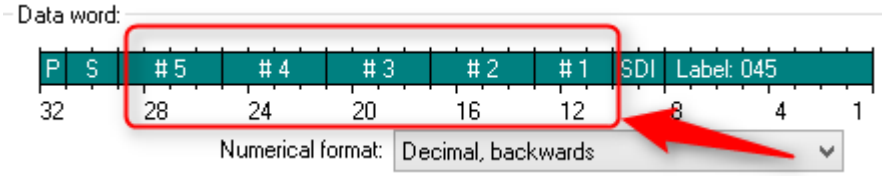


BCD data word

Settings	Description
Designation	Is an arbitrary, 64 character long string (may not contain quotation marks). It is assigned to the label and displayed in the Assistant-interface. It is also preserved in imc STUDIO as a channel comment. It is analogous to the *PARAMETER NAME* on the ARINC spec sheet, Attachment 2, and should always be specified.
Label-number	Here you can change the label number that was entered when the data word was created. 479

Data word

The order of the data word can be reversed for all numerical formats.



Settings - Data word	Description
Numerical format	Is a numerical format. There is a choice among three formats: **Format** / **Beschreibung** Decimal (, backwards) — The number is interpreted as a normal fixed-point number. This is the default case. Angle (, backwards) — The first three digits are interpreted as degrees, the next two as minutes of arc. Therefore, the minimum number of digits is 5. Six digits can also be specified if no SDI bits are present, in which case the sixth digit is interpreted as the decimal fraction of a minute of arc. Time (, backwards) — The first 2 digits are interpreted as hours (0.. 24) and the next 2 digits as minutes (0.. 60). Therefore, the minimum number of digits is 4. Five digits are also possible, in which case the fifth digit is interpreted as the decimal fraction of a minute.
Sign(S)	If this is active, the sign is evaluated. In general, it is good to leave this option "*off*" if only a positive value range was specified. The sign cannot be taken into account when the data exist in time-format.
SDI Bits	With this toggle you can determine whether or not SDI-bits are present in the label. This is not where the SDI address is set; that is done in the group *Transfer*. Instead, you can set whether or not the SDI is present at all.
Digit count	Number of valid BCD digits. • This value matches the column "*SIG DIG*" in the ARINC specs, Attachment 2. • The maximum possible number depends on the numerical format set and on the SDI bits. Example: In decimal representation, a maximum of 5 digits are possible if SDI bits are present. If the SDI bits are switched off (see above), up to 6 digits are possible. Some labels make use of this ability.
MSB-Position	The bit position of the most significant bit. In a BCD data word, the MSB position is usually Bit 28.

Scaling

Settings - Scaling	Description
Maximum	Largest measured data value. The highest raw data value is mapped to this value. The maximum is automatically computed from the Resolution, the Number of digits and the Offset.
Minimum	Lowest measured data value. The lowest value from the raw data maps to this value. Typically, with signed values, the input range is symmetric, i.e. Minimum = -Maximum. If the box for value sign is checked, the minimum it will be adjusted automatically.
Offset	This value can be freely selected. • The offset is computed with sign (S) as: *OFFSET = (Maximum+Minimum) / 2*. • The offset is computed without sign (S) as: *OFFSET = Minimum*.
Resolution	The order of magnitude of the least significant digit. It matches the column *RESOL* in the ARINC specs, Attachment 2. The input range (Maximum and Minimum) is computed automatically from the specified number of digits and the resolution.
Unit	The signal's unit. Corresponds to the column UNITS in the ARINC spec sheet, Attachment 2.

Transmission

Settings - Transmission	Description
Max. interval	The maximum time which may elapse between the transfer of two measurement values. • If this time is exceeded, a zero is transferred to denote a missed measurement value. • If this interval is not known or not important, enter 0. Then no misses will be recorded.
Failure value	The value which is transferred if the maximum permitted interval between two measured values is exceeded.
SDI	If necessary, you can change the SDI-bits for each channel in the ARINC-Assistant. To do this, click on the combobox from which the SDI address is selected.

9.10.2.3.4.3 Configuring a BCD free data word

The dialog is thematically arranged in groups each denoted by a gray frame.

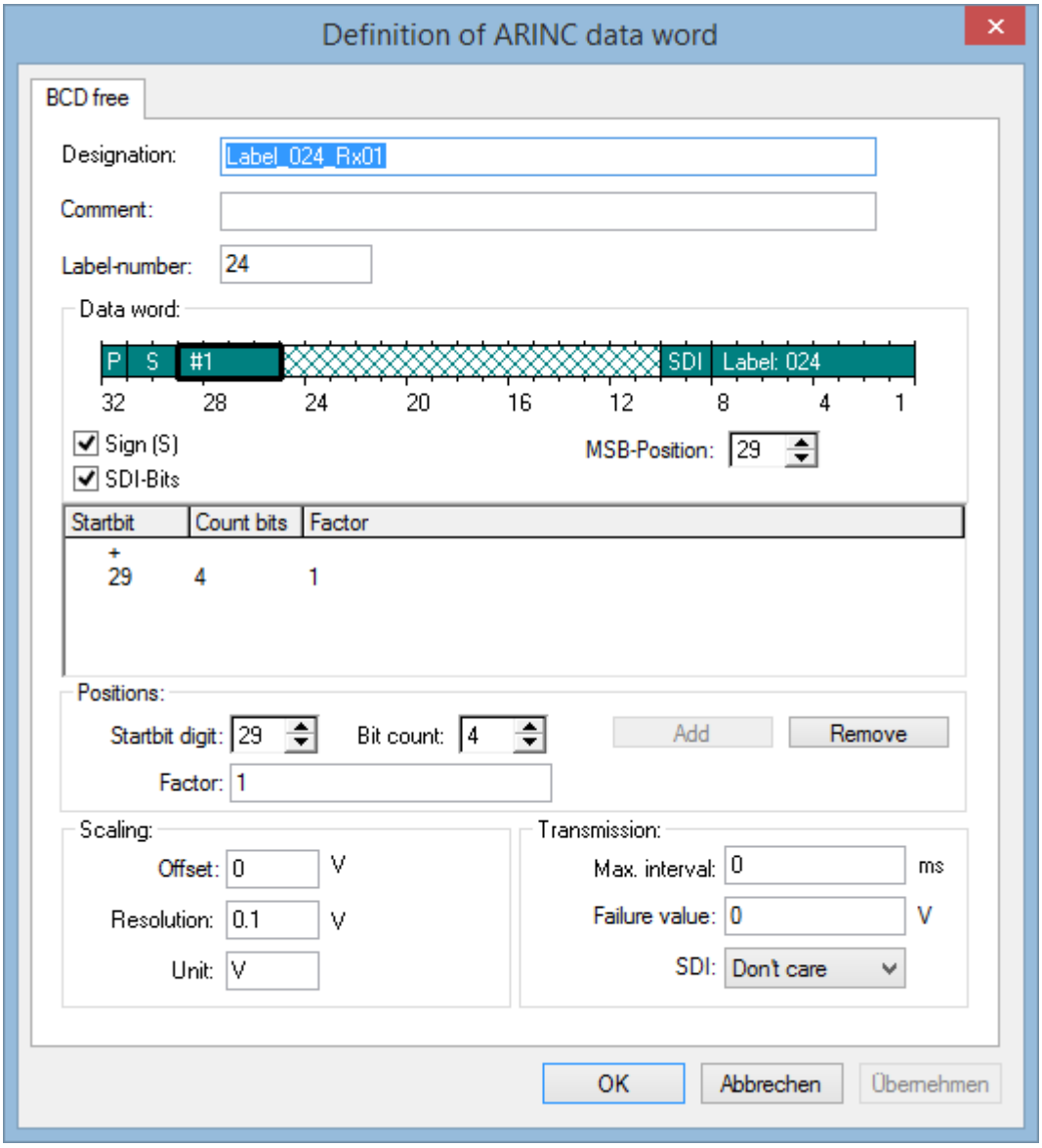
The group **Data word** comprises all settings concerning the structure of the ARINC data word. The currently set word structure is visually represented by a green horizontal bar.

In the **Positions** group the arrangement of the data bits is determined.

In the group **Scaling** the settings are grouped, which are needed for computing the ultimate measured value from the raw data.

In the group **Transfer**, settings are made concerning the timing conventions valid for the transfer of the ARINC data word.

The following is an explanation of the controls:



BCD data word

Settings	Description
Designation, Comment, Label-number, Sign (S), MSB-Position, SDI Bits, Scaling and Transmission	see BCD Dataword 483
Positions	To create a digit, select the "+"line in the list. Define the start with ***Startbit digit,*** the ***number of bits*** and the ***factor*** for the digit. With ***Add***, the settings are transferred to the list and thus to the data word. **Example of Factor:** In the decimal system, you use factor 1 to specify 0-9, factor 10 to specify the tens, and so on. You can use the factor to configure any number format.

9.10.2.3.4.4 Configuring discrete (DSC) data words

The dialog is thematically arranged in groups each denoted by a gray frame.

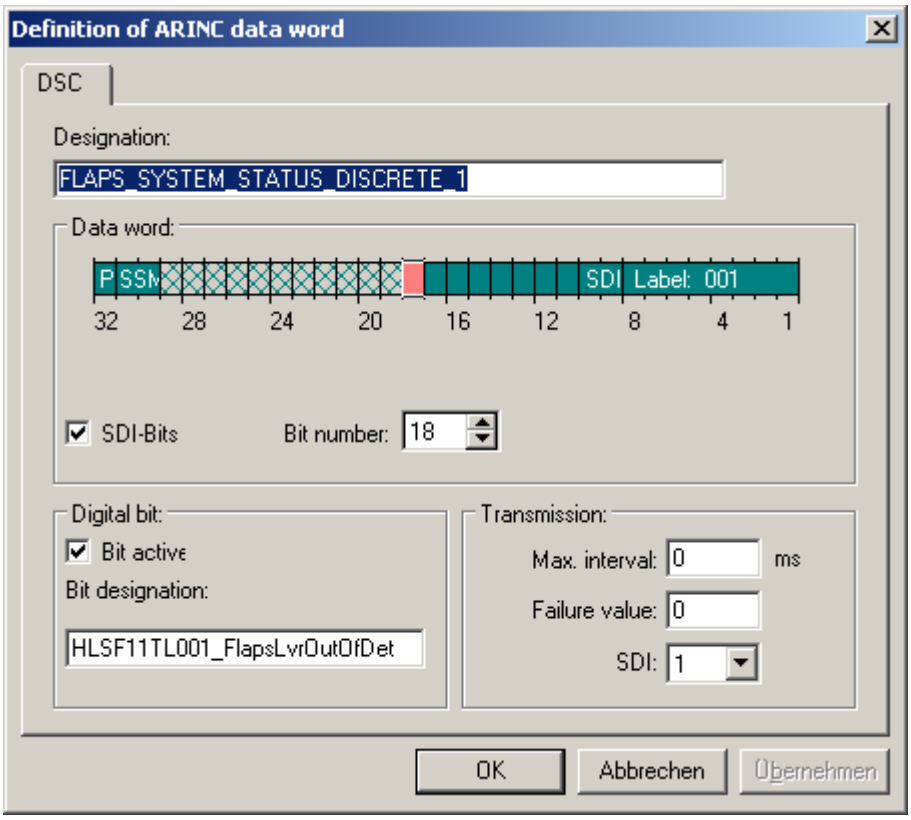
The group **Data word** comprises all settings concerning the structure of the ARINC data word. The currently set word structure is visually represented by a green horizontal bar.

Each bit in the data-carrying portion can be configured individually. To do this, select a particular bit using the control *Bit number* (the bit's image in the green bar will blink).

In the group **Digital bit**, the bit can be manipulated.

In the group **Transfer**, settings are made concerning the timing conventions valid for the transfer of the ARINC data word.

The following is an explanation of the controls:



DSC data word

Settings	Description
Designation	Is an arbitrary, 64 character long string (may not contain quotation marks). It is assigned to the label and displayed in the Assistant-interface. It is also preserved in imc STUDIO as a channel comment.

Data word

Settings - Data word	Description
SDI Bits	With this toggle you can determine whether or not SDI-bits are present in the label. This is not where the SDI address is set; that is done in the group *Transfer*. Instead, you can set whether or not the SDI is present at all. If the SDI bit are not in use, Bits 9 and 10 are available as data bits.
Bit number	Selects one of the bits in the data-carrying portion of the ARINC data word. The contents of the controls in group *Digital bit* are updated accordingly.

Digital bit

Settings - Digital bit	Description
Bit active	Switches the current selection under *Bit number* to an active state. A bit can only be received if it is active. The corresponding bit's image is displayed as a solid, green portion of the green bar. Inactive bits are shown there as green hatching on a gray background.
Bit designation	Here, a descriptive designation of the bit should be entered. Example: Bit 15 of the IRS Discrete Label 270 04 is called "Attitude Invalid". The relationship between the "Designation" for the bit and that for the channel is explained later.

Transmission

Settings - Transmission	Description
Max. interval	The maximum time which may elapse between the transfer of two measurement values. • If this time is exceeded, a zero is transferred to denote a missed measurement value. • If this interval is not known or not important, enter 0. Then no misses will be recorded.
Failure value	The value which is transferred if the maximum permitted interval between two measured values is exceeded.
SDI	If necessary, you can change the SDI-bits for each channel in the ARINC-Assistant. To do this, click on the combobox from which the SDI address is selected.

9.10.2.3.4.5 Mixed configurations

The ARINC standards allow the highest-valued bits in a label to carry a BNR data word, while discrete signals are assigned to the remaining bits. Conversely, several bits in a discrete label can be coordinated to encode a binary number.

Condition:

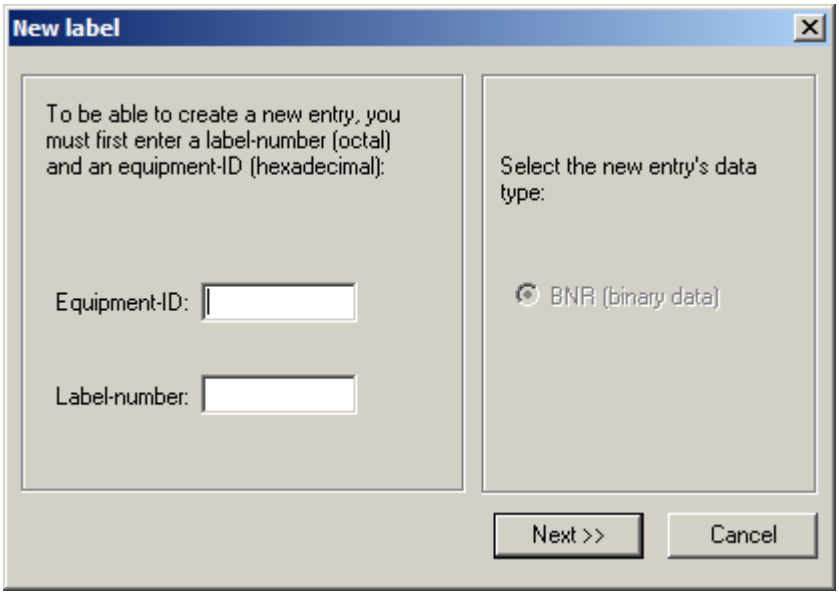
Both instances embody the same irregularity: More than one encoding definition must be made for a single label (one BNR and one DSC).

The Assistant allows any number of definitions to be set for a single label. In this context, even the data type (BNR, BCD or DSC) doesn't matter.

However, for ease of reference, it is good to specify a descriptive name in the **Designation**-control.

9.10.2.3.5 Transmitter Channel (Tx) - creating a ARINC label

Icon	Description
	Create a new Transmitter channel by selecting one of the four Tx-channels and click on the button with the speaker symbol.
	Change the properties of an existing channel using this button.



New label

On the **left side**, enter the **Equipment-ID** (hexadecimal) and the **Label-number** (octal). The correct entries can be determined from the ARINC specs.

The only data type available for sending purposes is **BNR** for binary encoding.

9.10.2.3.5.1 Configuring a BNR data word

The list **Signalname** (right) offers all data which the measurement device is able to send.

All settings affecting the structure of the ARINC data word appear together in the group **Data word**. The current configuration is denoted graphically as a green bar.

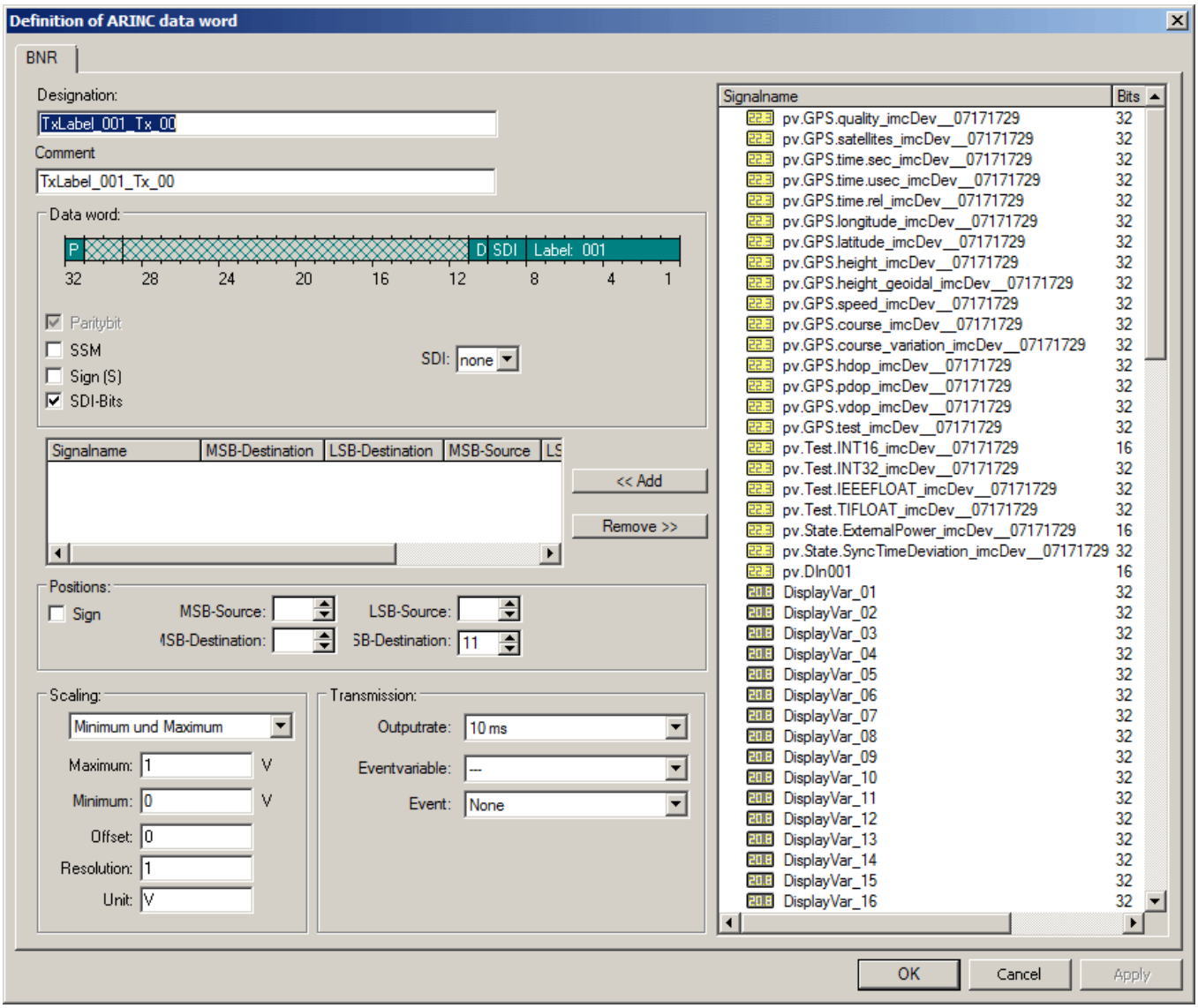
The table at left lists all data which can be added to the message to send.

In the group **Position**, the data type and its bit length are taken into account.

Settings required for calculating the ultimate measured value from the raw data appear in the group **Scaling**.

In the group **Transfer**, it is possible to make settings regarding the transferred ARINC data word's timing.

The following is an explanation of the controls:



Definition of ARINC data word

Settings	Description
Designation	Is an arbitrary, 64 character long string (may not contain quotation marks). It is assigned to the label and displayed in the Assistant-interface. It is also preserved in imc STUDIO as a channel comment. It is analogous to the *PARAMETER NAME* on the ARINC spec sheet, Attachment 2, and should always be specified.

Data word

Settings - Data word	Description
Parity bit	The parity bit is set automatically on the basis of the label contents and its control is therefore disabled.
SSM	• If this box is not checked, then depending on whether the **Paritybit** box is checked, a maximum MSB-position of 31 or 32 can be set. • If the box is checked, the maximum MSB-position which can be set is 28.
Sign (S)	• If there is a check in this box, the sign is transferred, otherwise it is ignored. • As a general rule: if you have specified only a positive value range for an ARINC-label, this option should not be selected. • If this switch is set, the maximum MSB-position is limited to 28.
SDI Bits	With this toggle you can determine whether or not SDI-bits are transferred with the label. This is not where the SDI address is set; that is done in the group *Transfer*. Instead, you can set whether or not the SDI is present at all. If SDI bits are not activated, it is possible to enter a variable at this position.
SDI	If necessary, you can change the SDI-bits for each channel in the ARINC-Assistant. To do this, click on the combobox from which the SDI address is selected.

Position

In the box *Position*, you determine how the variables to be sent are mapped to the label's data portion. If *SSM, Sign* and *SDI* are selected in the area *Data word*, then a maximum of 19 bits are available in the data portion. If the associated options are de-selected, a maximum of 23 bits are available.

The variables to be sent (source data words) take various data formats. Thus, process vector variables can be in the data formats 8, 16 or 32-bit Integer, signed or unsigned, as well as 32-bit Float.

- Display variable always take the data type 32-bit Float.
- DIO-ports consist of one 16-bit word.
- Bits consist of one 1-bit.

Settings - Position	Description
MSB-Source	• If the size of the source data word is **smaller** than the available data portion, then the **MSB-bit is set as the size of the source data word** and can not be changed. The same applies to the data type 32-bit Float. • If the size of the source data word is **greater**, it is possible to set here which payload data from the source data word are to be mapped onto the target data word. In that case, the MSB-source is represented by the highest bit to be evaluated.
LSB-Source	• If the size of the source data word is **smaller** than the available data portion, then the **LSB-bit is set as 1**. The same applies to the data type 32-bit Float. • If the size of the source data word is **greater**, it is possible to set here which payload data from the source data word are to be mapped onto the target data word. In that case, the MSB-source is represented by the lowest bit to be evaluated.
MSB -Target	• If the source data word's data type is Integer, the bits masked out of the payload data portion are copied directly to the target. The MSB-bit target indicates the top bit in the target data word.
LSB-Target	• If the source data word's data type is Integer, the bits masked out of the payload data portion are copied directly to the target. The LSB-bit target indicates the lowest bit in the target data word. The Assistant is used to ensure that the number of bits in the source and the target are the same, if the source is an Integer. If the source data word is of the data type Float, then it must be mapped to an Integer. The number of bits in the Integer is determined by the target data word's upper and lower bit. By means of the factor and offset, or the maximum and minimum, it is possible to change the scaling of the Float number on the mapping. The following rule is applied: `Result = Source word value * Factor + Offset.`

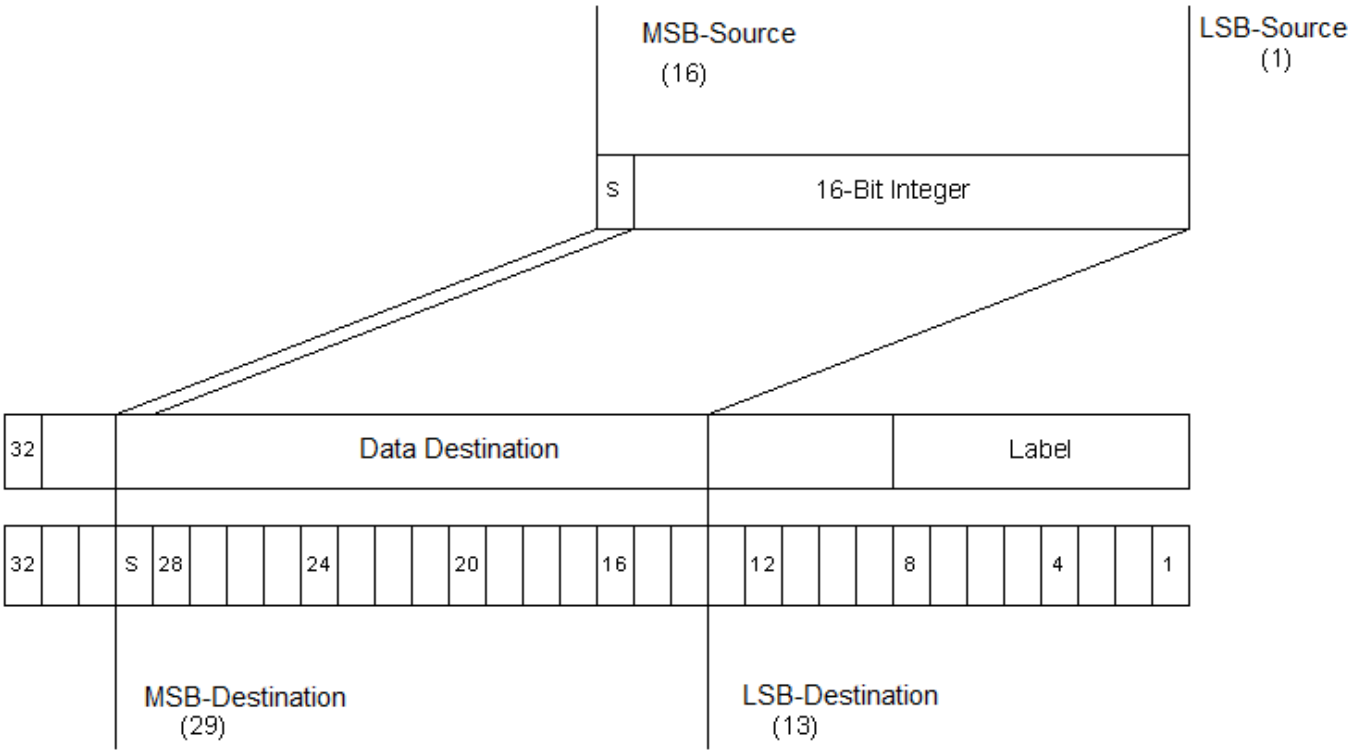
Sign

Multiple variables to be sent can be copied to the label's data portion. They may not overlap.

If the option *Sign (S)* is selected in the area *Data word*, then it is necessary to determine by which variable the sign is determined. I.e. if the variable is negative, then the sign bit is set; for zero or higher, the sign bit is not set.

Even if the option *Sign* is de-selected under *Data word,* it is possible to generate a label with the sign. This is done by arranging the source data word so that the sign (the top bit) is at Bit 29.

The following illustrates how the source word is copied to the data portion.



Copy the source word into the data destination

Scaling

Settings - Scaling	Description
Maximum	Largest measured data value. The highest raw data value is mapped to this value.
Minimum	Lowest measured data value. The lowest value from the raw data maps to this value. Typically, with signed values, the input range is symmetric, i.e. Minimum = -Maximum. If the box for value sign is checked, you must verify whether the minimum is correct, since it will not be adjusted automatically.
Offset	This box is computed automatically. • The offset is computed with sign (S) as: *OFFSET = (Maximum+Minimum) / 2*. • The offset is computed without sign (S) as: *OFFSET = Minimum*.
Resolution	This box's value is computed automatically. You may wish to verify that it matches the column *RESOL* of the ARINC specs, Attachment 2. The resolution is computed as: *RESOL = (Maximum-Minimum) / 2^(SIGDIG + S)* .
Unit	The signal's unit. Corresponds to the column *UNITS* in the ARINC spec sheet, Attachment 2.

Transfer

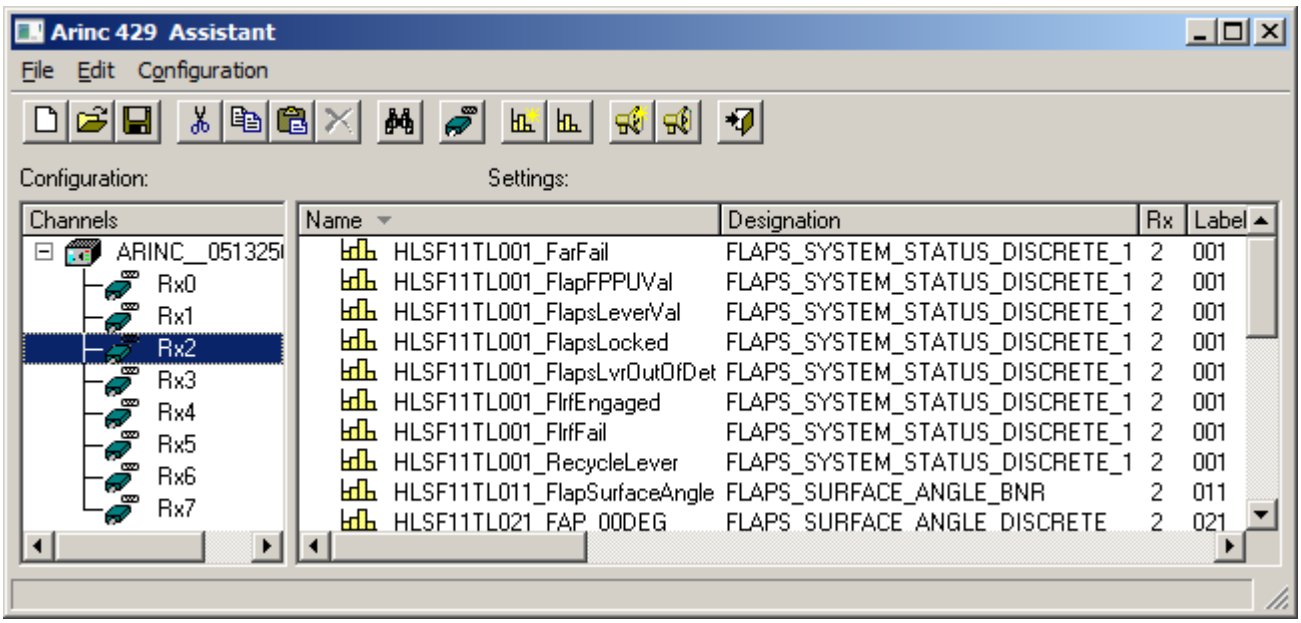
An output rate and a sending condition are defined. The condition is similar to a trigger event in the trigger machine, with the following distinction: there is only one event variable. Conditions defined as combinations of multiple variables linked by the operators AND or OR can only be made using imc Online FAMOS.

Sending is subject to the *Hold-off time* in seconds, where a value of 0 stands for "infinite duration".

Settings - Transfer	Description
Output rate	Cycle time for the output: 500µs - 1h, provided the Baud rate set is adequate.
Event variable	Sending only takes place once the variable entered here meets the condition of the following event.
Event	• For digital event variables: Signal change: 0 -> 1 or 1 -> 0; Signal = 1 or 0. • For analog event variables: Signal < or > Level; Positive or Negative Slope.

9.10.2.3.6 Configuration file list - settings

If the ARINC configuration file list has been filled with definitions for the desired channels, the list may look like the following:



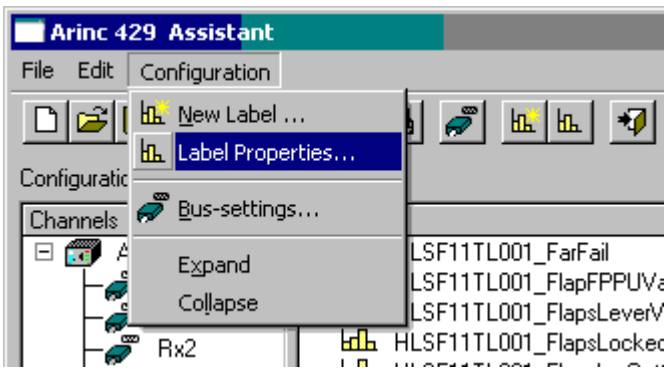
ARINC configration list

All definitions can be sorted as desired by clicking on the corresponding column header.

Each icon stands for one channel definition. After the icon, the name, designation, etc. of the label defined is displayed.

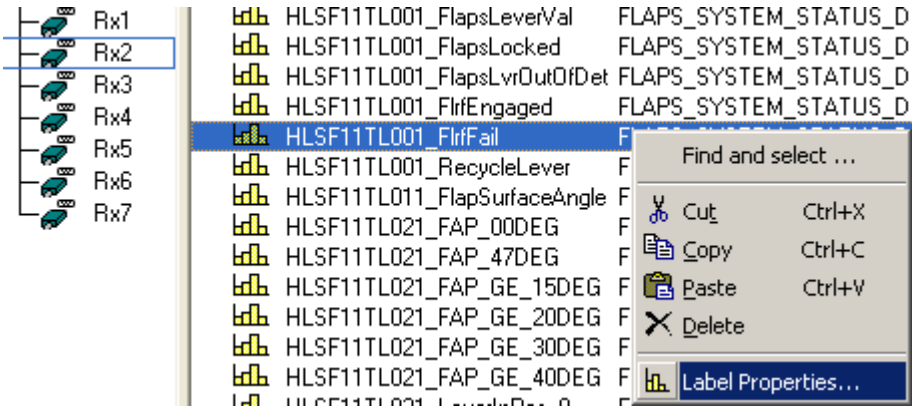
9.10.2.3.7 Editing a label definition

To edit an existing label definition, select a line and then select the menu item *Configuration > Label Properties*.



Entering the label properties

or right-click on the mouse to open the context menu and select the item for the configuration dialog.



Open the label properties using the context menu

The dialog for editing the data word then appears, as shown above.

Note

The transmission order is determined by the comment. The labels are processed in alphabetical order of the comment.

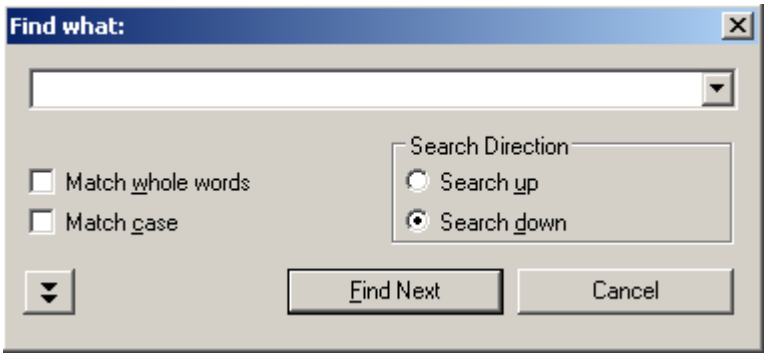
9.10.2.3.8 Cutting, copying, pasting or deleting a label definition

To cut, copy, paste or delete a label definition, it is necessary to first select an entry. Then it is possible to select either the corresponding toolbar button, Edit-menu or context menu item to cut, copy, paste, or delete the selected label definition.

9.10.2.3.9 Searching and selecting a label definition

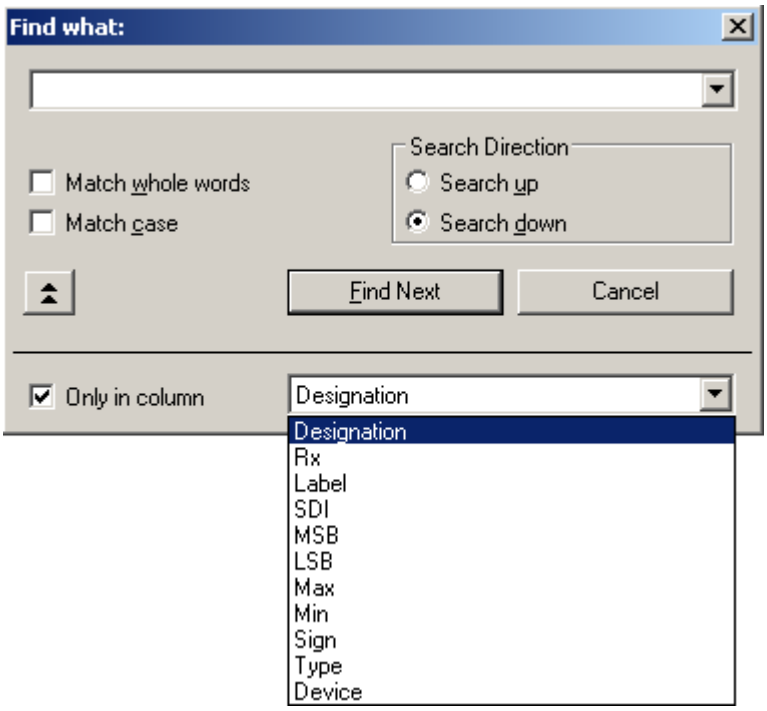
The context menu, called by right-clicking the mouse, also contains an item for the function **Search and select**. Using this function, you can search for a label definition directly.

The **Find** function is available from either the *Edit* menu or by pressing the combination of keys *Ctrl+F*.



Search and find a label definition

Additionally, it is possible to simultaneously search for multiple label definitions having a property in common by activating the function *Only in column*, which is reached via the double-arrow button.



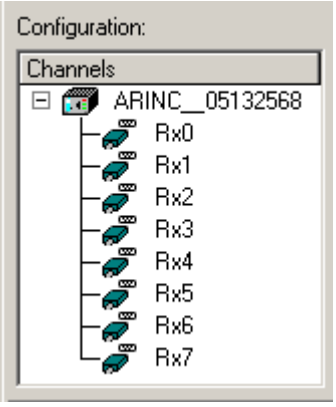
Search in a certain column

9.10.2.4 Configuration

The following sections describe the left side of the Assistant interface, entitled **Configuration**, since all settings made here will be saved by imc STUDIO as the experiment configuration. In the simplest case, a channel (a label definition) from the configuration file is entered when the Assistant is started. When you close the Assistant, this channel is adopted in imc STUDIO. In more complicated cases, the SDI bits and the ARINC bus speed must be set before closing the Assistant. To do this, proceed as follows.

9.10.2.4.1 Configuration list

Entries for all imc measurement devices and their ARINC interfaces are listed here. The figure below presents an example:



Configuration list

The icon represents a measurement device. Next to it is the device name.

The icons connected to it with lines symbolize the device's ARINC connections.

They are numbered and carry the title "Rx" (receiver).

9.10.2.4.2 Changing channel parameters

Channel parameters like channel name, sampling rate, measurement duration etc. are changed in the same way as with an analog channel.

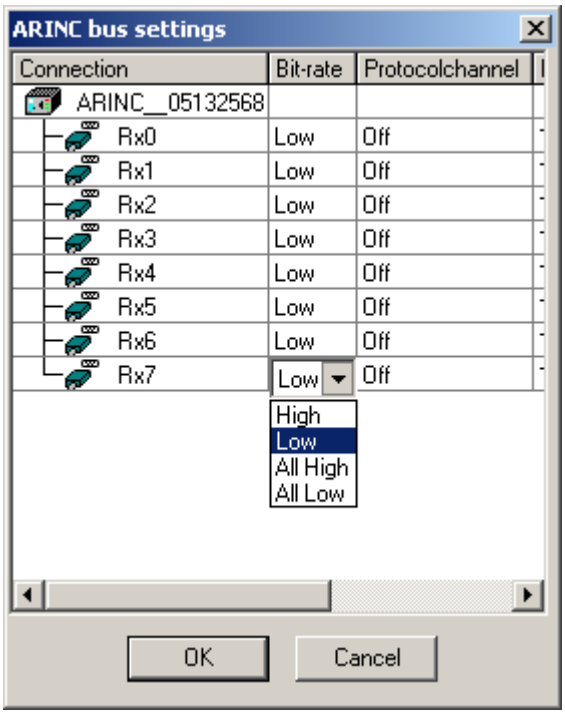
All channels configured with the Assistant are also listed as fieldbus channels.

9.10.2.4.3 Setting the ARINC bus speed

The ARINC bus can be configured for **Low** Speed (12 kbit/s) or **High** Speed (100 kbit/s), **All High** or **All Low**. The dialog for making settings to bus hardware can be called either by selecting the menu item *Configuration > Bus settings* or

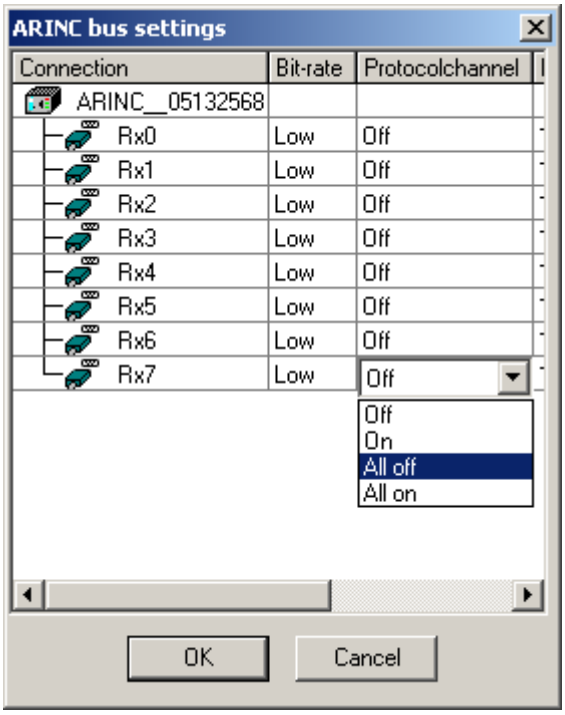
1. by clicking on the button .

2. By clicking in the column *Bit-Rate* on the corresponding row, a combobox, appears from which the associated bus speed can be selected.



Setting the bus speed

If the bus speed is set to the same value for all busses, the option *All High* or respectively *All Low* can be selected.



Record certain Log channels

It is also possible to make a record of a node's entire data traffic by selecting the option *On* for the respective node. The Log channel is then assigned the same name as the node. If either all nodes or no nodes are to be recorded, the option *All On* or *All Off* can be selected.

Log channels can be used for offline analysis of the data traffic, e.g. by means of the program imc FAMOS.

Note

When log channels are activated, these are indicated in the channel list.

If a new IDB channel is later imported via the IDB interface, the log channels disappear from the channel list, but are still activated in the Bus settings.

Upon each new import of an ARINC-configuration, these must then be first dactivated, then activated again; this will cause all the useal consequences of creating new channels (setting properties, storage options, etc.).

9.10.2.5 Specification IDB configuration file

9.10.2.5.1 ARINC BNR parameter

Integer	Label	210
Integer	Equipment ID	7
Integer	Subindex, for multiple "Label"	0
String	Parameter Name	"_278B345A10"
String	Parameter Commentary	"RAM AIR INLET TMP"
Integer	Reserved	0
Integer	Time delay to set drop out value	0
Integer	Reserved	0
Integer	ARINC - FORMAT (0=BNR, 1=BCD, 2= DSC)	0
Integer	Bus Number (0-7) when 8 channel card available	0
Integer	Reserved, bus baudrate (0 Low, 1 High)	1
Integer	SDI Setting: (4=0 00, 0=1 01, 1=2 10, 2=3 11)	4
Float	ARINC MAX WORD RANGE	1000.0
Float	ARINC MIN WORD RANGE	-1000.0
Float	Resolution	0.03
Integer	Number of used Bits	19
Integer	MSB	28
Integer	SDI-bits acitve (0: no 1: yes)	1
Integer	BIT SIGN (0=no, 1=yes)	1
String	UNIT	DEG

Example:

210 7 0 "_278B345A10" "RAW AIR INLET TMP" 0 0 0 0 0 1 4 1000 -1000 0.003 19 29 1 1 DEG

9.10.2.5.2 ARINC BCD parameter

Integer	Label	210
Integer	Equipment ID	7
Integer	Subindex, for multiple "Label"	0
String	Parameter Name	"_278B345A10"
String	Parameter Commentary	"RAM AIR INLET TMP"
Integer	Reserved	0
Integer	Time delay to set drop out value	0
Integer	Reserved	0
Integer	ARINC - FORMAT (0=BNR, 1=BCD, 2=DSC)	1
Integer	Bus number (0-7) when 8 channel card available	0
Integer	Reserved, bus baudrate (0 Low, 1 High)	1
Integer	SDI Setting: (4->0 00, 0->1 01, 1->2 10, 2->3 11)	4
Float	ARINC MAX WORD RANGE	1000.0
Float	ARINC MIN WORD RANGE	-1000.0
Float	Resolution	0.03
Integer	BIT SETTING (5=11; 29, 4-=5;29, 3=19;29, 2=23;29, 1=27;29) number of digits	3
Integer	MSB	29
Integer	SDI-bits acitve (0: no 1: yes)	1
Integer	BIT SIGN (0=no, 1=yes)	1
Integer	BCD-type: (0=decimal, 1=time, 2=angle)	1
String	UNIT	DEG

Difference to BNR 499

Example:

210 7 0 "_278B345A10" "RAW AIR INLET TMP" 0 0 0 **1** 0 1 4 1000 -1000 0.003 **3** 29 1 **1** 1 DEG

9.10.2.5.3 ARINC DSC parameter

Integer	Label	210
Integer	Equipment ID	7
Integer	Subindex, for multiple "Label"	0
String	ARINC _LABEL NAME	"Label_210_Rx_00"
String	ARINC _LABEL NAME COMMENTARY	"Label_210_Rx_00"
Integer	Reserved	0
Integer	Time delay to set drop out value	0
Integer	Reserved	0
Integer	ARINC - FORMAT (0=BNR, 1=BCD, 2=DSC)	2
Integer	Bus Number (0-7) when 8 channel card available	0
Integer	Reserved, bus baudrate (0 Low, 1 High)	1
Integer	SDI Setting: (4=0 00, 0=1 01, 1=2 10, 2=3 11)	4
Integer	Bitmask, absolut for complete 32 bit-word	8000
Integer	Number of bits	1
Integer	MSB	20
Integer	SDI-bits acitve (0: no 1: yes)	1
String	BIT NAME	"-279C111B0000"
String	BIT COMMENTARY	"FALSE VENTO A1"

Example:

210 7 0 "_ Label_210_Rx_00" " Label_210_Rx_00 " 0 0 0 2 0 1 4 8000 0 1 20 1 DEG "_ 279C111B000" "FALSE VENTIL A1"

9.10.3 AFDX-bus interface

Introduction - What is the AFDX-Bus?

AFDX: Avionics Full Duplex Switched Ethernet

The ARINC-664 standard uses the fundamental elements of the AFDX, which is based on IEEE 802.3. Since the main elements consist of AFDX end systems, AFDX switches and AFDX links, the AFDX is a kind of network rather than a bus system.

The end systems communicate via so-called Virtual Links (VL), which define the connection in one direction between a source and a destination. The monitoring of the data transfer rates on the AFDX is also performed on the basis of these Virtual Links, since each Virtual Link has a particular bandwidth.

A frame on the AFDX fundamentally consist of the network data such as Mac-address, IP and UDP, as well as the data payload. This payload can consist of up to 1472 8-bit characters. However, only 1471 of the 8-bit characters are used as real payload. The remaining 8-bit character is needed for the sequence number (SN).

The payload data themselves consist of a function status (FS) and the associated data set (DS). In the function status, the associated data set's properties are described. These properties and additional AFDX items can be defined in the AFDX-Assistant.

Note

Also observe the general notes on file buses in regard to control philosophy 474 and the sampling rate either as fixed sampling rate 752 or time stamp 752.

9.10.3.1 AFDX-assistant

Before it is possible to record an AFDX channel with the imc measurement device, the first step is to define the AFDX data word to be captured. This task is handled by the AFDX-Assistant.

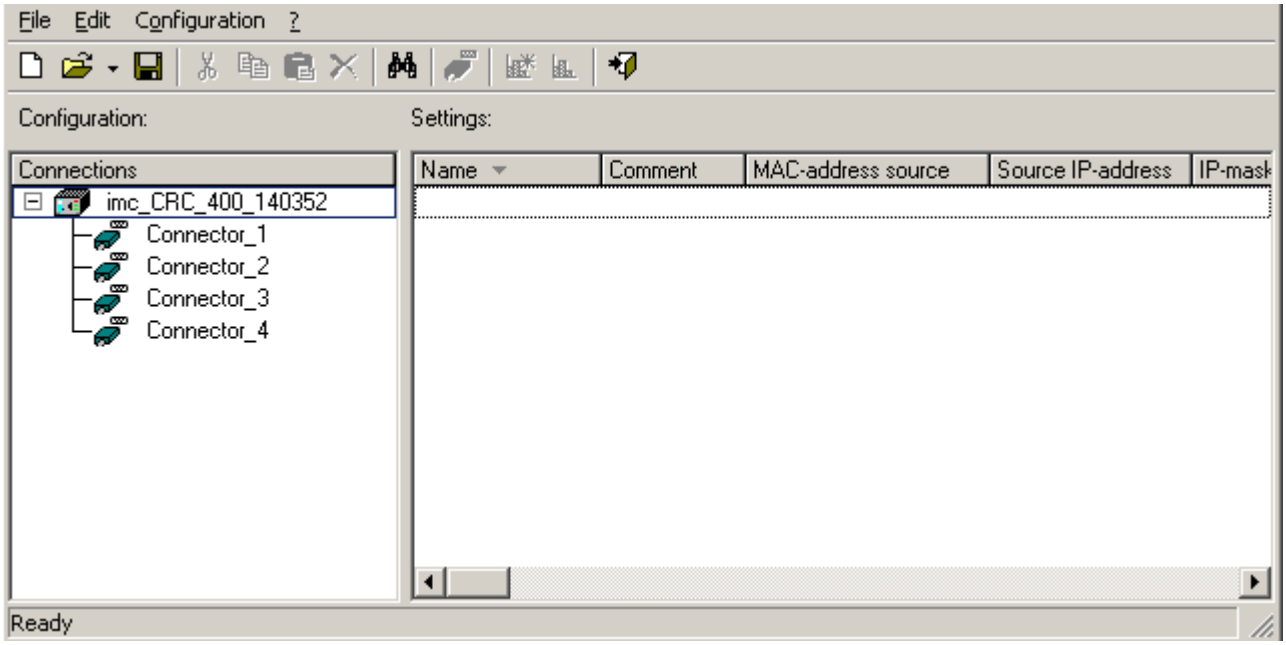
Once this step has been completed and the Assistant closed, an additional measurement channel is available in imc STUDIO, which can be used in a very similar way to any analog or digital channel. This means that the channel can be subjected to operations in imc Online FAMOS, for example, or linked together with analog channels to a trigger and recorded in response to it.

Thus, only the settings specific to the AFDX-bus must be made with the AFDX Assistant. All other properties such as the measurement duration or the trigger linkages are handled in the exact same way as if working with a regular analog or digital measurement channel.

9.10.3.2 Starting the assistant

In order to launch the Assistant from imc STUDIO, click in the menu ribbon on *Setup-Configuration > AFDX-Assistant*.

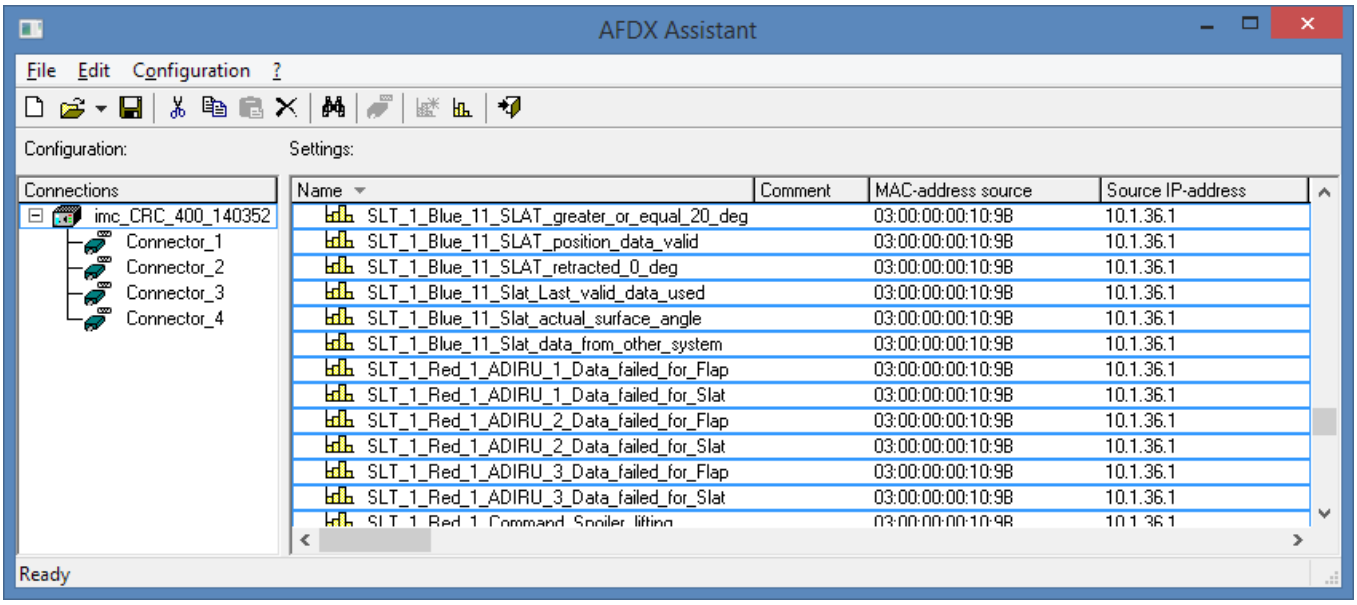
After launching, the Assistant appears with the following dialog:



AFDX-Assistant

The AFDX-Assistant's main window has two tiles:

- The left tile under the heading **Configuration** lists the device and the receiver channels belonging to it.
- The right tile under the heading **Settings** lists the AFDX channels which belong to the currently selected connector-channel and have been defined.



AFDX-Assistant, main window (example)

The definitions for the respective AFDX channels made here under **Settings** are saved in a dedicated configuration file (.XML format), where they are independent of the particular experiment. This configuration file can be regarded as a sort of database containing all defined AFDX-channels along with their respective channel assignments. Thus, to configure an experiment with the AFDX-bus, one only needs to load an AFDX configuration file and to close the Assistant. All AFDX channels defined then appear in the imc STUDIO operating software as channels and can be either displayed or subjected to further processing in imc Online FAMOS.

9.10.3.3 AFDX-configuration file

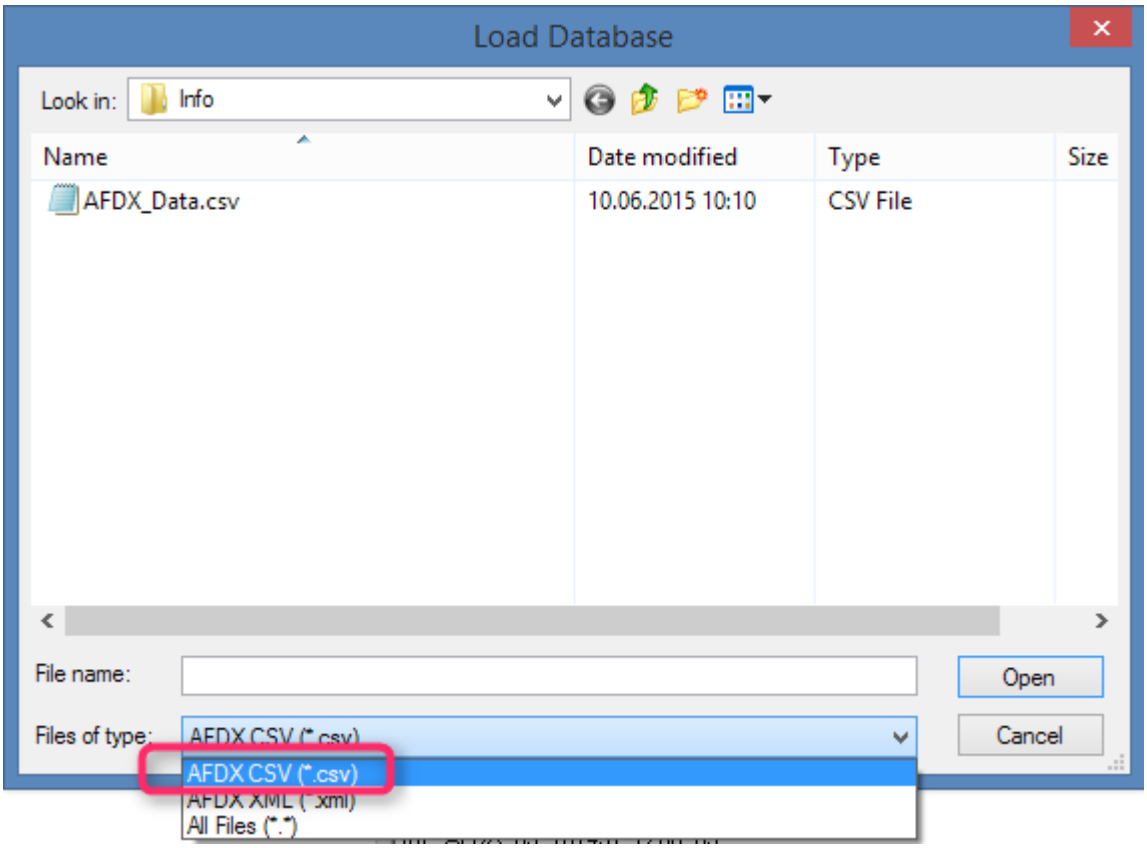
All AFDX channel definitions are saved in a configuration file. This configuration file can be used as a central database. The file can be loaded for every experiment, the AFDX channels can be deleted, copied, or moved as desired. Thus, the signal definitions are independent of the particular experiment. Based on this principle, a custom configuration for a new experiment can quickly be generated at any time from an existing configuration file.

9.10.3.3.1 Creating a new configuration file

To create a new configuration file, either click on the button in the toolbar, or select the menu item *File> New*.

9.10.3.3.2 Loading a configuration file

to load an existing configuration file, either click on the button in the toolbar or select the menu item *File > Load...*. By default, the file has the extension *.xml . It is also possible to load *.csv-files, in order to import one's own personal AFDX channel databases.



Loading a configuration file

9.10.3.3.3 Saving a configuratoin file (Save As...)

To save a configuration file, either click on the button in the toolbar or select the menu item *File > Save*.

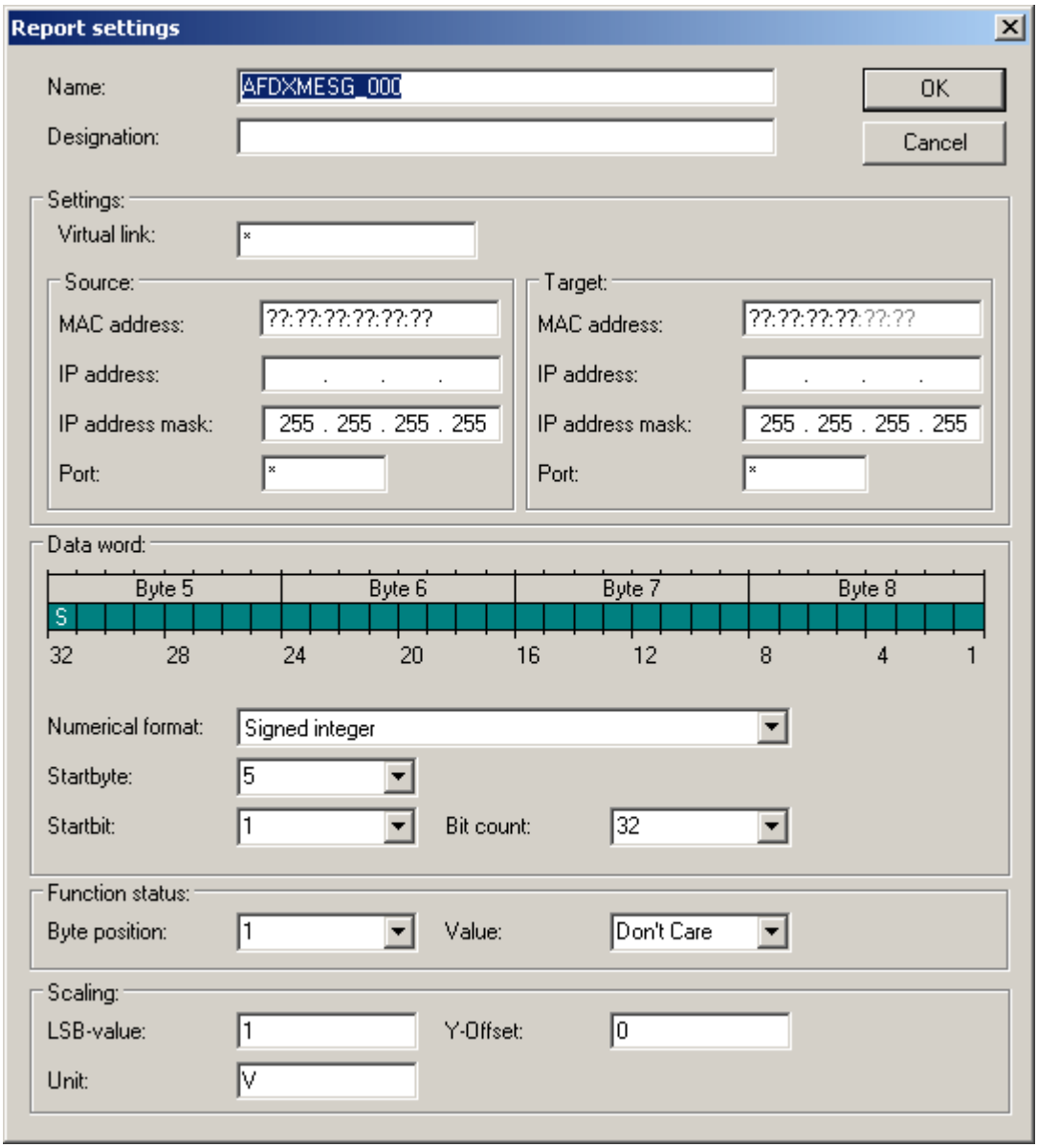
If you wish to save the loaded configuration file under a different name or path, use the menu item *File > Save As...*. In the Windows "File" dialog, you must ehn enter the filename without an extension. The extension *.xml is appended automatically.

9.10.3.3.4 Creating a new AFDX-channel

A new AFDX-channel can only be created in the configuration file and never directly in imc STUDIO: Setup.

- To create a new AFDX channel, either load an existing configuration file or create a new one.
- Next, in the left section of the dialog, Configuration, select the connector-channel on which the AFDX channel is to be received.
- Then either click on the button in the toolbar or select the menu item *Configuration > New Channel*.

A dialog then appears:



Creating an AFDX channel

The dialog is subdivided into 5 regions. In the top region, you can enter the channel's name plus an extra designation for the newly created channel.

The regions Settings 507, Data Word 508, Function Status 508 and Scaling 509 are described further below.

Settings	Description
Name	An arbitrary string of up to 255 characters in length. The usual limitations governing channel names in imc STUDIO apply. The name is assigned to this channel and displayed in the Assistant interface. it will also be used as the channel name in imc STUDIO.
Designation	An arbitrary string of up to 255 characters in length which may contain any characters desired. The designation is assigned to this channel and displayed in the Assistant interface. It is also noted in imc STUDIO as a channel comment.

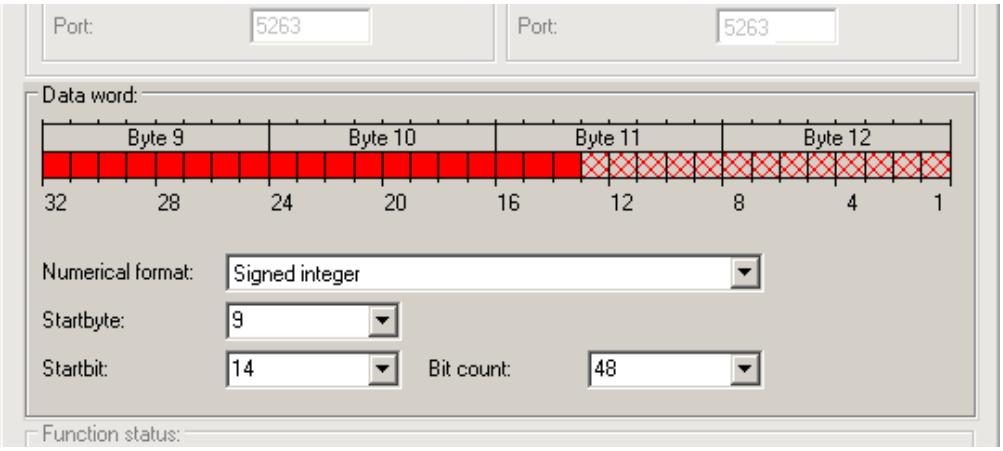
Settings

Settings	Description
Virtual Link	Here, a virtual link can be entered, which defines the connection in one direction from a source to a destination.

Settings - Source and Target	Description
MAC Address	Here, the MAC address of the source/target in the network is specified. The MAC address serves as unique identification in the network and enables more specific monitoring in conjunction with the use of the virtual links. The MAC addresses are 48 bits in size and are entered in hexadecimal format.
IP Address	Here, the IP address of the source/target in the network is specified. The IP address serves the purpose of addressing a subscriber within a network.
IP Address Mask	Here, the IP address mask of the source/target is entered, within which the corresponding IP address is located. The IP address mask denotes the network in which the source/target is located.
Port	Here, the source's/target's port is entered.

Data Word

In a data word, 128 bits are available. Since the data are always sent in 4 Byte-sized Function-Status-sets, the dialog also always displays only 4 Bytes. Bits which are not used are indicated in the dialog by hatched boxes, while bits which are used are represented by filled boxes. The available numerical formats for the data word are Digital, Signed Integer and real.

Settings - Data Word	Description
Numerical format	The available options are **Digital**, **Signed Integer** and **Real**. For all three numerical formats, the start byte can be selected.
Start byte	In the single-line text box, the desired start byte can be selected. Due to the limitation of the message size, only values up to 1496 are possible for the start byte.
Start bit	The start bit can only be selected for the *numerical formats Digital* and *Signed Integer*. The start bit must be ≤ 32.
Bit count	In the numerical format *Signed Integer*, it is additional possible to select the bit count. If the sum of the start bit and the bit count is greater than the number of the 32 available bits, the color of the fill in the bit boxes changes from green to red. In this case, the setting is **not accepted** by the AFDX-Assistant.  *Bit count*

Function Status

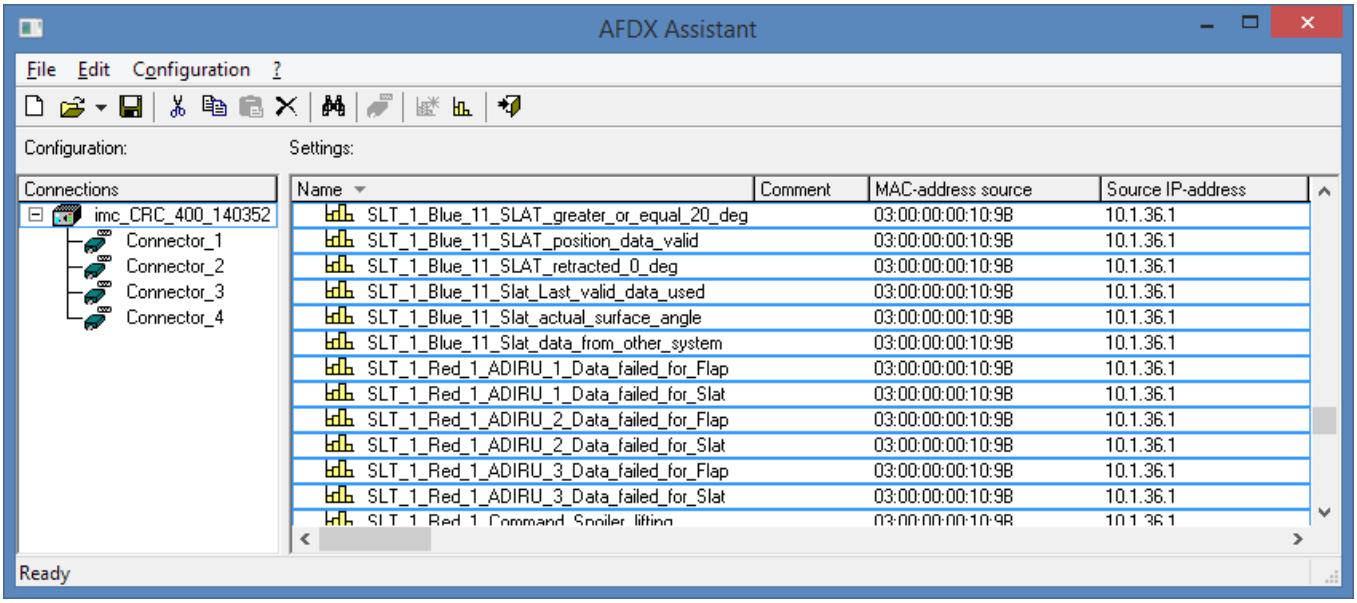
Settings - Function Status	Description
Byte position	Here, the byte position of the corresponding function status can be set. Note that the byte position of the function status proceeds the associated data package. Due to the limitation on message size, the byte position must be below 1500.
Value	Here, it is possible both to select among the three function statuses **No Data**, **Normal Operation** and **Functional Test**, as well as to enter a value ≤ 255 corresponding to the 8 available bits.

Scaling

Settings - Scaling	Description
LSB-value	The LSB-value specifies the factor with which the actual value is multiplied. Along with integer values, it is also possible to enter values in exponential representation such as 1.02e+3 or 1.02e-02.
Unit	The signal's unit of measure.
Y-Offset	Here, an offset can be specified, which in conjunction with the LSB value produces the ultimate result. Result = Value x LSB-Value + Offset

9.10.3.3.5 Configuration file list - settings

If the AFDX configuration file list was filled with AFDX-channels for the desired connector-channels, the list will appear similar to the example below:



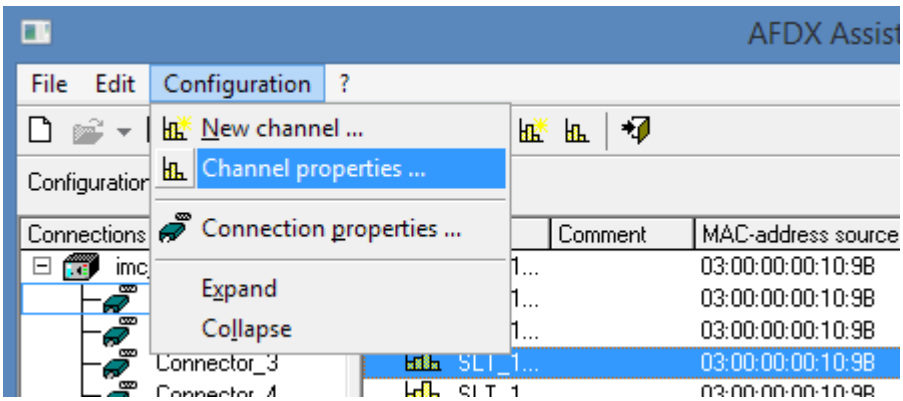
Configuration file list

All AFDX-channels can be sorted however desired by clicking on the corresponding column.

All icons represent the definition of a channel. After the icon, the name, designation etc. of the defined AFDX channel is stated.

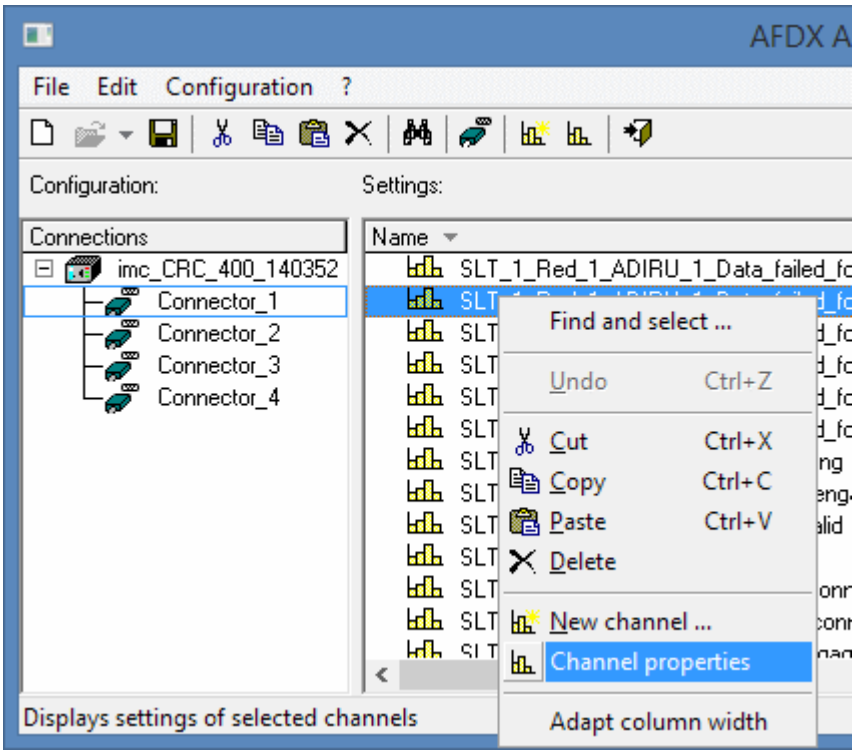
9.10.3.3.6 Editing an AFDX-channel

To edit an existing AFDX-channel, one selects a line and then selects the menu item *Configuration > Properties Channels*



Properties of an AFDX channel via the menu

or one opens the Configuration dialog by right-clicking on the corresponding line.



Properties via the context menu

The dialog for setting an AFDX-channel; see above.

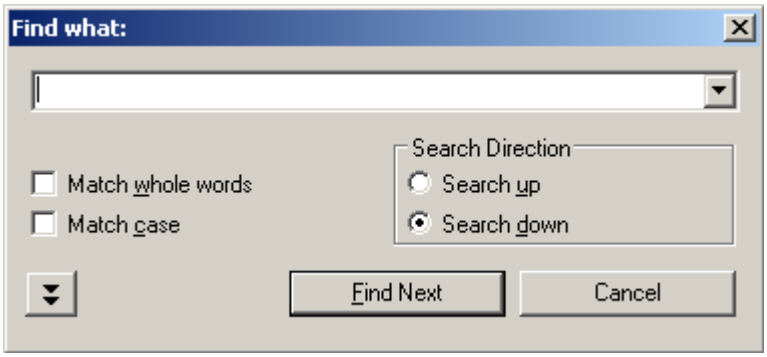
9.10.3.3.7 Cutting, copying, pasting or deleting an AFDX-channel

To cut, copy, paste or delete an AFDX-channel, it is necessary to first select an entry. Then, either by clicking on the corresponding button in the toolbar or by right-clicking on the selected d, it is possible to select one of the choices for editing the AFDX-channel: Cut, Copy, Paste or Delete.

9.10.3.3.8 Finding and selecting an AFDX-channel

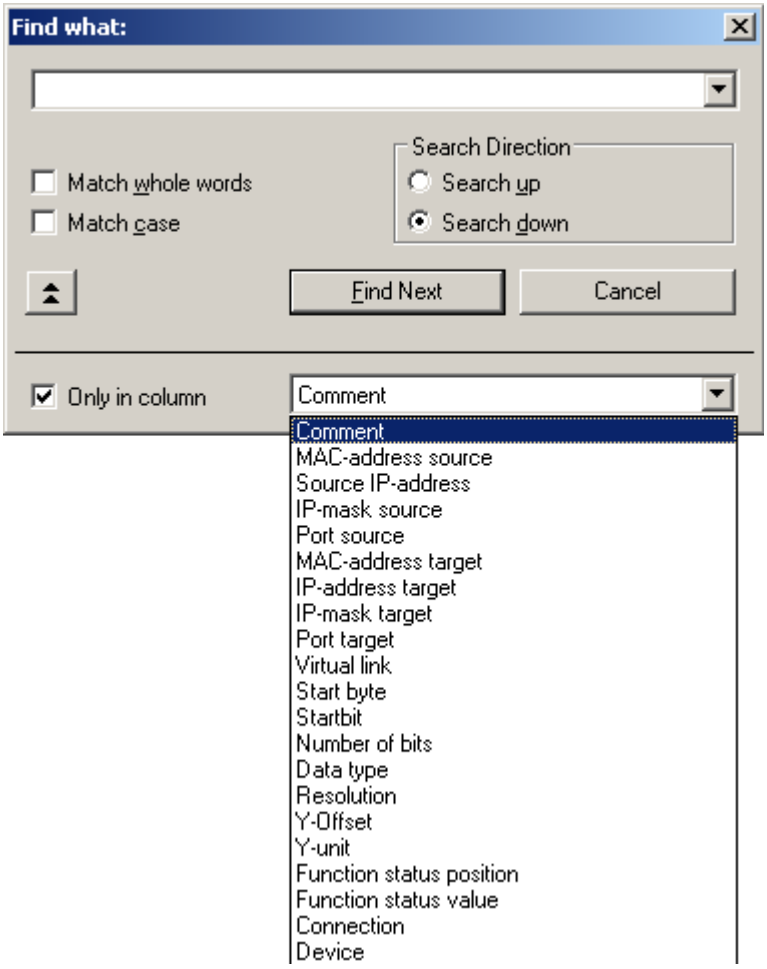
The context menu (accessed by right-clicking the mouse) also offers the function *Find and select...*. Using this, it is possible to directly search for particular AFDX-channels.

By means of the menu item *Edit*, or the keyboard combination *Ctrl+F*, the function **Find** is available.



Searching for an AFDX channel

Additionally, it is possible to select multiple channel definitions having a property in common, by selecting the function **Only in Column**, which is accessed by clicking on the double-arrow button.



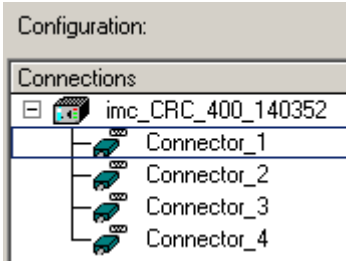
Searching in a particular column

9.10.3.4 Configuration

The next sections describe the left side of the Assistant, which is designated **Configuration**, since everything set in this half of the dialog is saved in the configuration of the imc STUDIO experiment. In the simplest case, after starting the Assistant, an AFDX-channel is added from a configuration file. Closing the Assistant then causes this channel to be adopted in imc STUDIO. The details of how this is done are described next.

9.10.3.4.1 Configuration list

In the (empty) configuration list, all imc devices with their AFDX-interfaces are listed. The following image provides an example:



Configuration list

The icons each represent a measurement device. After each one is the respective device name.

The icons joined by lines to it symbolize the AFDX-channels, which are serially numbered.

9.10.3.4.2 Editing channel parameters

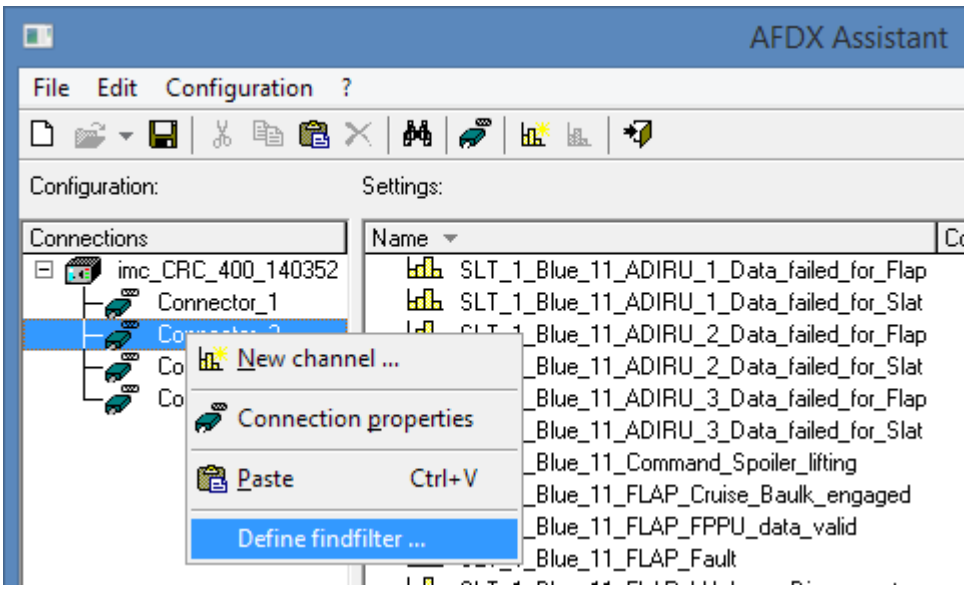
Channel parameters such as the channel name, sampling time, measurement duration and etc. are edited in the same way as for an analog channel.

All channels configured in the Assistant appear with the Fieldbus inputs.

9.10.3.4.3 Defining search filters

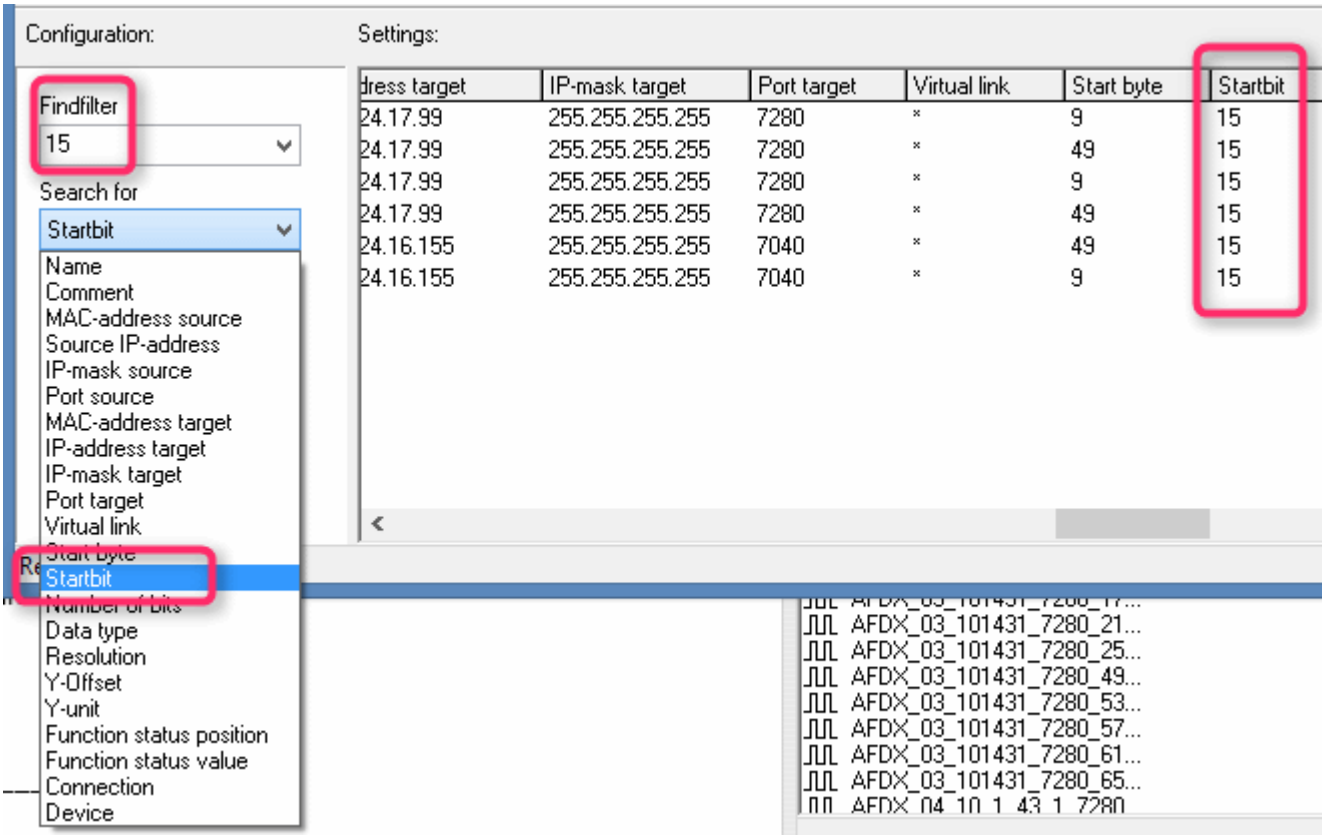
To start the search filter, open the context menu in the Connections list and select **Define FindFilter**.

This Filter supports the display and editing of channels belonging to a particular group by means of a wildcard-based search.



Calling the Search Filter

In the pop-down list **Search for**, select the property, and in the box above it, the value for which to search.



Using a search filter

9.10.4 CAN-bus interface

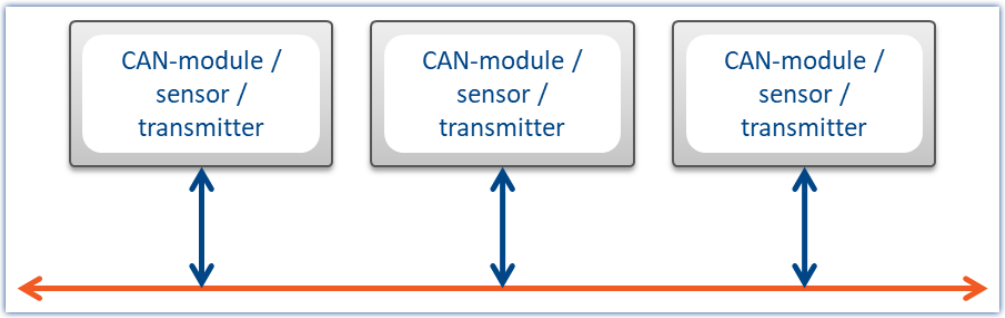
What is the CAN-Bus?

A CAN-Bus interface is optionally available for imc measurement devices, offering full support of CAN-Bus sensors using basic data formats. It is thus possible to take measurements via the CAN-Bus and via the analog and digital inputs practically synchronously.

The CAN-Bus (Controller Area Network) is a serial transmission line connecting all modules in a two-wire circuit. To prevent signal reflection, terminators are connected at the cable ends. The module holder is designed to conform to CAN-Bus operation per CiA standards (CiA Draft Standard 102 Version 2.0, CAN Physical Layer for Industrial Applications).

Multiple sensors and devices can be connected to the CAN-Bus, and can output their measured values at regular intervals onto the bus, for instance. Every device and every sensor sends its data as a ***message*** on the bus. Up to 8 bytes of data can be sent with one message. Every message possesses a unique ***identifier***. A signal's identifying information such as the channel name and units is not sent with the message. Instead it is usually recorded in a database which provides a key to how the message's data is encoded, based on its identifier.

Every CAN-module is connected to the CAN-Bus at a node.



CAN-Bus

9.10.4.1 Informations

Further references

- CiA® Draft Standard 102 Version 2.0: CAN Physical Layer
- CAN Controller Area Network, Wolfhard Lawrenz, Hüthig Verlage, 1994 Heidelberg
- ISO / DIS 11898 (ISO 11519-2) for bus driver
- ISO / OSI reference model
- CAN FD (ISO 11898-1:2015)

Exception: no support of Remote Frames (RTR)

Bus features

The CAN-Bus interface includes the following features:

- CAN Transceiver in accordance with ISO / DIS 11898, isolated
- Baud rate can be adjusted per software
- Standard identifier on CAN-Bus: 11-bit identifier (0..2047)
- Extended identifier on CAN-Bus: 29-bit identifier (0 .. 536870911)
- CAN FD (ISO 11898-1:2015)

Note

Also observe the general notes on file buses in regard to control philosophy 474 and the sampling rate either as fixed sampling rate 752 or time stamp 752.

9.10.4.2 Connection to the device

9.10.4.2.1 CAN port pin configuration

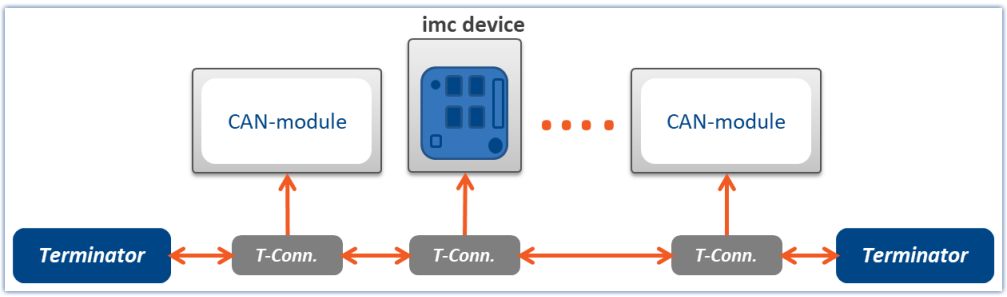
The device comes with a pair of terminals for each CAN-node, configured as follows:

DSUB-PIN	Signal	Description	Usage
1	+CAN_SUPPLY	depending on hardware	supply I < 1 A
2	**CAN_L**	**Dominant low bus line**	**Connected**
3	**CAN_GND**	**CAN Ground**	**Connected**
4	nc	Reserved	Not used
5	-CAN_SUPPLY	depending on hardware	
6	**CAN_GND**	**Optional CAN Ground**	**Connected**
7	**CAN_H**	**Dominant high bus line**	**Connected**
8	nc	Reserved (error line)	Not used
9	nc		Not used

9.10.4.2.2 CAN cabling (Y-adaptor)

Device connection with a Y-adaptor

The CAN interfaces from imc provide exactly one DSUB9 connection for each node. Connection to a CAN bus therefore requires a Y-adapter.



Device connected with Y-adapter

To guarantee a CAN data transfer rate of 1 Mbit/s, the **maximum length** for the pathway through a Y-adaptor is **30 cm**. Usually the wiring within the device is already 30 cm. If using an external Y-adaptor it has to be attached directly at the connector.

In this case it doesn't matter whether the other CAN participants are connected with or without a Y-adaptor. The diagram shows just one of several connection possibilities.

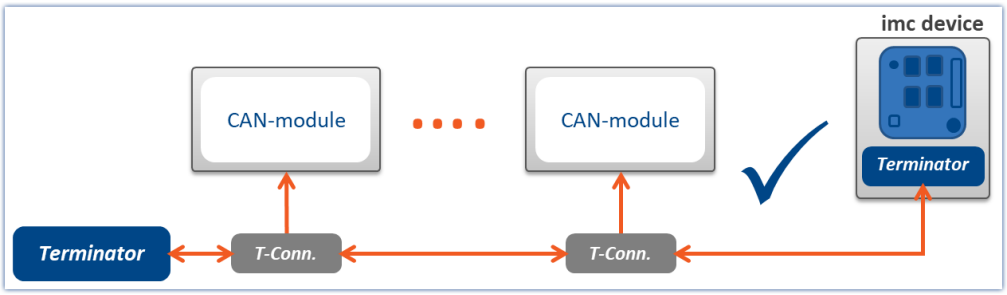
9.10.4.2.3 Connecting terminators

- Terminators with 124 Ω in accordance with CiA.
- Terminators are connected between pin 2 and 7.
- Terminators must be used at both ends of the bus. No further terminators are allowed.

Terminator in device

Devices equipped with CAN-bus interface, can switch a terminator resistor internal via software. If the device is connected at one end of the CAN line, an external Y-cable with terminator is not needed.

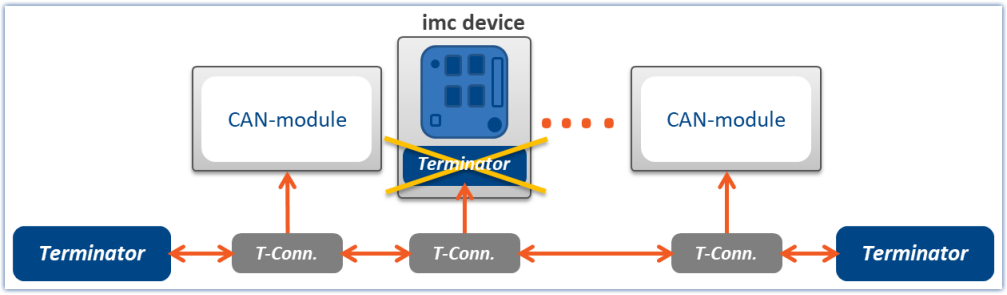
- See: Setting for a device of firmware group A 528
- See: Setting for a device of firmware group B 564



Correct termination

Warning

If the CAN-bus is already terminated correctly, that option cannot be used!



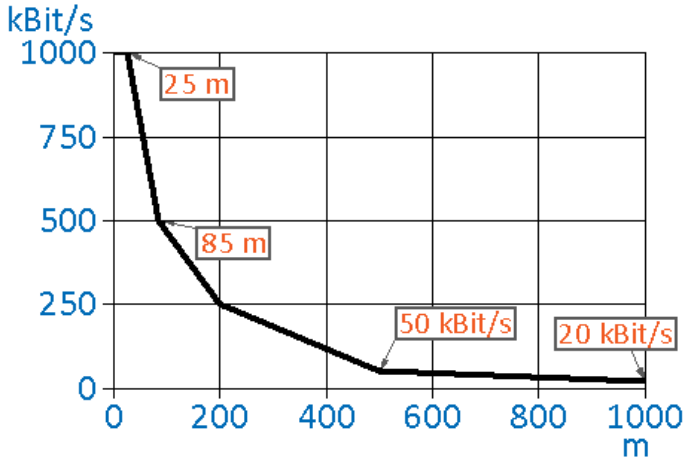
Wrong termination

9.10.4.3 Technical specification

9.10.4.3.1 CAN transfer rate

With the transfer rate, it is necessary to distinguish between standard CAN and CAN FD (CAN with flexible data rate). The following description pertains primarily to the classic, standard CAN-Bus.

As signal lead lines become longer, the maximum reliable data transfer rate decreases.



Data rate for imc devices

Line length [m]	Data rate [kbit/s]	
3	8000	CAN FD
5	5000	CAN FD
12,5	2000	CAN FD
25	1000	
85	500	
200	250	
500	50	
1000	20	

Net data transmission rate

Conversely, the net data rate depends on the package size and on whether the Standard Frame or the Extended Frame is used.

A high net data rate is achieved by using the maximum number of data bytes for each message. If each message in the **CAN standard** is optimally filled with **8 bytes** and the standard frame is used, the result is **576.6 kbit/s**.

The table shows the net data rates for **CAN standard** 1 Mbit/s.

Data length	Net data rate with CAN-standard 1Mbit/s	
	Standard Frame	Extended Frame
0	-	-
1	72,1 kbit/s	61,1 kbit/s
2	144,1 kbit/s	122,1 kbit/s
3	216,2 kbit/s	183,2 kbit/s
4	288,3 kbit/s	244,3 kbit/s
5	360,4 kbit/s	305,3 kbit/s
6	432,4 kbit/s	366,4 kbit/s
7	504,5 kbit/s	427,5 kbit/s
8	576,6 kbit/s	488,5 kbit/s

With **CAN FD**, even **without** the faster data transmission in the data phase, the larger data volume alone provides considerable advantages. At a nominal bit rate of **1 Mbit/s**, the net data transmission rate for a CAN FD frame with a data length of **64 bytes** is **903 kbit/s**.

9.10.4.3.2 CAN FD

CAN FD (CAN with flexible data rate) offers a **8 higher data rate of up to 8 Mbit/s** and up to **64 Bytes of payload data per message**.

Fundamentally, the same network topology can be used as with the classic CAN-protocol. However, the previous CAN-controllers are not able to send or receive CAN FD messages > 8 Bytes. For this reason, CAN and CAN FD can work together in one bus-system if:

- no Remote-Frame is sent (something which imc software does not generate),
- no CAN-messages > 8 Bytes are used
- no Switch Bit Rate 542 is used.

Remarks on CAN FD

- It is possible to send and receive larger messages with imc Online FAMOS 543.
- Fundamentally, all CAN-experiments can be transferred to devices with CAN FD.
- Messages can be sent at a higher baud rate using the option "*Switch Bit Rate* 542".
- CAN FD configurations can be imported to CAN-devices. The configuration is then converted to CAN and resulting errors must subsequently be eliminated by the user.

9.10.4.3.3 Number of CAN nodes

Correlation between line length, number of nodes and cable cross section. The cross section of the line should increase in size as more CAN-nodes are served by the system.

Line length	Number of nodes/ Cable cross section		
	32	64	100
100 m	0.25 mm²	0.25 mm²	0.25 mm²
250 m	0.34 mm²	0.5 mm²	0.5 mm²
500 m	0.75 mm²	0.75 mm²	1 mm²

9.10.4.3.4 Specifications from CAN bus in imc STUDIO

- Baud rate up to 1 Mbit/s; 8 Mbit/s with CAN FD
- Multiple independent CAN nodes (ports)
- Aggregate sampling rate on CAN channels: typically up to 15000 samples/s, depends heavily on data types, no. of channels and processing. Analog aggregate sampling rate is not affected.

Devices of the firmware group A 187 (e.g. CRONOS family, C-SERIES, SPARTAN, BUSDAQ)

- The devices can manage up to 512 channels (analog, virtual and CAN channels).
- CAN channel triggers are computed together with analog channel triggers on the unit's motherboard. This can affect the max. aggregate sampling rate on CAN channels!

Devices of the firmware group B 187 (e.g. ARGUS*fit*)

- The devices can manage up to 1000 channels (analog, virtual and CAN channels).
- The CAN bus channels are triggered on the CAN module. The maximum aggregate sampling rate is not influenced by this.

Note Bus Decoder

More channels can be acquired using the Bus Decoder.

9.10.4.3.5 What one needs to know to connect a CAN-sensor

All CAN-Bus sensors transmit messages on the CAN-Bus using a specific data structure. In order for the messages to be understood, a consistent structure must be used. The following parameters (values serve as an example) must be contained in a parameter:

What is supported?

Baud rate: 500 kbit/s

ID: 455

Length: 5 bytes

Repeat rate: every 10 ms

The contents of the various messages are usually listed in a table. Consider the following typical CAN application:

Signal Name	Byte	Bit	Value Range	Conversion
Speed	1	0..7	0...255	50*value
Status	2	4	0-1	---
Compressor ON	2	5	0-1	---
Clutch moment specification	3	0..7	-128..127	*20, in N×m
Braking moment	4	0..7	0..255	*0.39
ABS Indicator	5	0	0..1	---

Note

- Bytes are numbered from 0 to 7 in the imc STUDIO software. Byte 0 is the first byte and Byte 7 is the last byte in an 8-byte message. Bit 0 of each byte is the LSB and Bit 7 is the MSB. Some tables (such as the one above) may differ slightly, in that the bits and bytes are indexed from 1 to 8. In these cases, you need to subtract 1 from these values.
- Also observe the general notes on file buses in regard to control philosophy 474 and the sampling rate either as fixed sampling rate 752 or time stamp 752.

9.10.4.3.6 Supported CAN protocols

What is supported?

- Only layers 1 and 2 of the layer model
- channels which can be identified uniquely from a single identifier and contain exactly one sample per identifier
- sensors which must be commanded via the CAN-Bus to transmit or measure
- device receives data
- device sends data
- data structure as follows:

  Identifier indicates which byte contains channel data and the format of the data. Analog sensor data and binary signals (only 0 or 1) can be read.

Example

The following table illustrates a type of application which is supported:

Signal Name	Byte	Bit	Value Range	Conversion
Speed	1	0..7 unsigned	0...255	50*value
Status	2	4	0-1	---
Compressor ON	2	5	0-1	--

What is not supported?

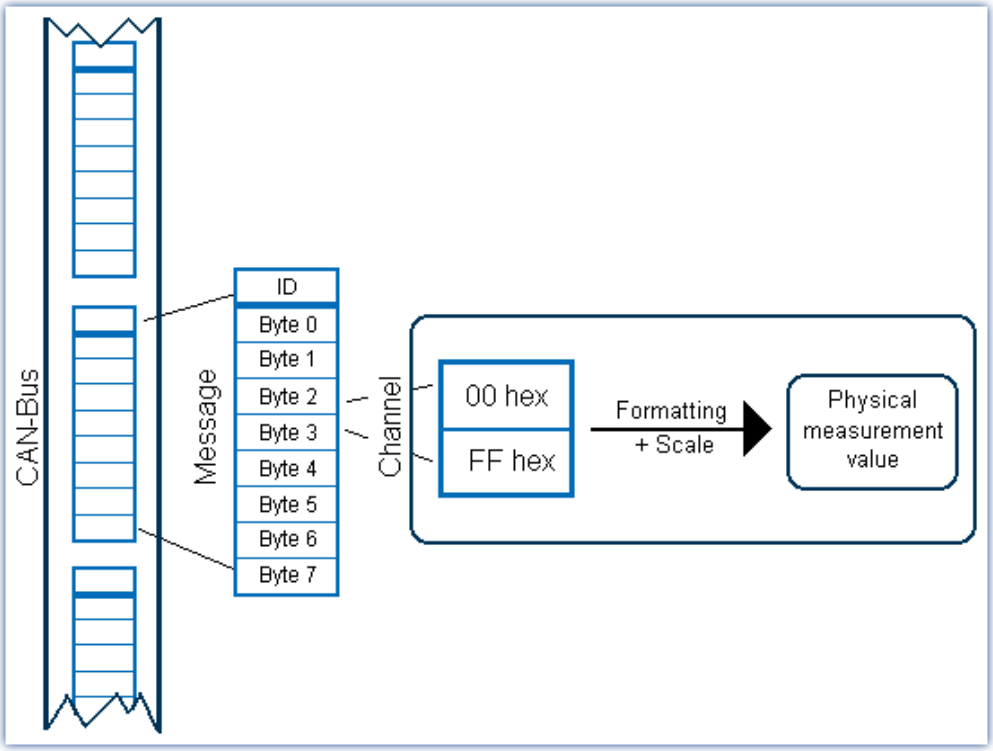
- terminal channels
- recording and analysis of 100-byte error protocols
- control of other devices on the CAN-Bus by the measurement device
- consideration of complex conditions
- compensation for CAN-Bus delay
- multiple protocols
- higher protocols

All major imc STUDIO features available for CAN channels

- Online processing using imc Online FAMOS
- Analog data can be mixed with CAN data for processing
- Data storage on the device internal hard drive and/ or PC hard drive.
- Complete access to the units trigger machine
- The device can be operated as a stand-alone unit, i.e. without a connected PC.
- Binary signals from CAN-Bus possible

How are CAN-Bus channels created in devices?

Several sensors and devices can be connected to a CAN-Bus. Each of these transmits data with an identifier at a constant clock rate to the CAN-Bus. The identifier indicates the origin and meaning of the data. An 8-byte data packet is attached to each of these (64 Byte with CAN FD).



Format

Individual bits or bytes can be read from a message

- These are usually specific measurement values or conditions
- Supported number formats include: 1 bit, 1-8 Byte, signed/ unsigned integers, real numbers, Intel or Motorola byte orders
- Integer values can be converted to physical quantities

Time stamp alternative

Time stamping enables you to protocol the exact transfer time of a CAN-Bus message

- This data can be saved and visualized
- Data can also be equidistantly resampled for digital filtering, computation of FFT's etc.

CAN error handling

Bus clients, individual client sensors or the bus line itself may be defective

- Complete configuration of error handling

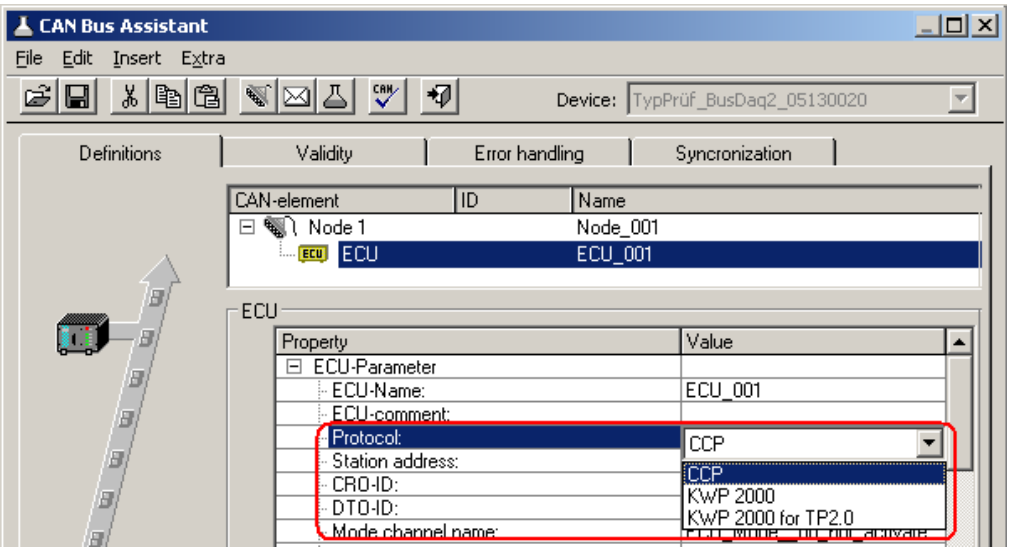
9.10.4.3.7 ECU control unit protocols

The device software comes with functions for (motor-) control units (ECUs) and special control unit protocols. Using these, it is possible to query and record internal control unit values, and to start special control unit values.

Motor control units are connected via the CAN-Bus. The ability to query internal control unit values requires compliance with special protocols.

System requirements	
Hardware	• A device with CAN-module is necessary with the common device control functions and all desired protocols being enabled or • Devices of firmware group A 187
Software	If the special imc Online FAMOS functions are used, then the device must be enabled for imc Online FAMOS.

The support ECU protocols can be found here 584.



Properties of ECU messages

To capture internal control unit values, it is necessary that the control units and the values (channels) to acquire be configured with the help of the CAN-Bus Assistant.

If the return values of the control unit commands are to be analyzed, or if special control unit functions are to be run, then it is necessary to use special functions in an imc Online FAMOS program with control commands.

For the description of the configuration see Control units in the CAN-Bus Assistant 584.

Basic functions

Keyword Protocol 2000 with ISO Transport Protocol (ISO 14230-3)		Keyword Protocol 2000 with VW TP 2.0 (ISO 14230-3)	
A subset of keyword protocol 2000 is used with addressing modes Normal and NormalFixed.			
10h	StartDiagnosticSession	10h	StartDiagnosticSession
13h	ReadDiagnosticTroubleCodes	13h	ReadDiagnosticTroubleCodes
14h	ClearDiagnosticInformation	14h	ClearDiagnosticInformation
17h	ReadStatusOfDiagnosticTroubleCodes	17h	ReadStatusOfDiagnosticTroubleCodes
18h	ReadDTCsByStatus	18h	ReadDTCsByStatus
20h	StopDiagnosticSession	20h	StopDiagnosticSession
21h	ReadDataByLocalIdentifier	21h	ReadDataByLocalIdentifier
22h	ReadDataByCommonIdentifier	22h	ReadDataByCommonIdentifier
23h	ReadDataByAddress	23h	ReadDataByAddress
27h	SecurityAccess	27h	SecurityAccess
2Eh	WriteDataByCommonIdentifier	2Eh	WriteDataByCommonIdentifier
31h	StartRoutineByLocalIdentifier	31h	StartRoutineByLocalIdentifier
33h	RequestRoutineResultsByLocalIdentifier	33h	RequestRoutineResultsByLocalIdentifier
38h	StartRoutineByAddress	38h	StartRoutineByAddress
3Ah	RequestRoutineResultsByAddress	3Ah	RequestRoutineResultsByAddress
3Bh	WriteDataByLocalIdentifier	3Bh	WriteDataByLocalIdentifier
3Dh	WriteDataByAddress (Maximal zwei Byte Daten)	3Dh	WriteDataByAddress (Maximal zwei Byte Daten)
3Eh	TesterPresent	3Eh	TesterPresent

CCP		XCP		Diagnostics on CAN (ISO 15765)	
01h	CONNECT	D3h	ALLOC_ODT_ENTRY	10h	StartDiagnosticSession
03h	DNLOAD	D4h	ALLOC_ODT	12h	ReadFreezeFrameData
02h	SET_MTA	D5h	ALLOC_DAQ	14h	ClearDiagnosticInformation
04h	UPLOAD	D6h	FREE_DAQ	17h	ReadStatusOfDiagnosticTroubleCodes
07h	DISCONNECT	E0h	SET_DAQ_LIST_MODE	21h	ReadDataByLocalIdentifier
0Fh	SHORT_UP	E1h	WRITE_DAQ	22h	ReadDataByIdentifier
12h	GET_SEED	E2h	SET_DAQ_PTR	23h	ReadMemoryByAddress
13h	UNLOCK	E3h	CLEAR_DAQ_LIST	27h	SecurityAccess
14h	GET_DAQ_SIZE	F0h	DNLOAD	2Ch	DynamicallyDefineLocalIdentifier
15h	SET_DAQ_PTR	F4h	SHORT_UPLOAD	31h	StartRoutineByLocalIdentifier
16h	WRITE_DAQ	F5h	UPLOAD	3Bh	WriteDataByLocalIdentifier
		F6h	SET_MTA	3Eh	TesterPresent
		F7h	UNLOCK		
		F8h	GET_SEED		
		FEh	DISCONNECT		
		FFh	CONNECT		

Standards

Keyword Protocol 2000 with ISO transport protocol

- Keyword protocol 2000: ISO 14230-3
- ISO transport protocol: ISO 15762-2

Options and add-on's

The different protocols are optional.

9.10.4.4 CAN channel data formats

The following data formats are available for recording CAN channels:

CAN-Bus Format	Recording with sampling	Recording with time stamping
Unsigned/ signed integer: 1 to 16 bit	2 byte integer	2 byte integer + 6 byte time stamp
Unsigned/ signed integer: 17 to 32 bit	4 byte real	4 byte real + 6 byte time stamp
Real 4, 8 byte	4 byte real	4 byte real + 6 byte time stamp
Digital port with 1 to 16 bit tracks	2 byte word for compact 16 bits	2 byte word for compact 16 bit + 6 byte time stamp

As you can see, each format has different memory requirements. This is important to know when you are estimating how much memory you will need and how precise the measurement values will be.

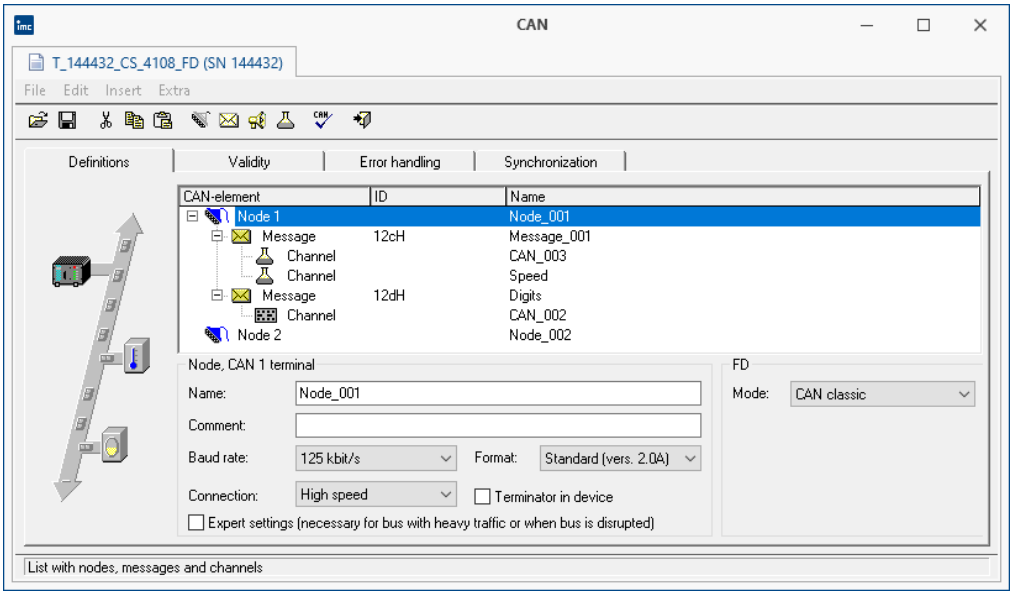
Example

A CAN sensor is sending messages consisting of 8-bit numbers. These are to be directly interpreted as the measurement values and equidistantly sampled (i.e. without time stamping). Therefore, a 2 byte integer is stored for each value.

9.10.4.5 CAN-Bus Assistant (firmware group A)

For the purpose of configuring nodes, messages and channels belonging to the device of the firmware group A 187 (CRONOS, BUSDAQ, SPARTAN, etc.), the *CAN-Bus-Assistant* is used. The assistant for firmware group B 187 (ARGUS) is located here 562.

Ribbon	View
Home > CAN (CAN)	all
Setup-Configuration > CAN (CAN)	Complete



CAN-Bus Assistant

Together, the CAN nodes, messages and channels form a CAN configuration. These can be saved for re-use as ASCII configuration files (*.CBA). We recommend creating a separate folder for these files. Each line in a CBA file consists of a code word and a corresponding value. If necessary, some the values can be edited directly using a text editor like the Windows Notepad. In general, we do not recommend changing information concerning the bitmaps - these properties should only be edited using the CAN-Bus Assistant.

The CAN-Bus Assistant does not contain an *OK* button for confirming changes. Instead, all changes are applied directly. Invalid values are shown in red in the status bar. You can also perform a Configuration Check 556 .

Note

It is possible, however, to exit the Assistant without correcting the changes - however, no CAN channels will be created! Therefore, if you wish to create CAN channels, the configuration must be correct before you exit the Assistant.

9.10.4.5.1 Quick tutorial

The Quick Tutorial shows you how to create a simple CAN-Bus configuration for a single CAN channel. After completing the tutorial, you should be able to create more complex configurations as well.

Step 1: Start the CAN Assistant

Select the Ribbon "*Home*" (or "*Setup-Configuration*") > "*CAN*" to start the Assistant. A single node is visible the first time you start.

Step 2: Define a node

Change the node name to a more suitable name for your project. Specify the baud rate from the drop-down list. Select a format for the node (Standard or Extended format).

Step 3: Create a message

Next, you need to define a message. Select the node and click on the message icon. A message object is added below the node.

Step 4: Configure the message

Change the message name to suit your project. Assign an ID to the message (between 0 and 2031 for Standard format, between 0 and 536870911 for Extended format).

Step 5: Create a channel

Next, you need to define a channel. Select the message and click on the channel icon. A channel object is added below the node.

Step 6: Configure the channel

Change the channel name to suit your project and specify the numerical format of the channel from the drop-down list.

Step 7: Check configuration

Perform a configuration check by clicking on the checkmark button. The results of the check are displayed in the status bar.

Step 8: Quit the Assistant

Once you have exited the CAN-Bus Assistant, you can now use the CAN configuration when measuring via the CAN interface. The CAN-channel you defined now appears in the software interface.

9.10.4.5.2 Editing a CAN configuration

The properties of various CAN objects are arranged in drop-down lists and text boxes. By selecting several objects in the list, you can change their properties all at once. This is only possible, of course, when objects of the same type have been selected. Right-clicking in the list opens a context menu where you can select all objects of a certain type (e.g. nodes, messages or channels).

Note

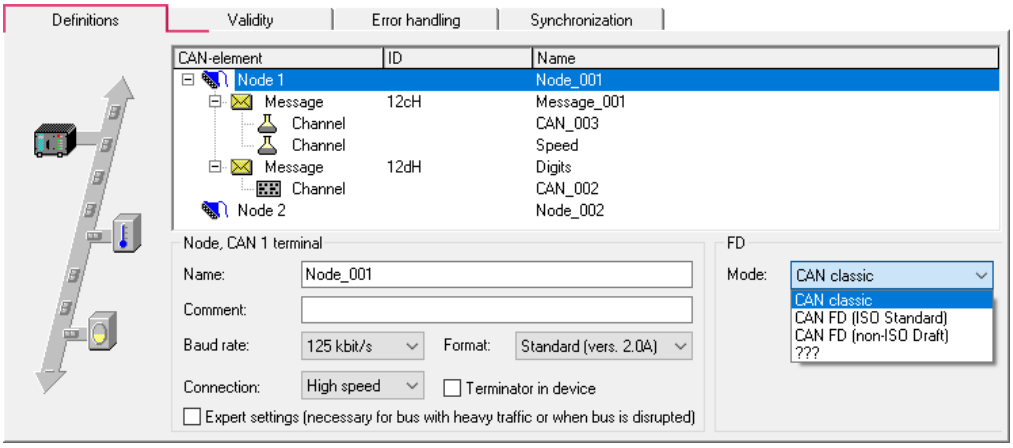
Any changes you make to the properties are applied immediately. Even incorrect settings (e.g. channel names) are applied. Any errors are shown in the status bar at the bottom of the list.

It is not necessary to click on an <*OK*> button to apply any changes or new settings - however, there is no <*Cancel*> button to undo any unwanted changes!

In general, we recommend defining the properties for new objects one-by-one. It is also possible to change settings for several like objects simultaneously, but properties which must be unique (e.g. channel name) will be grayed out.

9.10.4.5.3 Nodes

9.10.4.5.3.1 Definition



Definition nodes

FD-Mode:

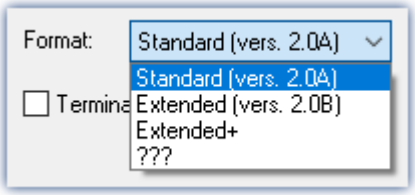
If the device is equipped with a CAN FD interface, the entry for FD appears on the *Definitions* card. Here is where the transfer protocol is defined:

Mode	Description
CAN classic	CAN High Speed in accordance with ISO 11898, CAN Low Speed in accordance with ISO 11519. The symbol for the node is shaded grey-blue.
CAN FD (ISO Standard)	in accordance with ISO 11898-1:2015 The symbol for the node is shaded red-blue.
CAN FD (non-ISO Draft)	according to an earlier draft from BOSCH. The symbol for the node is shaded red-blue.

Note

Mixed operation with both classical CAN-modules and CAN FD are covered in the ISO standard 11898-1:2015, if the CAN-controllers involved support it. Mixed operation with modules which work with "old" controllers, however, is not possible. When receiving CAN FD messages, these modules would generate error frames.

Parameter	Description
Name	A unique name for each node with up to 65 characters. The default name for a new node is "Node_" + index (e.g. 001). If several devices are being used, a _j is appended to the name, where j is the device index number. All names must be comply with imc FAMOS variable syntax, just as all other device channel names.
Comment	This text accompanies the node name and is usually intended to add a more detailed description of the node. The maximum length of this field is 255 characters.
Baud rate	The baud rate specifies how fast the bits are transferred over the serial bus. All modules in a given CAN-Bus must use the same baud rate! Select the baud rate from the drop-down list: 5kbit/s to 10Mbit/s. The default value is 125kbit/s.
Data rate (with CAN FD)	If CAN FD is set for the FD-mode, the entry for the data rate is shown. If the bus exclusively has subscribers whose controllers support CAN FD, it is possible to transfer different settings for the Baud rate and data rate. While transferring at the higher data rate, the subscribers using the slower Baud rate rate are then switched to passive.

Parameter	Description
Format	 *Format: Standard or Extended* "*Standard*", "*Extended*" or "*Extended+*". The difference between Standard and Extended is that the Standard-format ID length is 11 bits (IDs from 0 .. 2031 are permitted) and in Extended-format 29 bits (IDs from 0 .. 536870911 are permitted). Extended-format allows for greater distances between consecutive IDs. This makes it possible to extend the system without major changes to the configuration. The "*Extended+*" format is able to receive "*Standard* " and "*Extended*" identifier. For sending the message then select in the Assistant 540 either "*Device sending message*" for a 29-bit identifier or "*Device sending message in Standard-format*" for 11 bits.
Connection	This parameter determines which node is assigned to which port. Node 1 is generally assigned to CAN 1 and Node 2 to CAN 2, no other nodes are assigned to any port and must be deleted from the CAN-configuration in order for it to be consistent.
Terminator	A terminal resistor can be connected to a node via software. If the measurement device is located at the end of the CAN line, this eliminates the need for a Y-cable with an external terminator. See also "*Connecting terminators* 516".

Expert settings

CAN-protocol (classic)

This option is only needed for very rare cases. If there are disturbances on the bus, you can make adjustments for each node here.

Parameter	Description
Slope detection (soft synchronization), width of the time window	All subscribers to the bus have the same basic clock rate. However each module comes with its own quartz. Thus, phase differences can arise. The first edge of a message is used to get all subscribers in phase (hardware synchronization). Edges within a message can be used to compensate subsequently arising phase differences within a message. The width The width of the time window determines by what value a bit is shortened or lengthened in order to re-align the phase. Using this option it's possible to alter the time window in which the edges are expected.
Number of samples per bit	If the bus has achieved a steady state, the level is read by the receiver by means of sampling, and the bit is determined to be either 1 or 0. If there is disturbance on the bus, it's possible that the disturbance will be picked up in a sample. This option helps by increasing the number of samples to three. If the samples show different levels, the majority is taken as the bus level.
Sample location in bit time	Here, the position within the bit time can be altered. By default, it is set to 50%. With long leads, an impulse reaches its level only after a certain delay. With this option, the sampling time can be shifted backward in order to compensate for the delay. The setting is expressed in percent of the total time.

Note

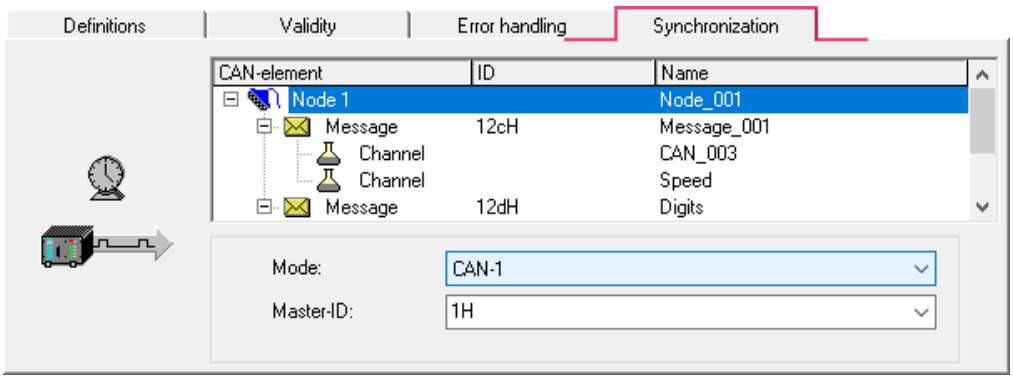
- In most cases, disturbance on the bus can be compensated by shifting the ***position of the sampling time*** backwards.
- Disturbance is usually the result of long cables. The low-pass effect of such long lines prevents voltage levels from changing rapidly. In extreme cases, this can mean that the correct level at the center of the bit time (50%) has not been reliably reached.

CAN FD

For transfer at the CAN FD data rate, deviating settings can be made. Only the number of samples per bit is omitted here.

9.10.4.5.3.2 Synchronization

For the purpose of synchronizing any CAN-modules connected, such as imc CANSAS, the measurement system is capable of sending a message in CAN-1 protocol. In such a case, your measurement system functions as the Master. If the imc CANSAS module is configured accordingly, it synchronizes itself to this message.



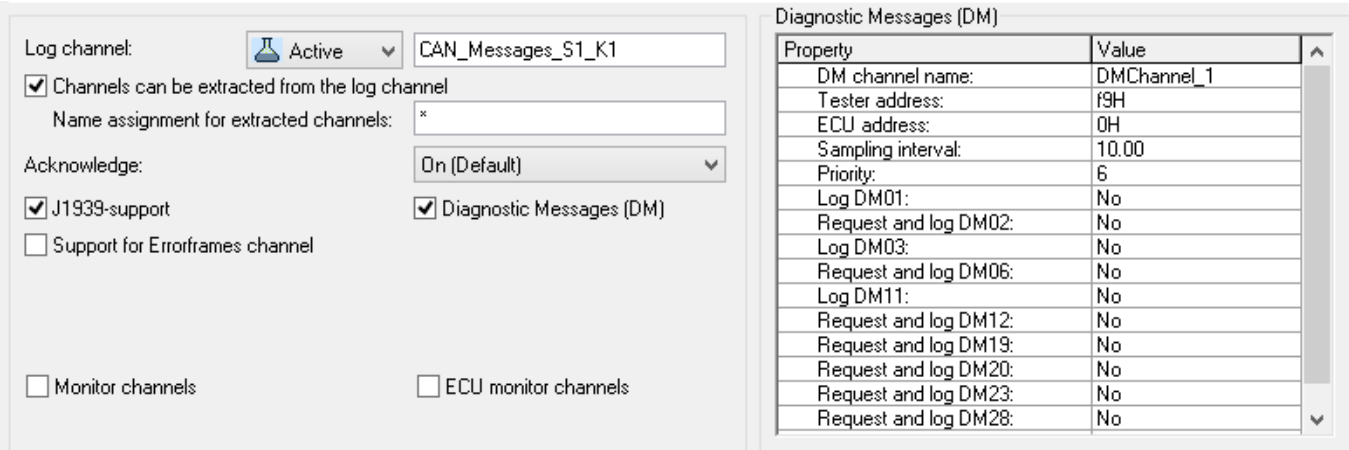
Synchronization of a node

The **CAN-1 Protocol** message is sent at a one-second clock rate by the device working as the Master. imc CANSAS modules working as Slaves receive this CAN message and synchronize themselves to it. The message used takes a format corresponding to the CAN-1 protocol. Other imc CANSAS modules, as well, can work as the CAN-1 Master. It is recommended that the device capturing the measured data works as the CAN-1 Master.

Conditions

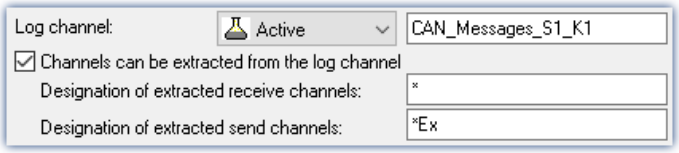
- All devices (in other words the Master and all Slaves) are connected to the same CAN-Bus line.
- No gateway or router may be connected between devices. In the CAN-1 protocol, special consideration is given to achieve very high precision.
- For imc BUSDAQ (group A4 187): The imc measurement device may not be operated in Sleep/Resume mode.

9.10.4.5.3.3 Validity



Validity of a node

Log channel



Parameter	Description
Log channel	With Log channel "*Active*", all messages belonging to the node are logged in imc STUDIO. In the input field you can enter the name of the log channel. By default, a name with slot and node number is generated. There is exactly one log channel per node. Message IDs and data bytes are logged. You can filter a selection of messages by creating receiving messages 541 with specific IDs whose "*Message log*" is set to "*Log CAN bus message*". In this case, only these messages are logged.
Extract all channels from log channel	The option allows the data stream to be unpacked as individual channels in imc STUDIO by the Bus Decoder 539 or via imc FAMOS. For reasons of data security, this option is deactivated by default. In consequence, sensitive information will not be inadvertently incorporated into the log channel's file of measured data. **Beware of the following:** Extraction of individual channel signals from the data stream by means of the Bus Decoder or via imc FAMOS is only possible if the box is check marked. In this case, the option must be explicitly activated.

A global channel for logging CAN-messages is also available in ASCII-time stamp data format. By default, imc STUDIO creates this global channel with the name CAN_Messages_S(i)_N(j), i: slot index (1..8), j: node index (1,2). If you have set messages of different slots and nodes to be logged, a corresponding number of global channels will be created.

Under "*Table*" menu item of the curve window context menu (accessed by right mouse-click), it is possible to set the text format to "*imc DEVICES CAN-message*". At the left of the row in the table, the message ID is stated, to the right of it the message data bytes. The values are displayed in hexadecimal format.

Message log channel can be saved as MDF files to the device disk. See also section "*Storage options* 1154".

Acknowledge

Disables the transmission of Acknowledge messages. The CAN controller usually sends this packages to confirm successful received messages.

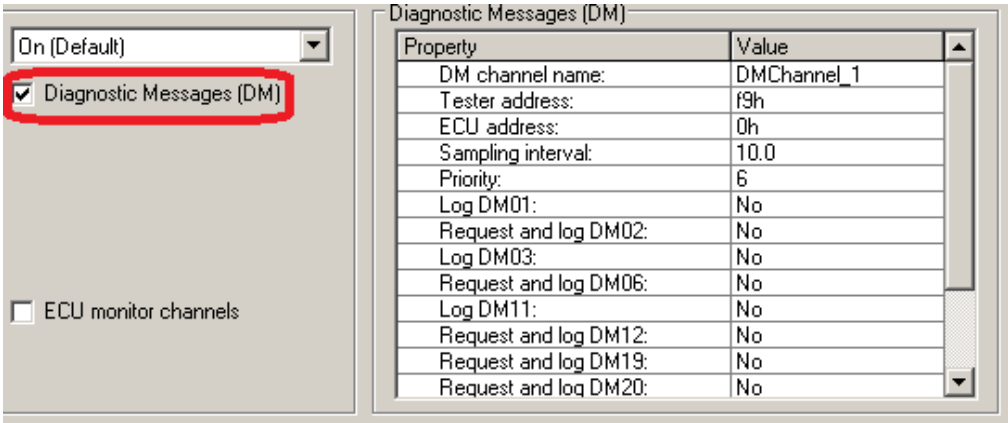
J1939-support

Activated the J1939-support, see also Messages: Validity 544

Note

This option is only available, if the identifier is set to *Extended Format 2.0B* or *Extended+.*

Also, DM-services for J1939 can be set:



Diagnostic Messages (DM)

The error codes are captured with imc STUDIO as Time Stamp ASCII channels. To display the channel in the curve window, select the item "*Table*" in the menu *Configuration\Display*. There, set the *Text Format*: *Hex*.

The error code format is structured as follows:

Byte	Description
Byte: 1: Format specifier	A log entry's first byte is specifies the format • **ach** = ASCII (VIN) • **d1h** = DM1 messages and others with the same structure (DM1, DM2, DM6, DM12, DM23, DM28) • **d0h** = record of a "Clear" command (DM3, DM11) • **d2h** = binary (DM19, DM20) • **dfh** = for errors (currently not used)
Byte: 2: for format specifiers **d1h**, **d2h** and **d0h**	DM-number, pattern: DM# with # equaling 1, 2, 6, ... • 1 = DM1 • 2 = DM2 • 6 = DM6 • ...
Byte: 3: for format specifiers **d1h**, **d2h** and **d0h**	Source
Byte: 4: for format specifier **d0h**	Target
Byte: 4 and subsequent: for format specifiers **d1h** and **d2h**	according to J1939

Example of DM1 and similar:

Byte	Bit	Description
Byte: 1	bits 8-7	Malfunction Indicator Lamp Status
	bits 6-5	Red Stop Lamp Status
	bits 4-3	Amber Warning Lamp Status
	bits 2-1	Protect Lamp Status
Byte: 2	bits 8-7	Flash Malfunction Indicator Lamp
	bits 6-5	Flash Red Stop Lamp
	bits 4-3	Flash Amber Warning Lamp
	bits 2-1	Flash Protect Lamp
Byte: 3	bits 8-1	SPN, 8 least significant bits of SPN (most significant at bit 8)
Byte: 4	bits 8-1	SPN, second byte of SPN (most significant at bit 8)
Byte: 5	bits 8-6	SPN, 3 most significant bits (most significant at bit 8)
	bits 5-1	FMI (most significant at bit 5)
Byte: 6	bit 8	SPN Conversion Method
	bits 7-1	Occurrence Count

Note

When the occurrence count is not available it should be set to all ones which is a value of 127.

Example

The following illustrates the message format for when there is more than one diagnostic trouble code.

Given:

- a=lamp status
- b=SPN
- c=FMI
- d=CM and OC

Message form will be as follows: a,b,c,d,b,c,d,b,c,d,b,c,d....etc.

With format specifier ach = ASCII

Byte	Description
Byte: 2	Higher-value byte of PGN
Byte: 3	Lower-value byte of PGN
Byte: 4	Source
Folgende Bytes	contents according to J1939 standard (ASCII string)

Abbreviations

CM	SPN Conversion Method	
DM1	Diagnostic Message 1	Active Diagnostic Trouble Codes
DM2	Diagnostic Message 2	Previously Active Diagnostic Trouble Codes
DM3	Diagnostic Message 3	Diagnostic Data Clear/Reset for Previously Active DTCs
DM6	Diagnostic Message 6	Emission Related Pending DTCs
DM11	Diagnostic Message 11	Diagnostic Data Clear/Reset for Active DTCs
DM12	Diagnostic Message 12	Emissions Related Active DTCs
DM19	Diagnostic Message 19	Calibration Information
DM20	Diagnostic Message 20	Monitor Performance Ratio
DM23	Diagnostic Message 23	Previously Active Emission Related Faults
DM28	Diagnostic Message 28	Permanent DTCs
FMI	Failure Mode Indicator	
MI	Malfunction Indicator	
MIL	Malfunction Indicator Lamp	
OC	Occurrence Count	
PG	Parameter Group	
PGN	Parameter Group Number	
PID	Parameter Identifier (SAE J1587 or SAE J1979)	
SPN	Suspect Parameter Number	

Errorframes channel

With this option the CAN controller creates a new channel to record the number of error frames within a sample interval. A name can be defined by the edit field *Channel name.*

Wake On CAN

Prerequisites

This function is only available for the new **imc BUSDAQ**, group A4 187. The devices must be configured via the Remote connection in accordance with the description in the device manual.

Description

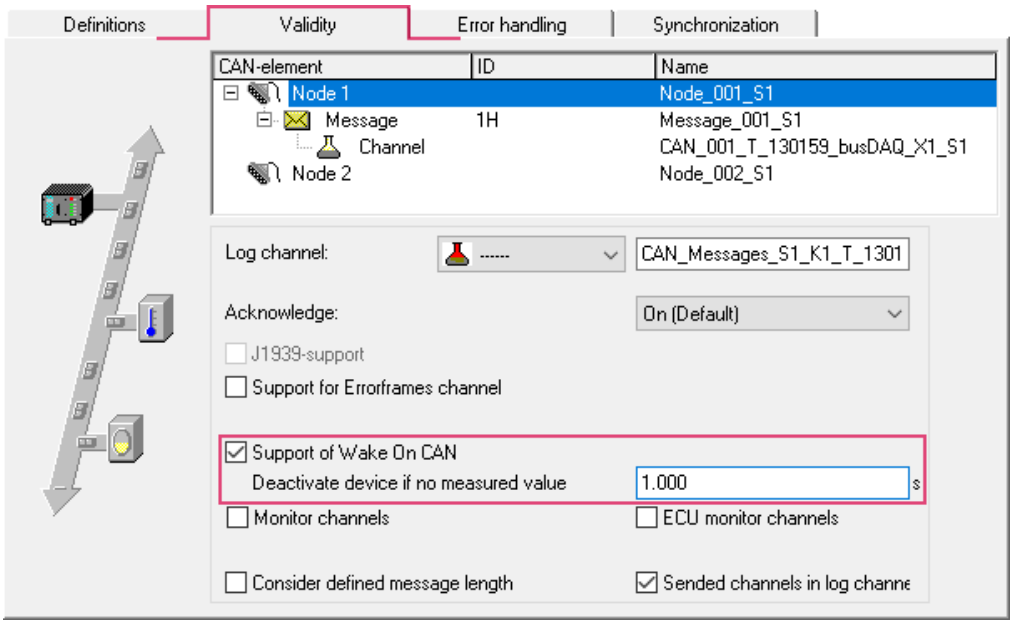
Wake On CAN enables the functionality of Sleep mode in dependence on the CAN-Bus activity. As soon as data arrive at the CAN-node, the device starts measurement. After a specifiable time without bus activity, the device returns to the Sleep mode.

Systems supply or activate the measurement device separately from the connected CAN-bus thus avoid premature startup of data capture and the resulting unnecessary power consumption.

Each node can be adjusted individually.

The prerequisite is the appropriate configuration of the Remote connection, described previously, which enables the hardware-driven Sleep mode.

- Without *Wake On CAN*, the Sleep mode is available without any additional preparation in the operating program.
- The *Wake On CAN* function is a supplemental AND condition and must be activated in CAN-Bus-Assistant.



Activation of Wake on CAN

Parameter	Description
Support of Wake On CAN:	The option *Support of Wake On CAN* appears on the page Validity if a node is selected in the CAN-element tree diagram. If it isn't shown, this indicates that your device's hardware isn't set up for this function.
Deactivate device if no measured value x s:	Enter the time interval whose elapse without the arrival of data activate the sleep mode. Intervals between 1ms and 14h (50400s) are possible.

Monitorchannels

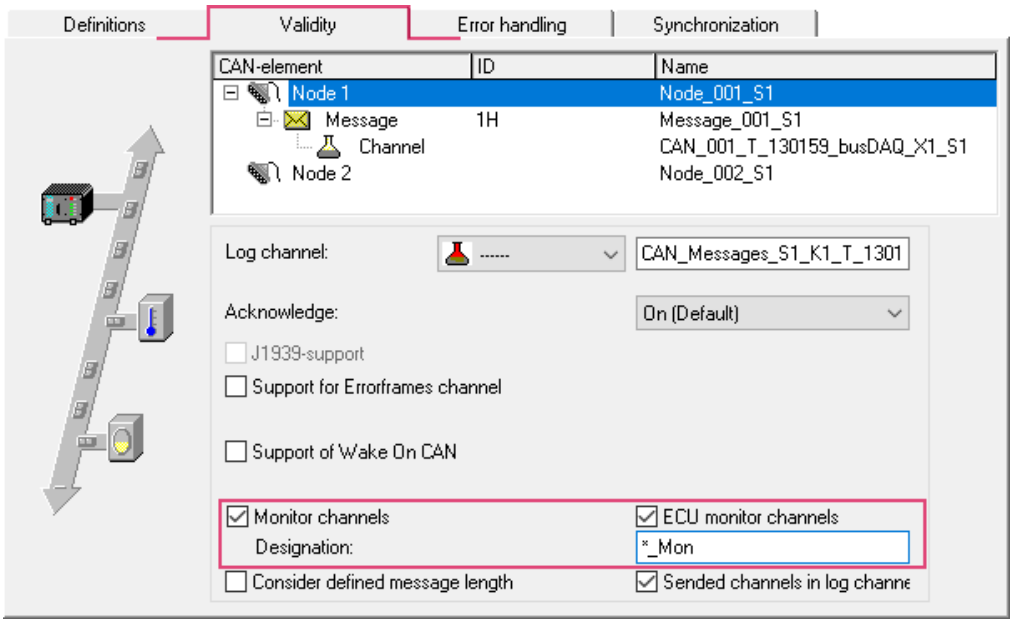
Triggered Fieldbus channels are not visible prior to the trigger event. Therefore, for analog channels there are monitor channels which can be started independently of the original channel's trigger.

Create monitor channels for all channels of a node:

It is possible to create a monitor channel for each CAN channel or ECU channel. To do this, activate the option *Monitor channels* or respectively *ECU monitor channels* on the node's *Validity* page.

Designation:

Monitor channels receive a name extension which can be freely defined in the box *Designation*.



Activate monitor channels for CAN- or ECU-channels

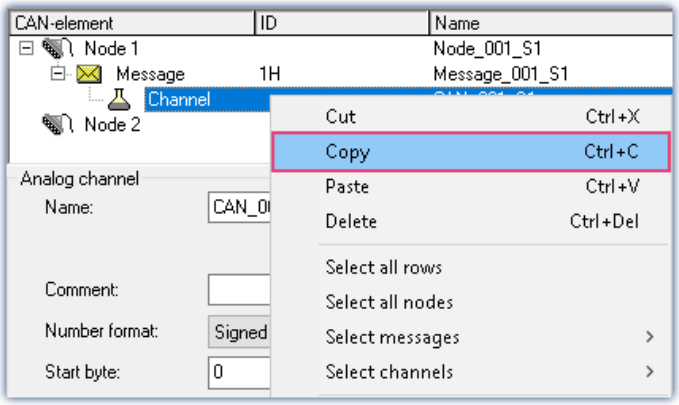
Note

Monitor channels are created for all of a node's CAN- or ECU-channels.

Create monitor channels for single channels:

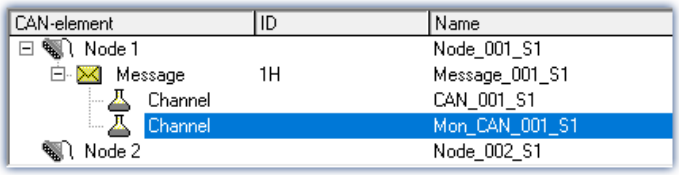
Use the Assistant to create within a channel's message an additional channel having a different name and the exact same settings. This channel can then be captured without triggering.

Copy:



Create individual fieldbus monitor channels: copy

and Paste:



Create individual fieldbus monitor channels: paste

Note

Monitor channels, created by modules that are synchronized by CAN-1, may not be sampled with a lower sampling rate.

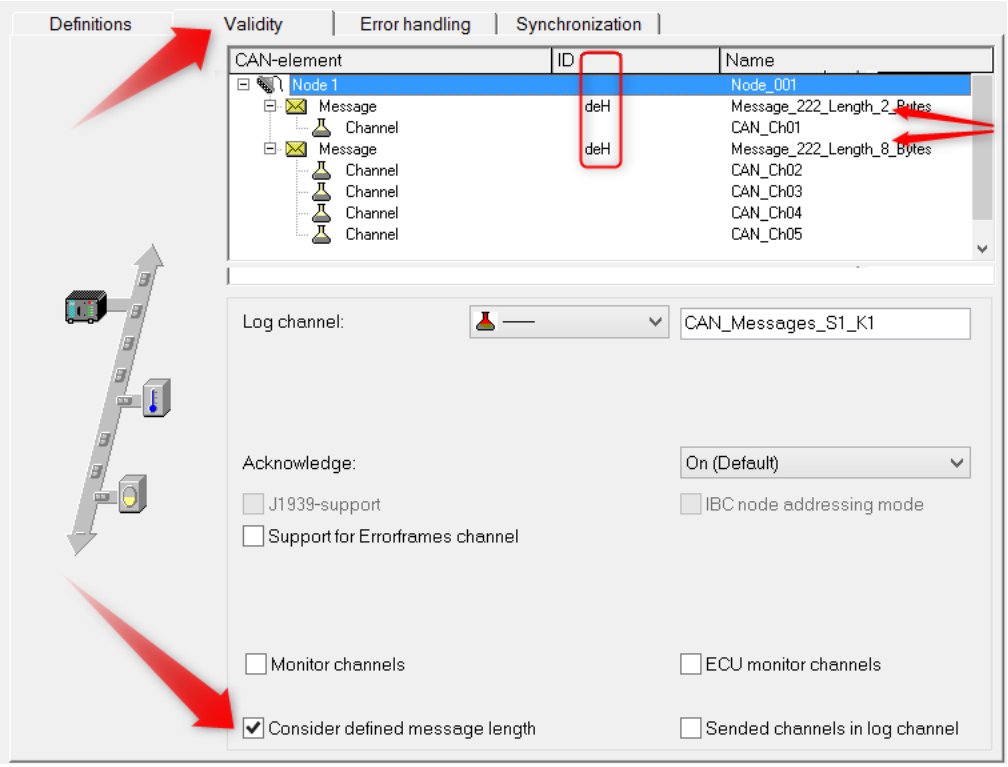
Consider defined message length

Per default, the length of a message defined in the CAN Assistant is not strictly observed and is only checked in some situations. Messages of an ID can only be created once per node, otherwise the syntax check of the CAN Assistant reports an error.

If option "*Consider defined message length*" is activated for a node, the message length transmitted on the bus is compared against the length defined in the CAN Assistant. If different, no channels are extracted from this message. However, the message protocol (log channel, CAN bus dump, log file) also records messages with differing lengths.

This option now allows a **message** with **a certain ID** to be assigned to a node **several times with different lengths**. The same ID with the same length is not allowed more than once.

Example:



Consider defined message length

Sended channels in log channel

Transmitted CAN channel data do appear in the log channel and therefore they can be extracted from the log channel and by using `OnCanMessageReceive` in imc Online FAMOS. With this option transmit channels can be extracted from the data stream.

Sended channels in log channel

9.10.4.5.3.4 Sensor initialization

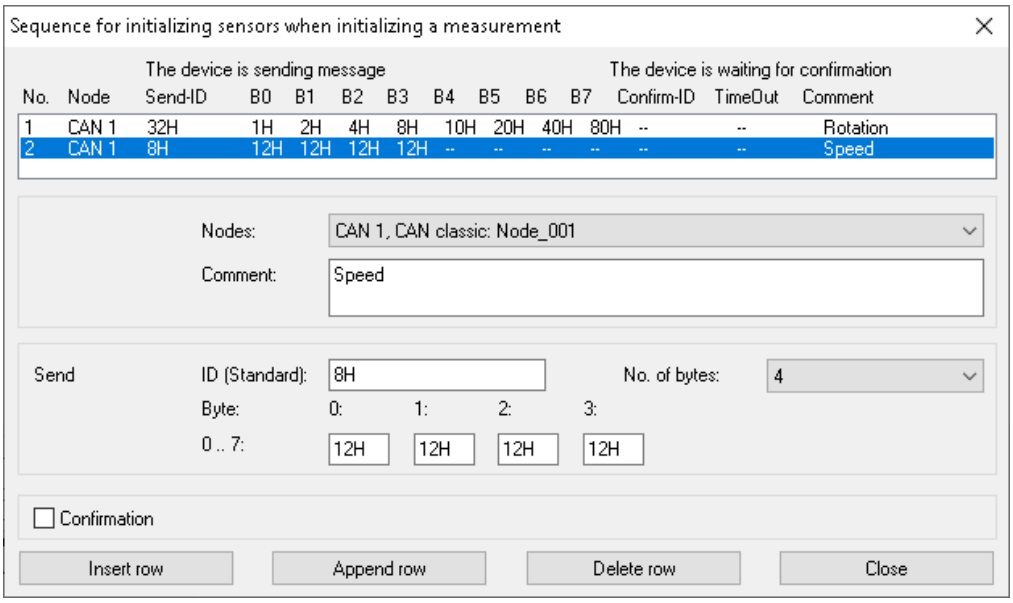
Many CAN sensors need to be initialized after being turned on. The initialization usually consists of one or more messages.

The device can send a defined set of messages to the CAN-Bus. The initialization sequence is performed only once at the very beginning of a measurement once the configuration is complete. Restarting a measurement (i.e. stopping it and starting it again) does not usually re-initialize the sensors. An exception to this is when the configuration is changed and the measurement is reinitialized.

The same applies to Autostart measurements. The initialization consists of a sequence of messages. If hand-shaking (i.e. confirmation from the sensor) is required, time-outs can be used. Many sensors perform hand-shaking by sending a return message. The unit waits for the sensor's response before sending the next message.

Initialization dialog

The initialization sequence can be defined using the *Edit > Sensor-initialization*:



Sensor initialization

For initialization, messages are sent to the sensors. It is possible to specify confirmation of receipt of the messages. The list in the dialog shows the entire initialization sequence.

Parameter	Description
Nodes	Specify which of the nodes (ports) to use.
Comment	Additional description of the initialization sequence or other information - maximum 200 characters.
ID (Send)	An identifier for the message with the initialization sequence. Allowed values are from 0 to 2031 or 0H .. 7efH for Standard-format, for Extended-format 0 .. 536870911 or 0H .. 1fffffffH. The ID is always displayed in hex format but you can enter it as a hexadecimal or decimal number. Numbers entered in Hex-format must be designated as such by using H or h (e.g. 2ah).
No. of bytes (Send)	The number of bytes to use for the message - allowed are 0 to 8 bytes.
Byte 0 to Byte 7 (Send)	The contents of the individual bytes in the initialization sequence. The allowed value range is 0 - 255 (decimal) or 0H - ffH (hexadecimal). The values are always displayed in hex format but you can enter them as a hexadecimal or decimal numbers. Numbers entered in Hex-format must be designated as such by using H or h (e.g. 2ah).
ID (Confirmation)	The identifier for the confirmation (hand-shaking) message which the sensor returns to the device to indicate that the initialization sequence has been carried out. The unit waits for this message before continuing with the next initialization sequence in the list.
Time-out (Confirmation)	The device waits for the confirmation message until the time-out limit has been reached. If the confirmation message is not received, the next initialization sequence in the list is performed. The sensor is then not initialized. The allowable values range is 1ms to 1000ms.

9.10.4.5.3.5 Blob - Decoding with the Bus Decoder

All data associated with a CAN-node can be captured completely as a log channel. This is a way around the limitation of 512 channels per device instance. Along with separation into individual channel with imc FAMOS, this can be accomplished already with imc STUDIO's Bus Decoder 1101.

To do this, you make the following settings in the CAN-Bus-Assistant:

- A log channel which logs all messages 530.
  - On the node's page "*Validity* 530", the option "*Channels can be extracted from the log channel*" must be activated.
- The CAN-channels to be extracted in the Bus Decoder from the log channel must be set to "*Has not been set up in the device, in imc STUDIO, extractable from log channel*" on the "*Validity* 551" page (light blue highlighting).
  To select all channels, use the menu item "*Edit*" > "*Select channels*" > "*Select all receiving channels*".

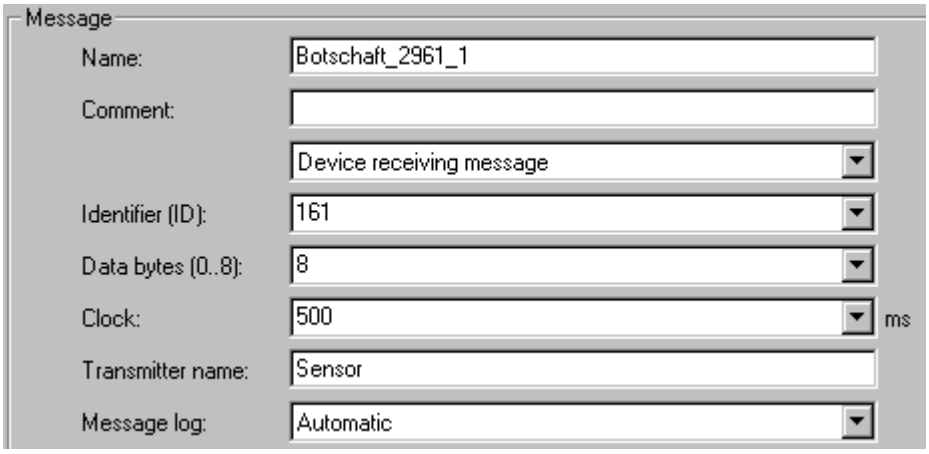
If the log channel is then extracted in the imc STUDIO, then all CAN-channels thus highlighted appear int he Bus Decoder.

If the log channel is extracted in imc FAMOS, then **all** CAN-channels appear in imc FAMOS (not only those specially highlighted).

9.10.4.5.4 Message

9.10.4.5.4.1 Definition

Receiving a message

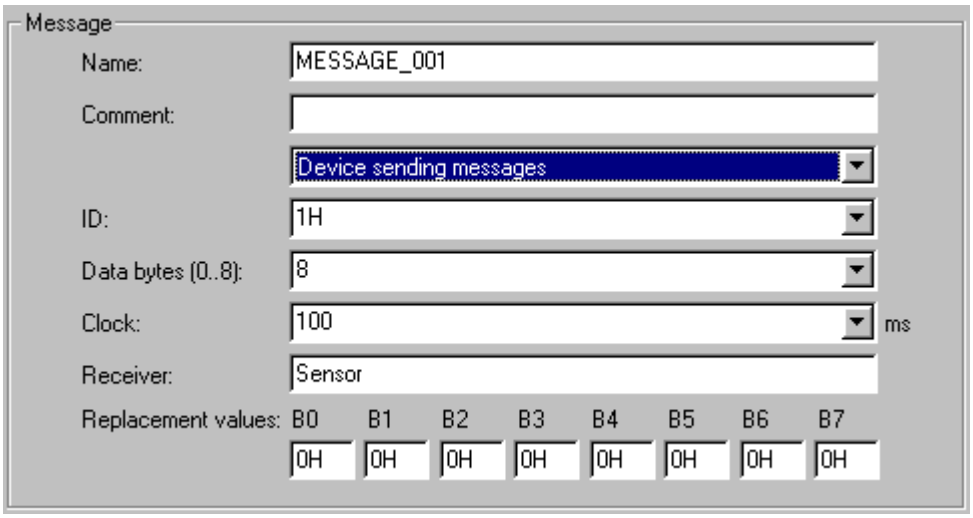


Receiving a message

Parameter	Description
Name	A unique name for each message with up to 65 characters. The default name for a new message is Message_" + index (e.g. 001). If several devices are being used, a _j is appended to the name, where j is the device index number. All names must be comply with imc FAMOS variable syntax, just as all other device channel names.
Comment	This text accompanies the message name and is usually intended for adding a more detailed description of the message. The maximum length of this field is 255 characters.
Device receiving / sending messages	A message can only be reset from sending to receiving or vice versa if there is **no channel** contained in the message. This can be converted via the menu item "*Edit*" > "*Transform message* 556". Depending on whether the *node is set with the format Extended+* 528, there can be an entry "*Device sending message in Standard-format*", which makes it possible to send messages in the **Extended+** format with an 11-bit identifier. In this case, if the setting "*Device sending message*" is selected, the extended identifier having 29 bits is used.

Parameter	Description
ID	Each message on the CAN-Bus contains an identifier (ID) which indicates the origin and meaning of the data. No two messages in a node can have the same ID. The identifier also indicates the priority of the message and, once again, no two messages can have the same priority. When two messages are to be sent at the same time, the message with the lower ID is sent first. Allowed values are: 0 to 2031 (Standard ID format) and 0 to 536870911 (Extended ID format). Identifiers can also be specified as a hexadecimals. Append an "h" or "H" to the hexadecimal, e.g. 2ACH or e4h. If set to *All messages*, information on all messages of the current node are logged. For this purpose, only channels having the format *Log channel* may be assigned to the message (see below for a description of how to make log channel settings). Alternatively, the "*Message log*" control can be set to "*Log CAN-Bus message*". Then all node messages will be logged along with their data types.
Data bytes	This is the number of bytes in the message, 0 to 8 bytes or 64 with CAN FD are allowed. It is not possible to send partial bytes in a message but it is also not necessary that every bit in the byte be occupied.
Clock	Messages are sent via the CAN-Bus at a regular rate. The value for the clock specified here is a set point, not necessarily the actual value. The clock is a property of the sensor. The allowed range is 0.001ms to 100,000ms. The sampling rate of the CAN channel to be created is determined by the clock value specified here.
Transmitter name	The transmitter name describes the transmitting sensor. This name is not used for any other purposes - therefore it need not be unique.
Message log	If you wish to **log all of a node's messages**, activate the log channel on the page with the tab "*Validity* 530". You don't have to do anything else. If you wish to **log individual messages**, again, activate the log channel in the "*Validity* 530" tab first. Then create an additional receiving message and select "*Log CAN bus message*" under *Message log*. Select the desired message using the identifier. **Note — "Identifier (ID)" as "All Messages"** The setting *"Identifier (ID)"* on "*All Messages*" originates from the previous setting method and leads to an error message with the current version. Instead, activate the log channel in the Validity tab 530 of the node and remove all protocol messages with specific IDs if necessary. **Note — Manual evaluation TSA** If the log channel is evaluated by the TSA functions in FAMOS manually, the CAN-ID is handled as user date: 4 bytes without sign in INTEL order. If the Most Significant Bit is set, it is an extended-ID, otherwise standard ID. To get the ID itself, the value must be masked with 1ffffffffH.

Sending messages



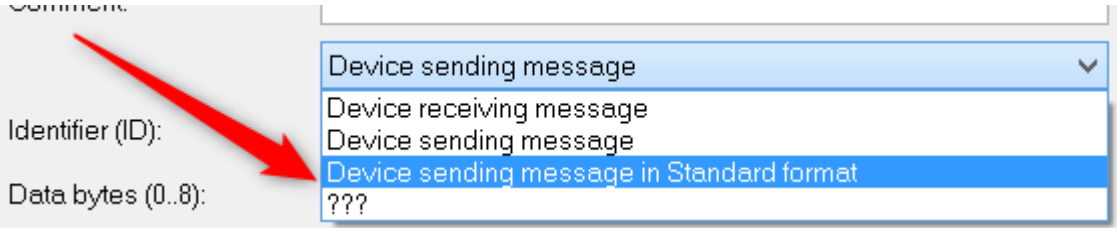
Sending a message

The CAN-Bus-Assistant is used to create a formal framework of messages and channels. The message can be set by imc Online FAMOS or with process vector variables, see below.

Parameter	Description
Name, Comment, ID, Bytes	As for messages to be received. **Data bytes for CAN FD**: Messages can be sent at the data rate set for FD 527. For this purpose, the option ***Bit Rate Switch*** must be activated, which appears at the CAN FD node. Data bytes (0..64): 20 ☑ Bit Rate Switch
Receiving / sending messages	A message can only be reset from sending to receiving or vice versa if there is **no channel** contained in the message. This can be converted via the menu item *Edit\Transform message* 556.
Clock	Messages are sent via the CAN-Bus at a regular rate. The *Clock*-value is the interval between transmission of messages. The interval must be at least 1ms. The value set here serves in imc Online FAMOS as the default setting for the clock rate for cyclic transmission in the Formula-Assistant.
Receiver	The receiver name describes the receiving sensor. This name is not used for any other purposes - therefore it need not be unique.
Replacement values	When messages are transmitted with the help of imc Online FAMOS, the replacement values serve as data bytes if there are no current data or if certain message bytes are not filled. With CAN FD, from the 9th Byte onward, additional boxes for entering replacement values appear.

Note

If a node 528 is set to ***Extended*** or ***Extended+*** **format**, a message can be send as **extended** or as **standard**-format! By default, the message is then sent in **extended** format. To send a message on an **Extend(+)** node in **standard format**, select "*Device sending message in Standard format*".



Device sending messag in Standard format.

Send channels with imc Online FAMOS

The CAN-Bus-Assistant is used to create a formal framework of messages and channels. Using imc Online FAMOS, this framework is filled with data and transmitted. The channels or single values are transmitted in the format set in the CAN-Bus-Assistant. As for the reception of CAN-messages, channels are defined for each message with their data types, Start-bits, Start-Bytes, number of bits etc. imc Online FAMOS uses these data to generate a transmission function for each message. With the help of this function, for instance, a virtual channel in imc Online FAMOS is assigned to the channel defined here in the CAN-Bus-Assistant and in sent in the specified data format.

With imc Online FAMOS, it is also possible to send larger messages in conjunction with **CAN FD**.

CAN messages can be sent **asynchronously** or **synchronously** in imc Online FAMOS.

Asynchronous transmission is performed **without control commands** by the *SendMessage_Message_name* after a transmission message has been created in the CAN assistant. If **control commands** are **activated**, transmission takes place in the *OnTriggerMeasure* of the channel to be transmitted.

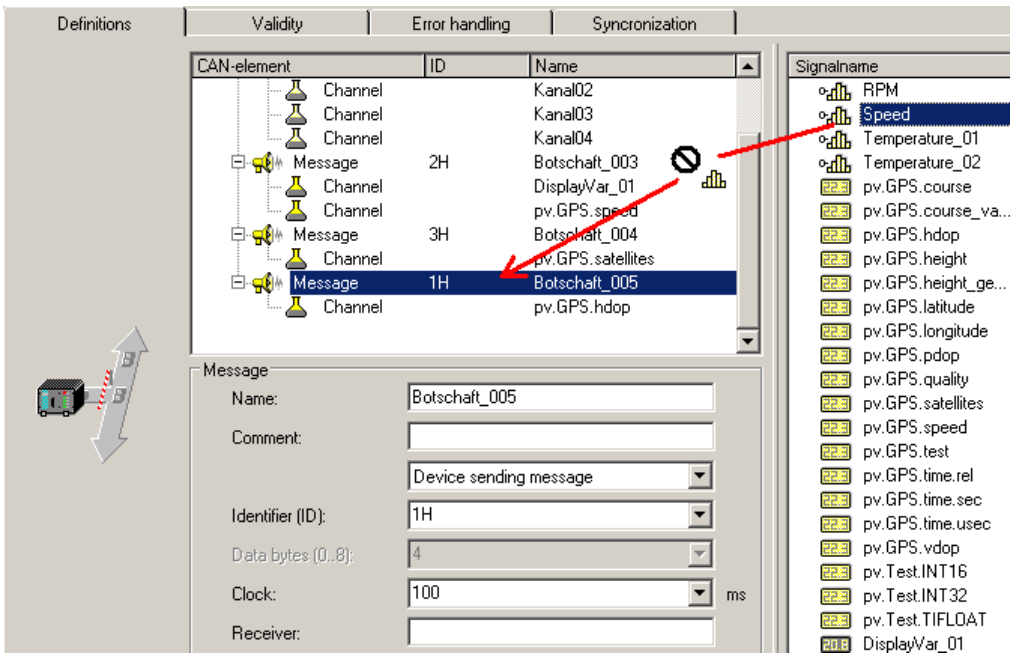
Sending is **synchronous** in a *SyncTask*. Due to the system, only **one** *SyncTask* can send CAN messages at a time. Sending in different synchronous tasks is not allowed.

For examples to send messages, see Sending CAN messages 577.

Send channels with process vector variables

Channels, process vector variables and display variables can be assigned out of the CAN-Bus-Assistant directly. If a message is set to send mode, a list with *Sending channel name comes up.* Via double click or drag&drop the channel is assigned to the send message. No further settings in imc Online FAMOS are necessary.

Prerequisite is the activation of imc Online FAMOS, even if the editor is not needed.



Sending process vector variables via CAN

Note

- Only messages can be copied. Single send channels can not be copied.
- When sending messages, if there is no recipient (yet), an error message sometimes appears, indicating that CAN-messages have been lost.

9.10.4.5.4.2 Validity

J1939

To work with J1939, the setting ***J1939-support*** is activated on the control *CAN-node* on the *Validity* page. After that the setting options on the *Validity for messages* card become visible:

J1939 settings at Message-Validity

Both the **Source-address** and the **Priority** of messages can either be accounted for or ignored. With J1939, the source address and priority are ignored by default. The source address is in the bottom 8 bits of the extended identifier and the priority in the top 3 bits.

With *Ignore (Default)*, the corresponding bits are masked out.

In J1939, the extended message identifier is comprised as follows:

$$ID_Extended = Source\text{-}Address + ProgramGroupNumber * 28 + Priority * 2^{26}$$

For standard IDs, the following applies:

$$ID_Standard = PGN\text{-}number + Priority * 2^{8}$$

When importing from a DBC-file, J1939 messages could be greater then 8 Byte. Those messages can be received from several messages.

Note

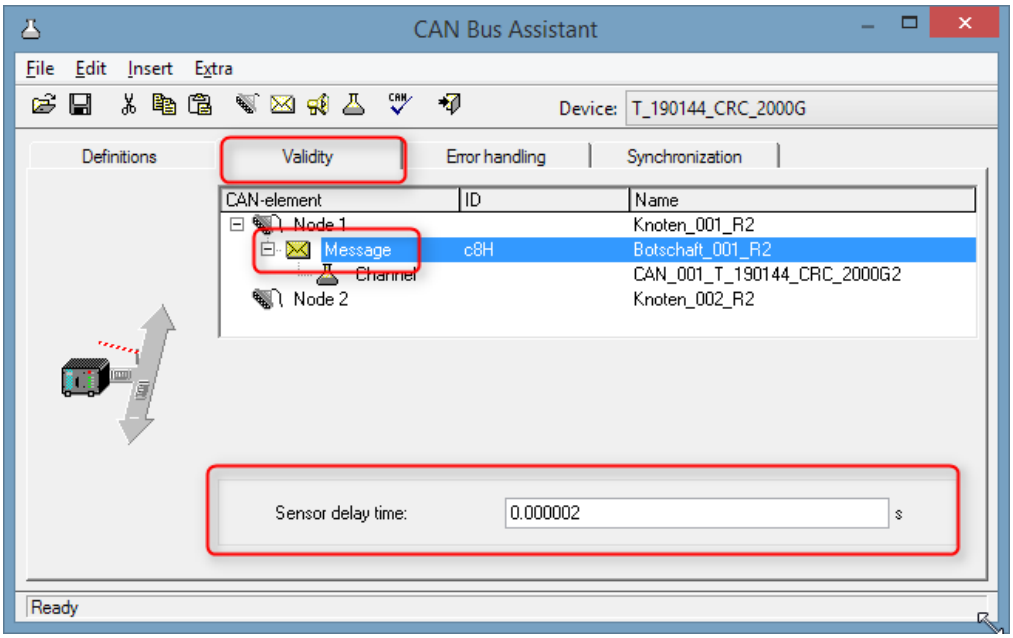
With the J1939 protocol, by default only bits 8 ... 25 (Parameter Group Number, PGN) of CAN messages are taken into account by default, bits 0 ... 7 (Source Address, SA) and bits 26 ... 28 (Priority) of CAN messages are ignored.

If the bits 8 ... 25 (Parameter Group Number, PGN) do not lead to uniqueness of the message IDs, the bits 0 ... 7 (source address, SA) or / and the bits 26 ... 28 (priority) of the CAN messages must also be taken into account.

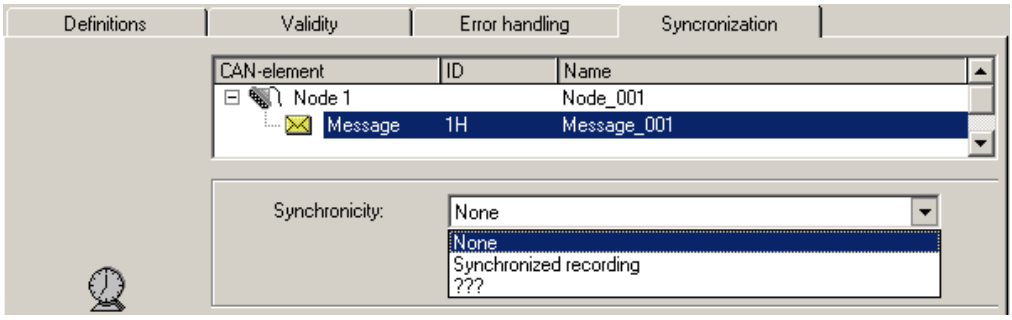
These settings must be changed in the J1939 node in the CAN messages that prevent uniqueness of the message ID (in case of doubt for all CAN messages).

Sensor delay time

Here it is possible to take into account the sensor's delayed response upon reception of CAN-messages. Differences in transit times can be compensated.



9.10.4.5.4.3 Synchronization



Synchronization a message

Synchronization is activated at the node. At this node, other devices which do not understand the CAN-1 protocol can be used alongside imc CANSAS. It should be possible to record these devices' messages, even without monitoring of the synchronization. If synchronized recording is selected for such messages, error messages will be posted whenever the messages fail to arrive at the clock rate expected.

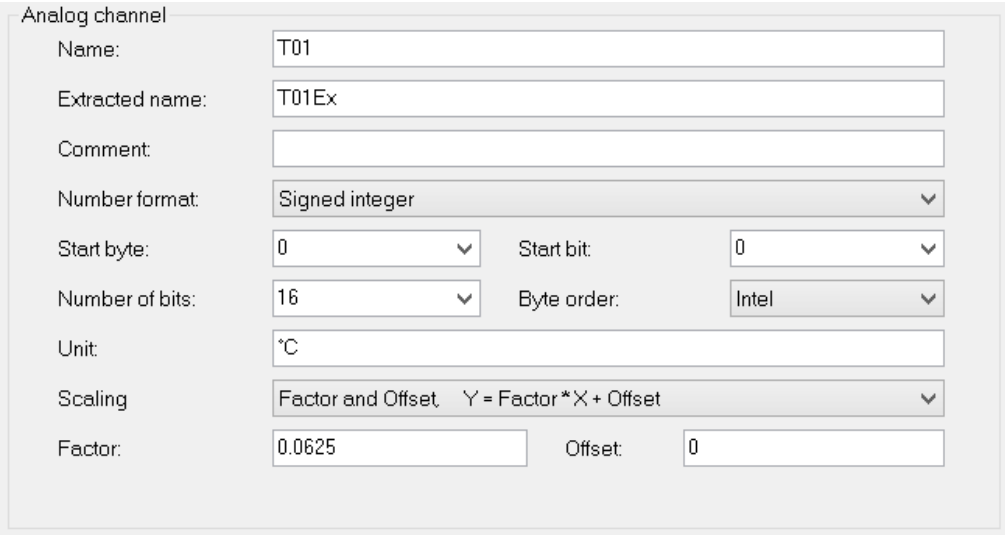
Since the integrated imc CANSAS software knows the scope of the imc CANSAS modules' functions, their messages are automatically pre-set appropriately. For example, if a imc CANSAS module is set as the CAN-1 Slave in the imc CANSAS software, its message is automatically set in the CAN-Bus-Assistant to "Synchronized recording".

9.10.4.5.5 Channel

9.10.4.5.5.1 Definition

Analog CAN channels

Analog signals are digitized by the CAN-sensors with a specific number of bits. The numerical format can be set to *signed integer, unsigned integer or real number, or digital (1 bit)*.



Definition of a analog channel

Parameter	Description
Name	See Message 540
Extracted name	Name for channel 551 for transmission channels extracted from the log channel 551. To avoid duplicate names in the send dialog 544.
Comment	See Message 540
Numerical format	Data can be sent and received as signed or unsigned integers, real numbers and in digital (1Bit) format. The option 16bits (see below) is only available for receiving mode.

Note Information about the numerical format

Some manufacturers may not indicate clearly in their CAN tables which number format they use.
For example, the table may indicate that Bytes 1 and 2 represent the guidance angle and that this value is to be multiplied with 0.0639 to arrive at the actual value in degrees. If no other information has been given, there is no definite way of determining the number format other than trial and error. Our tech support cannot help you in such cases! Sometimes however, there are implicit clues as to what the format is.

Examples:

- If the range minimum is 0 and the maximum is ffff hex, then unsigned integers are being used because they always start at zero.
- If the range is given as 7fff = -700°C and 8000 hex = +700°C, then signed integers in two's complement form are being used. The zero (0 hex) lies directly in the middle between positive and negative integers.

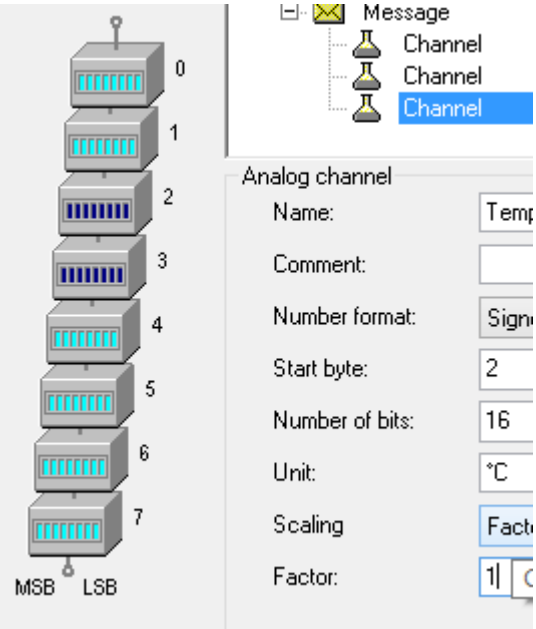
The most certain method of determining the number format is to run the sensor through its entire range.
If the correct number format has been selected, the curve that you measure will be continuous. Otherwise, the upper and lower halves are exchanged, which appears as a discontinuity. A discontinuous curve is therefore an indication of the wrong integral format being used.

Parameter	Description
Start Byte	The start byte is the location of the message byte where the channel bits actually begin. It is allowed to have overlapping channels within a message. In other words, two channels can share the same start byte in a message.
Start bit	The start bit specifies where the number starts within the start byte.
Bits	This is the number of bits used in the channel. The remaining bits continue in the direction of the increasing byte indices. For signed and unsigned integers, 1 to 32 bits are allowed. Real numbers can range from 32 (floating point) to 64 (double).

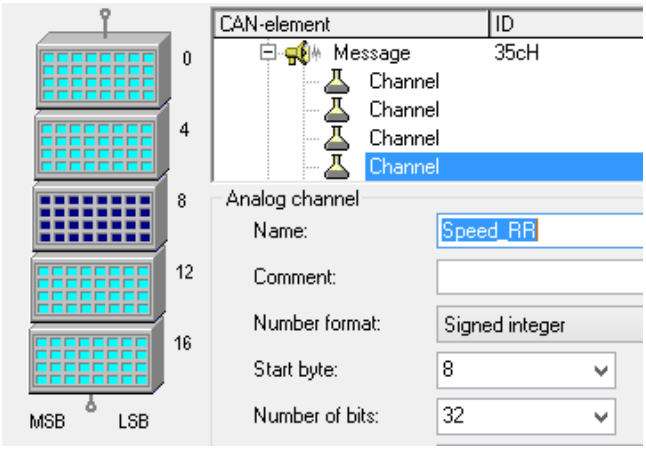
Note Information about the graphical display and the format

Graphical display of the position within the message

The Assistant displays the specified position within the message graphically. In this way, any overlapping position allocations can easily be detected:



Bit allocation in classic CAN-Bus



Bit allocation in CAN-FD

Note Byte Order | Format

This is the bit order used by the CAN module, either **Intel format** or **Motorola format**.

Intel format: The start bit is the LSB. The **LSB** has a **lower byte index** than the MSB.

Example: start byte = 5, start bit = 3, bits = 18

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 5	X	X	X	X	**LSB**			
Byte 6	X	X	X	X	X	X	X	X
Byte 7				**MSB**	X	X	X	X

Motorola format: The start bit is the MSB. The **LSB** has a **higher byte index** than the MSB.

Example: start byte = 5, start bit = 3, bits = 18

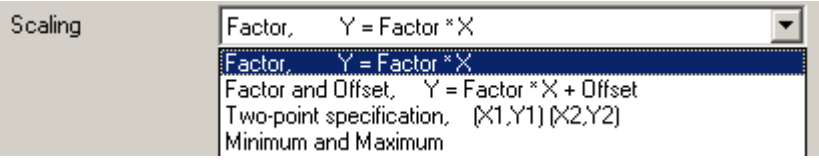
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 5					**MSB**	X	X	X
Byte 6	X	X	X	X	X	X	X	X
Byte 7	X	X	X	X	X	**LSB**		

Note on recognizing Intel or Motorola byte order

- **The CAN tables from manufacturers do not always indicate whether Intel or Motorola byte order is used.** It is sometimes possible to determine the byte order indirectly. Within one message typically only one of the variations will be used.
- *Example:* Two byte data packet, message Byte 3 is LOW and message Byte 4 is HIGH
  Intel byte order because HIGH byte contains MSB.
- *Example:* Bits 0 to 14 contain data and Bit 0 is LSB
  Data is contained in Bytes 0 and 1 but only the lowest seven bits in Byte 1 are occupied. Therefore, LSB is in the byte with the smaller index which means that Intel format is used.

Parameter	Description
Unit	The physical units of the channel, maximum 24 characters. The units are used when displaying the channel, e.g. on the Y-axis in a curve window. Note that units should be specified without prefixes. Examples of units that you can use are: m, s, kg, N, m/s. If you need to use prefixes (e.g. n-, µ-, M-) or non-SI units (e.g. inches, pounds), always place these between quotation marks (e.g. "inches").
Scaling	Here is determined how to convert the physical value to a CAN bus integer. There are different choices available, to determine the conversion.

Note Notes for scaling



The Scaling options

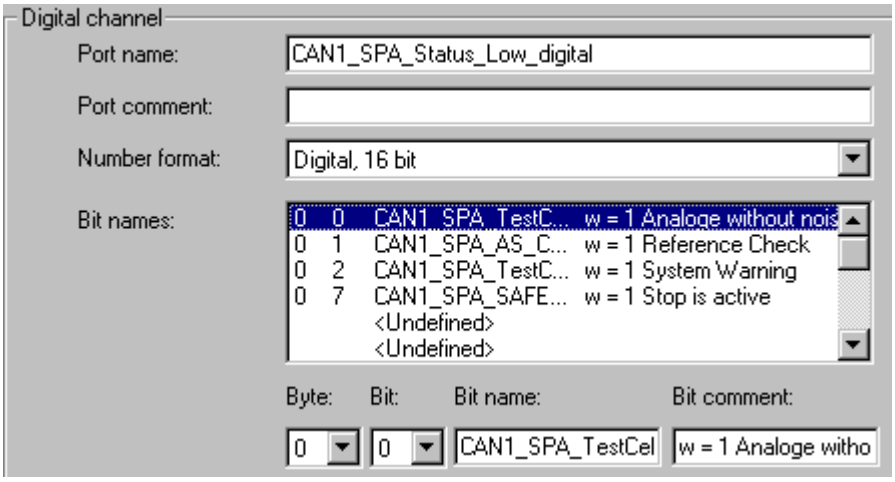
Y = physical measurement value

X = CAN bus integer value

Scaling	Description
Factor, Y=Factor*X	**Receiving**: for received signals (X) the following applies: Y = Factor * X **Sending**: for physical values (Y) to be sent, the integer value is calculated as follows: X = Y / Factor
Factor and Offset, Y = Factor * X + Offset	**Receiving**: for received signals (X) the following applies: Y = Factor * X + Offset **Sending**: for physical values (Y) to be sent, the integer value is calculated as follows: X = (Y - Offset)/Factor
Two-point specification, (X1, Y1), (X2, Y2)	X1: x-coordinate of 1st point Y1: y-coordinate of 1st point X2: x-coordinate of 2nd point Y2: y-coordinate of 2nd point
Minimum and Maximum	The lower limit of the specified number format with the specified number of bits corresponds to this minimum (e.g. signed integers, 8 bit, range: 0 to 255, minimum corresponds to 0, maximum corresponds to 255)

Digital CAN channels

The number format is set to Digital, 16 bit (only to be set for reception of CAN-Bus messages).



Definition of a digital channel

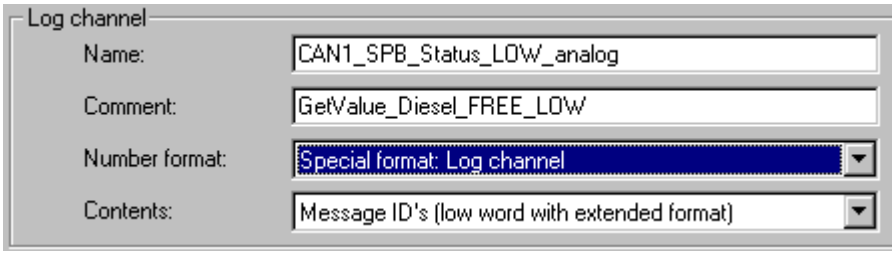
A digital CAN channel is analogous to a digital 16-bit port in imc STUDIO - it consists of 16 separate bits or one **word**. Each bit is of course a binary value, i.e. either 0 or 1. Each bit in the word does not actually have to be used. The entire 16 bit port receives a name as do each of the bits therein. These names are employed when using them in other components of the software, e.g. in triggers and imc Online FAMOS.

Parameter	Description
Name, Comment	see Message 540
Numerical format	must be set to 16 bit
Byte	determines in which of the message's bytes the digital bit is located.
Bit	position of the bit within the selected byte
Bit name, Bit comment	see Message 540

Log-channel

The numerical format is set to *Special format: Log-channel.* This special log channel allows filtering by low or high word of the message IDs or the number of data bytes. A log channel that contains the complete data transfer is set in Node/Validity 529.

This log channel is only available for messages in **receive mode**. At the same time, the message affected must be set to *All messages*.



Definition of a log-channel

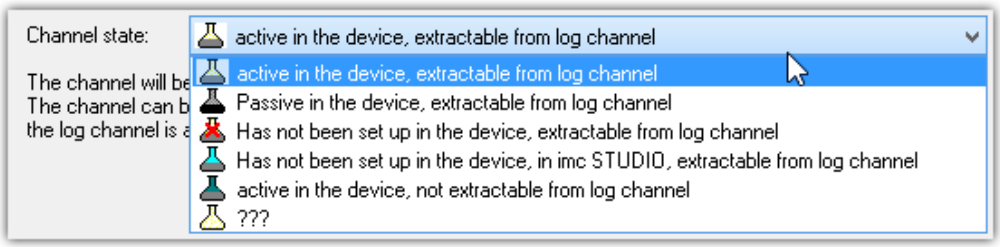
Information about all of the node's messages is logged. The message-IDs or the number of data bytes can be logged. In default format (11-bit message IDs), the message ID's are logged directly. In Extended-format (29-bit message-IDs), the High and Low word of the message ID's can only be logged separately. Each ID must then be constructed by combining the Low word and High word.

Contents: Message-IDs (low word for Extended-format); High word of message-IDs (only for Extended-format) or Number of data Bytes.

9.10.4.5.5.2 Validity

The dialog illustrated below appears if the system is set to receive CAN-Bus messages.

Channel-State



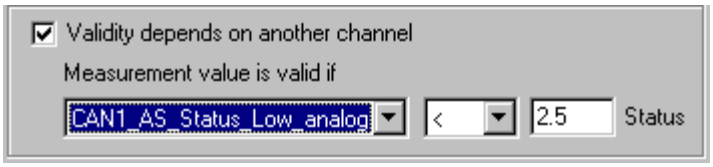
		log channel 541 extractable with		
device setup		**e.g. imc FAMOS**	**imc STUDIO**	**Comment**
	active	yes	no	Normal data acquisition on a CAN-channel. Default: active
	passive	yes	no	Normal data acquisition on a CAN-channel. Default: passive
	no	yes	no	Channel extracted offline, with imc FAMOS for instance.
	no	yes	yes	Channel extracted with imc STUDIO's Bus Decoder (see also "Blob - Decoding with the Bus Decoder 539") or offline, with imc FAMOS for instance.
	active	no	no	Normal data acquisition on a CAN-channel. No data transfer to the log channel.

If a channel is present in the *Device channel list*, it can be either activated or deactivated for recording. I.e., the channel appears as a Fieldbus channel in the Setup.

Extensive CAN-configurations (e.g. *.DBC) may exceed the maximum channel count which the device can administer. In such a case, it is possible to exclude certain channels from the device using the channel status "*Has not been set up in the device*". Then these ones don't appear in the Setup. The data belonging to "not activated" channels can instead be extracted subsequent to the measurement. For this purpose, a ***log channel*** must be activated for the associated node; see here 541.

With imc STUDIO, you also have the option to extract channels from the log channel during a running measurement. This is accomplished by the Bus Decoder 1101. The channels to be extracted must be registered for the Bus Decoder with the channel status "*Has not been set up in the device, ...*".

Validity depends on another channel



Depending on another channel

If the current channel is to depend on another channel in the same message, check the option *Validity depends on another channel*.

The channel on which the current one depends is referred to as a *mode channel*. A mode channel must be contained in the same CAN message and use the unsigned integer number format with less than 16 bits. The drop-down list to the left contains all of the valid mode channels.

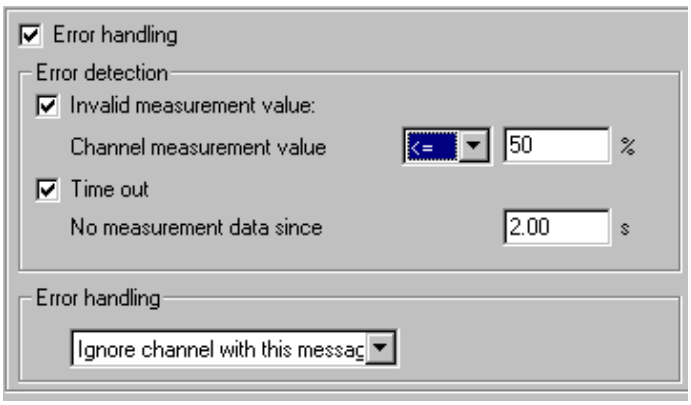
The following two conditions must also be fulfilled by a mode channel:

- A channel cannot depend on itself.
- a mode channel cannot depend on another mode channel.

The value of the current channel is only valid when the mode channel condition (e.g. CAN_001 ≤ 5) has been satisfied. Otherwise, your measurement device behaves as if the message has not been received. Enter the comparison value in real number format using the "analog" range of the mode channel - do not enter an unsigned integer value!

9.10.4.5.5.3 Error handling dialog

The dialog illustrated below appears if the system is set to *receive* CAN-Bus messages.



Error handling

This dialog allows you to define error handling for invalid measurement values and time-out errors. If an invalid measurement value appears, or if after a certain amount of time has elapsed no further message has arrived in the device, you can perform error handling. To enable it, check the Error handling box and the desired options Invalid measurement value and/or Time out.

Invalid measurement value:

Enter the comparison value for analog channels in real number format using the display range. Select a comparison operator from the drop-down list.

Enter the comparison value for digital CAN channels (ports) using the decimal equivalent of the binary value, i.e. no bits = 0, add $2^{bitindex}$ for each bit which is HIGH. For example if the first and third bits (Bits 0 and 2) are HIGH and all others are LOW, the decimal equivalent is $0 + 2^0 + 2^2 = 5$. If the comparison condition is not met, error handling is performed (see below).

Time out:

Define a time out limit here. This is the maximum time which the sensor has to send the message. Many sensors send messages at regular intervals - this option is often useful for monitoring them.

Error handling:

The error handling box is displayed whenever any of the two above options is checked. Select one the following error handling options from the drop-down list:

- Ignore channel with this message
- Last channel value
- Replacement value

The error handling functions slightly differently, depending whether the channel is continuously sampled or whether time-stamping.

Error correction	Effect with continuous sampling	Effect with time stamping
Ignore channel with this message	Same as *Last channel value*	System behaves as if the message was never received - no time-stamped sample is created.
Last channel value	The last valid channel value is generated each sampling interval until a valid sample arrives.	*Time out error:* a new sample is generated using the last valid channel value and is stamped with the time-out limit. *Invalid value:* a sample is generated using the last valid channel value and is stamped with the message time stamp.
Replacement value	The replacement value is generated each sampling interval until a valid sample arrives.	*Time out error:* a new sample is generated using the replacement value and is stamped with the time-out limit. *Invalid value:* a sample is generated using the replacement value and is stamped with the message time stamp.
No error correction used	Same as *Last channel value*	Same as *Ignore channel with this message*

Note

- **Input range of substitute values for error handling**
  Certain substitute values may lie outside of the input range for the CAN channel.
- Channels with 16 bits or less can be given substitute values having a 16-bit input range. Example: For a CAN channel with 8 bits, a substitute value of up to 16 bits is permitted.
- Channels with more than 16 bits and less than or equal 32 bits can be given substitute values having a 32-bit input range. Example: For a CAN channel with 24 bits, a substitute value of up to 32 bits is permitted.
- Consequently, the following exceptions exist: Channels with exactly 16 bits or 32 bit-channels only permit substitute values having the same input range.

9.10.4.5.6 Menu

File

Menu item		Description
	Restart (Ctrl + N)	Clears the node-, message- and channel lists. A single node is created with default settings.
	Load (Ctrl + O)	Loads a complete CAN-configuration. A dialog is called which enables you to select a CBA-file from a folder of CBA-files. For the selected device (or the selected slot if there are several in the device), the current CAN-configuration is completely replaced by the one from the file. The current configuration is first deleted and then the new one is loaded from the CBA-file.
	Save (Ctrl + S)	Saves the current CAN-Bus configuration as a *.CBA file. If the configuration has not yet been saved, the Save As... dialog is started.
	Save as	Saves a CAN-configuration under a new filename.
	Import	Loads can configuration a selected node. Beside the CAN assistant format CBA, the DBC is supported, if the PC has the DBC option installed or the device is prepared for ECU (A2L).
	Export	With Export, the sendable messages of one node can be saved as DBC file.
	Quit	This quits the CAN-Bus Assistant. The CAN-Bus Assistant automatically checks all of the settings first before closing and informs of any errors. You are then given the option of quitting or not. If you decide not to, The CAN-Bus Assistant shows you the error or errors so that you can fix it/ them easily. Note: If you quit the CAN-Bus Assistant without fixing the errors, no CAN channels are created.

Edit

Menu item		Description
	Cut (Ctrl + X)	Cuts the selected lines to the Windows Clipboard. Cutting a message also copies all of the channels contained in the message; cutting a node copies all of the messages and all of the channels therein. Once copied to the Clipboard, the lines are deleted from the list.
	Copy (Ctrl + C)	Copies the selected lines to the Windows Clipboard. In contrast to the Cut function, only those lines are copied to the Clipboard which have been selected.
	Paste (Ctrl + V)	Pastes lines from the Clipboard into the current configuration.
	Delete (Ctrl + Del)	Deletes the selected lines from the configuration. Note: Deleting a message also deletes all of the channels contained in the message; deleting a node copies all of the messages and all of the channels therein.
	Find	Using the *Find* dialog, you can search by names of channels, messages, nodes, etc.
	Select All Lines	This selects all nodes, messages and channels in the current configuration. This makes it easy to cut or copy the entire list.
	Select All Nodes	This selects all nodes in the current configuration. This makes it easy to adjust settings in all of nodes simultaneously or to see which settings are identical in all nodes and which differ.
	Select All Messages	This selects all messages in the current configuration.
	Select All Channels	This selects all channels in the current configuration.
	Show Messages	In this display mode, only the nodes and messages are visible.
	Show Channels	In this display mode, all of the nodes, messages and channels are visible, i.e. the entire configuration.

Menu item	Description
Sort	Messages can be ordered by the following criteria: • messages by name • messages by IDs • channels by name • channels by start bits
Transform messages	A receive message is converted into a send message. This means that a configuration with reception messages can be used to simply create a sending station. "Empty" transmission messages can also be transformed into receiving messages. This conversion may generate (automatically) unique message names. Therefore, the same CAN Assistant content does not necessarily result from the conversion of receiving to sending messages and subsequent conversion of sending messages into receiving messages.
Sensor Initialization	See detailed description 538.
Apply	This applies the current settings immediately and checks that they're correct. If a setting is incorrect, a message in the status bar indicates this. Most settings in the CAN-Bus Assistant are applied automatically or when you exit the dialog, so this menu option should be rarely needed.
Complete Check	We recommend that you check the configuration often. If any errors are found, they are indicated in the status bar and the incorrect field or fields are highlighted. This makes it easy to fix any incorrect or implausible settings.

Insert

Menu item	Description
Nodes	Inserts a new node into the configuration list. A node is only inserted when exactly one line is selected. If a node is selected, the new node is inserted above it. If a message or channel is selected, the new node is inserted above the next node which follows, or to the bottom of the list if no other nodes exist. The first node is always assigned to CAN Port 1 and the second node to CAN Port 2. None of the other nodes can be assigned to any of the ports. If you create any more than two nodes, they can only be used temporarily and should be deleted before quitting the CAN-Bus Assistant. Otherwise the configuration will be "incorrect" and no CAN channels will be created.
Message	Inserts a new message into the configuration list. A message is only inserted when exactly one line is selected. If a node is selected, the new message is created below any existing ones. If a message is selected, the new message is inserted above it. If a channel is selected, the new message is inserted below the message to which the channel belongs. By default, the message created is always a receiving message. See also Receiving a message 540.
Sendable message	Inserts a new message to send channels. See also Sending messages 542.
Channel	Inserts a new channel into the configuration list. A channel is only inserted when exactly one line is selected. If a node is selected, a message is created with the new channel. The new message is created below any existing ones. If a message is selected, the new channel is inserted below any other existing channels in the message. If a channel is selected, the new channel is inserted below it. By default, the channel created is always an analog channel.

Menu item	Description
ECU	Shows the control unit properties 584 and the values to acquire. That entry is only enabled for devices prepared for the support of ECU.

Extras

Menu item	Description
Options	See "*Options* 560"

9.10.4.5.6.1 Import

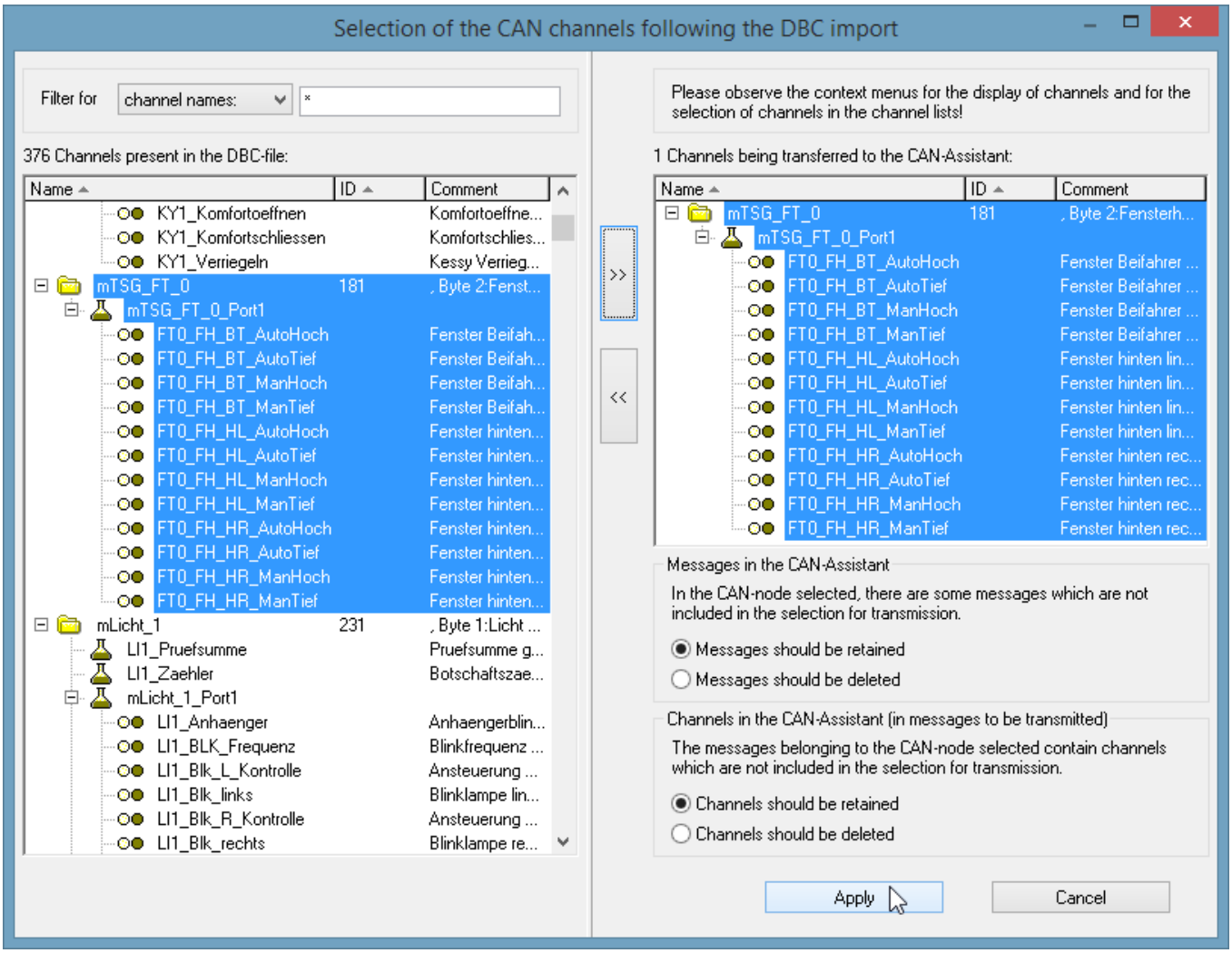
Note

At this time, direct import of more than 1000 channels is not possible. In order that channels from such an import file can still be used, any channels not needed must be blocked for the system. Load the import file, go to the page Channels-Validity 551 and assign to the unneeded channels the status "Do not adopt in system" –.

DBC

The import of CAN-Database (*.DBC) requires the purchased and installed expansion module "Import CAN-Database (*.DBC)". DBC-files can be created using the program CANdb (© Vector Informatik GmbH) from Vector Informatik GmbH. CANdb is a program for administrating the CANalyzer's (© Vector Informatik GmbH) CAN system data base. DBC-files which were created using Versions 3.03 to 3.20 of the CANdb program are supported. These versions of CANdb belong to Versions 2.0a to 3.0 of the CANalyzer.

DBC-files mostly contain all channels sent via a single CAN-node. Upon importing such channels, you have the option to filter for targeted channels relevant to your experiment.



DBC-import filter

The J1939 544 standard will be adhered to with DBC files.

CBA

As with loading a **CBA**-file, the current CAN-configuration for the selected device (or the selected slot if there are several in the device) is completely replaced by the data from the CAN-Database, i.e. the current CAN-configuration is deleted and then the CAN-Database (DBC-file) is imported. In the CAN-Bus-Assistant, a node is added as a default because the CAN-Database does not contain imc STUDIO node data such as the Baud rate. If the current configuration already includes a node, the node remains and retains its properties even after the DBC-file is loaded. But the node settings should be re-checked.

ECU (A2L)

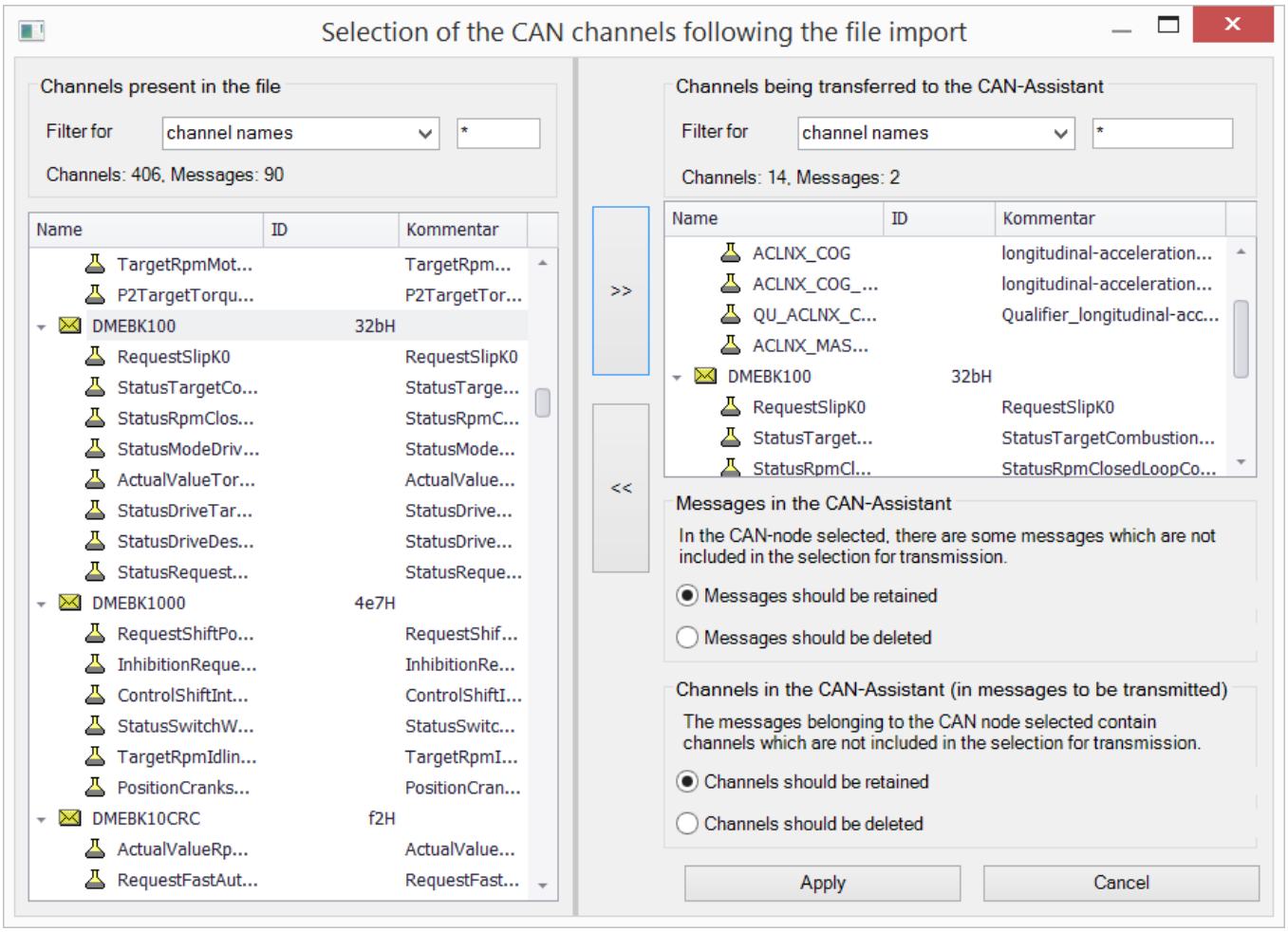
For users of the *Import CAN-Database (*.DBC)* and ***ECU (A2L)***, the same applies as for loading a complete configuration. Additionally when importing ECUs a single ECU can be selected (not only a node). Then that ECU will be replaced. In case the name of the ECU and the one in the file is different, a message box appears to confirm the replacement. If the A2L file contains short and long event names, the long ones are used.

The extended format XCPplus is supported.

Further settings for too long channel names and characteristic curves of type CURVE and VAL_BLK are available in menu Extra\Options 561.

ARXML

The ARXML format was defined by the development partnership **AUTOSAR** (***AUT***omotive ***O***pen ***S***ystem ***AR***chitecture). Once the file has been opened, it is possible to import individual messages/channels in the selection assistant.



CAN FD

CAN FD configurations can be exported to CAN-devices. The configuration is then converted to CAN and any resulting errors must be subsequently eliminated by the user.

9.10.4.5.6.2 Options

For changed message clock rate:

If the option "*Apply as sampling interval for respective channels?*" is active, the sampling rate of the Field bus channels in the imc STUDIO software is automatically set.

Example: A CAN database is imported. Further settings for the Fieldbus channels are made in the CAN-Bus-Assistant. The sampling rates on the Base-page are at first set to match the associated channel's message rate. However, the user can make settings on the Base page which deviate from that clock rate. If you then return to the CAN-Bus-Assistant and re-load the database, or if you change the message clock rate directly, the sampling rate in the Base page is set equal to the message rate, if the option "Apply as sampling interval for respective channels?" is active. If it isn't active, the sampling rates previously set in the Base page are retained.

CAN-database (*.DBC)import:

Treatment of duplicate channel names

Treatment of duplicate channel names are not allowed in imc STUDIO. In order to load a CAN-configuration with a duplicate name anyway, there are these possibilities:

Configuration	Description
Retain	The names are adopted. But the configuration cannot be applied. The Assistant highlights the duplicate names, which must then be edited manually by the user.
Don't accept	Duplicate channels are marked as defective and ignored by the administration.
Append message ID / Append message name	The message ID or message name is appended to the duplicate names, which are thus rendered unique.
Don't accept in CAN-Assistant	The duplicate channels are not imported to the CAN-Bus-Assistant and are thus no longer displayed. The channel with the lowest ID, in other words, with the highest priority, is adopted. Alternatively, these channels can be set to "*Select channels (not in device extractable)* 539", and then selected and deleted.

Channels with invalid properties

Configuration	Description
Don't accept	With "*Don't accept*", channels with incorrect properties (e.g. invalid bit count, invalid start bit, invalid characters in the channel name,...) are not adopted in the administration. The check then doesn't acknowledge any error. But the channels are correspondingly marked in the CAN-Bus-Assistant display. The channels can be re-activated if required.
Don't accept in CAN-Assistant	The invalid channels are not imported to the CAN-Bus-Assistant and are thus no longer displayed.

Calls dialog for the name extension

If this is selected, when importing a DBC-file, the dialog for the name extension is displayed after the file is selected. This dialog is for the purpose of being able to import the same DBC-files on different nodes (slots) of a device and still have unique names.

Set J1939-Protocol, if byte number of a message > 8 and extended-format

With CAN FD, the protocol can be set to J1939 in Extended Format automatically.

Import the specified devices as CAN-Senders, filter for device name

Messages of marked device names for send devices will be read as "*Send messages*".

When importing A2L files:

- Replace
  - characteristic curves of the type CURVE and VAL_BLK with type VALUE. Further information see here 602.
  - Measurement vectors with ARRAY_SIZE > 0 and matrices as single vector elements (measurement channels).
- Option for saving original names of the imported channels in the channel comments.

9.10.4.5.7 imc STUDIO - Configuration / processing of fieldbus channels

Analog inputs

The channels created using the CAN-Bus-Assistant appear in the channel list under **Analog channels > Fieldbus: Analog inputs** for further processing.

Digital I/O's

The digital channels created using the CAN-Bus-Assistant appear in the channel list under **Digital channels > Fieldbus: Digital inputs / outputs**.

9.10.4.5.7.1 Settings

If fieldbus channels are selected in the channel list (assuming the device currently allows this), then additional setting options are available.

Parameter	Description		
	long name	*short name*	*column ID*
X-axis	*X-axis*		*eXFormatVariable*
	• *Sampling rate*: The setting Sampling time is related to the settings **Duration** or **Sample count** and **Sampling rate/time** as previously discussed. • *Time stamp*: Data acquisition is handled by external, asynchronous sampling within the fieldbus. The corresponding time coordinate for the data point is passed as a time stamp along with the data for subsequent processing. In this case, the parameters **Sample count** and **Sampling rate/time** are irrelevant.		
Duration	*Duration*		*eDuration*
	If the parameter X-axis is set to *Time stamp*, the duration of the measurement depends only on the setting in **Duration**.		
Assignment	*Assignment*		*eAllocation*
	The data can additionally be assigned to a Display variable (only for analog Fieldbus inputs).		

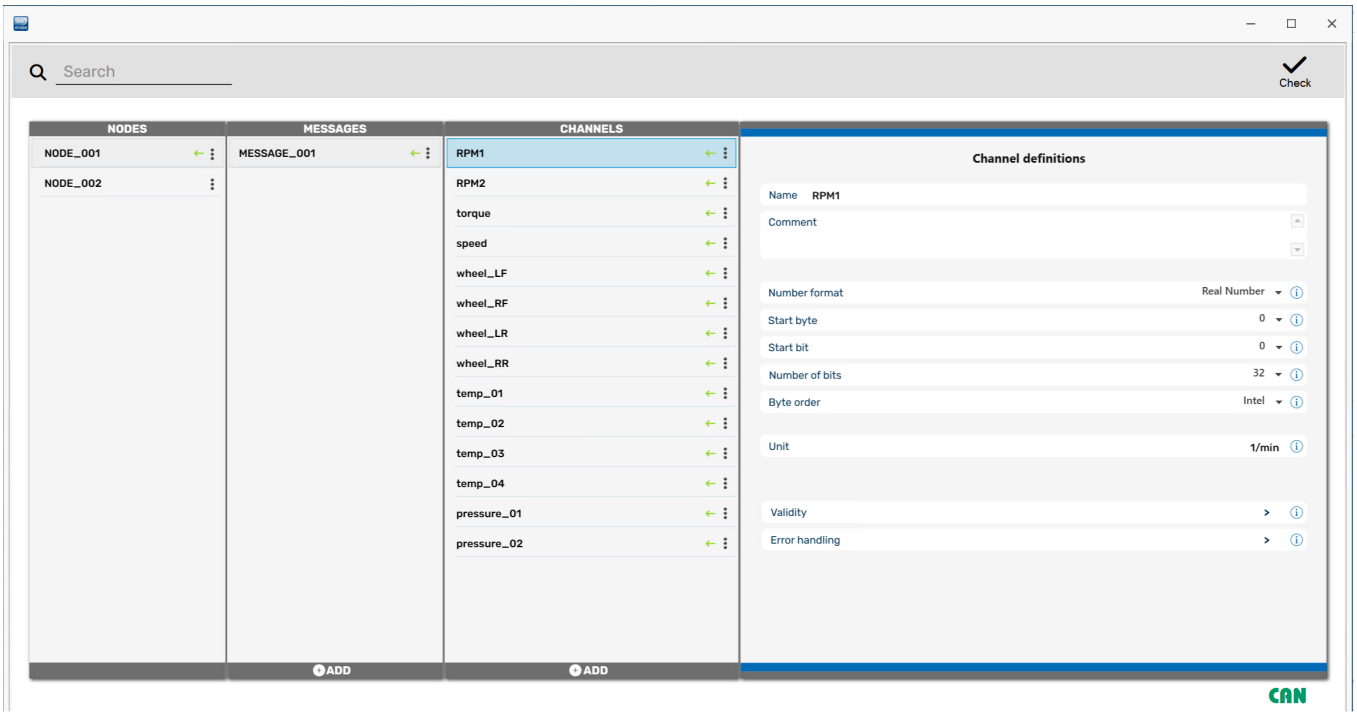
9.10.4.5.7.2 Global channels for logging CAN-Bus messages

These channels are for collecting the data from corresponding sensors which are connected to a CAN-Bus. The data are captured via a special CAN-Bus interface. The necessary definitions can be made using the CAN-Bus Assistant. The IDs and data bytes of CAN-messages are logged in these channels. The channels are expressed in ASCII time stamp format. The measurement duration can be adjusted.

9.10.4.6 CAN-Bus-Assistant (firmware group B)

For the purpose of configuring nodes, messages and channels belonging to the device of the firmware group B 187 (ARGUS), the "*CAN-Bus-Assistant*" is used. The assistant for firmware group A 187 (CRONOS, BUSDAQ, SPARTAN, etc.) is located here 525.

Menu ribbon	View
Start > CAN (CAN)	all
Setup-configuration > CAN (CAN)	Complete



CAN-Editor

Together, the CAN nodes, messages and channels form a CAN configuration. This configuration can be exported as a JSON-file. JSON-files are saved in the ASCII-format. Thus the content of these files can be viewed using a text editor.

In the *CAN-Bus-Assistant*, there is no *OK*-button. Any entries are adopted immediately. If you make any invalid entries, a notification will appear in the status bar at the bottom in red script upon making the change. In order to detect any errors, it is recommended that you perform a configuration check from time to time. Check

Note

It is possible, however, to exit the Assistant without correcting the changes - however, no CAN channels will be created!

9.10.4.6.1 Editing a CAN configuration

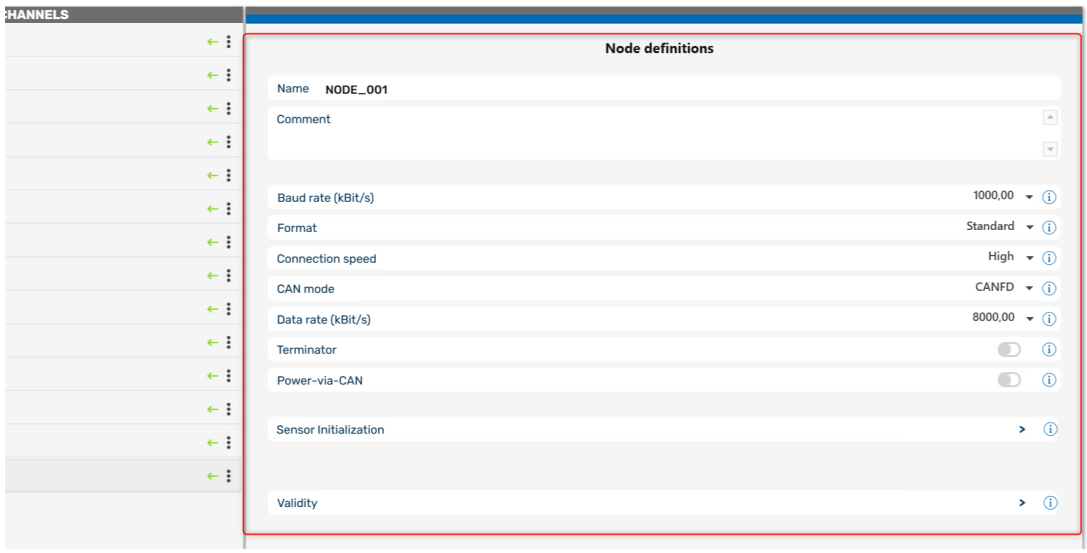
The assistant can be operated both by mouse and by touchscreen. On the right side, the properties are shown of whichever element is selected under Node, Messages or Channels.

Notes

Any changes made are applied immediately without the need to click on the *Enter*- or *OK*-button, even if they are incorrect, such as for the channel name. Any error is indicated in the status bar.

There is also no *Cancel*-button to correct unintentional changes!

9.10.4.6.2 Nodes



Definition nodes

Parameter	Description
Name	Unique name by which to distinguish the node, up to 65 characters long. The default is "NODE_" + Index (e.g.: 001) of the node is suggested. If multiple devices are present, the device index is additionally appended. The name must conform to the syntax of an imc FAMOS-variable. Whether a node sends and/or receives messages is indicated by a red/green arrow: NODE_001
Comment	Freely defined text describing the node, up to 255 characters.
Baud rate (kbit/s)	The baud rate specifies the interval between individual bits. All modules belonging to the same node must have the same baud rate. The drop-down list "Baud rate" offers a choice of rates from 5kbit/s to 1Mbit/s. The default value is 125kbit/s.
Format	"***Standard***", "***Extended***" or "***Extended+***". The difference between Standard and Extended is that the Standard-format ID length is 11 bits (IDs from 0 .. 2031 are permitted) and in Extended-format 29 bits (IDs from 0 .. 536870911 are permitted). Extended-format allows for greater distances between consecutive IDs. This makes it possible to extend the system without major changes to the configuration.
Connection speed	**High** Speed is the default and complies with ISO 11898. **Low** Speed corresponds to ISO 11519 and is applied in the automotive industry among other areas.

Parameter	Description
CAN mode	CAN classic: CAN High Speed as per ISO 11898, CAN Low Speed as per ISO 11519. CAN FD (ISO Standard): in accordance with ISO 11898-1:2015. CAN FD (non-ISO Draft): based on an earlier design from BOSCH.
Data rate (kbit/s) (only with CAN FD):	In *CAN FD* mode, the input box for the data rate appears. If the controllers of all subscribers in the Bus support CAN FD, then different settings for the Baud rate and data rate can be transmitted. When transmitting at the high data rate, the subscribers having the slower Baud rate are switched to passive status. Caution: In order for the data of a send message to be sent at the data rate set here, the message's option Bitrate switch 569 must be activated.
Terminator	A terminal resistor can be connected to a node via software. If the measurement device is located at the end of the CAN line, this eliminates the need for a Y-cable with an external terminator. See also "*Connecting terminators* 516".
Power-via-CAN	Connectable power supply for imc CANSAS modules. By means of this switch, the power supply for the CAN FD module is activated via the CAN-Bus cable (Pin 1(+) and Pin 5(-)). This voltage is isolated from the CAN-Bus. Caution: The module power supply and its pinout are imc proprietary accessories only appropriate to imc CANSAS modules.

9.10.4.6.2.1 Sensor initialization

Many sensors require a one-time initialization in the form of one or more messages. This initialization is usually necessary only once, upon activation of the sensors.

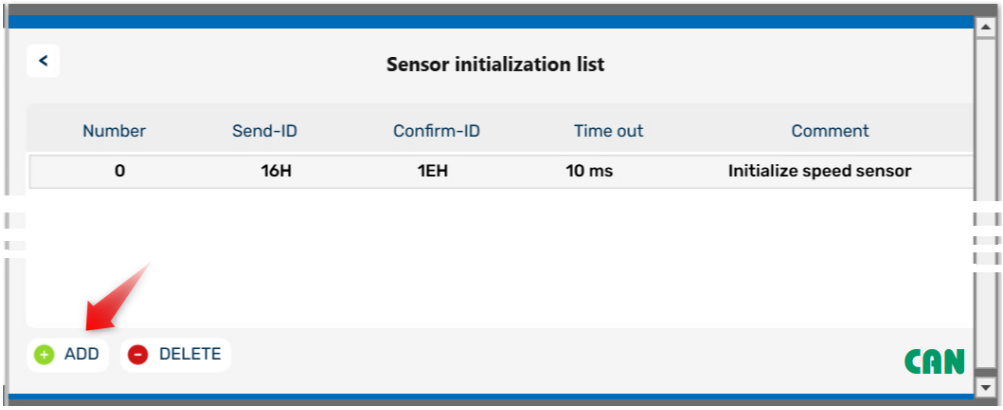
For one single time, the device can output a specified sequence of messages on the CAN-Bus. This initialization sequence is performed once only upon preparing for measurement after configuring the device. If the configuration isn't change, no initialization is performed any more upon next starting.

For devices that start automatically, the initialization is performed in the same way upon at the first start.

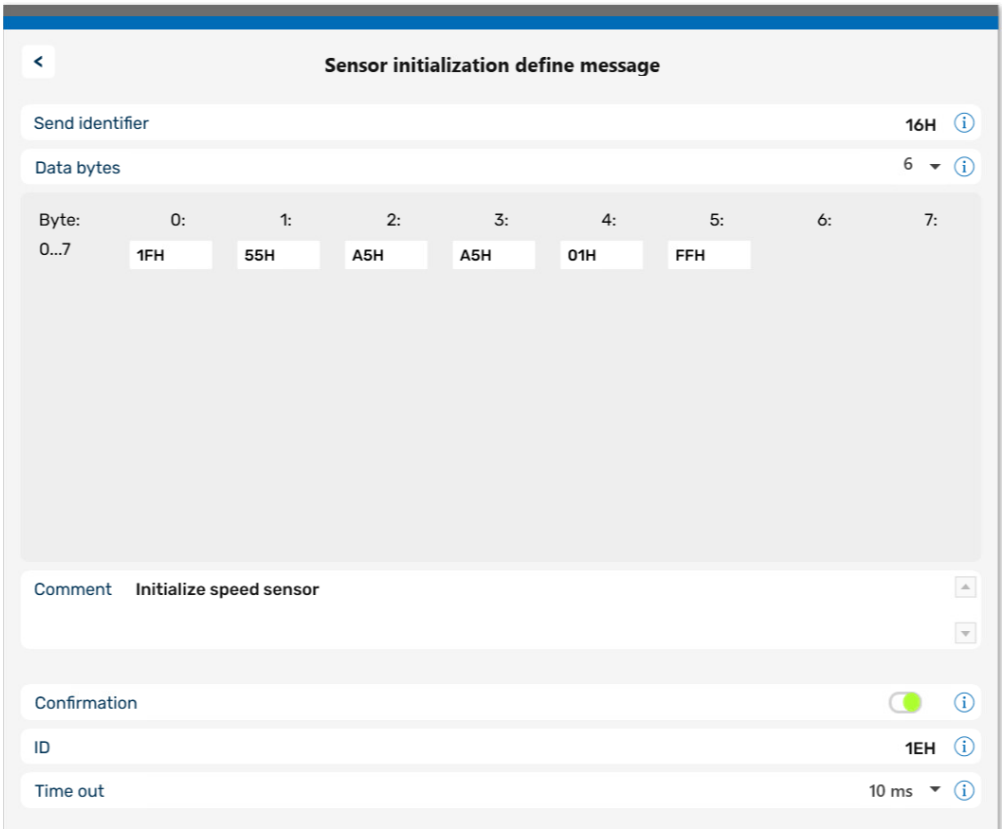
The initialization consists of a sequence of messages. After each message, it is optionally possible to wait for a response from the sensor by means of *Time out*. Many sensors acknowledge a parameterization message with an additional message. The device then waits for the sensor's response before sending the next message.

Defining the initialization message

The Sensor Initialization list is accessed from the node's Properties page.



Multiple messages can be created. New messages can be created by clicking on ADD.

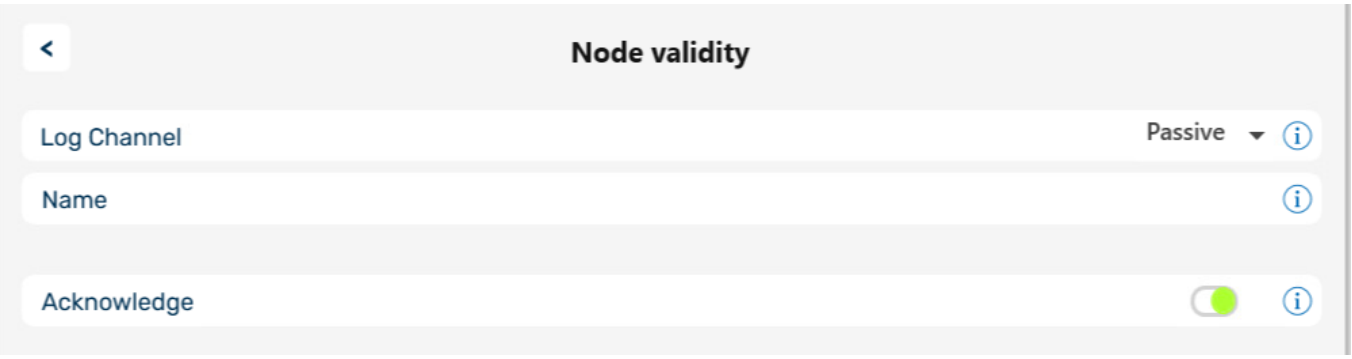


For the purpose of initialization, messages are sent to the sensors. These messages can be acknowledged. The list in the dialog shows the entire initialization sequence.

Sensor initialization

Parameter	Description
Send identifier	An identifier for the message with the initialization sequence. Allowed values are from 0 to 2031 or 0H .. 7efH for Standard-format, for Extended-format 0 .. 536870911 or 0H .. 1fffffffH. The ID is always displayed in hex format but you can enter it as a hexadecimal or decimal number. Numbers entered in Hex-format must be designated as such by using H or h (e.g. 2ah)
Data bytes	The number of bytes to be used for the message. Up to 8 bytes on the classic CAN-Bus; up to 64 bytes on CAN FD.
Byte 0 .. 7 (63)	Content of the individual byte which is sent to the sensor. 0 .. 255 or 0H .. ffH are allowed.
Comment	Additional description of the initialization sequence or other information - maximum 200 characters.
Confirmation	Activation of an acknowledgement message.
ID	Identifier of the acknowledgement message. Both Standard- and the Extended-format are possible. This message must be sent by the sensor. The device waits for arrival of such a message before resuming at the next line of the list.
Time out	If after this time (1 ms .. 1000 ms) no acknowledgement is received, the initialization sequence is continued. In this case, the sensor may not be initialized.

9.10.4.6.2.2 Validity



Validity of a node

Log Channel

Parameter	Description
Log Channel	When Log channel "*Active*", all messages belonging to the node are logged in imc STUDIO. In the input field *Name* you can enter a unique name for the log channel. By default, a name with slot and node number is generated. There is exactly one log channel per node. Message IDs and data bytes are logged.

Acknowledge

Disables the transmission of Acknowledge messages. The CAN controller usually sends this packages to confirm successful received messages.

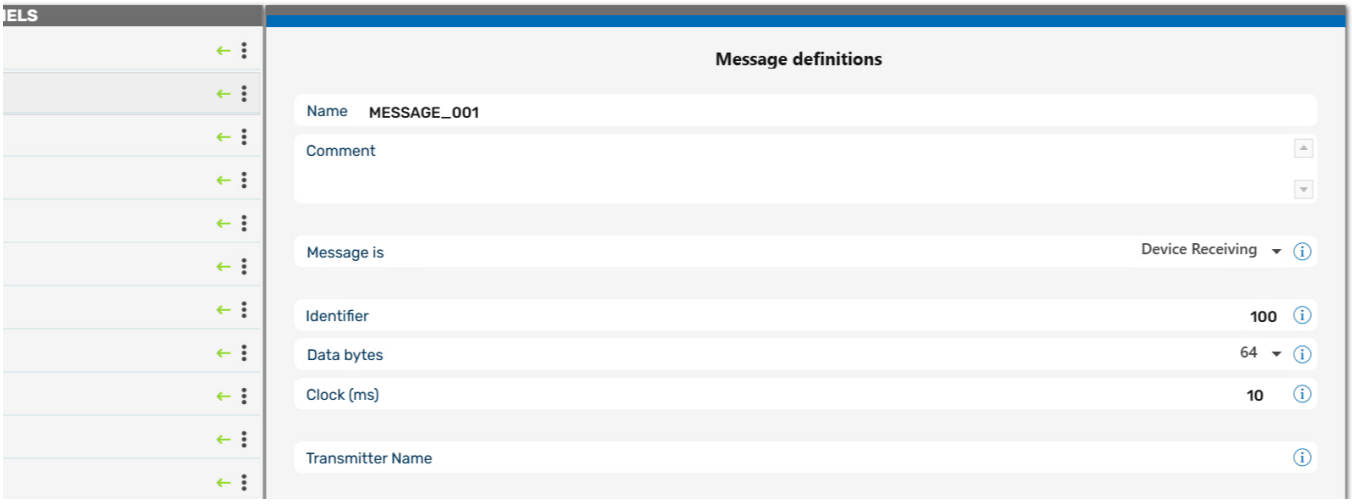
9.10.4.6.2.3 Context menu

Right-click the node name or the three dots to display a shortcut menu with the following options:

Parameter	Description
Import	Restores settings from a JSON file. The available formats are "JSON" and DBC".
Export	Exports the settings of a node, including all messages and channels, as a "JSON" or "DBC" file.
Revert	All changes to the node are discarded and the settings are restored back to their state at the time the assistant was opened.
Clear	Discard settings. All messages with channels are deleted.

9.10.4.6.3 Messages

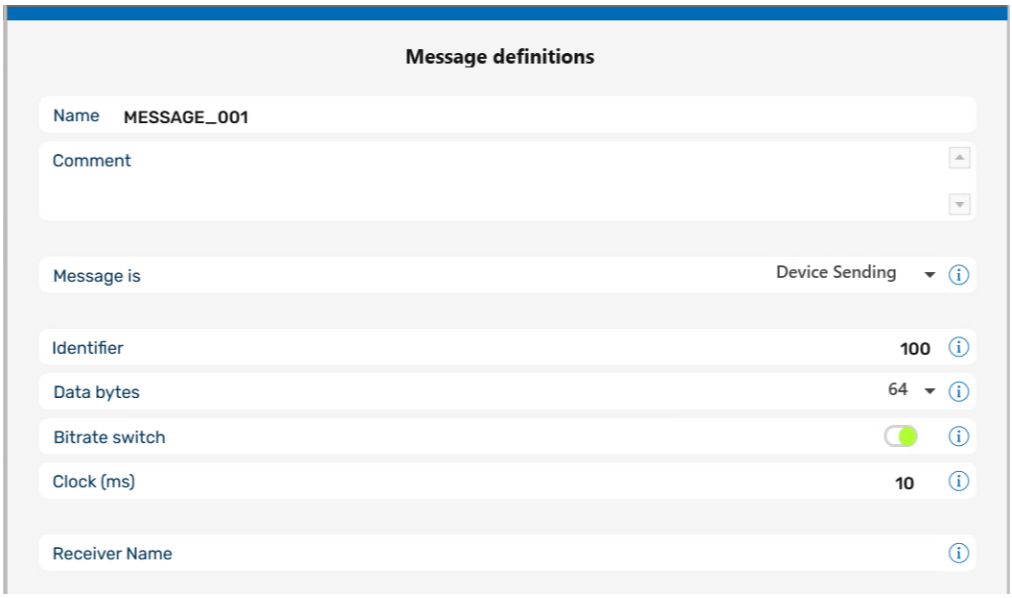
Receiving a message



Received message

Parameter	Description
Name	A unique name for each message with up to 65 characters. The default name for a new message is Message_" + index (e.g. 001). If several devices are being used, a _j is appended to the name, where j is the device index number. All names must be comply with imc FAMOS variable syntax, just as all other device channel names..
Comment	Freely definable text describing the message and having a maximum of 255 characters.
Message is	Specifies whether the message is to be sent or received. Received messages are indicated by a green arrow. MESSAGE_001
Identifier	Each message on the CAN-Bus contains an identifier (ID) which indicates the origin and meaning of the data. No two messages in a node can have the same ID. The identifier also indicates the priority of the message and, once again, no two messages can have the same priority. When two messages are to be sent at the same time, the message with the lower ID is sent first. Allowed values are: 0 to 2031 (Standard ID format) and 0 to 536870911 (Extended ID format). Identifiers can also be specified as a hexadecimals. Append an "h" or "H" to the hexadecimal, e.g. 2ACH or e4h. Otherwise the number is interpreted as decimal.
Data bytes	Number of bytes in the message, 0 to 8, or 64 if the node has been created in CAN FD mode 564. For instance, if 4 bytes is entered, 4 bytes will be sent per message. It is not possible to send partial bytes in a message but it is also not necessary that every bit in the byte be occupied.
Clock (ms)	Messages are sent via the CAN-Bus at a regular rate. The value for the clock specified here is a set point, not necessarily the actual value. The clock is a property of the sensor. The allowed range is 0.001ms to 100,000ms. The sampling rate of the CAN channel to be created is determined by the clock value specified
Transmitter name	Name of the sender of the message. Only for description.

Sending messages



Sendende Botschaft

Parameter	Description
Name, Comment, Identifier, Data bytes, Clock	As for messages to be received. 568
Bitrate switch	**Data bytes for CAN FD**: Messages can be sent at the data rate set for FD 527. For this purpose, the option ***Bit Rate Switch*** must be activated, which appears at the CAN FD node.
Message is	Messages to be sent are indicated by a red arrow.
Clock	Messages are sent via the CAN-Bus at a regular rate. The *Clock*-value is the interval between transmission of messages. The interval must be at least 1ms. The value set here serves in imc Online FAMOS as the default setting for the clock rate for cyclic transmission in the Formula-Assistant..
Receiver Name	The receiver name describes the receiving sensor. This name is not used for any other purposes - therefore it need not be unique.

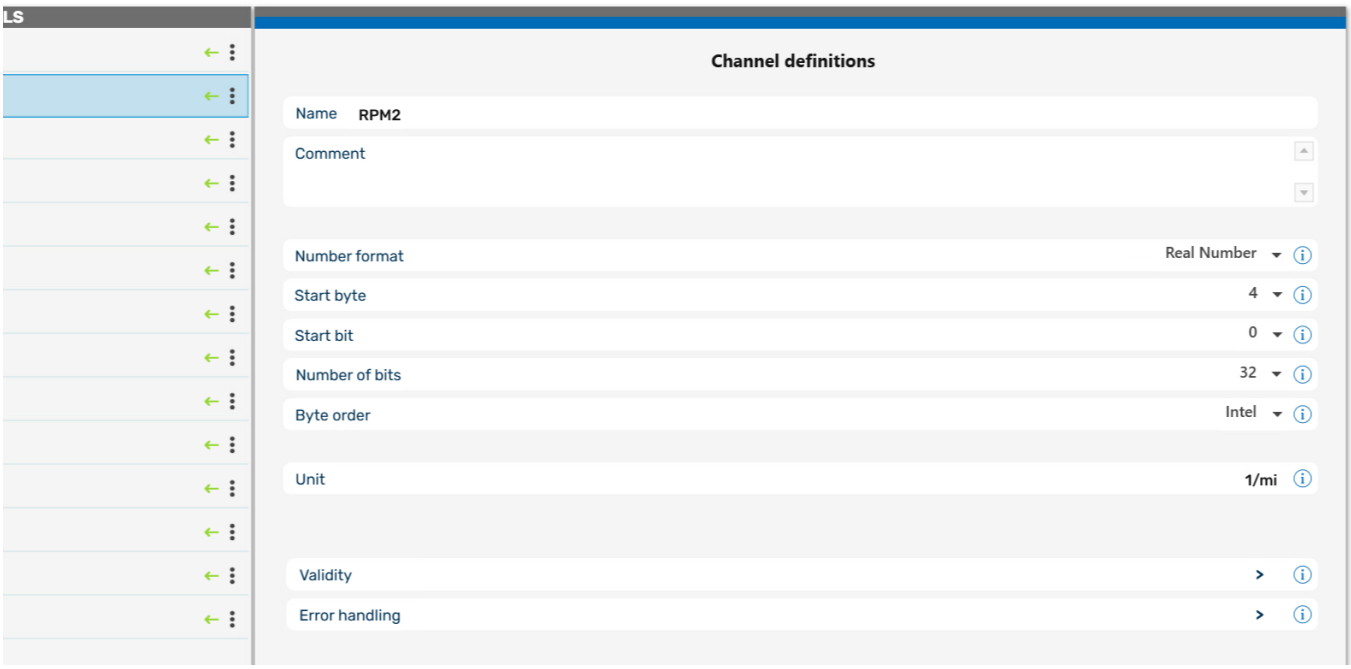
Context menu of Messages

Right-clicking on the name of the message or on the three dots accesses a way to delete the message.

9.10.4.6.4 Channels

Analog CAN-channels

Analog signals are digitized by the CAN-sensors with a specific number of bits. The numerical format can be set to *signed integer, unsigned integer or real number*.



Definition of a analog channel

Parameter	Description
Name	Unique name with up to 65 characters. Default is "CHANNEL_" + channel's index. The name must conform to the syntax of a imc FAMOS-variable.
Comment	Freely defined text describing the message; up to 255 characters
Numerical format	Data can be sent and received as signed or unsigned integers, real numbers or in digital (1Bit) format.

Note Information about the numerical format

Some manufacturers may not indicate clearly in their CAN tables which number format they use.
For example, the table may indicate that Bytes 1 and 2 represent the guidance angle and that this value is to be multiplied with 0.0639 to arrive at the actual value in degrees. If no other information has been given, there is no definite way of determining the number format other than trial and error. Our tech support cannot help you in such cases! Sometimes however, there are implicit clues as to what the format is.

Examples:

- If the range minimum is 0 and the maximum is ffff hex, then unsigned integers are being used because they always start at zero.
- If the range is given as 7fff = -700°C and 8000 hex = +700°C, then signed integers in two's complement form are being used. The zero (0 hex) lies directly in the middle between positive and negative integers.

The most certain method of determining the number format is to run the sensor through its entire range.
If the correct number format has been selected, the curve that you measure will be continuous. Otherwise, the upper and lower halves are exchanged, which appears as a discontinuity. A discontinuous curve is therefore an indication of the wrong integral format being used.

Parameter	Description
Start byte	Specifies in which byte of the message the number begins. Byte 0 is the first transmitted byte of the CAN-message.
Start bit	The start bit specifies where the number starts within the start byte.
Number of bits	This is the number of bits used in the channel. The remaining bits continue in the direction of the increasing byte indices. For signed and unsigned integers, 1 to 32 bits are allowed. Real numbers can range from 32 (floating point) to 64 (double).
Byte order	Bit order used by the CAN-module: either Intel-format or Motorola-format

Note Byte Order | Format

Intel format: The start bit is the LSB. The **LSB** has a **lower byte index** than the MSB.

Example: start byte = 5, start bit = 3, bits = 18

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 5	X	X	X	X	**LSB**			
Byte 6	X	X	X	X	X	X	X	X
Byte 7				**MSB**	X	X	X	X

Motorola format: The start bit is the MSB. The **LSB** has a **higher byte index** than the MSB.

Example: start byte = 5, start bit = 3, bits = 18

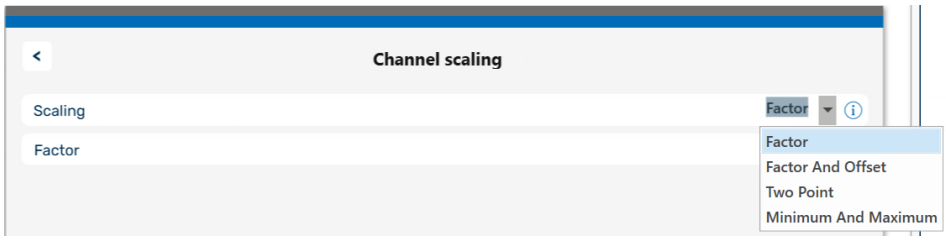
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 5					**MSB**	X	X	X
Byte 6	X	X	X	X	X	X	X	X
Byte 7	X	X	X	X	X	**LSB**		

Note on recognizing Intel or Motorola byte order

- **The CAN tables from manufacturers do not always indicate whether Intel or Motorola byte order is used.** It is sometimes possible to determine the byte order indirectly. Within one message typically only one of the variations will be used.
- **Example***:* Two byte data packet, message Byte 3 is LOW and message Byte 4 is HIGH
  Intel byte order because HIGH byte contains MSB.
- **Example***:* Bits 0 to 14 contain data and Bit 0 is LSB
  Data is contained in Bytes 0 and 1 but only the lowest seven bits in Byte 1 are occupied. Therefore, LSB is in the byte with the smaller index which means that Intel format is used.

Parameter	Description
Unit	The physical units of the channel, maximum 24 characters. The units are used when displaying the channel, e.g. on the Y-axis in a curve window. Note that units should be specified without prefixes. Examples of units that you can use are: m, s, kg, N, m/s. If you need to use prefixes (e.g. n-, µ-, M-) or non-SI units (e.g. inches, pounds), always place these between quotation marks (e.g. "inches").
Scaling	Here is determined how to convert the physical value to a CAN bus integer. There are different choices available, to determine the conversion.

Note Notes for scaling



The Scaling options

Y = physical measurement value

X = CAN bus integer value

Scaling	Description
Factor, Y=Factor*X	**Receiving**: for received signals (X) the following applies: Y = Factor * X **Sending**: for physical values (Y) to be sent, the integer value is calculated as follows: X = Y / Factor
Factor and Offset, Y = Factor * X + Offset	**Receiving**: for received signals (X) the following applies: Y = Factor * X + Offset **Sending**: for physical values (Y) to be sent, the integer value is calculated as follows: X = (Y - Offset)/Factor
Two-point specification, (X1, Y1), (X2, Y2)	X1: x-coordinate of 1st point Y1: y-coordinate of 1st point X2: x-coordinate of 2nd point Y2: y-coordinate of 2nd point
Minimum and Maximum	The lower limit of the specified number format with the specified number of bits corresponds to this minimum (e.g. signed integers, 8 bit, range: 0 to 255, minimum corresponds to 0, maximum corresponds to 255)

Context menu of Channels

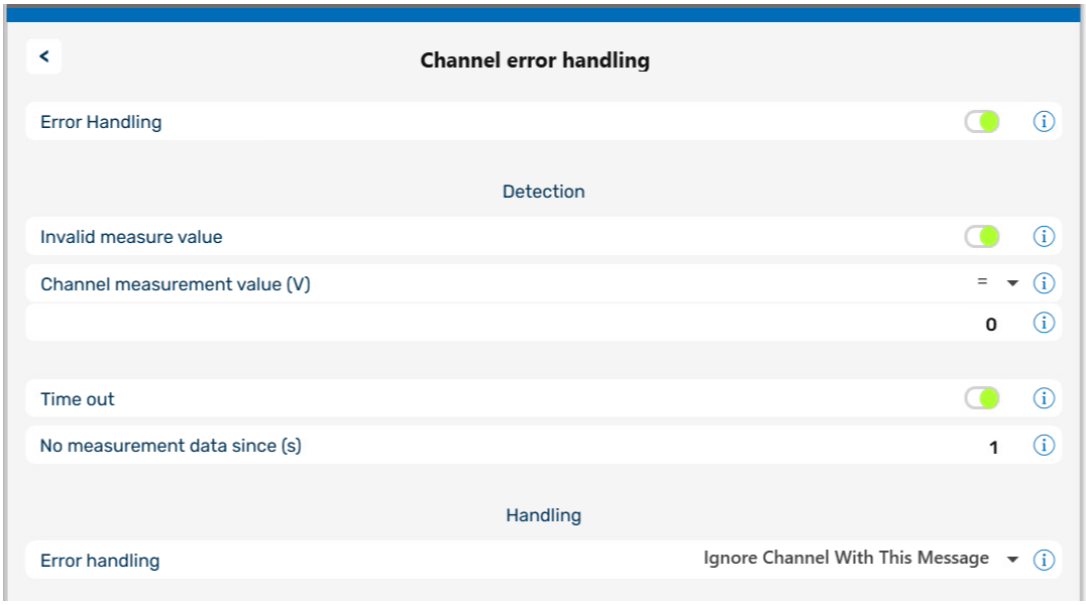
By right clicking on the channel name or the three dots, you can access to the command for deleting the channel.

9.10.4.6.4.1 Validity

Parameter	Description
Channel state	Active by default. The channel can be set to passive. The settings remain intact, but the channel is not set up in the data logger and does not demand any resources.

9.10.4.6.4.2 Error handling

The dialog illustrated below appears if the system is set to *receive* CAN-Bus messages.



Error handling

If an invalid measurement value occurs or if no more messages are received by the device after a certain amount of time, *Error handling* can be used to notify the user. When *Error handling* is activated, the error handling option *Invalid measurement value* and/or *Time out* are displayed.

Invalid measurement value

The comparison value (Channel measurement value) for an analog channel is expected as a real number with physical units.

Time out

Define a time out limit here. This is the maximum time which the sensor has to send the message. Many sensors send messages at regular intervals - this option is often useful for monitoring them.

Error Handling

The available options are

- *Ignore Channel with this message*
- *Last channel value*
- *Replacement Value* - Value indicating an error, e.g. -999

The error handling functions slightly differently, depending whether the channel is continuously sampled or whether time-stamping.

Error correction	Effect with continuous sampling	Effect with time stamping
Ignore channel with this message	Same as *Last channel value*	Es wird kein Sample erzeugt.
Last channel value	The last valid channel value is generated each sampling interval until a valid sample arrives.	*Time out error: A* new sample is generated using the last valid channel value and is stamped with the time-out limit. *Invalid measurement value:* a sample is generated using the last valid channel value and is stamped with the message time stamp.
Replacement value	The replacement value is generated each sampling interval until a valid sample arrives.	Like *Last value,* but with *replacement value.*
No error correction used	Same as *Last channel value*	Same as *Ignore channel with this message*

Note

- **Input range of replacement values for error handling**
  Certain replacement values may be outside of the input range for the CAN channel.
- Channels with 16 bits or less can receive replacement values having a 16-bit input range. Example: For a CAN channel with 8 bits, a replacement value of up to 16 bits is permitted.
- Channels with more than 16 bits and less than or equal to 32 bits can be given replacement values having a 32-bit input range. Example: For a CAN channel with 24 bits, a replacement value of up to 32 bits is permitted.
- Consequently, the following exceptions exist: Channels with exactly 16 bits or 32 bit-channels only permit replacement values having the same input range.

9.10.4.6.5 imc STUDIO - Configuration / Processing of fieldbus channels

Analog inputs

The channels created using the CAN-Bus-Assistant appear in the channel list under **Analog channels > Fieldbus: Analog inputs** for further processing.

Digital I/O's

The digital channels created using the CAN-Bus-Assistant appear in the channel list under **Digital channels > Fieldbus: Digital inputs / outputs**.

9.10.4.6.5.1 Settings

If fieldbus channels are selected in the channel list (assuming the device currently allows this), then additional setting options are available.

Parameter	Description		
	long name	*short name*	*column ID*
X-axis	*X-axis*		*eXFormatVariable*
	• *Sampling rate*: The setting Sampling time is related to the settings **Duration** or **Sample count** and **Sampling rate/time** as previously discussed. • *Time stamp*: Data acquisition is handled by external, asynchronous sampling within the fieldbus. The corresponding time coordinate for the data point is passed as a time stamp along with the data for subsequent processing. In this case, the parameters **Sample count** and **Sampling rate/time** are irrelevant.		
Duration	*Duration*		*eDuration*
	If the parameter X-axis is set to *Time stamp*, the duration of the measurement depends only on the setting in **Duration**.		
Assignment	*Assignment*		*eAllocation*
	The data can additionally be assigned to a Display variable (only for analog Fieldbus inputs).		

9.10.4.7 Application examples

9.10.4.7.1 Sending CAN messages

For this example, an imc measurement device with a CAN-interface is necessary.

9.10.4.7.1.1 Balancing a bridge module via CAN

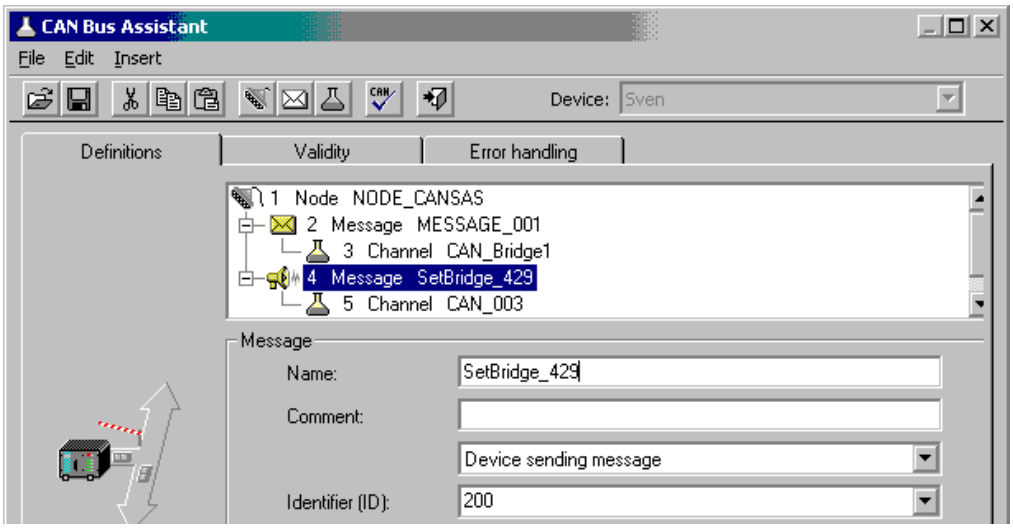
A CANSAS bridge module is used to take a strain gauge measurement. The bridge is to be remotely balanced in its rest state.

Some imc CANSAS-modules arte able to receive CAN-messages for balancing and shunt calibration. The devices must previously be prepared for this by means of the CANSAS configuration software, refer to the CANSAS manual for details. The example assumes that balancing takes place whenever the number 10Hex is sent in a message with the identifier 200.

CAN-Bus Assistant settings:

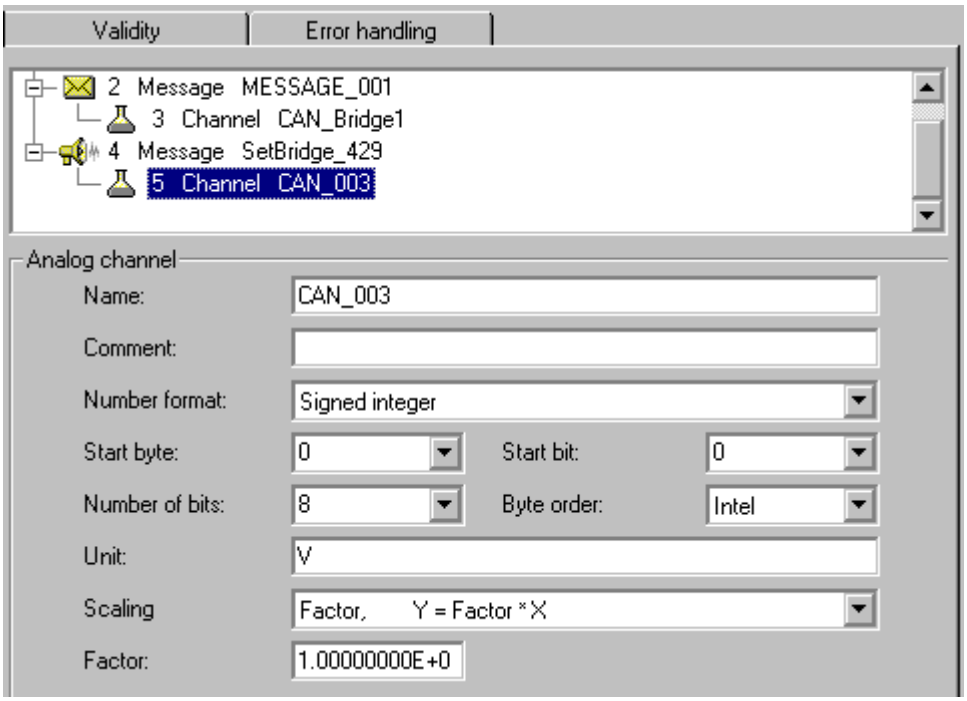
As message "SetBridge_429" is created. The name is arbitrary, the suffix _429 is an aid in locating the module by the serial number. The message must be given the ID which was previously set in imc CANSAS-module for reception purposes, the default is 200.

The channel CAN_Bridge1 measures the strain gauge signal, and thus we will be able to observe how the offset disappears if we balance the bridge while the measurement is running.



Create a message for sending the balance command

The channel "CAN_003" is also created. We intend to send the number 10 Hex via this channel and reserve 8 bits for it.



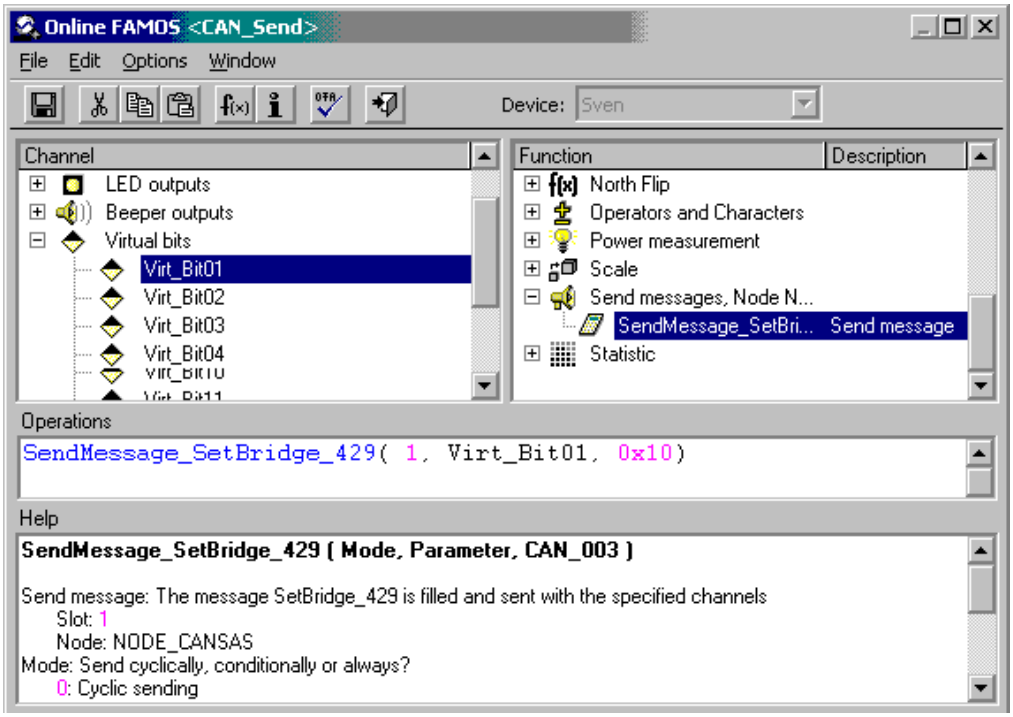
Channel for the balance command

The balancing process is to be carried out on command. For this purpose we use a virtual bit which we rename in the configuration. Later, the DIODAC dialog will provide a convenient way to set this bit. Of course, it would also be possible to use a digital input bit which is set by means of a button.

imc Online FAMOS:

Once a message has been created in the CAN-Bus-Assistant (and only then!), a new entry appears in imc Online FAMOS, *Send messages, Nodes N…*. There you will find a function which starts with *SendMessage_* and bears the name of your message, in this case *SendMessage_SetBridge_429*. This function is able to send a message in three different ways: cyclically, condition-dependently, and, if you are working with control commands, permanently.

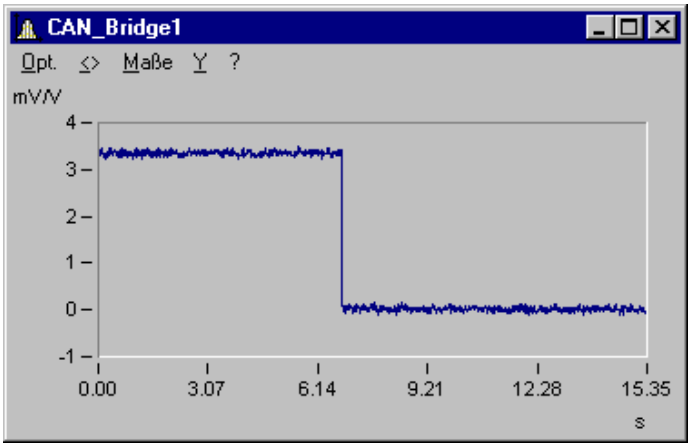
We select the operating type *With condition* and select our virtual bit *Virt_Bit01* as the condition. The number we send is 10 Hex, which causes imc CANSAS-module to be balanced.



SendMessage-command in imc Online FAMOS

We create a Panel page showing the strain gauge and a push button linked with the virtual bit for the balancing. Start the measurement. At first the bridge is unbalanced. Set the virtual balance bit with the push button.

The bridge is balanced:



The bridge is balanced and shows 0 mV/V

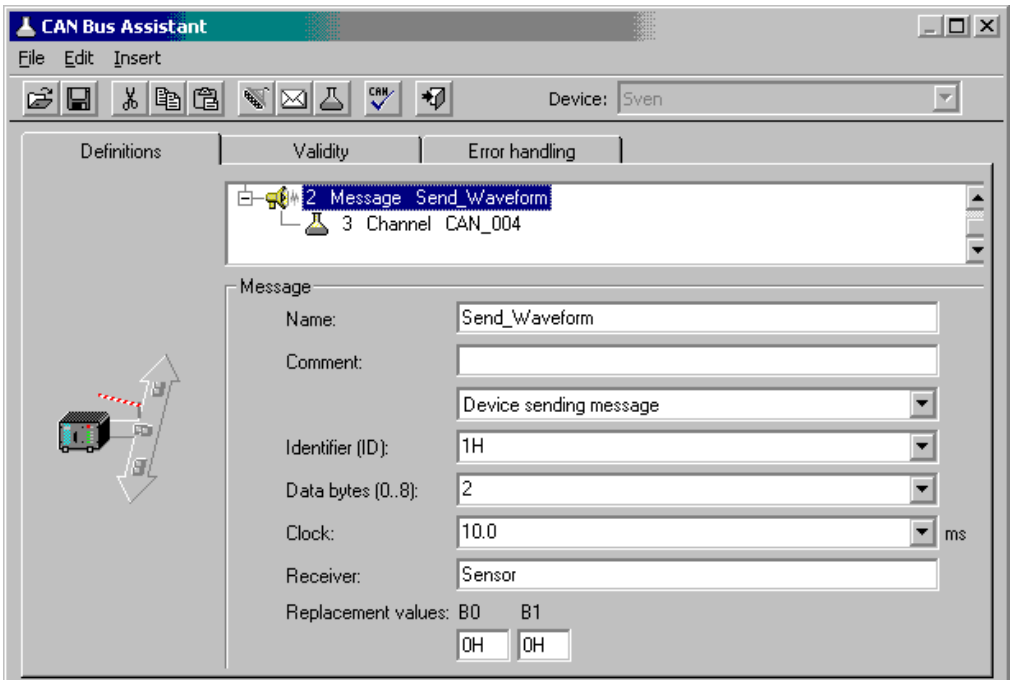
9.10.4.7.1.2 Initiating a shunt calibration for a bridge module via CAN

Proceed as for Balancing a bridge module via CAN 577. Simply substitute the HEX-code to send with 20Hex.

9.10.4.7.1.3 Outputting a waveform to a DAC module

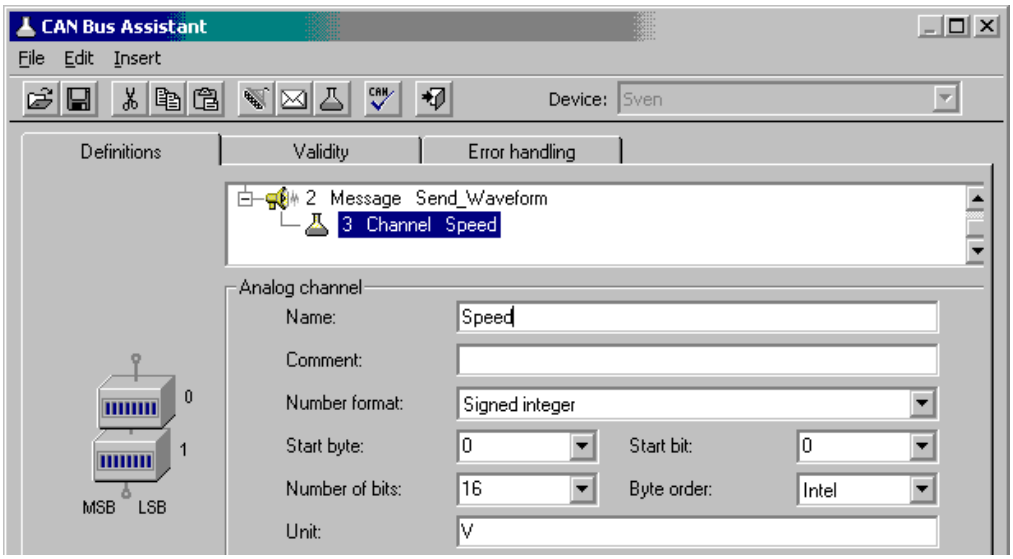
Cyclical output of values is another example. In this example, a ramp is generated. The data set to be outputted is loaded in the form of a characteristic curve. The `Charact` function is used to convert the ramp to the output values. This is sent as a message to a DAC module.

CAN-Bus-Assistant settings: Message



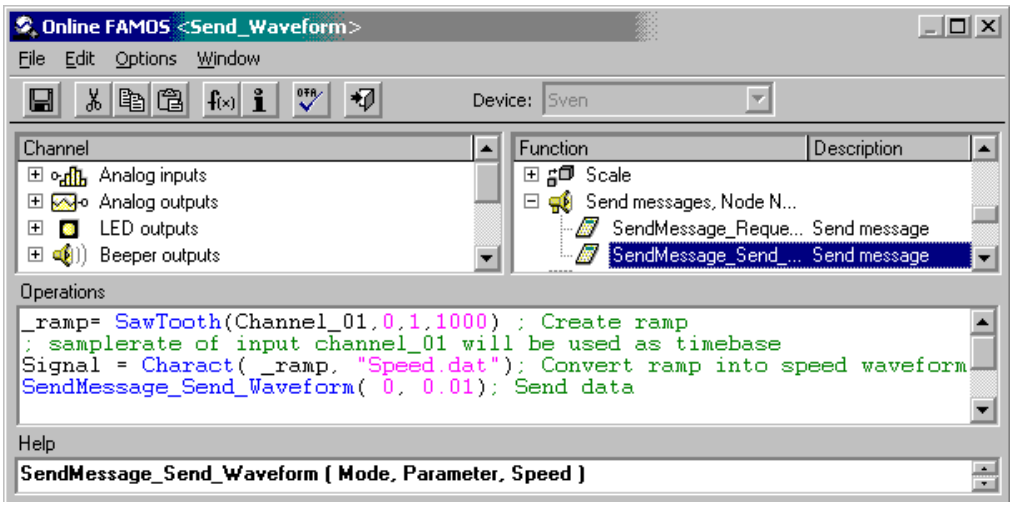
Create a send message with a channel

Channel:



The send channel is configured to the format of the data, that will be transfered by imc Online FAMOS afterwards

The text in imc Online FAMOS then looks like this:



Transfer the data in imc Online FAMOS

Note

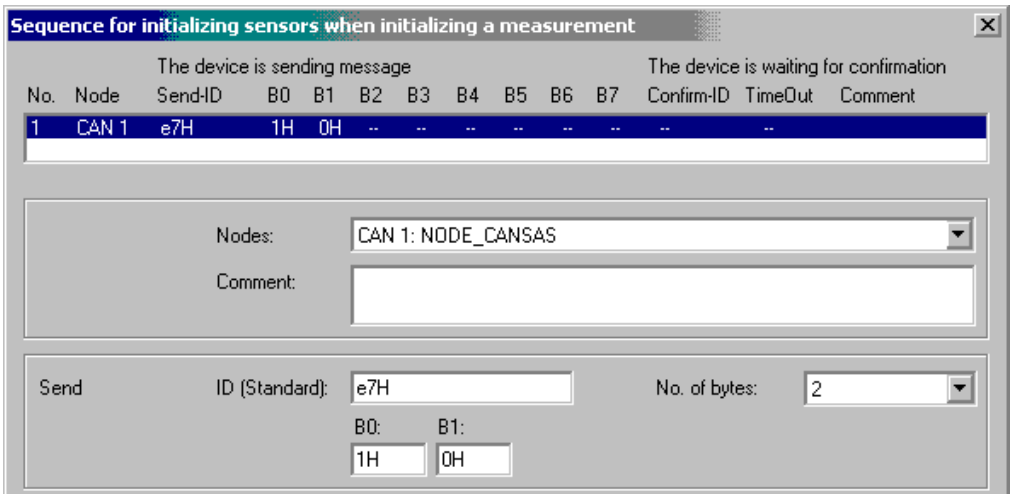
- The ramp must have the same length and same data rate as the data set to be outputted.
- In this example, an underline "_" was prefixed to the ramp. This causes the ramp to be retained in the device as an local auxiliary variable and cannot be found by the curve window.
- The data set to output must be in the folder which is specified under Options-Folders.
- The data set to be outputted must be expressed in imc FAMOS format. You can create the data set using imc FAMOS or simply copy data from an already recorded measurement.

9.10.4.7.2 Initializing a sensor and requesting data

Many sensors don't operate cyclically and must be "started". For this purpose, they need information about the initialization ID as well as on the numbers which are sent for the initialization. Both are noted in the description of your sensor.

In order to create an initialization sequence, open the CAN-Bus-Assistant. Once you are there, select the menu item Edit > Sensor initialization 538.

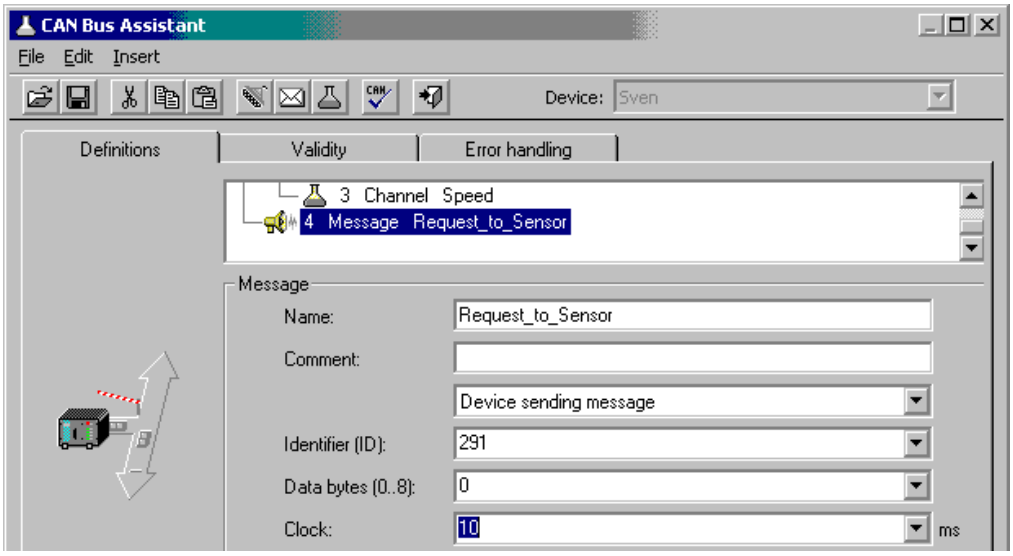
A possible sensor initialization code to enter could be as follows:



Example for initializing a sensor

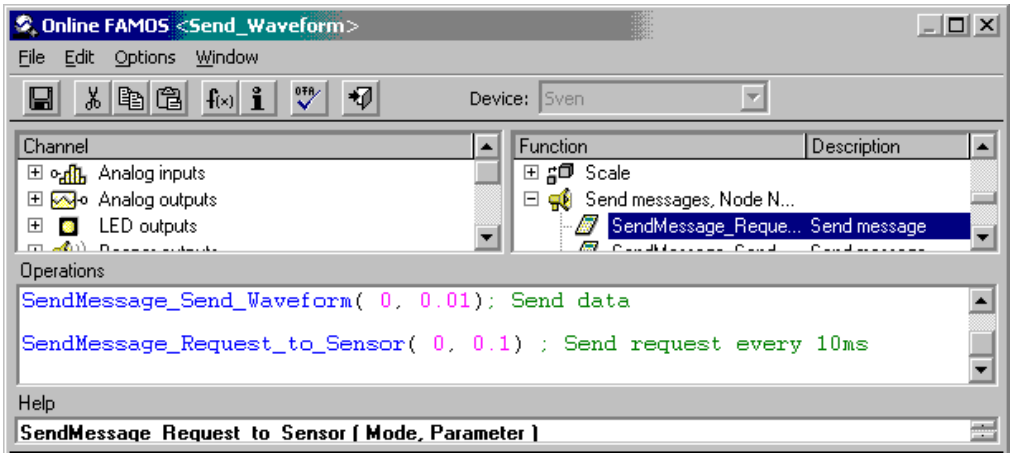
If your sensor requires a request for data, you will need to send a message to it. The ID and contents of the message are stated in the sensor's documentation.

In the CAN-Bus-Assistant, you must create a message and then put it in sending mode:



Create a send message

In imc Online FAMOS, select the SendMessage function and specify whether to make the data request cyclical or to make it depend on a state transition:



Example of cyclical sending at a 10ms rate

```
SendMessage_Request_to_Sensor( 0, Virt_Bit02) ; Send request, if
    ; Virt_Bit02 changes from 0 to 1.
Help
```

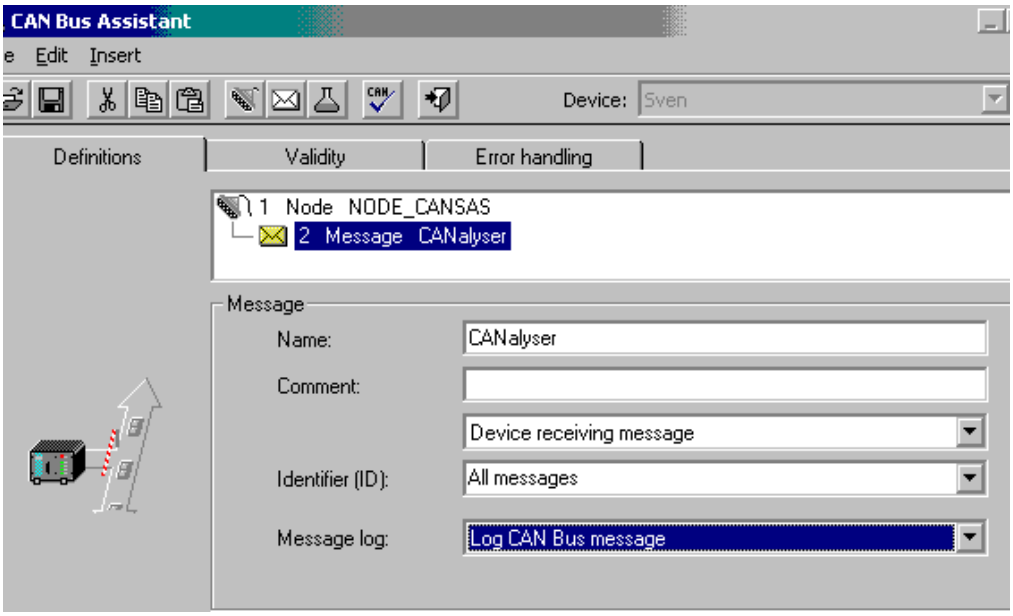
Example of an event-dependent request

```
; Send request, if Channel_01 is greater than 5V.
_Channel_01GreaterThan5Volt= Greater( Channel_01, 5)
SendMessage_Request_to_Sensor( 1, _Channel_01GreaterThan5Volt)
Help
```

Example of a request in response to a virtual bit, for instance by means of the DIODAC dialog.

9.10.4.7.3 Setting imc STUDIO as a CAN analyzer

It is possible to make a comprehensive log of all messages. For this purpose a message must be created in the CAN-Bus-Assistant which receives all the IDs. Furthermore you must set *Log CAN-Bus message* under Message log:

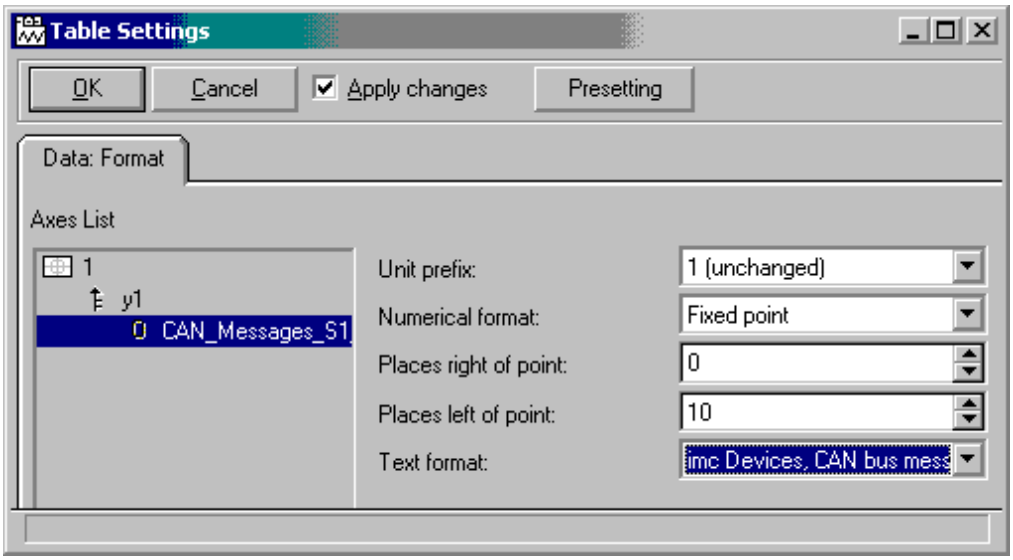


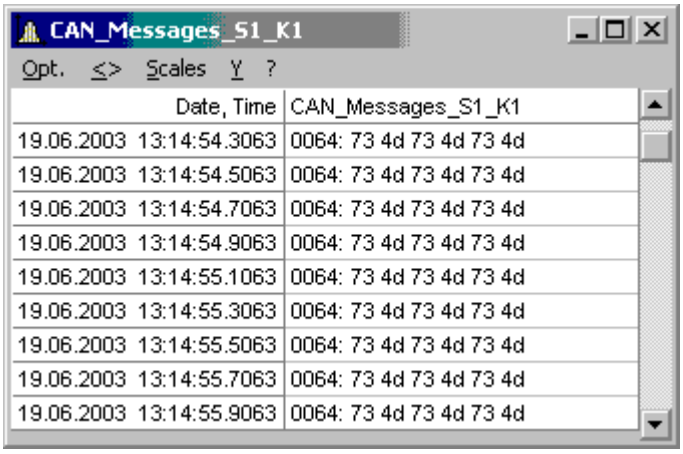
Create a log message

At this point, a channel is created whose tree-diagram entry appears together with the Fieldbus inputs.

Display this channel in a curve window in tabular form. Select the menu item *Options Display Absolute Date/Time*.

Right-click the mouse in the table and select *Table...* Select as the text format *imc DEVICES, CAN-messages*.





Date, Time	CAN_Messages_S1_K1
19.06.2003 13:14:54.3063	0064: 73 4d 73 4d 73 4d
19.06.2003 13:14:54.5063	0064: 73 4d 73 4d 73 4d
19.06.2003 13:14:54.7063	0064: 73 4d 73 4d 73 4d
19.06.2003 13:14:54.9063	0064: 73 4d 73 4d 73 4d
19.06.2003 13:14:55.1063	0064: 73 4d 73 4d 73 4d
19.06.2003 13:14:55.3063	0064: 73 4d 73 4d 73 4d
19.06.2003 13:14:55.5063	0064: 73 4d 73 4d 73 4d
19.06.2003 13:14:55.7063	0064: 73 4d 73 4d 73 4d
19.06.2003 13:14:55.9063	0064: 73 4d 73 4d 73 4d

Display of CAN messages as a CAN analyzer

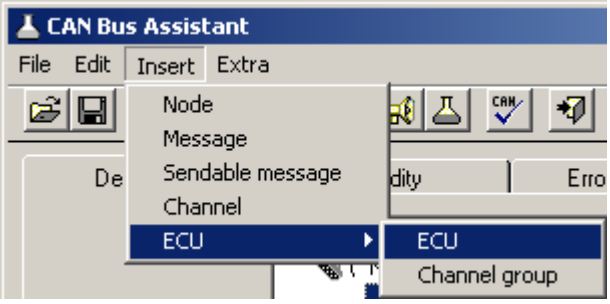
9.10.4.8 Control units in the CAN-bus assistant (ECU)

Using the CAN-Bus Assistant, the control unit properties and the values to acquire are selected. This can also be accomplished by importing a file describing the control unit or by configuring step by step.

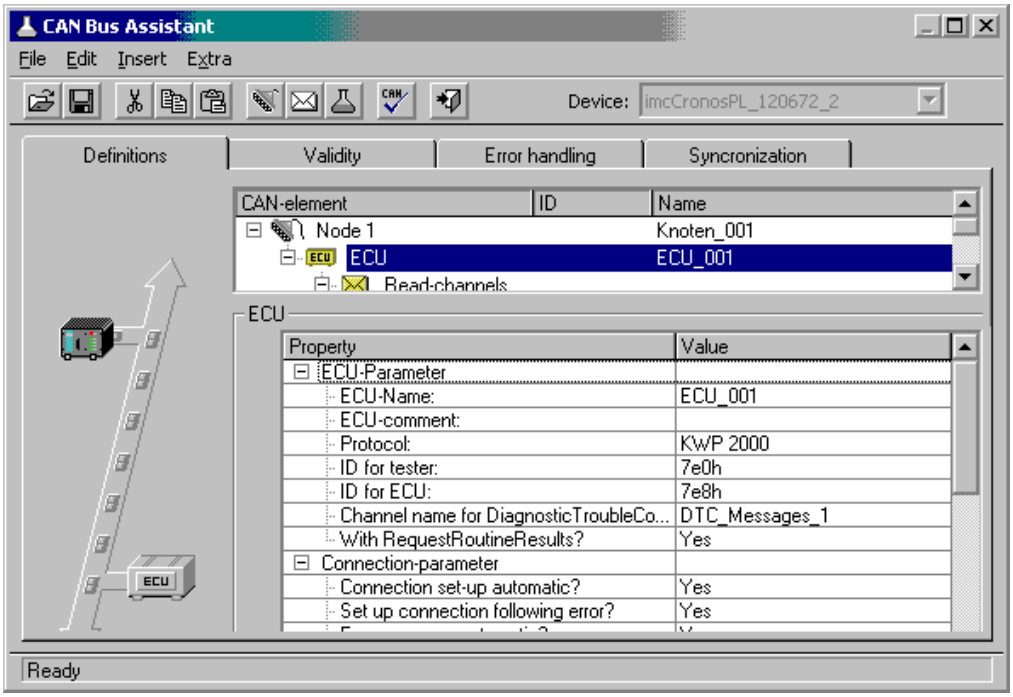
Requirements see here 522.

9.10.4.8.1 Configuring a new control unit

Start by assigning a new ECU to a CAN-node. Next, the ECU's properties must be configured in accordance with the control unit and protocol used. To open the control unit properties select menu *Insert - ECU* 556.

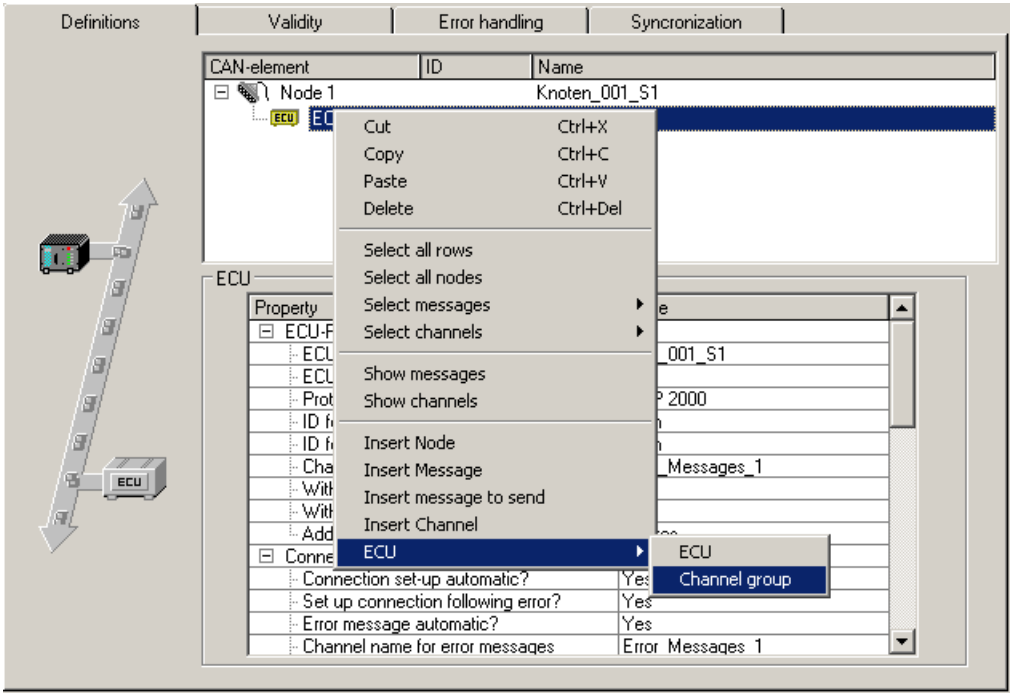


Open control unit properties



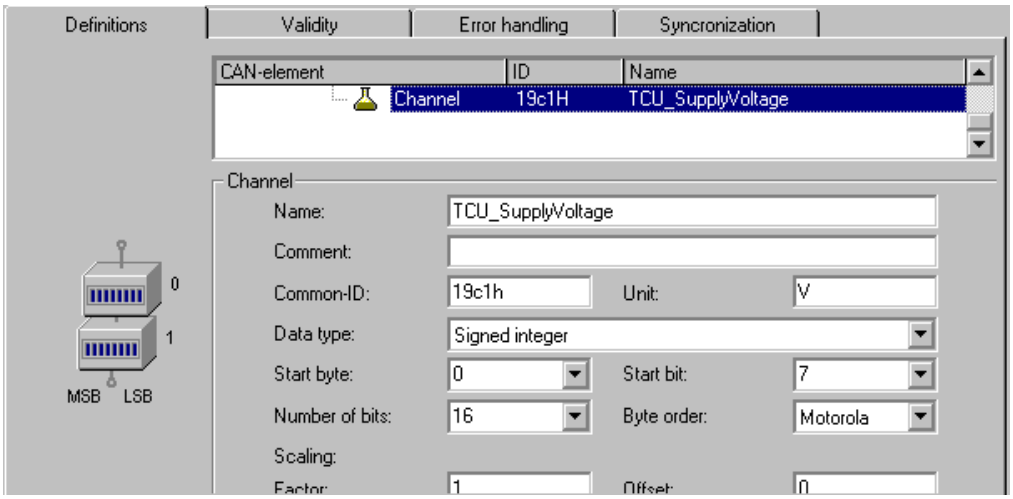
ECU properties

Channels representing the values to be captured can be assigned to the new ECU. For this purpose, a new channel group is created.



Create a channel group

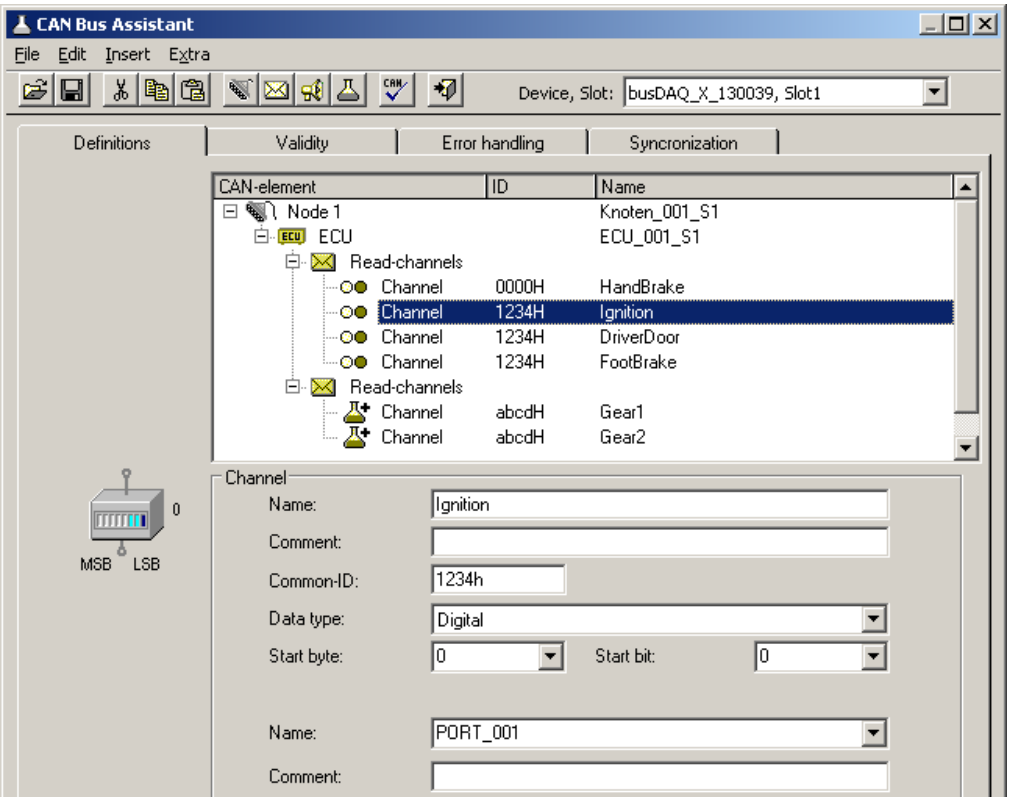
A new channel is automatically created for a new channel group.



ECU channel

An ECU's channels are arranged in channel groups. A channel group determines how the ECU's values are addressed. If the ECU's values are to be addressed using varying addressing modes, ten use multiple channel groups.

It is also possible to use channel groups for one's own organization purposes. If one of the ECU's addresses returns multiple, combined values, for example multiple individual bits each with a different interpretation, then these can be gathered together into one channel group.



Combining several bits

The object (channel) assigned to an ECU-address normally has a fixed size, in bytes. In order for such an object's values to be captured, the right quantity must be used when querying it. To do this, define all of an object's values. Configure any extra channels as passive. Alternatively, define at least one channel in the object's highest-value byte and configure it as passive if no capture is desired.

If the protocol supported by the ECU used allows it, it is also possible to create channel groups for the purposes of importing and/or exporting channels and for starting routines.

Create a separate channel group for each routine, in which the routine's parameters are to be defined.

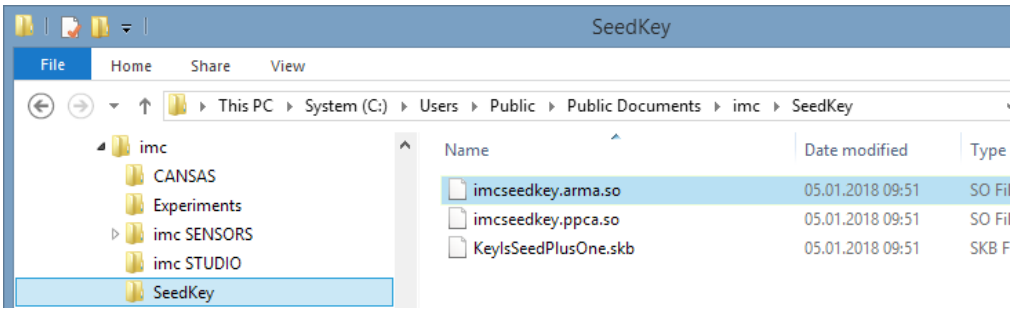
The channels then appear in the Channel-Table.

9.10.4.8.1.1 Seed & key

Access to certain diagnostic services can be protected using the **See-and-Key method**. This is done either via a **fixed key** or a **random seed value**. The algorithm required to generate the random seed value is stored as a file in the device. For each algorithm, a file must be created according to the following scheme:

- "*name.architecture.so*" (all lower case)
- ***name***: freely defined by user
- *architecture*: designates the platform on which the file can be executed
- "arma" for all platforms to date (devices as of group A4 187 and higher)
- ***so***: under Linux, typical file extension for dynamically loadable libraries.
- ***.skb***: Alternative format of Vector-CANape.

This file must be located in the directory "*C:\Users\Public\Documents\imc\SeedKey*".

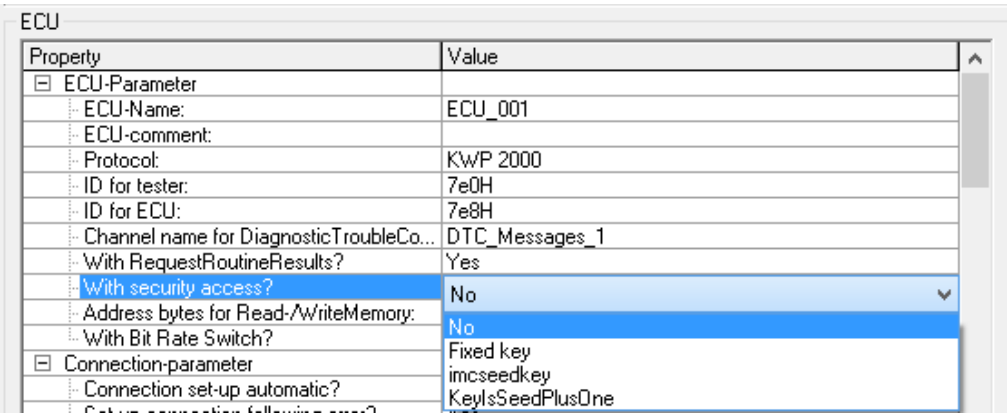


Directory "SeedKey" for "so" and "skb" files

Settings

The following parameters must be set in the settings of the ECU message, whose protocol is supported by the seed-and-key procedure:

- The "*With security access*?" property offers the options "*No*","*Fixed key*" and all file names which are available in the SeedKey directory. If a file is selected, it is transferred within the configuration during preparation.



Settings for "With security access?

- Selection "*Fixed key*": If "*With security access?*" is set to *Fixed Key*, *LogInKey* must be set to the key for the ECU. Only one fixed seed/key pair is supported.
- Selection of an encryption file: If there are ***. so*** or ***. skb*** (*KeyIsSeedPlusOne*) files in the SeedKey directory 588, they are also available at "*With security access?*"

9.10.4.8.2 Control units in imc Online FAMOS

The use of special functions for controlling control units is only possible with **imc Online FAMOS with Control Commands**.

In order to be able to evaluate return values from commands for the control unit, the special control command OnECUCmdReturn_* must be used (* stands for the control unit's designation (e.g. ECU_01)).

OnECUCmdReturn_* (Return, ECUCmd, CmdID)

***Result*: Return value**

The return value is composed of two parts.

kt * 256 + k

	Description
kt = 0	Communication error k = 0: no error, command successfully issued and acknowledged. k = 1: timeout, the control device did not respond in time. k = 2: sequence error, the ECU responses were not received in the correct sequence. k = 3: command buffer overflow. Only twenty commands (function calls) per ECU can be buffered.
kt = 1	protocol error The ECU has responded to the command with the error code k.
kt = 2	Return status The control unit has announced the end of a function which could not be executed successfully with the return status k.

***ECUCmd*: ID of the command**

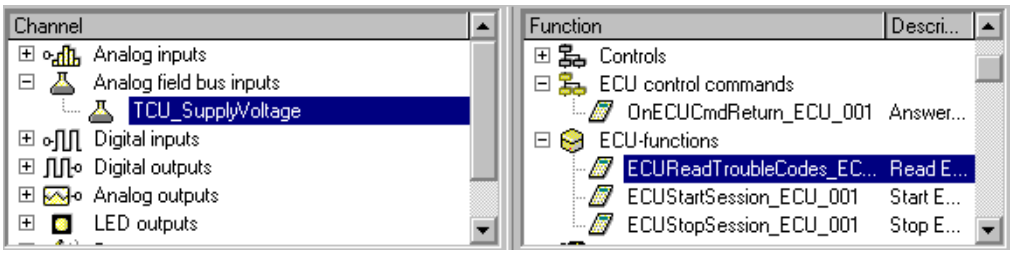
The command's ID consists of three parts.

b * 64 * 1024 + i * 256 + c

	Description
b	the bus node to which the control device is assigned b = 0: the first node in the CAN-Bus Assistant b = 1: the second node
i	index of the ECU i = 0: the first ECU assigned to the bus node
c	the protocol-specific command code

***CmdID*: Identifier or address of the command**

In addition, there are special functions for the control units:



Control units in imc Online FAMOS

```
; Executed during measurement
; By setting the virtual bit Virt_Bit01,
; a function with the ID= 0x68 in the control unit is started.
OnTriggerMeasure(BaseTrigger)
    if Virt_Bit01 = 1
        Virt_Bit01 = 0
        ECURoutine_ECU_001( 0x68, 9, 1)
    End
End
; Execute when receiving an answer
OnECUCmdReturn_ECU_001( OECR_Return, OECR_ECUCmd, OECR_ProcCmd )
; Checking the answer of the function with ID = 0x68
    If OECR_Return = 0 AND OECR_ProcCmd = 0x68
        ECUReadTroubleCodes_ECU_001( )
    End
End
```

9.10.4.8.3 KWP2000 (on CAN)

For this protocol, it is generally necessary to configure:

- *ID for tester:* Messages are sent to the ECU with this Identifier.
- *ID for ECU:* Response messages from the ECU having this Identifier are expected.
- Either values, or starting of the desired functions.
- Seed-and-Key settings, see here 587.
- *With RequestRoutineResults?:* If the system is to wait for the results of a routing of long duration, having a particular protocol, enter "Yes".
- *With Bit Rate Switch?*: With CAN FD 527 all send messages can be send with higher baud rate.

Additional parameters

- *Channel name for DiagnosticTroubleCodes:* Unique name for the channel, to which the requested DTCs are written.
- *StartDiagnosticSession:* Here the user enters the byte sequence for a diagnostics session, which can be understood by the ECU. A diagnostics session must be selected which meets the desired requirements. This means exporting of values and starting the desired functions.
- *Addressbytes for Read-/WriteMemory:* The default value for KWP2000 is the address made of three bytes.
- *Tester present:* How to respond to the message notifying the ECU that test personnel is present.
- *don't send:* the message is not sent.
- *send if needed:* only send if no other message has been sent to the ECU for the period of "Tester present cycle".
- *always send:* The message is always set at the rate of the "Tester present cycle", without regard to other messages.
- *send at the start and always*: At the beginning there is no delay by the period "Tester present cycle"; instead the message is sent immediately.
- *Standard-timeout:* Time span in seconds within which the ECU must respond.

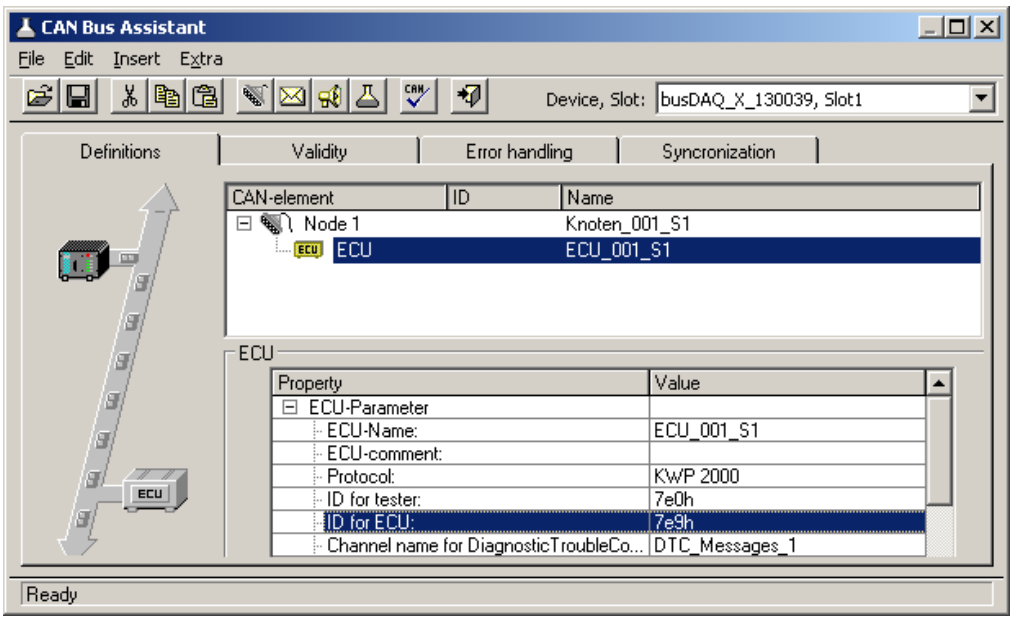
- *Expanded timeout:* Time span in seconds within which the ECU must conclude any routines started, which could not be concluded within the Standard-Timeout period.
- *Tester present cycle:* Cycle at which the measurement device sends a life-sign message to the ECU.
- *StartDiagnosticSession:* The second byte determines the "Diagnostic Session". This property is specific to the ECU.
- *LogInKey:* For cases where "*With security access?*" was set to Yes, the key for the Seed/Key procedure is entered here.
- *ReadDiagnosticTroubleCodesByStatus:* The second byte is the status queried. The third and fourth bytes are the High Byte and Low Byte of the queried DTCs group.
- *ClearDiagnosticInformation, ReadDiagnosticTroubleCodes, ReadStatusOfDiagnosticTroubleCodes:* The second and third bytes are the High Byte and Low Byte of the DTCs group.
- *P3 time out*

Special imc Online FAMOS functions for ECU's:

Function	Description
ECUStartSession_*	Performs the service StartDiagnosticSession.
ECUStopSession_*	Performs the service StopDiagnosticSession.
ECUReadTroubleCodes_*	The list of DTCs is imported and entered in the channel for DTCs (e.g. DTC_Messages_1)
ECUSend_*	Exports to ECU objects or starts routines

Sample TCU with KWP2000 protocol (on CAN)

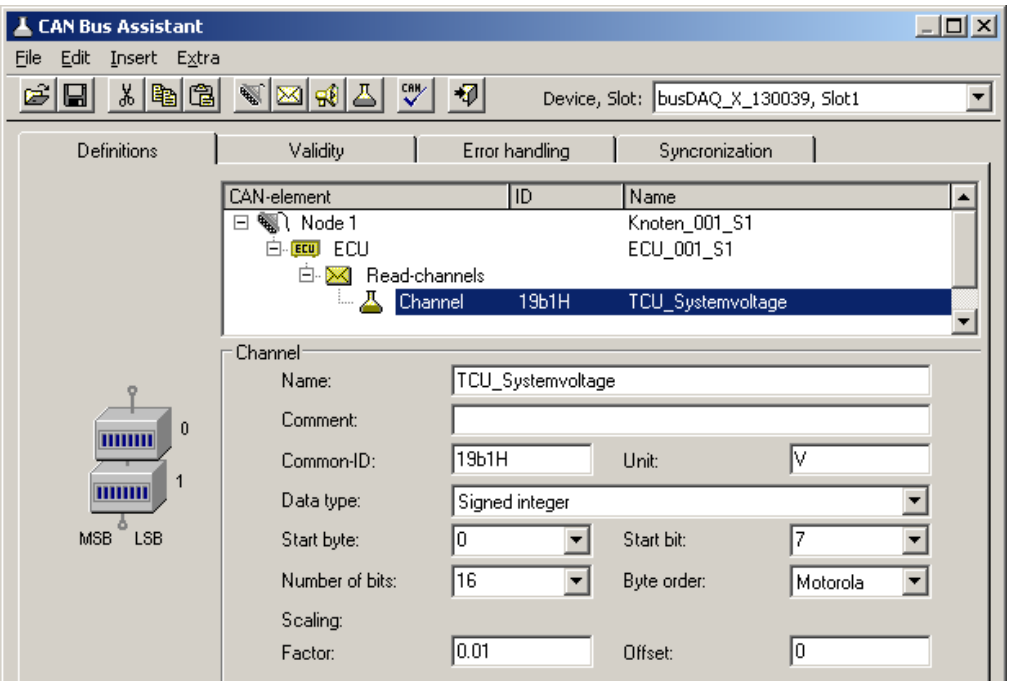
Once a new ECU has been added, the protocol is selected and the ID configured.



Set the identifier of the control unit

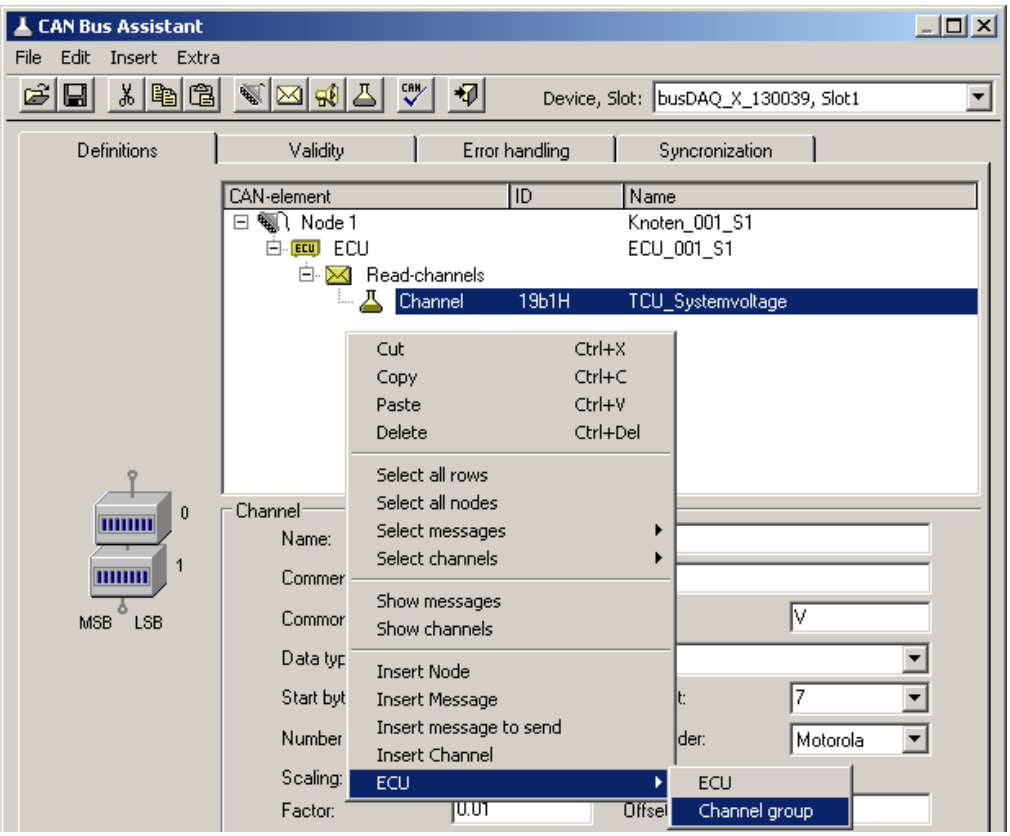
The value for StartDiagosticSession is left at the default setting.

Now the first channel is configured with the name "TCU_Systemvoltage" and with the Common-ID 0x19C1. This is an unsigned integer with 16 bits and a factor of 0.01.

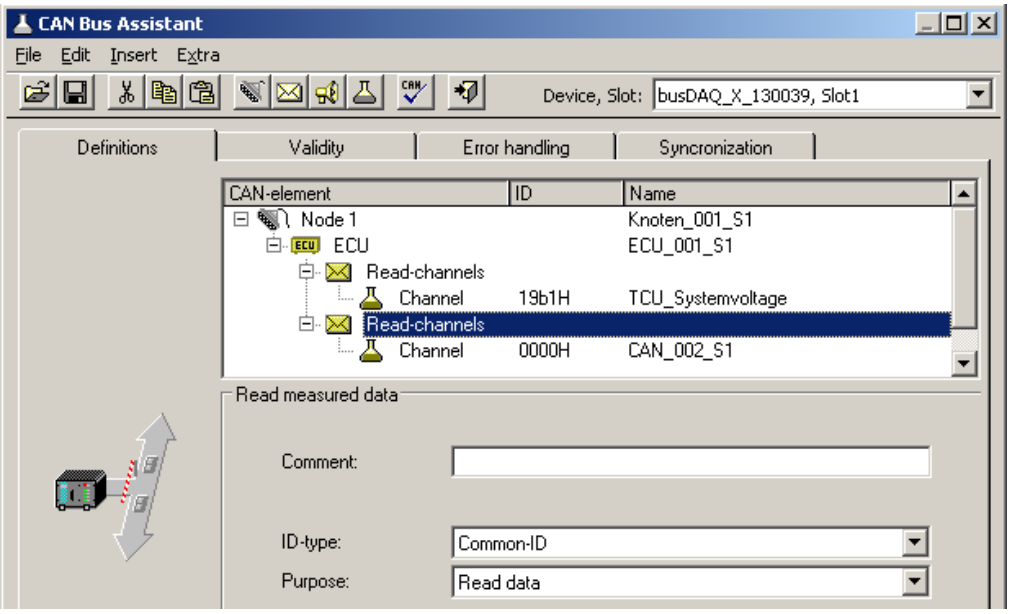


Configuring the channel

In order to be able to start an ECU routine from imc Online FAMOS, the routine must be configured in the CAN-Bus Assistant. For this purpose, a new channel group is first created.

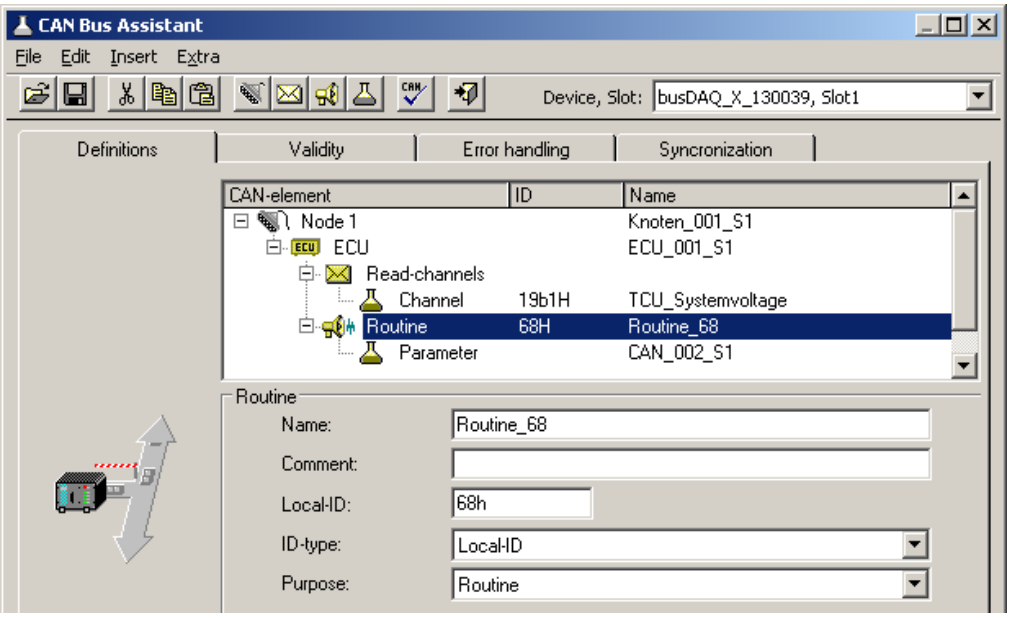


Create a channel goup



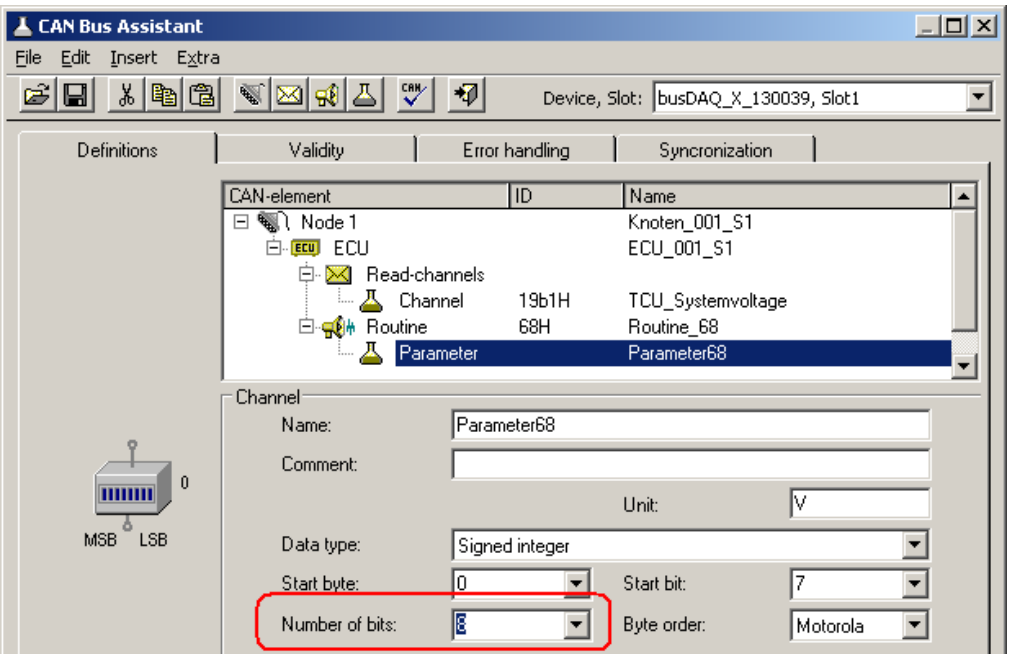
Channels of the channel group

The new channel group s configured as the routine's *Purpose* with a "Local Identifier" of 68h.



Setting the local identifier

Next, the parameter is given a name and configured for 8 bits.



Set paramter to 8 bit

This concludes configuration of the CAN-Bus Assistant.

imc Online FAMOS is now configured so that when the virtual bit 1 is set, the ECU's Routine_68 starts with Parameter 9. If the routine concludes successfully, the list of DTCs is imported.

```
; Initialization before the first measurement
OnInitAll
End

; continuous running
OnAlways
End

; run at start of measurement
OnTriggerStart(BaseTrigger)
End
```

```
; run during measurement
OnTriggerMeasure(BaseTrigger)
    if Virt_Bit01 = 1
        Virt_Bit01 = 0
        ECUSend_Routine68( 9 )
    End
End

; run at end of measurement
OnTriggerEnd(BaseTrigger)
End

; run in response to command
OnECUCmdReturn_ECU_001( OECR_Return, OECR_ECUCmd, OECR_CmdID )
    If OECR_Return = 0 AND OECR_CmdID = 0x68
        ECUReadTroubleCodes_ECU_001( )
    End
End
```

Note

- If the values of the channels to be captured are queried one-by-one from the ECU, then the sampling interval configured for the channel is used. As for other CAN channels as well, the sampling interval can be set to values from 100µs on. ECUs normally require longer to respond (typ. 30 ms). For this reason, configuring a shorter sampling interval than the ECU's capabilities is not sensible.
- An ECU normally only executes one command at a time. If ECU values are captured, it is not possible to query them if any ECU function of long duration has been started. Therefore, the channels displaying the ECU's values can not receive any new values for the duration of the function. Depending on the configuration, they either display the last known value or a dummy value.
- In KWP2000 (on CAN), if a LogInKey is specified, the Key is also sent in response to Seed 00h 00h if the ECU is already enabled.

9.10.4.8.3.1 KWP2000 for TP2.0

Unless otherwise specified, the same descriptions of the properties apply as appearing under KWP2000.

source-address: The source address for the dynamic channel structure. Default: zero; for the tester.

target-address: The target address for the dynamic channel structure. Default: one; for the motor ECU.

Dynamic setup?: States whether to have dynamic channel structure, or whether to use permanent Identifiers instead.

9.10.4.8.3.2 Diagnostic On CAN

For *Diagnostic On CAN*, the same notes apply as for KWP2000, with the following differences:

- The service *ReadFreezeFrameData* is supported.
- The service *DynamicallyDefineLocalID* is supported.
- The service *RequestRoutineResultsByLocalId* is not supported.

DynamicallyDefineLocalID

- For a standards-conformant ECU, the properties for *Dynamically defined lists* can be set as follows:
- *Length of definition message:* -1 for automatic uses the maximum possible length. If the length is limited by the ECU's requirements, make this setting.
- *Defined data length*: -1, for automatic, uses the maximum possible block length. If the block length is limited by the ECU's requirements, make this setting.
- *Nested lists?:* If the lists cannot be used independently of each other, then set *Yes here*. Then, only one list will be queried during import. However, it is possible to use the previous list for definition purposes.

- *With Local- and Common-ID?:* If all Identifier types can be used for definition purposes, set "*Yes*" here. "*No*" is to be set if only addresses can be used.
- *Delete dynamic lists?:* If the dynamic lists need to be deleted before a new definition, then set "*Yes*" here.

Special imc Online FAMOS functions:

Function	Description
ECUStartSession_*	Runs the service StartDiagnosticSession.
ECUReadStatusOfDTC_*	The list of DTCs is imported by Service 17h and adopted in the channel for DTCs (e.g. DTC_Messages_1)
ECUClearDiagInformation_*	Deletes the list of DTCs.
ECUReadFreezeFrameData_*	The data of a particular FreezeFrame-number associated with a 'Diagnostic Trouble Code' are imported.
ECUSend_*	Writes to ECU objects or starts routines

Note

For the DiagOnCAN-protocol 3-byte- and 4-byte-addresses with one or two length bytes are supported.

9.10.4.8.4 OBD-2

OBD-2 enables reading of standardized ECU channels. The ECU protocol OBD-2 is implemented in accordance with the SAE J1979DA:2021-04-21.

In imc Online FAMOS, there is for OBD-2 the function `ECUClearDiagInformation_*`, which deletes diagnostics information saved in the control unit.

Note

Operation requires skills in the OBD-2 protocol.

General notes

On**B**oard**D**iagnostik, abbreviated OBD-2, is a standardized control mechanism enabling quick access to a vehicle's control instruments and sensor data. This interface is legally required in new vehicles.

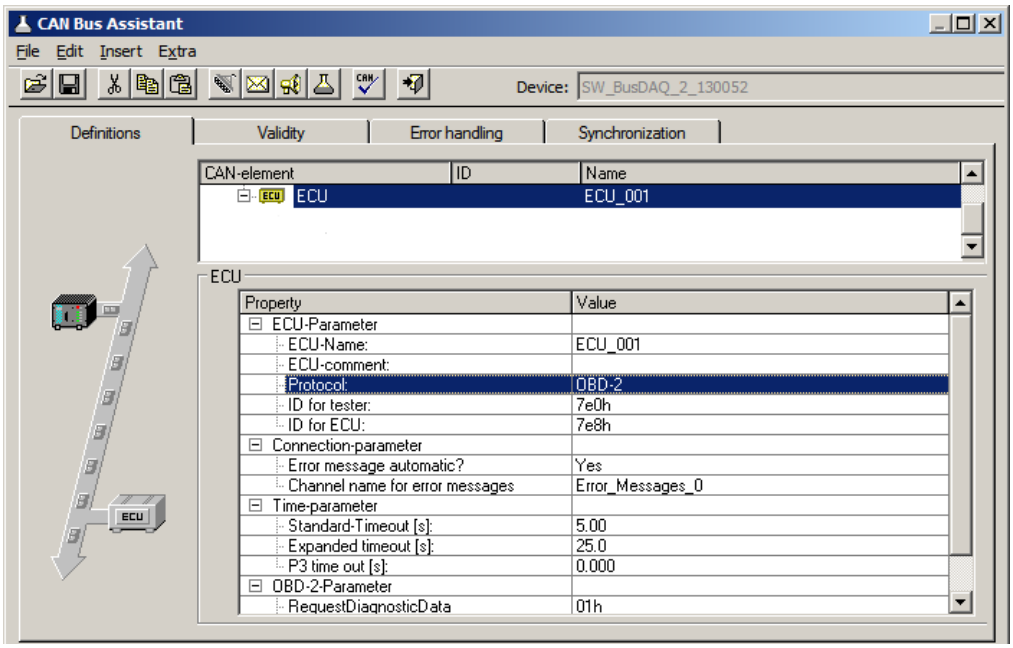
OnBoardDiagnostik monitors and saves even errors occurring during vehicle operation. Certain sensor data are recorded at fault condition.

These so-called freeze frames make quite effective troubleshooting and diagnostics possible. Necessary information must be supplied be the vehicle manufacturer. By this means, the vehicle data can be read out and used for diagnostics purposes.

The signal connection terminal is located in the passenger compartment, never in the hood or trunk.

9.10.4.8.4.1 Settings in the assistant

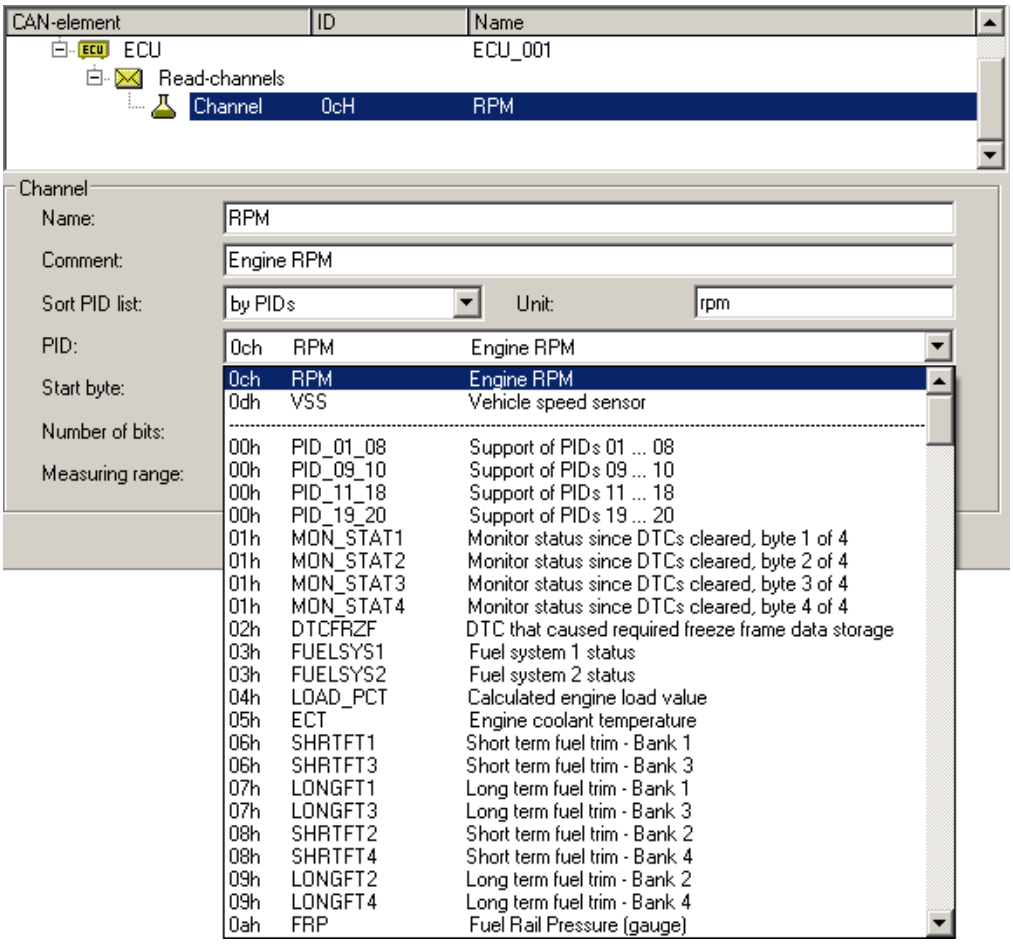
To select OBD-2 channels, add a ECU 585 to the configuration. Under ECU-Parameters, set the protocol type to OBD-2, before adding any channel:



ECU-protocol: OBD-2

With this type, there are only a few ECU parameters to set, e.g. there is no log-in command, no Security and no dynamic lists.

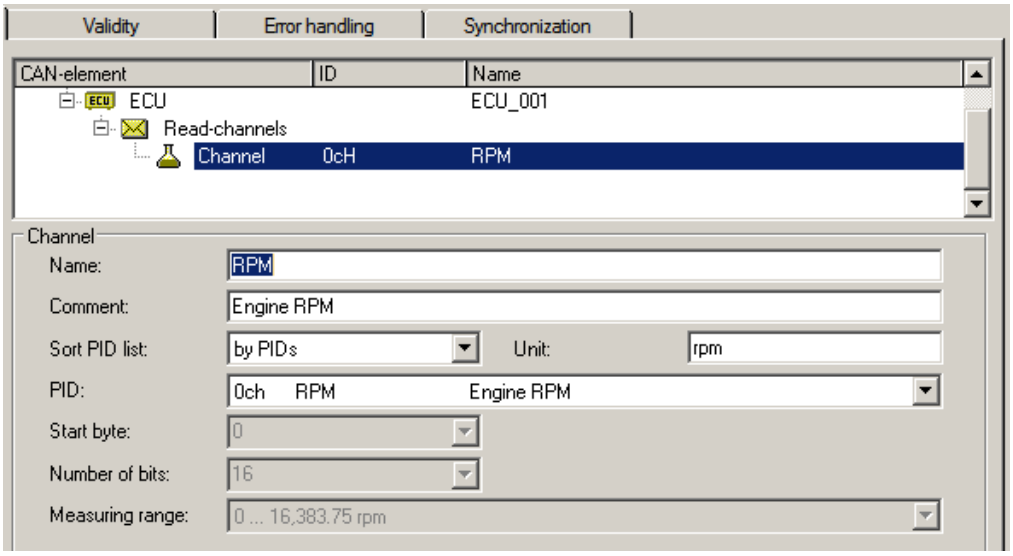
For selection of the ECU channels, permanent PIDs are set. Create a message and a channel and select the channel via the PID list:



Select a OBD-2 channel

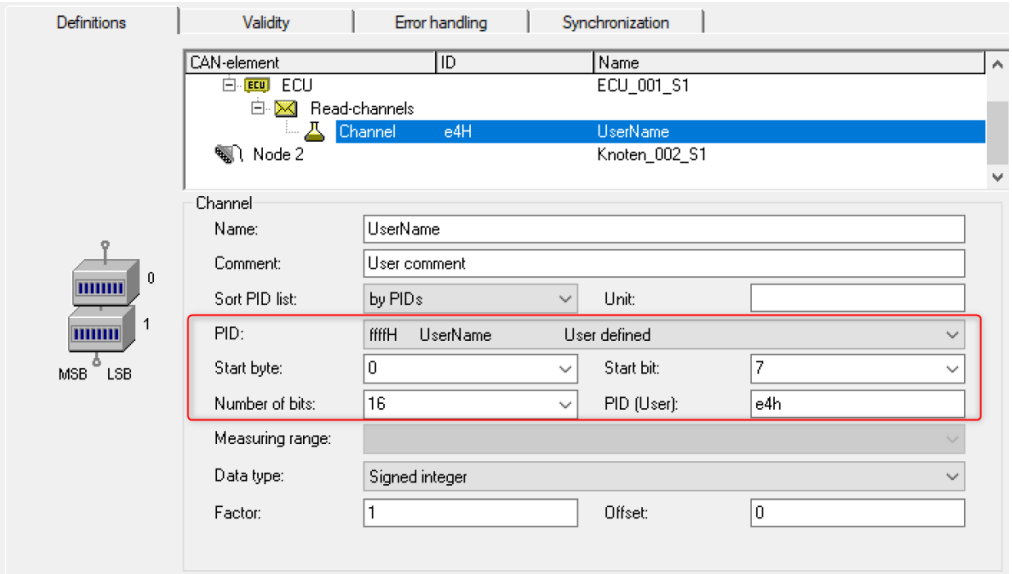
The PID can also be sorted by channel names, channel description or PID.

After selecting a PID, the *Name, Scaling, Unit, Comment, Start byte, Number of bits* and *Measuring range* for this PID are set automatically.



Properties of a OBD-2 channel

The last selection in the PID list is the entry "*fffH UserName User defined*". This selection can be used to add further PIDs that are not listed as standard. Another input field "*PID (User)*" appears here, in which the PID can be freely selected.



user defined PIDs

9.10.4.8.5 CCP

For this protocol, you must usually configure the following:

- *CRO-ID (ID for Tester):* With this Identifier, messages are sent to the ECU.
- *DTO-ID (ID for ECU):* Response messages from the ECU are expected to have this Identifier.
- *Station address:* The station address serves to distinguish the ECUs in cases where multiple ECUs are using the same Identifier.
- Seed-and-Key settings, see here 587.
- *DAQ-lists:* If the ECU supports DAQ-lists, it is possible to use DAQ-lists to relieve the burdens on the ECU and the CAN-Bus.

Additional parameters

- *DAQ-lists only with bytes:* It is necessary to enter "Yes", if only single bytes can be defined in the DAQ-lists. In that case, two entries must be used for a 16-bit wide value.
- *Byte order:* Which sequence of bytes is to be used by default.
- *Mode channel name:* Unique name for the additional channel required for the use of DAQ-lists.
- *With SHORT_UP?:* If the ECU understands the command SHORT_UP, it is an easy way to query data.
- *With START_STOP_ALL?:* If the ECU understands the command START_STOP_ALL, then all DAQ-lists can be started together.
- *ECUs use same IDs?:* If multiple ECUs use the same Identifier, then each time the ECU is accessed, a connection having a specified station address, and subsequent disconnection, are necessary.
- *Standard-Timeout:* Time duration in seconds within which the ECU must respond.
- *LogInKey for send channels:* For the case that "*With security access*?" is set to *Yes*, the key for enabling write access is entered here.
- *LogInKey for DAQ-lists:* For the case that "*With security access*?" is set to *Yes*, the key for enabling the DAQ-lists is entered here.

- *Events:* In order to be able to use DAQ-lists, it is first necessary to define the associated events. An ECU event is a cyclical prompt to send specific DAQ-lists. An event can be triggered by timer or by an RPM-value, for instance, of the crankshaft. imc STUDIO can only process time-domain events.
- *Properties of event:* Name: for designating an event
- *ID:* The identification used by the ECU. Usually begins with zero and ends with the number of events minus 1. For an entry which isn't used, enter -1.
- *base clock:* The minimum time period between two consecutive events.
- *Connection set-up automatic?*: Yes or No

DAQ-lists

In order to be able to use DAQ-lists, it is necessary to define the ECU's 'Events'. These are the sampling intervals available for the ECU. An 'Event' is defined with an arbitrary name, the Event-ID and the event's base clock rate.

To define a DAQ-list, the following must be specified:

- *Number of list:* The identification used by the ECU. Normally begins with zero and ends with the number of DAQ-lists minus 1. For an entry which isn't used, enter -1.
- *Event:* Specifies which event can trigger the DAQ-list. *Variable* is the setting in which the DAQ-list can be triggered by any event. The name of a particular event must be specified if the DAQ-list is to be triggered by only that event.
- *CAN-ID:* Identifier with which messages can be sent; -1 for the DTO-ID. If the DAQ-list can use other Identifiers instead of the DTO-ID, it is possible to set an unused Identifier. For many DAQ-lists, a different, permanent Identifier must be set.
- *MAX_ODT:* The maximum number of tables (Object Descriptor Table) per DAQ-list.
- *MAX_ODT_ENTRY:* The number of possible entries in a table (ODT).
- *FirstPID:* The PID of this DAQ-list's first ODT. The PID is used to identify the associated ODT. For each of the subsequent ODTs, a 1 is added. -1, for automatic, uses 0 for the first DAQ-list, MAX_ODT of the first DAQ-list for the second DAQ-list, etc.
- *With reduction:* Is *Yes*, if the basic clock rate needs to be subdivided. Then, messages are only sent for every n events. An ECU's channels are arranged in the DAQ-lists according to their sampling intervals. The channels with the shortest sampling interval are used first. If the DAQ-lists aren't sufficient for all channels, then the remaining channels are polled (read directly at every sampling instant).

Special imc Online FAMOS functions:

Function	Description
ECUSend_*	writes to ECU objects

Example

1. There are no DAQ lists ore they are not to be used.

The parameter number is set to -1 for all DAQ-lists.

2. A DAQ-list

```
Number:  0
Event:  Variable
CAN-ID:  -1
MAX_ODT: 10
MAX_ODT_ENTRY: 7
First PID: -1
With reduction? No
```

3. Two DAQ-lists each having a fixed event, special "First PID" and different size

```
Number:  1
Event:  0
CAN-ID:  -1
MAX_ODT: 12
MAX_ODT_ENTRY: 7
First PID: 0
With reduction? No

Number:  0
Event:  1
CAN-ID:  -1
MAX_ODT: 8
MAX_ODT_ENTRY: 7
First PID: 16
With reduction? No
```

9.10.4.8.6 XCP

For XCP, the same notes apply as for CCP, with the following differences:

- XCP does not have station addresses
- Seed-and-Key settings, see here 587.
- *Dynamic DAQ-lists:* If DAQ-lists can be created as needed by the ECU, enter "*Yes*" here.

Additional parameters

- *Alignment in bytes:* Which alignment to adhere to in defining the ODTs.
- Messages with XCP standard are supported (short format: only the necessary bytes will be sent). Also messages are supported that are extended to 8 bytes.

9.10.4.8.7 GMLAN

The ECU protocol **GMLAN** resembles KWP2000. Import of A2L-files is possible.

The GMLAN-protocol has been implemented according to GMW 3110 Version 1.5 specifications (04.02.2004).

For dynamic lists, the following commands are additionally supported:

DynamicallyDefineMessage: assignment of a PID-list to a DPID

DefinePIDByAddress: assignment of unique PIDs to the (channel) addresses

ReadDataByPacketIdentifier: definition of DPID-list

With dynamic lists, the parameters in the group *'Dynamically defined lists*' must be set appropriately.

ModeChannelName: For dynamic lists, an additional channel is required (for internal purposes, such as with CCP).

Addressbytes: 2-, 3- and 4-Byte addresses are supported.

Supported imc Online FAMOS functions:

*ECUStartSession_**

*ECUStopSession_**

9.10.4.8.8 UDS

The UDS protocol is similar to the KWP2000 protocol and is implemented according to the ISO 14229-1 standard (04.2007).

- Seed-and-Key settings, see here 587.

Additionally supported commands for dynamic lists:

DynamicallyDefineDataID: Assignment of addresses (channels) to unique PIDs

ReadDataByPeriodicIdentifier: Assignment of a PID-list to one of the specified sampling rates

With dynamic lists, it is necessary to set the parameters in the group *'Dynamically defined lists'* appropriately. The dynamic lists serve the purpose of transmitting multiple channels on the bus as quickly as possible while loading the bus as little as possible.

ModeChannelName: For dynamic lists, an additional channel is needed for internal purposes.

AddressBytes: 2-, 3- and 4-Byte addresses are supported

Functions supported in imc Online FAMOS:

ECUStartSession_* : Executes the service StartDiagnosticSession.

ECUStopSession_* : Executes the service StopDiagnosticSession.

ECUClearDiagInformation_* : The list of diagnostics information is deleted

ECUSend_* : Writes to ECU objects or starts routines

9.10.4.8.9 A2L: CURVE and VAL_BLK

Characteristic curves of the types *CURVE* and *VAL_BLK* of an A2L-file can be imported for the ECU-protocols CCP and XCP.

The option *Replace characteristic curves of the type CURVE and VAL_BLK with type VALUE* (see the menu item *Extra > Options* 557), all characteristic curves of both these types are replaced with characteristic curves of the type *VALUE*. This generates as many characteristic curves of this type as there were values in the original characteristic curve. For each selected characteristic curve of the type VALUE, one write-function is generated in imc Online FAMOS.

Without this option, for each characteristic curve of the type *CURVE* or *VAL_BLK*, a channel with the additional dialog element *Vector length:* is created, which can only be exported. For each characteristic curve, one write-function is automatically generated in imc Online FAMOS, in which the vector index must be supplied when it is called. Thus, there is one write-function imc Online FAMOS per characteristic curve.

Example: imc Online FAMOS:

```
; run during the measurement
OnTriggerMeasure(BaseTrigger)
     if IsV02 > 0
         ECUSend_CCP_struct_gen_vector_02( 54.5, 1) ; Index: 1,2
         ECUSend_CCP_struct_gen_vector_02( 58.5, 2)
         IsV02 = 0
     end
end
```

9.10.4.8.10 Properties in common

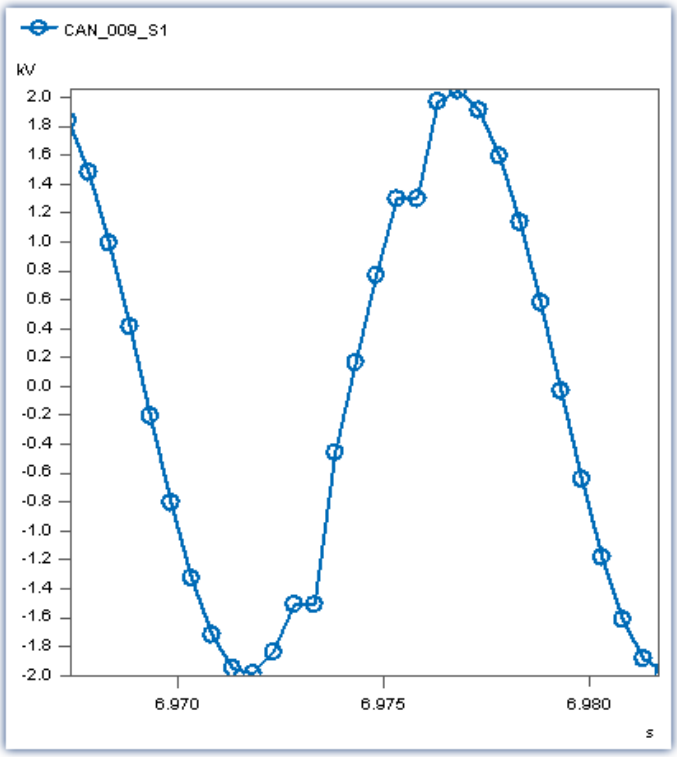
Property	Description
ECU-name, ECU-comment	An arbitrarily specifiable designation for the ECU, and a comment on the same. If the ECU's settings are imported from an A2L-file, then the ECU-name should not be changed. Upon re-importing a changed A2L-file, it is possible to adopt either only the changes, or also additional channels.
Protocol	This property should be set first, since the protocol used determines the particular properties. The protocol to be used in communicating with the ECU is to be chosen. As soon as a channel group has been created for the ECU, it is no longer possible to change the protocol.
Error messages automatically?	If the setting is "*Yes*", any error messages are automatically outputted in the error message channel. With "*No*", it is possible to evaluate errors occurring in the section "*OnECUCmdReturn*" in imc Online FAMOS.
Connection after error?	For the setting "*Yes*", the system attempts to set up a new connection whenever protocol timeouts occurred or messages containing measured data were delayed too long.
Channel name for error messages	Unique name for the error message channel.
Protocol-parameter	In the protocol-parameters, the command codes for the selected protocol are defined. These settings can mostly be left intact. Settings which need to be edited or checked are explained in conjunction with the protocol-specific properties.

9.10.4.9 Information and tips

Question: Why is the CAN-TimeStampASCII history not displayed completely?

Answer: The pretrigger for CAN-TimeStampASCII is never greater than 512Bytes. Even if a longer pretrigger time is set, a maximum of 512Bytes history is shown.

Question: At high sampling rates with CAN-Bus modules, some indicated values are duplicated or, in rarer cases, incorrect.



Answer: The CAN-Bus modules feature a considerable amount of jitter in outputting the measured data due to factors affecting the overall system. If the data are recorded at a high sampling rate, some values may be duplicated or, in rarer cases, erroneous data may appear.

Solution:

a) **Universal**:
Sample the CAN-Bus channel with a time stamp instead of equidistantly.

b) **When using imc STUDIO**:
Use the CAN-1 synchronization. This is absolutely necessary if subsequent data processing is performed with in imc Online FAMOS.

9.10.5 EtherCAT Slave Interface

The fieldbus module EtherCAT Slave Interface (imc ECAT-Slave-IF) provides the integration of imc measurement devices and fieldbus loggers into an automation system based on an EtherCAT fieldbus.

The complete imc CRONOS device is integrated via the interface into the EtherCAT fieldbus as a Slave module, which is operated by an external EtherCAT Master.

Data from the imc measurement device are thus available throughout an EtherCAT system and integration of the device into other system environments is achieved. This interface is not limited to readout of data and variables from the imc system. Moreover, variables and system parameters of the imc system (Slave) can also be set and manipulated by the external EtherCAT Master, which allows for close interaction and further control schemes.

Note

The **EtherCAT-Slave-IF** is **not** to be mistaken with the ***imc* CRONOS*flex*-base unit** (the ***imc EtherCAT-Master***), which allows connection of imc CRONOSflex 343 modules. imc CRONOS*flex* is a distributed measurement system, in which the EtherCAT Fieldbus is used as a transfer medium between imc amplifiers and the imc basis device. In such applications, the connected EtherCAT-bus is to be viewed as an internal imc bus, to which only EtherCAT modules are connected.

Software and operation:

For full functionality (especially pv and sending functions) imc Online FAMOS Professional OFA Pro (additional license) is required.

- As the signal source, the imc device's process vector is used. This means that the measurement channels and virtual channels represented by process vector variables are available to external subscribers and systems via EtherCAT. The data are available and valid directly after preparation, independently of trigger releases.
- Via the interface, it is possible to read and write process vector variables (pv-variables) of the imc system. Reading and writing can occur either cyclically or irregularly.
  - Cyclic data are transmitted periodically after the fieldbus and the imc measurement system start.
  - Irregular data can be queried asynchronous by the EtherCAT Master by means of the protocol CoE (CANopen over EtherCAT), which is supported by the imc system.
- The imc measurement device is parameterized via the device software imc STUDIO. The configuration is transferred to the EtherCAT master via an "EtherCAT-Slave-Information" document (ESI) in XML format.

9.10.5.1 EtherCAT general

EtherCAT (**Ether**net for **C**ontrol **A**utomation **T**echnology)

EtherCAT is an Ethernet-based Fieldbus. EtherCAT is distinguished by high performance and real-time capability and thus enables open-and closed-loop control mechanisms which classical Fieldbus systems could not achieve. EtherCAT uses the principle of a Master-Slave communication system. The Master in this case is a regular commercially available PC with a network card. The Slaves use special EtherCAT-hardware as the interface which enables data exchange on the hardware level. This saves considerable time in processing the protocol by means of an interface program.

CoE (**C**ANopen **o**ver **E**therCAT)

To map the process data and to set up a communication channel to each bus subscriber, EtherCAT uses the communication profile CANopen (*Controler Area Network*). By the use of the communication channel, the cyclical real-time operation is configured whenever necessary, and the acyclical transfer of data is enabled. The basis for communication by means of the CANopen protocol is the CANopen object directory, which exists on every bus subscriber. Each object in the object directory is uniquely defined by means of a so-called CANopen object index (UINT16 value). In the EtherCAT standard, use of the CANopen standard is referred to as *CANopen over EtherCAT* (CoE*).* If communication by CoE is not supported by a Slave, the data mapping is saved in an ESI document.

Data map

In the EtherCAT protocol, data exchange between the EtherCAT-Master and EtherCAT-Slave takes place via so-called Sync-Managers. A Sync-Manager defines a storage area of specific length within the EtherCAT Slave hardware, from which the Master can read or write data cyclically. To map the structure and size of a Sync-Manager's process data to be transferred, an EtherCAT-Slave's CANopen object directory contains SyncM objects (SM-objects), which assign process data objects (PDOs) to the individual SyncManagers. Together the SM-objects and PDOs produce the process data mapping.

A SyncManager object has the structure of a *UINT16 array*. Every entry in the *SM-object* corresponds to a PDO-index. A PDO has the structure of a *UINT32 array*. Each PDO entry references a subobject of an object in the object directory. The entry comprises the object's index, the subobject's subindex and its bit length.

Example: Sync Manger3 (0x1c13) , 2 PDOs (0x1A00;0x1A01) , 2 data objects (0x2000 , 0x2001) subobjects 1-3 data length 16 bits

```
<0x1c13>
        < 0x1A00>
                <0x20000110>(0x2000;1,16)
                <0x20000210>(0x2000;2,16)
                <0x20000310>(0x2000;3,16)
        < 0x1A00/>
        < 0x1A01>
                <0x20000110>(0x2000;1,16)
                <0x20000210>(0x2000;2,16)
                <0x20000310>(0x2000;3,16)
        < 0x1A01/>
<0x1c13/>
```

9.10.5.2 Prerequisites

For operation of imcECAT-IF the module **CRC/ECAT-Slave** is required.

The module is supported by imc CRONOS*compac*t, imc CRONOS-XT, imc CRONOSflex or imc ECAT-GATEWAY.

9.10.5.3 Configuring the EtherCAT interface

Configuration of the Fieldbus module is performed with the help of the imcECAT-IF Assistant which comes with the device software. By means of this Assistant, the user sets which PVVs on the imc device are mapped in the imcECAT-IF's CANopen object directory. Additionally, the Assistant offers the ability to define the data mapping (CoE-mapping) of the cyclical data transfer.

The following two configuration variants are possible:

Variant 1:

No complete mapping is defined. The EtherCAT Master performs the mapping via CoE.

Configuration work performed with the Assistant:

- Definition of PVVs
- Definition of CoE process data objects (PDOs)

Configuration work for the EtherCAT-Master (via CoE):

- Definition of the PDO-contents
- Definition of the Sync-Manager contents (see CoE)

Variant 2:

Complete mapping is defined (no CoE communication necessary).

Configuration work performed with the Assistant:

- Definition of PVVs
- Definition of CoE process data objects (PDOs)
- Definition of PDO contents
- Definition of Sync-Manager contents (see CoE)

The data mapping created in the Assistant can be imported by the Master via CoE, or be exported to the Master by means of an ESI document supplemented with the mapping. The Assistant can be used to create this device-specific ESI-document. The assignment of the device-specific ESI-document to the corresponding imcECAT-Slave-IF can be determined by the EtherCAT-Master on the basis of the device's VendorID, ProductCode and **serial number**.

9.10.5.4 CANopen object directory

The *CANopen* objects contained in the object directory can be subdivided into two function groups. To the first group, objects are assigned whose function, CANopen-index and name are permanently determined by the CANopen-standard or EtherCAT-standard. To the second group, all *imc*-specific objects are assigned. To which group an object belongs can be found out on the basis of the CANopen-index. All objects whose index lies in the range 0x2000 …..0x6000 belong to the group of *imc*-specific CANopen objects.

The main part of the object directory consists of objects which describe defined PVVs in the configuration. A channel is completely described by one data object and one parameter object. The association of an "object pair" is determined by an object's name. An *imc* measurement channel's data object consists of at most 1 sub objects, where a subobject maps the instantaneous value of the PVV. An *imc* parameter object consists of multiple subobjects. The individual subobjects contain the properties of the PVV, such as the sampling interval, scaling etc. Each subobject can be imported via CoE. Parameter objects can not be assigned ("mapped") to a PDO.

Object directory contents

Index	Name	Access	Amount	Function
0x1000	device type	RO	1	specific to device type
0x1008	device name	RO	1	device name
0x1009	Hardware version	RO	1	indicates the device's hardware version
0x100A	Software version	RO		indicates the currently operable software version in the device
0x1018	Identity	RO	1	object indicates the vendor ID, product code, revision number and serial number
0x1600 0x1601 ... 0x17ff	RxPDO	RW	depends on configuration	defines process data received by EtherCAT-Slave
0x1A00 0x1A01 ... 0x1BFF	TxPDO	RW	depends on configuration	defines process data sent by EtherCAT-Slave
0x1C00	SM-Comm Type	RW	1	Defines the communication type of the individual Sync-Managers, where each subindex stands for one SyncManager. Subindex 1 = SM 0 Subindex 2 = SM 3 etc.
0x1C12	SM2-PDO-assign	RW	1	contains the index of the PDOs assigned to the Sync-Managers
0x1C13	SM3-PDO-assign	RW	1	contains the index of the PDOs assigned to the Sync-Managers
0x1C32	SM2-Parameter	RW	1	contains the Sync-Manger's transfer properties
0x1C33	SM3-Parameter	RW	1	contains the Sync-Manger's transfer properties
0x2000 0x2001 ... Ox5ffff	pv_"name"_DATA pv_"name"_DATA ... pv_"name"_DATA	RW	depends on device type	maps a PVV's measurement data
0x2000 0x3002 ... Ox5ffff	pv_"name"_PARAMETER pv_"name"_PARAMETER ... pv_"name"_PARAMETER	RW	depends on device type	contains a PVV's properties

9.10.5.5 Parameterobject

An imc parameter with its associated sub-objects represents the properties of a PVV. The object contains the properties of the channel associated with the PVV (e.g.the sampling rate), as well as transmission properties pertaining to the EtherCAT interface. The seven sub-objects in the object have a variety of data types (in accordance with CANopen REC structure).

The parameter object contains the following properties:

Subindex	Description
0	Subobjekt 0
1	Status
2	imc Source
3	CANopen Data Type
4	Factor
5	Offset
6	Unit
7	Sample Rate

Subobject 0

The function of the value in Subobject zero, as well as its data type, has a fixed definition for all CANopen objects in the CANopen standard. The subobject's value specifies how many subobjects are contained in a CANopen object.

Status

The subobject "Status" indicates whether the PVV mapped in the object library is saved in the device configuration of the imc STUDIO Software.

Example

A PVV defined in the imc Online FAMOS source code is mapped in the EtherCAT-Slave-IF's object library. If the PVV definition is deleted due to modification of the imc Online FAMOS source code, the PVV in the EtherCAT-Slave-IF's object library is indicated as "*passive*". If the representation of the deleted PVV is also to be deleted in the object library, the EtherCAT-Slave-IF configuration must be changed.

Data type: UNSIGNED 8

Property representation type: ENUM

Value	Description
1	active
0	passive

imc Source

imc Source contains the information on which resource type the PVV is associated. By means of this information, it is possible to determine, for example, whether a PVV represents an amplifier channel's measurement value.

Data type: UNSIGNED16

Value	Description
6	not definied
7	Amplifier-channel
8	imc Online FAMOS
9	Test variables
10	Fieldbus-channel
11	Device variable
12	Device variable
13	Device variable
14	Test variable
15	Synthesizer
16	CoE Dummy
17	multiple sources

CANopen Data Type

Contains the PVV's data type.The data type is represented in the form of a CANopen index. The data types and the assigned ("associated"?) indices are defined in the CANopen standard.

Data type: UNSIGNED16

Value	Description
0x0003	INTEGER16
0x0004	INTEGER32
0x0006	UNSIGNED16
0x0007	UNSIGNED32
0x0008	REAL32

Factor (only defined for Resource-Type Amplifier channel and fieldbus-channel)

The content of the subobject pertains to the channel associated with the PVV.

Data type: REAL32

Offset (only defined for Resource-Type Amplifier channel and Fieldbus-channel)

The content of the subobject pertains to the channel associated with the PVV.

Data type: REAL32

UNIT (only defined forResource-TypeAmplifier channel and Fieldbus-channel)

The content of the subobject pertains to the channel associated with the PVV.

This contains the unit of the physical variable applied at the channel input/output. The condition for displaying the unit is prior definition of the unit in the imc Device Configuration.

Data type: String

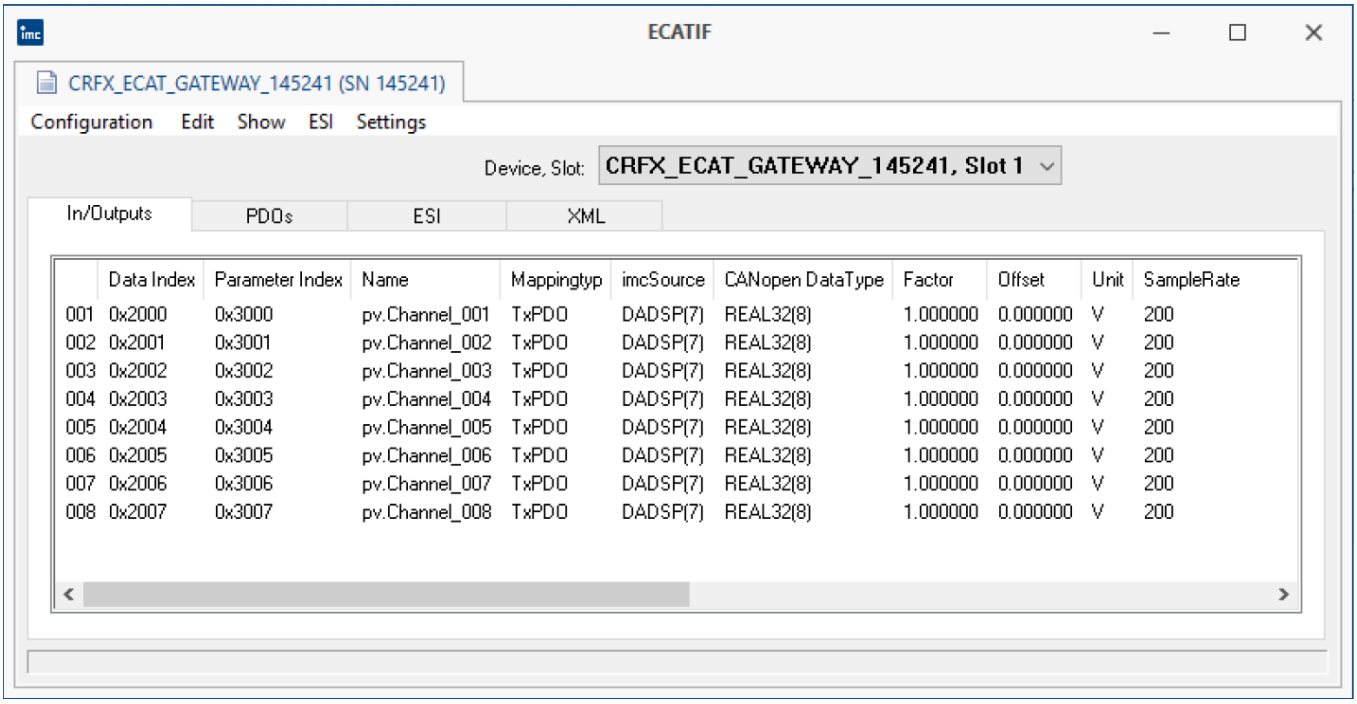
Sample Rate (only defined forResource-TypeAmplifier channel and Fieldbus-channel)

The content of the subobject pertains to the channel associated with the PVV.

Data type: UNSIGNED32

9.10.5.6 EtherCAT-IF assistant

The EtherCAT Assistant uses the experiment set to generate an XML file which makes all process vector variables available as EtherCAT channels.



	Data Index	Parameter Index	Name	Mappingtyp	imcSource	CANopen DataType	Factor	Offset	Unit	SampleRate
001	0x2000	0x3000	pv.Channel_001	TxPDO	DADSP(7)	REAL32(8)	1.000000	0.000000	V	200
002	0x2001	0x3001	pv.Channel_002	TxPDO	DADSP(7)	REAL32(8)	1.000000	0.000000	V	200
003	0x2002	0x3002	pv.Channel_003	TxPDO	DADSP(7)	REAL32(8)	1.000000	0.000000	V	200
004	0x2003	0x3003	pv.Channel_004	TxPDO	DADSP(7)	REAL32(8)	1.000000	0.000000	V	200
005	0x2004	0x3004	pv.Channel_005	TxPDO	DADSP(7)	REAL32(8)	1.000000	0.000000	V	200
006	0x2005	0x3005	pv.Channel_006	TxPDO	DADSP(7)	REAL32(8)	1.000000	0.000000	V	200
007	0x2006	0x3006	pv.Channel_007	TxPDO	DADSP(7)	REAL32(8)	1.000000	0.000000	V	200
008	0x2007	0x3007	pv.Channel_008	TxPDO	DADSP(7)	REAL32(8)	1.000000	0.000000	V	200

EtherCAT-IF assistant

In the first step, pv-variables appear as EtherCAT outputs. It is possible to subsequently remove any EtherCAT channels from the XML file.

Process vector variables created in imc Online FAMOS can be converted to EtherCAT inputs.

9.10.5.6.1 Procedure - configuration variants

Configuration variant I (no mapping)

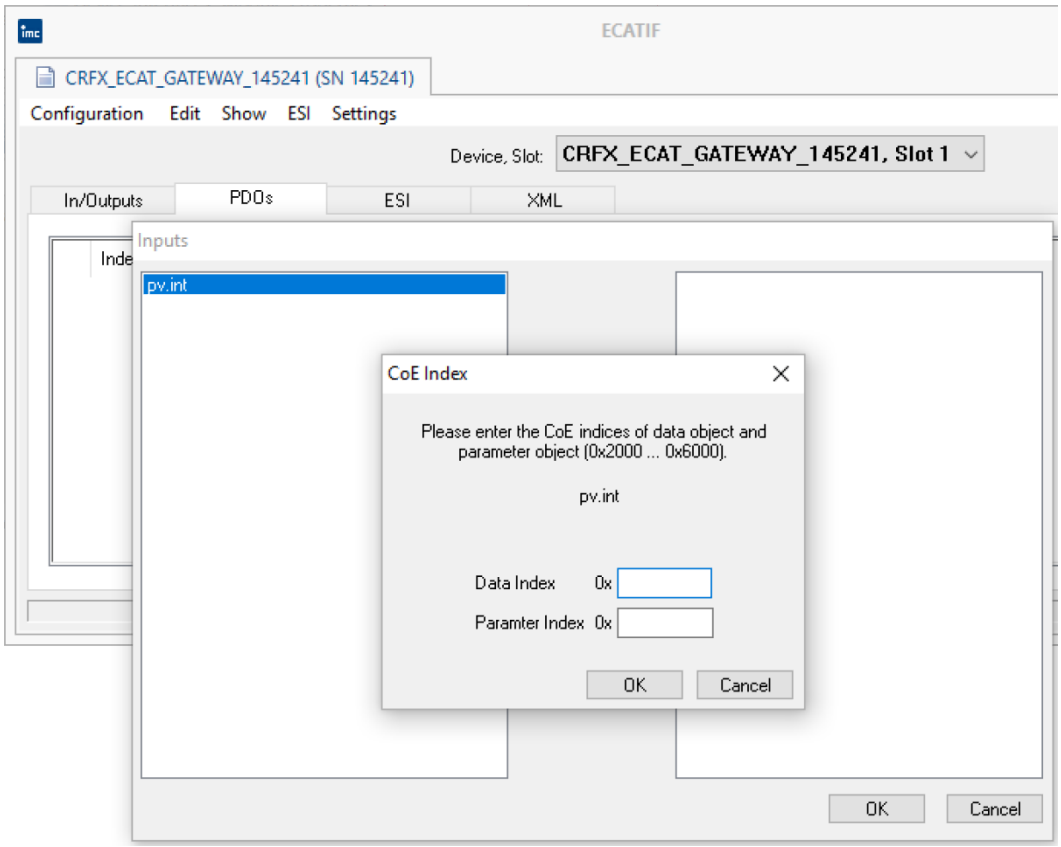
1. Create an experiment with all channel settings.
2. Under Edit > Outputs 615 or Edit > Inputs 615, define which PVVs in the imcECAT-IF's CANopen object directory are to be mapped.
3. Using Edit > Add PDO 615, define the PDOs needed for the mapping. If the PDOs don't correspond to any specific structure, they can be generated automatically using Edit > Create PDOs 615 automatically
4. Save the configuration created by means of Configuration > Accept changes 615.
5. With ESI > Save device specific ESI 616, you can, if desired, save an ESI document containing a map of the CANopen object directory.
6. Close the Assistant. Upon the next preparation, the EtherCAT interface is configured.

Example:

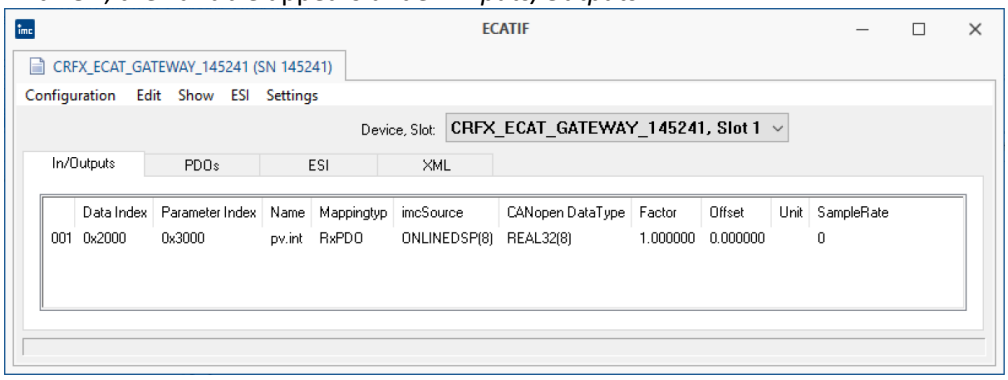
- Define target variable(s) (PV) in imc Online FAMOS (imc Online FAMOS Professional required). Activate "*With control commands*" under "*Extra*" and create the variable:

```
; Executed one single time after the task is started
OnInitAll
    pv.input= 0
End
```

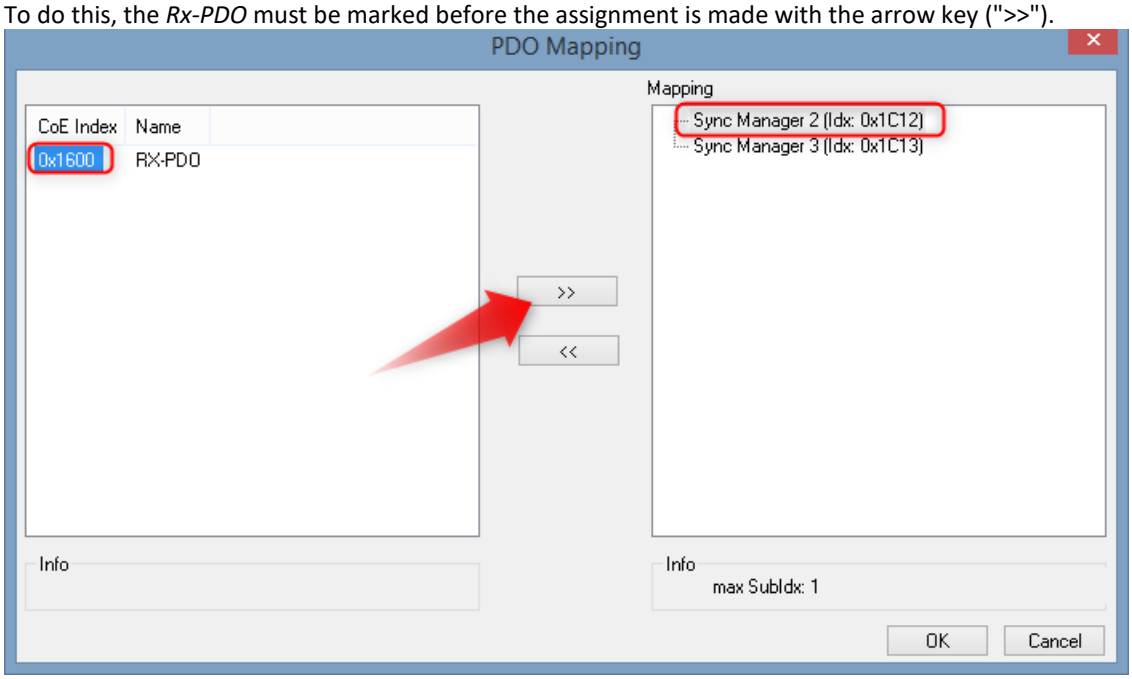
- Open the ECATIF Assistant in imc STUDIO menu *Configuration*. In the menu "*Edit* > *Inputs*" move the pv-variable to the right side (">>"). The information about *Data Index* and *Parameter Index* should be provided by the ECat Master.



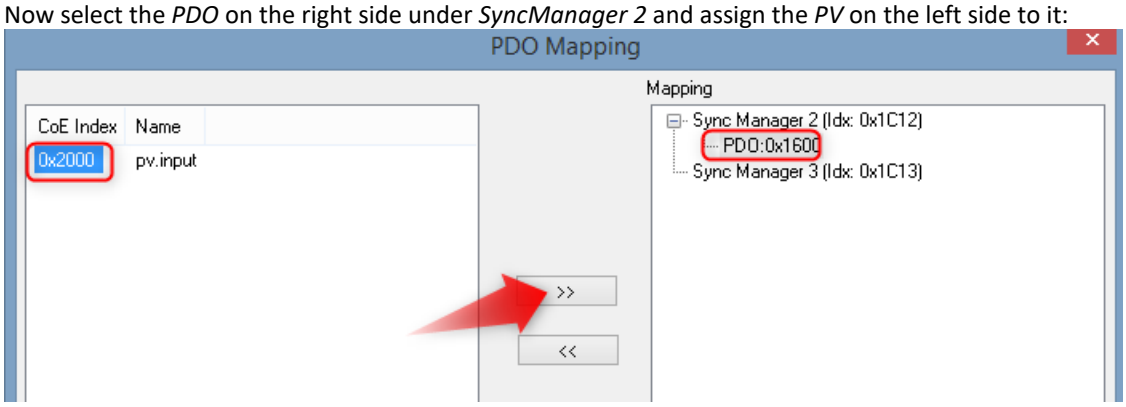
- With OK, the variable appears under "*Inputs/Outputs*":



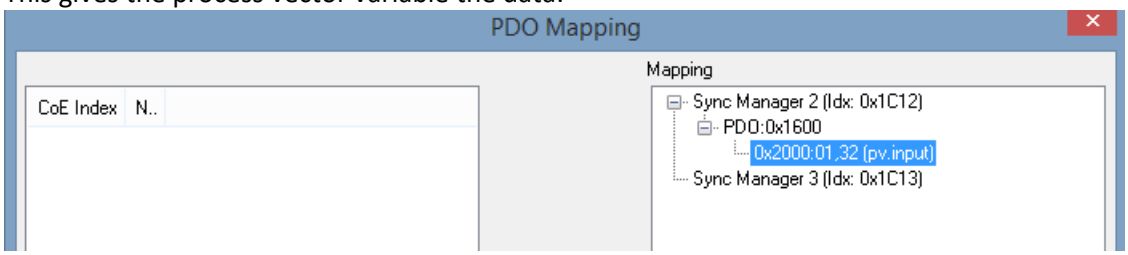
- The PDOs are then generated. Use the "*Edit*" menu to select either automatic ("*Automatic build PDOs*")" or manual ("*Insert PDO*"). In the example, it is selected automatically.
  Select "*Edit*" --> "*Mapping*" in the menu. Click on *SyncManager 2* on the right side and assign the *Rx-PDO* that appears. To do this, the *Rx-PDO* must be marked before the assignment is made with the arrow key (">>").



- Now select the *PDO* on the right side under *SyncManager 2* and assign the *PV* on the left side to it:



- This gives the process vector variable the data:



- With imc Online FAMOS a virtual channel can be created from this variable:

```
; Constantly repeated between the trigger start and the trigger end
OnTriggerMeasure(BaseTrigger_imc_CRFX_2000G_190453)
    vc_ECAT_Input= CreateVChannelInt(Channel_001, pv.input)
End
```

Configuration variant II

1. Create an experiment with all channel settings, imc Online FAMOS etc.
2. Under Edit > Outputs 615 or Edit > Inputs 615, define which PVVs in the imcECAT-IF's CANopen object directory are to be mapped.
3. Using Edit > Add PDO 615, define the PDOs needed for the mapping. If the PDOs don't correspond to any specific structure, they can be generated automatically using Edit > Create PDOs automatically 615.
4. Under Edit > Mapping 615 create the mapping of the cyclical data transfer.
5. Save the configuration created by means of Configuration > Accept changes 615.
6. With ESI > Save device-specific ESI 616, you can, if desired, save an ESI document containing a map of the CANopen object directory.
7. Close the Assistant. Upon the next preparation, the EtherCAT interface is configured.

9.10.5.6.2 Creation and editing of a configuration

The connection between the EtherCAT Interface and data acquisition channels is established by channel names. Configured channels which are subsequently deactivated or renamed remain in the EtherCAT configuration for later use without causing any problems.

Note

Once an EtherCAT configuration has been created, any changes to the experiment are **not** reflected in the EtherCAT configuration. This means that additional channels and channels having new names must be set up in the EtherCAT Assistant in a separate procedure.

Once the configuration set has been prepared for the EtherCAT Interface by means of the menu item New 615, the channels can be set as either outputs or inputs with data and parameter addresses.

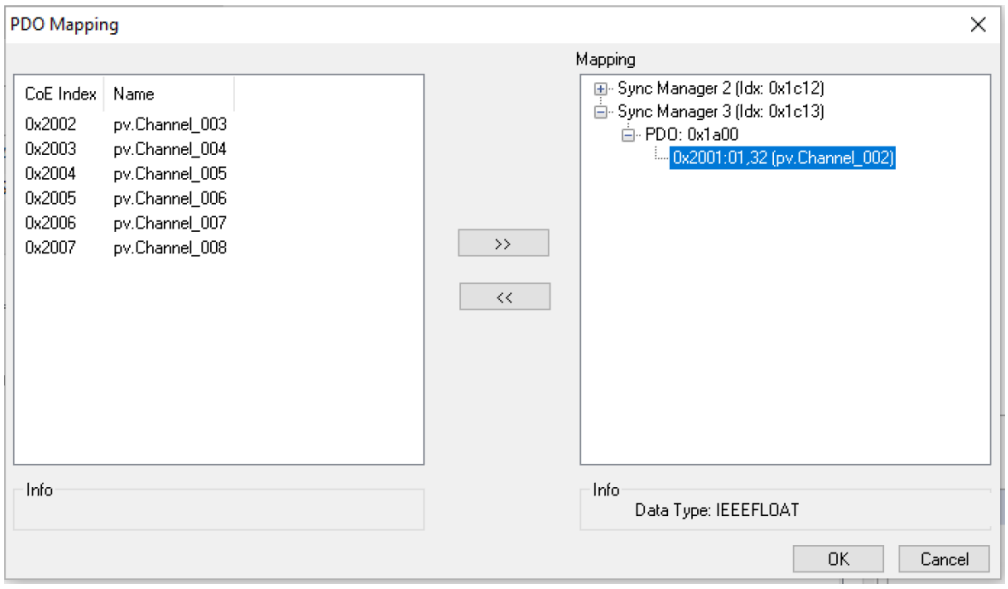
Alternatively, you can generate a complete EtherCAT configuration using the menu item Create PDOs automatically 615.

9.10.5.6.3 Menu

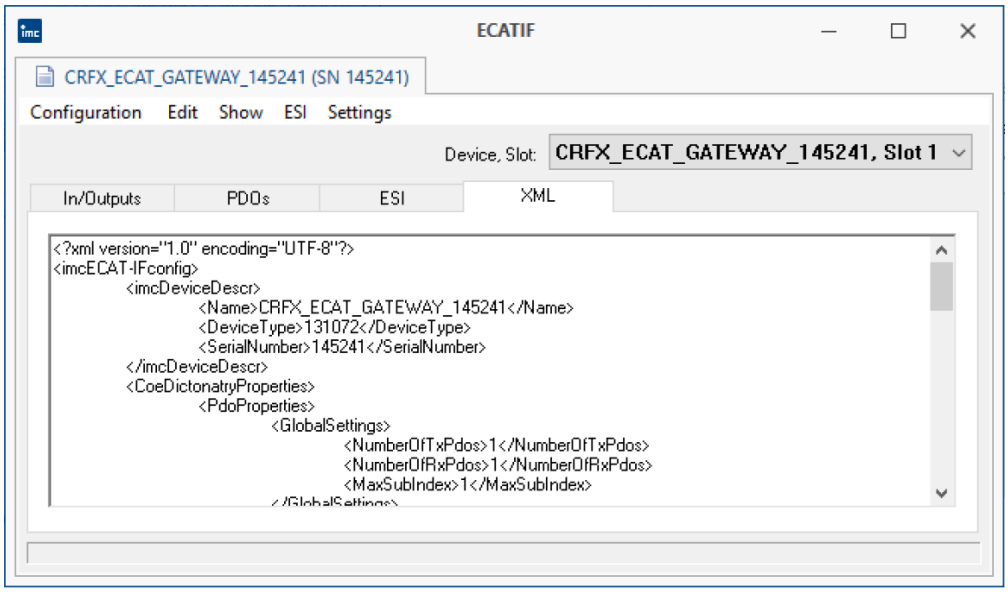
Configuration

Menu item	Description
New	This creates a new configuration. Existing settings are reset. Next, the channels must be set as either outputs 615 or inputs 615 under Edit.
Accept changes	Saves the current EtherCAT configuration within the experiment.
Import	Loads a previously saved configuration. Note that the channel names are decisive for the connection between the data acquisition channels and the EtherCAT Interface. If you use an existing EtherCAT configuration, make sure that the channel names match.
Export	Saves the current EtherCAT configuration. Thus the configuration can be used for an experiment that will be loaded for a different device.

Edit

Menu item	Description
Outputs	Device channels outputted via EtherCAT must be configured as outputs 617.
Inputs	Device pv-variables which are written via EtherCAT must be configured as inputs.
Add PDO	The PDOs act as the containers in which the inputs and outputs are organized. The menu item allows manual configuration.
Delete PDOs	Use this menu item to delete any PDOs no longer needed.
Automatic build PDOs	PDOs can also be generated automatically.
Mapping	 *Mapping dialog* Create the configuration as described in the section Configuration variant II 614. Subsequently, you can adapt the mapping.

Show

Menu item	Description
Inputs/Outputs	Display of all inputs and outputs in the Editor.
Inputs	Display of all inputs in the Editor.
Outputs	Display of all outputs in the Editor.
PDOs	Display of all Rx- and TxPDOs in the Editor.
RxPDOs	Display of all RxPDOs in the Editor.
TxPDOs	Display of all TxPDOs in the Editor.
Xml	Display of the XML-configuration generated.  *Show configuration as XML*

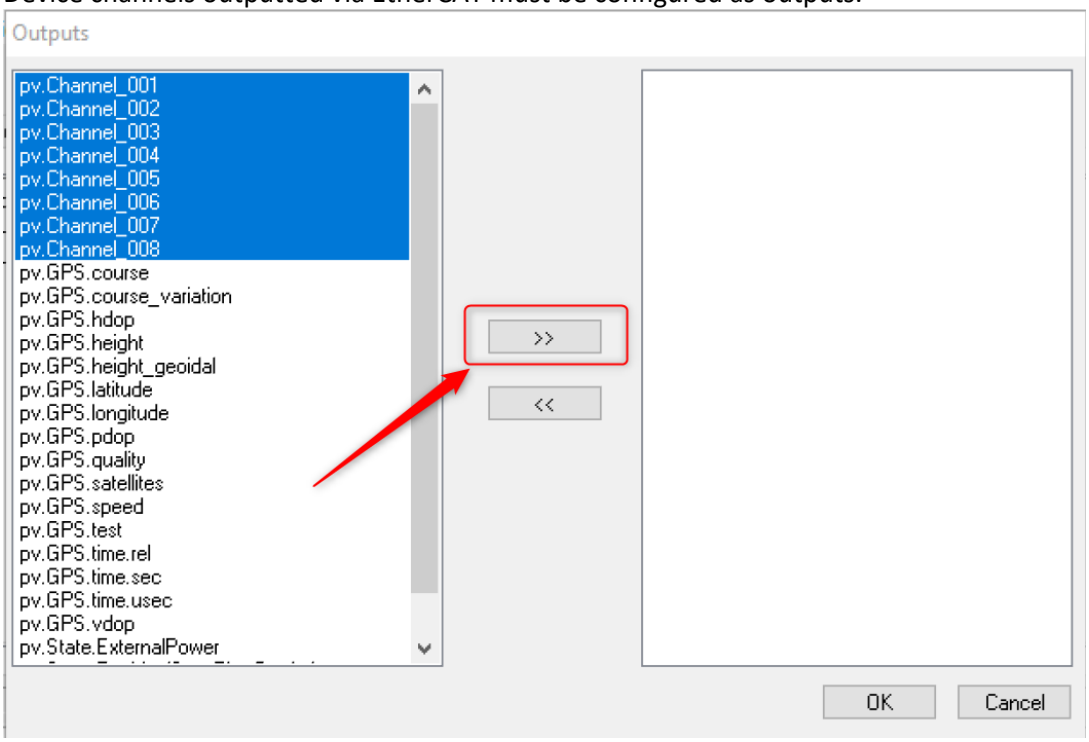
ESI

The EtherCAT interface's transfer properties can be transferred as an ***E**therCAT-**S**lave-**I**nformation* document (ESI).

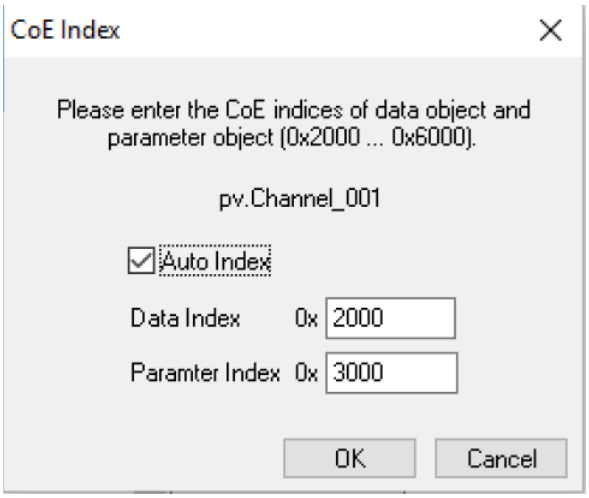
Menu item	Description
Save general ESI	Saves the current EtherCAT configuration as general EtherCAT-Slave-Information configuration.
Save device-specific ESI	Saves the current EtherCAT configuration as ESI configuration for a certain device. The device-specific ESI-documents include information on the VendorID, ProductCode and **serial number**.

9.10.5.6.3.1 Configured as outputs

Device channels outputted via EtherCAT must be configured as outputs.



Select one or more pv.channels and click >>



Set object index

A unique index must be specified both for the data part and for the parameters.

The parameters contain such information as unit, offset, etc.

When multiple channels are selected, the option *Auto Index* is offered. If it is active, you set the addresses for the first channel. Subsequent channels are automatically incremented up.

The addresses are specified in EtherCAT's usual numerical format **Hex**, in other words with "0x" preceding the number. The number range is fixed from **0x2000 through 0x6000**.

When you click on *OK*, the outputs are configured and listed on the right side.

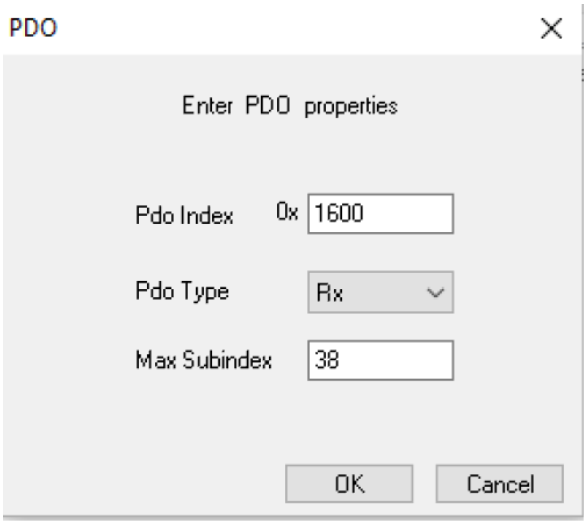
Outputs generated with *Create PDOs automatically* 615 can be brought to the left side using the button <<. Subsequently, these appear under *Edit > Inputs* 615 and can be configured as inputs there.

9.10.5.6.3.2 Insert 1600PDO

The PDOs act as the containers in which the inputs and outputs are organized. The menu item Create PDOs automatically 615 also generates these. The menu item *Edit > Add PDO* allows manual configuration.

PDOs for sending contain **outputs** and are called **RxPDOs**.

PDOs for receiving contain **inputs** and are called **TxPDOs**.



Add PDO

Creation of an RxPDOs with index 0x1600 and subindex 38.

9.10.6 FlexRay-bus interface

What is the FlexRay-Bus?

FlexRay is a Fieldbus system used for high-speed, deterministic data transfer in automobiles.

For configuration of imc devices connected with FlexRay, imc STUDIO offers the FlexRay Assistant 621.

9.10.6.1 Definitions

Term	Description
Bus	obsolete: see Cluster or Channel
Channel	(two interpretations) 1. FlexRay has two physical channels per cluster (Channel A and Channel B). A channel is an independently operating "DataBus". The two channels operate in synchronization, i.e. the cycles at Channel A and B begin and end simultaneously. 2. imc-specific: this refers to a measurement channel in imc STUDIO
Cluster	(two interpretations) 1. In accordance with FIBEX: All nodes involved with the Fieldbus system and their interconnections. 2. In the Assistant: an element in the overview window which makes the properties of the respective selected imc-FlexRay node accessible. ATTENTION: node and cluster properties are set jointly here

Term	Description
Cycle	The data stream in the channels is subdivided in cycles. Each cycle has a Cycle-ID [0…63]. If the cycle having the ID 63 was sent, then the cycle having the ID 0 is sent next. Thus, a period is 64 cycles long. The cycle subdivisions include the static and dynamic segments. In these two segments, slots are defined in which one can send or receive frames. See also [1] 620
Cycle-Multiplexing	This provides the ability to send different Frame types to the same slot of different cycles. See also [1] 620
Frame	A data package in which multiple signals can be defined. When and where a frame can be sent or received is determined by a frame triggering. Since a frame of a particular type may be scheduled to be sent at different times, multiple frame-triggerings can be assigned to the same frame..
Frame-Triggering	Determines in which slot (Slot-ID) and in which cycle (Cycle-Basis, Cycle-repetition) the assigned frame is to be received or sent.
KeySlot-ID	The Slot-ID of the slot in the cycle's static segment at which the cold-start or Sync frame is to be sent.
Node	Cluster (bus) subscriber. • A node is a Fieldbus terminal at a device. • A device can have multiple nodes. • Not every node must be connected with both channels (A and/or B). • Often there is exactly one node in a device.
Node-mode	A property of every FlexRay-node. see also: *FlexRay rules for KeySlotID and the NodeMode* 634 Normal Node • No KeySlotID needs to be specified, since no Cold-start frame or Sync frame is sent • Sync Node: A KeySlotID must be specified. The node will/must send its Sync frame to the specified slot. • Cold-start node: A KeySlotID must be specified. The node will/must send its Cold-start frame to the specified slot.
Signal	A bit series at a particular position in a repeating frame, interpreted as a physical measurement quantity.
Slot	(two interpretations) 1. imc-specific: bay for an imc Fieldbus board in an imc device 2. FlexRay-specific: A particular time segment on a channel, in which a specific frame having the associated Slot-ID is sent by a node in the cluster, or which can be received by all other nodes in the cluster.
Slot-ID	See also: Slot, Cycle

9.10.6.2 Reference literature and standards

[1]	FlexRay - Protocol Specification V2.1.rev A, Copyright © 2004-2005 FlexRay Consortium
[2]	FlexRay - EPL-Specification - V2.1.rev A, Copyright © 2004-2005 FlexRay Consortium
[3]	FlexRay - EPL-Application Notes - V2.1.rev A, Copyright © 2004-2005 FlexRay Consortium
[4]	FIBEX - Field Bus Exchange Format, AE[FBX] Version 2.0, Release Version, Association for Standardisation of Automation and Measuring Systems, © ASAM e.V

9.10.6.3 Capabilities / constraints

- **Recording** of the bus data received
- **Deconstruction** of the bus data into signals (measurement channels)
- Measurement of maximum 512 channels per device (total over all channels: analog, incremental etc.)
- Frame and data sending capability
- Sync and cold-start frame sending capability
- No cycle-multiplexing capability in cold-start or sync-frames (KeySlot-frames)
- Import capability for FlexRay-FIBEX files 633, Version 2.0.1 (*.xml)
- Import of FlexRay-FIBEX-Plus 633 files based on the FIBEX version 2.0.1 (*.xml)
- Import of Version 3.0.0 and 3.1.0 FlexRay FIBEX files (*.xml)
- XCP-Master
- Import of A2L 634-files for FlexRay
- Supported XCP-specifications:
  - ASAM_AE_MCD-1_XCP_BS_Protocol-Layer_V1-2-0.pdf "ASAM MCD-1 (XCP); Universal Measurement and Calibration Protocol; Protocol Layer Specification; Version 1.2.0 Date: 2013-06-20"
  - ASAM_AE_MCD-1_XCP_AS_Flexray-Transport-Layer_V1-2-0.pdf "ASAM MCD-1 (XCP on FlexRay); Universal Measurement and Calibration Protocol; FlexRay Transport Layer; Version 1.2.0; Date: 2013-06-20"

Constraints on FIBEX-import:

- no implementation of CODING.COMPU-METHOD.CATEGORY "TEXTTABLE", "SCALE-LINEAR","TAB-NOINTP", "FORMULA"
- no implementation of FRAME.MULTIPLEXERS (with FIBEX 2.0.1) and PDU.MULTIPLEXERs (with FIBEX 3.0.0, 3.1.0)
- no implementation of multiple COMPU-METHOD.COMPU-INTERNAL-TO-PHYS.COMPU-SCALEs per signal; only one COMPU-SCALE is evaluated.
- no implementation of UPDATE-BITs in PDUs and SIGNALs

Note

Also observe the general notes on file buses in regard to control philosophy 474 and the sampling rate either as fixed sampling rate 752 or time stamp 752.

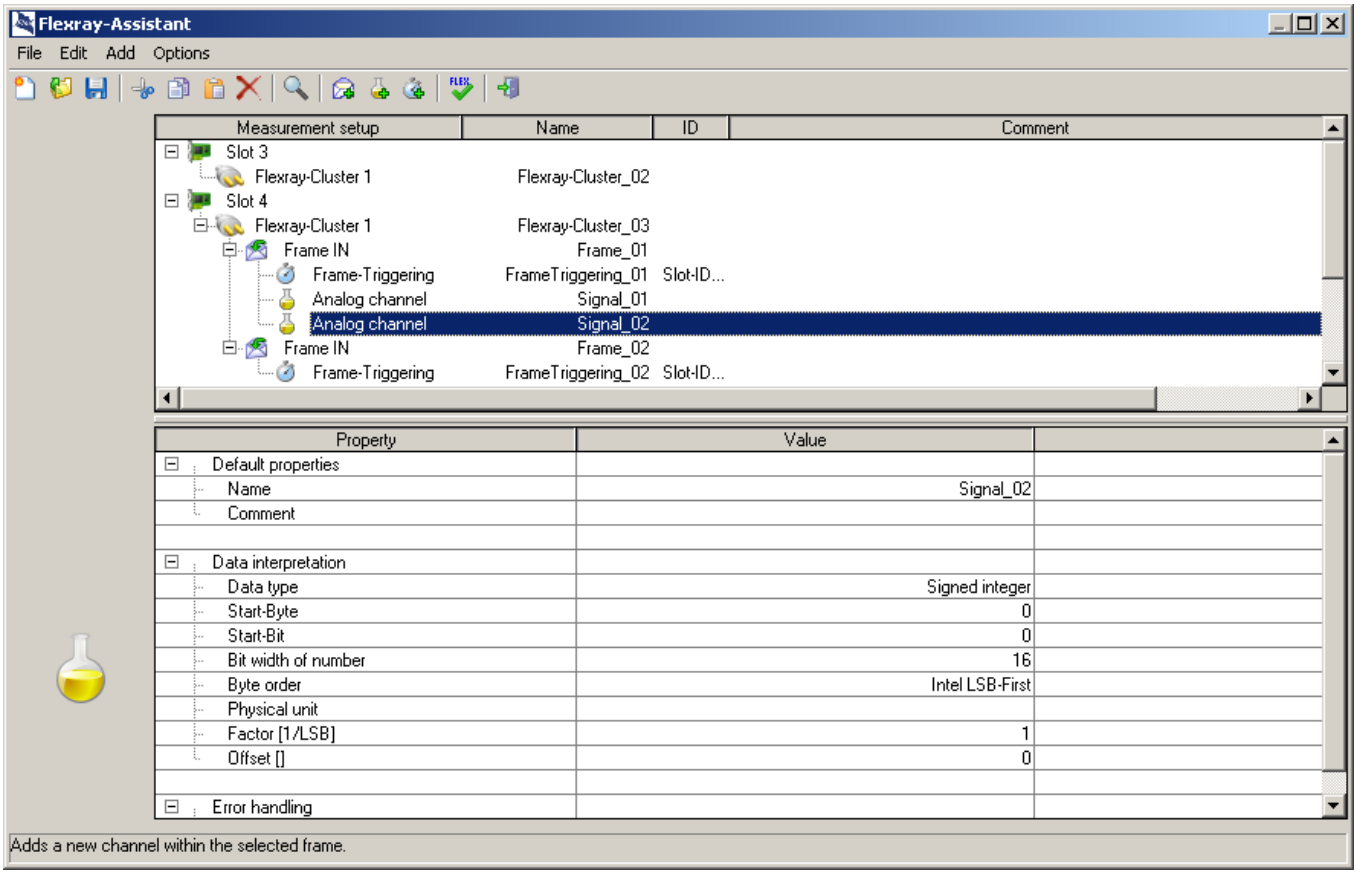
9.10.6.4 DSUB-9 Pin configuration

Pin	DSUB-9
1	n.c.
2	BM Channel A
3	GND
4	BM Channel B
5	GND
6	n.c.
7	BP Channel A
8	BP Channel B
9	n.c.

9.10.6.5 FlexRay assistant

To set the FlexRay connections (clusters), as well as to define frames and channels (signals), the *FlexRay-Assistant* is used.

Start the Assistant from within the imc STUDIO software via the Ribbon *Setup-Configuration > FlexRay-Assistant*.



FlexRay Assistant

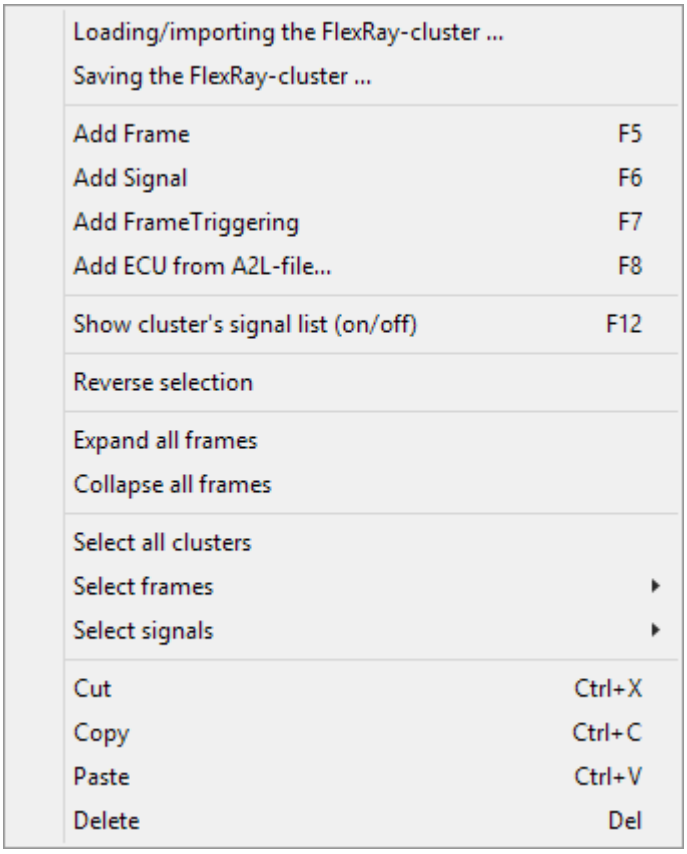
9.10.6.5.1 Overview window

The Overview window is the upper list box, in which the measurement setup is illustrated schematically as it relates to the FlexRay configuration. What is shown is the measurement devices ( ) with the experiment along with their names. If there is only one FlexRay-capable device in the experiment, it will not be displayed, in the interest of clarity.

Below each measurement device, its built-in FlexRay modules (slots) ( ) with their FlexRay connections (FlexRay clusters) ( ) are shown.

For each FlexRay cluster, the previously defined frames with their associated signals (channels) are shown.

The selection of frames and signals can be made by means of the context menu, also inverted.



Context menu of the overview window

9.10.6.5.2 Properties window

The Properties window is visible in the lower portion. It displays the editable properties and the settings options for the object currently selected in the Overview window. If multiple objects are selected at the same time, the Properties window shows a selection of suitable properties.

To change a property's value, click on the desired value.

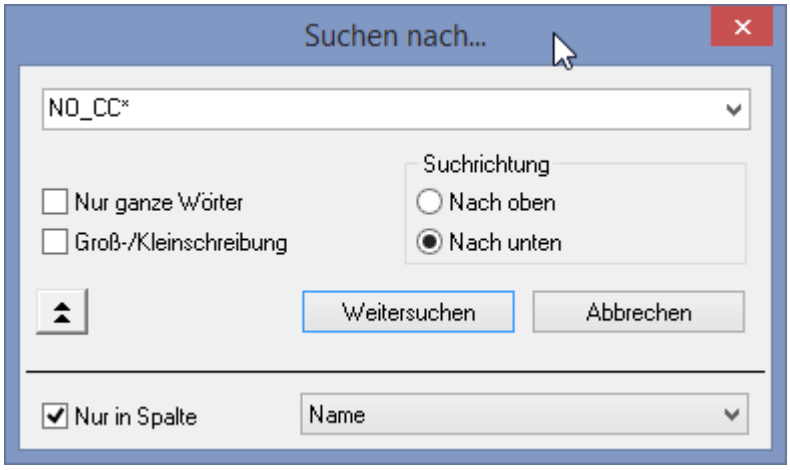
Depending on the property, either a pop-down list or a text box is used.

9.10.6.5.3 Items in the main menu

File

Menu item	Description
New	Deletes any previously created frames and channels and creates an empty FlexRay configuration.
Loading/Importing the selected FlexRay-Cluster	The configuration will be loaded from a file for the selected cluster. Enables the import of an entire FlexRay configuration from a file. Supported formats: • FRY: an imc proprietary format • XML: FIBEX-XML, see Loading FIBEX-files 633 up to version 4.x • ARXML: AUTOSAR v4.3.1
Saving the selected FlexRay-Cluster	The configuration will be saved into a file for the selected cluster. Only supports FRY; xml-FIBEX not supported.

Edit

Menu item	Description
Copy Paste Cut Delete	These commands only affect elements visible in the overview window. Frames, Frame-Triggerings and signals can only be copied, deleted and pasted. Devices, slots and clusters can not be pasted.
Find Find next [F3]	 In the overview window can be searched for words or part of words. When there are many defined frames/signals, it is reasonable to search for their names. It is also possible to perform a targeted search in the columns selected.

Add

Menu item	Description
Frame	Adds an additional frame to the selected cluster.
Frame- Triggering	Here you can add an additional Frame-Triggering to the selected frame.
Signal	This menu item lets you add an additional signal to the selected frame.
ECU from A2L-file	Adds an additional ECU channel 634.

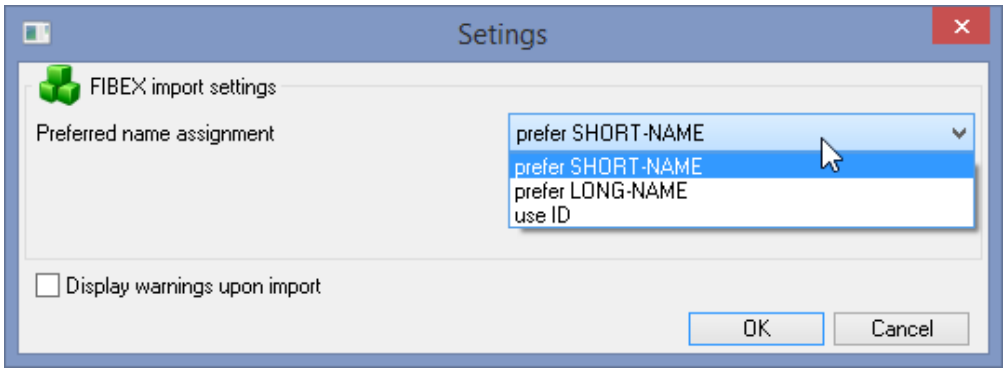
View

Menu item	Description
Display tree diagram Display the cluster's signal list	Toggling between the tree diagram and the list of signals.

Configuration

Menu item	Description
Test FlexRay configuration	Tests whether the settings are valid as they are made. If not, the overview window skips to the invalid object and a message is posted in the Status bar.
Create ECU configuration	Prepares the configuration for the ECUs.

Options

Menu item	Description
Font	Font for the overview window.
Settings	*Preferred name assignment*: As a matter of principle, you can set up the name assignment with either short or long names, or use an ID. All names are constructed in accordance with the construction rule in the cluster properties 625. To display warnings upon importing, activate the corresponding option.  *Options menu: Settings*

9.10.6.5.4 Properties of clusters

Here, parameters of FlexRay clusters and nodes can be set.

Cluster parameters are parameters which must be the same for all nodes in the cluster in order for the cluster to work without problems. The node parameters only pertain to the node (connection) of the imc device affected. They need not necessarily be the same in all of the cluster's nodes.

Some of the node parameters are only provided as information for certain users. These cannot be edited since they are derived automatically from the settings for the other parameters.

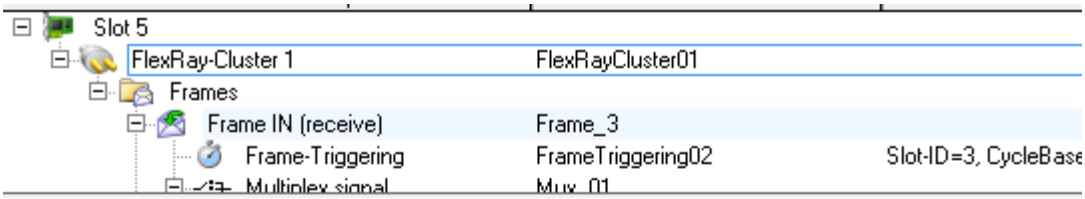
Describing the meaning of every parameter would go beyond present purposes. For more information on this, please see: [1] FlexRay - Protocol Specification 620.

Multiple frames can be created in a cluster.

Construction rule for names of channels

The channel names can be composed from the name entries with placeholders: {Clu} for **Cluster**, {Frm} for **Frame** 626 and {Sig} for **Signal** 628 and **ECU-channels** 634.

Example: Name of the clusters = "*FR_Cl01*", name of the frame= "*Frame01*" and name of the signal = "*Sig01*" from *{Clu}_abc_{Frm}_{Sig}_123,* the resulting name is: "*FR_Cl01_abc_Frame01_Sig01_123*"



Property	Value
Default properties	
Name	FlexRayCluster01
Comment	
Construction rule for channel names	{Clu}_{Fra}_{Sig}
Construction rule for names of Monitor channels	{Clu}_{Fra}_{Sig}_mon
Construction rule for BitPort channel names	{Clu}_{Sig}
Construction rule for names of BitPort Monitor channels	{Clu}_{Sig}_mon
Construction rule for ECU channel names	{Clu}_{Ecu}_{Sig}
Construction rule for names of ECU Monitor channels	{Clu}_{Ecu}_{Sig}_mon
Construction rule for channel names in the FrameDesciptionBlob	{ResultName}_Fdb
Construction rule for names of bus- and error-logging channels	{Clu}_{Sig}
Node properties	
loaded FIBEX-fileame	
loaded FIBEX-cluster name	
Behavior in the cluster	Cold-start node
Connected FlexRay channels	Channel A
Key Slot ID	1
Internal bus termination for channel A	No
Extra StartUp node mode	Disabled
Automatically re-connect upon disconnection?	Yes
(pWakeupPattern)	3
(pAllowHaltDueToClock)	Yes
(pAllowPassiveToactive) [even/odd CylcePairs]	0
Information: Automatic properties	
Global cluster properties	
Baud rate per FlexRay channel [MBit/s]	10.0
Expected bus load in [%]	40

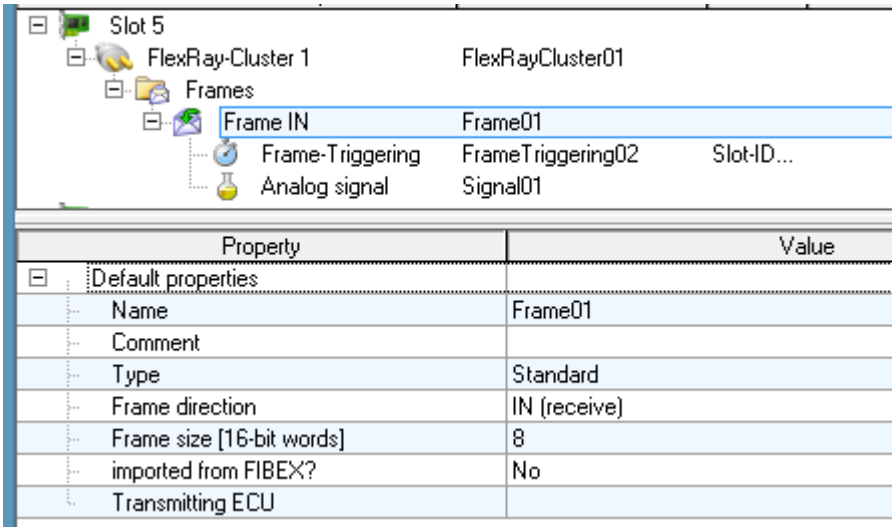
Properties of Clusters

Note

If *Recording of null- or data frames* is activated, then all frames are processed on the bus, which can lead to performance problems if the data load on the bus is high. If the logging is deactivated, then only such frames which are listed in the FlexRay Assistant are processed.

9.10.6.5.5 Properties of frames

In the frame parameters only the length and direction can be specified. Within a frame, multiple signals and multiple frame triggerings can be created.



Properties of Frames

Parameter	Description
Name	Here, an arbitrary name can be entered for the frame. It will also be displayed in the Overview window under the Name column. This property represents a brief description. With the option "always prefix with the frame name", then frame name is automatically prefixed to the channel name.
Comment	The comment is for providing a more detailed description of the frame.
Typ	Standard, XCP-pre configured or XCP-runtime configured.
Frame direction	This property determines whether the frame is to be received and interpreted by the measurement device (incoming, receiving, IN) or whether this frame is to be assembled and sent (outgoing, sending, OUT).
Frame size [16-bit-words]	This property sets the number of data bytes the frame has. This must be between 0 and 127.
imported from FIBEX?	Info on whether the frame is loaded from FIBEX.
Transmitting ECU	Designation of the ECU if available

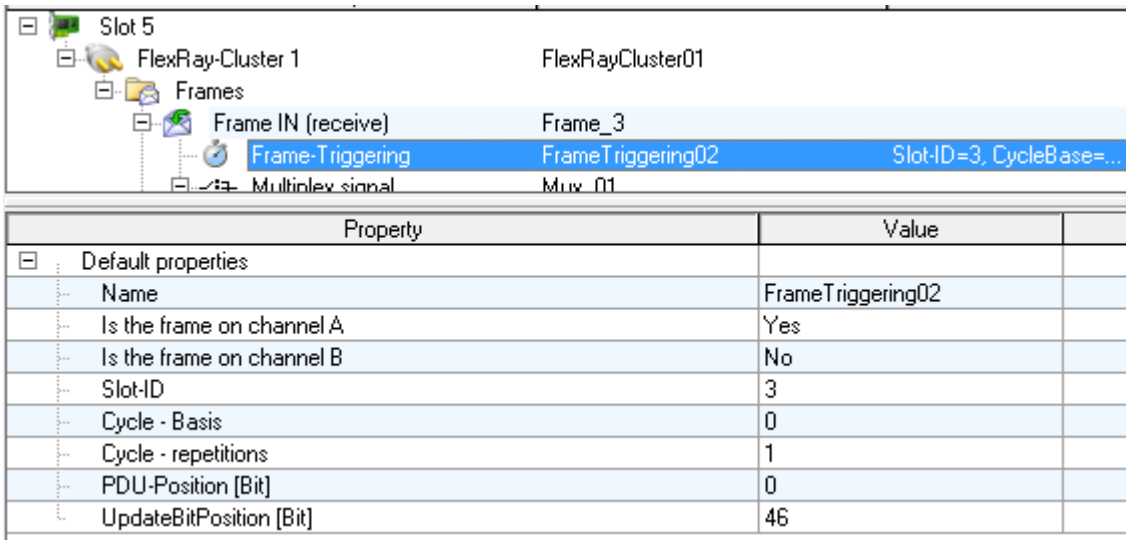
9.10.6.5.6 Properties of frame-triggerings

A Frame-Triggering determines when and where a frame is to be sent or received. For this reason, the Slot-ID, Cycle-Base, Cycle-Repetition and the channel assignment are found here.

The separation of frames and the triggering is due to the fact that it allows one to specify multiple points in time at which to send / receive a frame, without needing to define (enter) new frames with all the associated signals each time.

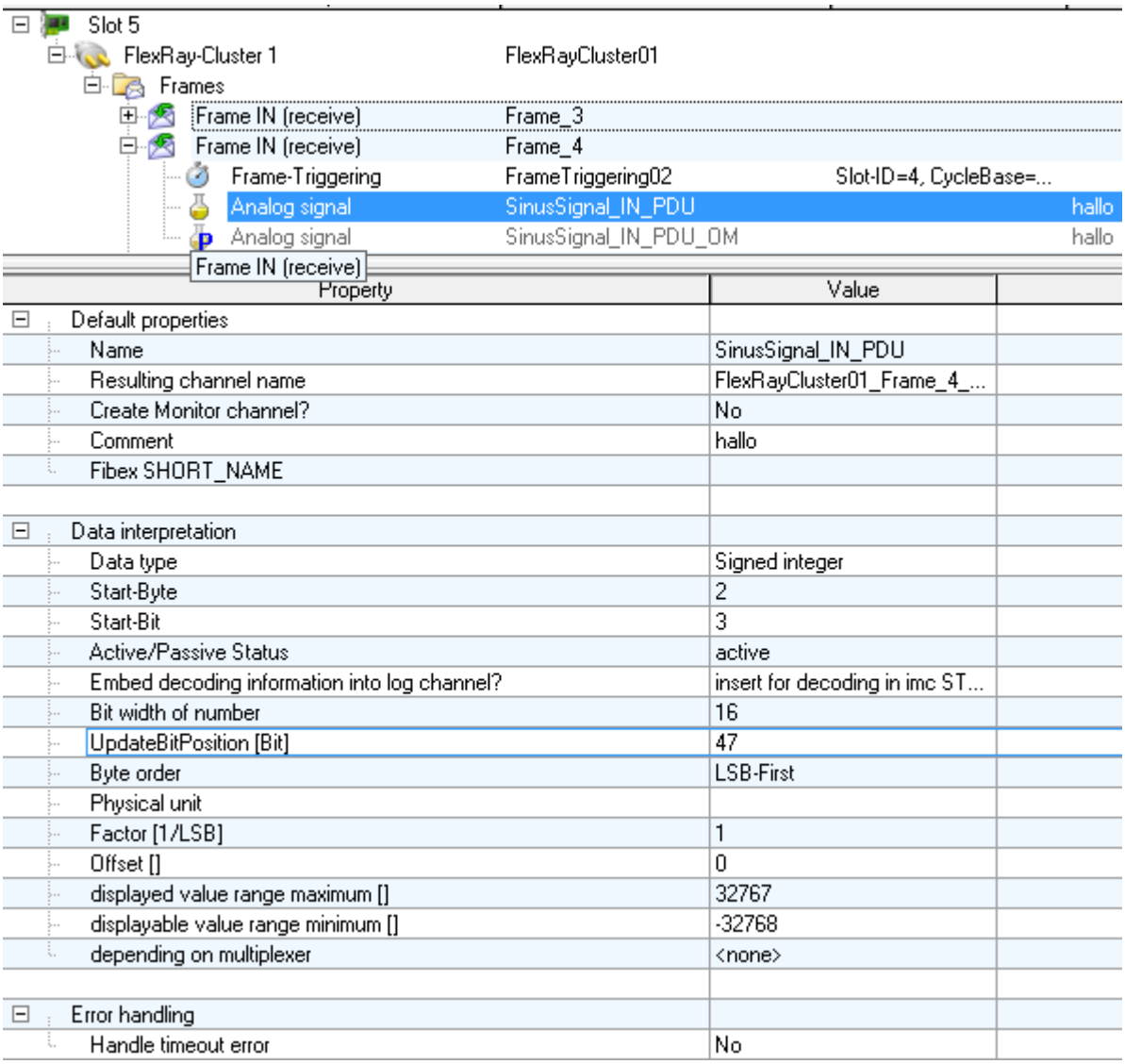
The property PDU-Position indicates a shift of all signals in the frame. The signals are shifted backwards by the specified value in bits. This means that the signal positions can be different for each Frame-Triggering.

The *UpdatedBitPosition* sets the position of the PDU's validity control bit. The signals in this frame are only evaluated if the bit (here at position 46) is set to "1". In addition, the signal-specific UpdateBit 629 must also be set to "1.



Properties of Frame-Triggerings

9.10.6.5.7 Properties of signals



Properties of Signals

Default properties	Description
Name	Sets the channel's name, with which one works in imc STUDIO. This name must conform to the rules and the format applicable to a imc STUDIO channel name.
Resulting channel name	The channel name which results from the construction rule 625 in the Cluster properties.
Create Monitor channel?	If "Yes", a copy of the channel is generated. The name is determined according to the construction rule 625.
Comment	Specifies the comment for this channel.
Fibex SHORT_Name	Specifies the short name of the signal as it appears in the FIBEX file.

Data integration	Description
Data type	This sets the signals type, or rather, how the data on the bus are interpreted. The following data types are possible:

Data type	Description
Signed Integer	The bit widths available for this data type are 2 to 32 bits.
Unsigned Integer	The bit widths available for this data type are 2 to 32 bits.
Floating point	Either 32-bit or IEEE 64-bit floating point format are available.
Bit	A single bit as a digital signal.

Active/Passive Status	Allows signals to be hidden The configuration files to be imported often contain multiple thousands of signals. Up to 990 Fieldbus channels can be registered for each imc device. There is a limit of 512 active channels per device. Once the count of signals belonging to a device has been exceeded, it is necessary to select which ones are important for the measurement.

Active/Passive Status	Remarks
active	The signal is set as an active channel and recorded.
passive	The signal is registered in the system as a channel, but set to passive. No measured data are recorded.
Do not register with administration	The signal is not registered in the system. This signal only exists in the Assistant. No measured data are recorded.

Embed decoding information into log channel?	The option allows the data stream to be unpacked as individual channels in imc STUDIO by the Bus Decoder 1101 or via imc FAMOS. This setting is independent of the setting: "*Active/Passive Status*". The exception is "XCP over Flexray"-channels.

Information	Description
don't insert	Decoding is not possible.
insert	The channel can be extracted offline, e.g. with imc FAMOS.
insert decoding in imc STUDIO	The channel can be extracted with the Bus Decoder or offline, e.g. with imc FAMOS.

Bit-width of number	This property sets the signal's length. Different maximum and minimum values can be entered, depending on the data type.
UpdateBitPosition (Bit)	If a bit position is entered here, the signal is only evaluated if the relevant update bit is set. With the value"-1", the signal is always evaluated.
Byte order	Intel (LSB first) or Motorola (MSB first)
Start-Byte and Start-Bit	These two properties determine at which position within the frame the signal begins. Outgoing signals may not occupy the same bits in the frame; for incoming signals this is not a problem.
Physical unit	Here you can enter a physical unit or its abbreviation, e.g. "V" for Volt. Use SI-units wherever possible. It is not advised to enter any scaling prefix such as m for milli-. Instead, adjust the factor correspondingly. The imc STUDIO curve windows automatically add such prefixes to correspond to the measured values.

Data integration	Description
Factor and Offset	If the signal currently selected happens to be of the integer data type, signed or unsigned, then both the properties **Factor** and **Offset** are displayed. These values are parameters for linear transformation, which are computed by the following formulas. **Incoming signal**: $y = f \cdot x + a$ **Outgoing signal**: $x = \frac{y}{f} - \frac{a}{f}$ Variable definition y - physical value x - integer on the Fieldbus f - factor a - offset An incoming signal is a 16-bit unsigned integer. Thus, the integer's numerical range is 0 LSB to 65535 LSB. The physical quantity which it represents is a voltage between 10 V and 20 V. The factor is thus given by the equation $f = \frac{20\ \mathrm{V} - 10\ \mathrm{V}}{65535\ \mathrm{LSB}} = 1.526 \cdot 10^{-4} \frac{\mathrm{V}}{\mathrm{LSB}}$ and the offset is $a = 10\ \mathrm{V}$ Supposing the following integer were imported from the bus, $x = 13483\ \mathrm{LSB}$ it would correspond to a voltage of $y = f \cdot x + a = 12.057\ \mathrm{V}$
displayed value range minimum, maximum	Boundaries of the value range. This is calculated in observance on the factor, the offset and the bit width.
depending on multiplexer	Reference to the Multiplexer signal on which this signal depends.

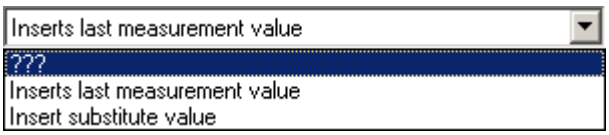
Handle timeout error

The term timeout-error refers to a case where no measured value is received from the fieldbus after a certain period; error handling can be set to deal with such an occurrence.

Parameter	Description
Handle timeout error	This settings option determines whether error handling of any timeout error would be performed.
Timeout Interval [s]	This is where to set the time limit after which a timeout-error is considered to have occurred.

Error handling action

This setting determines the action to be performed in case the timeout time limit elapses.



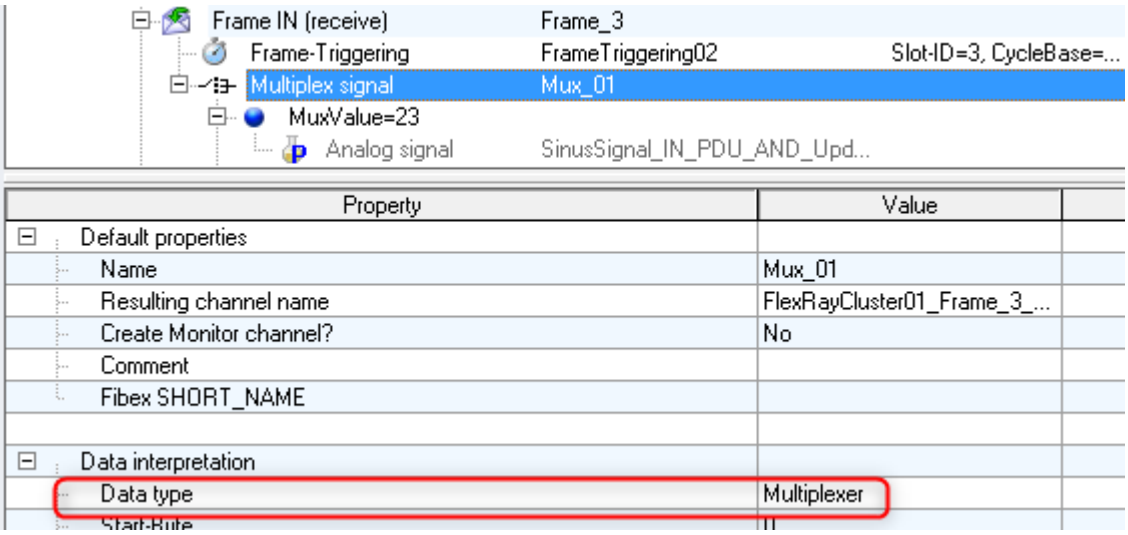
For this purpose, the following options are available, whose effects depend on whether a channel is sampled equidistantly or recorded with a time stamp.

Error handling	Equidistant sampling	Samples with time stamp
Last value	The last valid value is returned until a new value arrives.	After elapse of the timeout time, a new sample is generated with the last value.
Substitute value	The substitute value is returned until a new value arrives.	After elapse of the timeout time, a new sample is generated with the substitute value.

9.10.6.5.7.1 Multiplexer signal

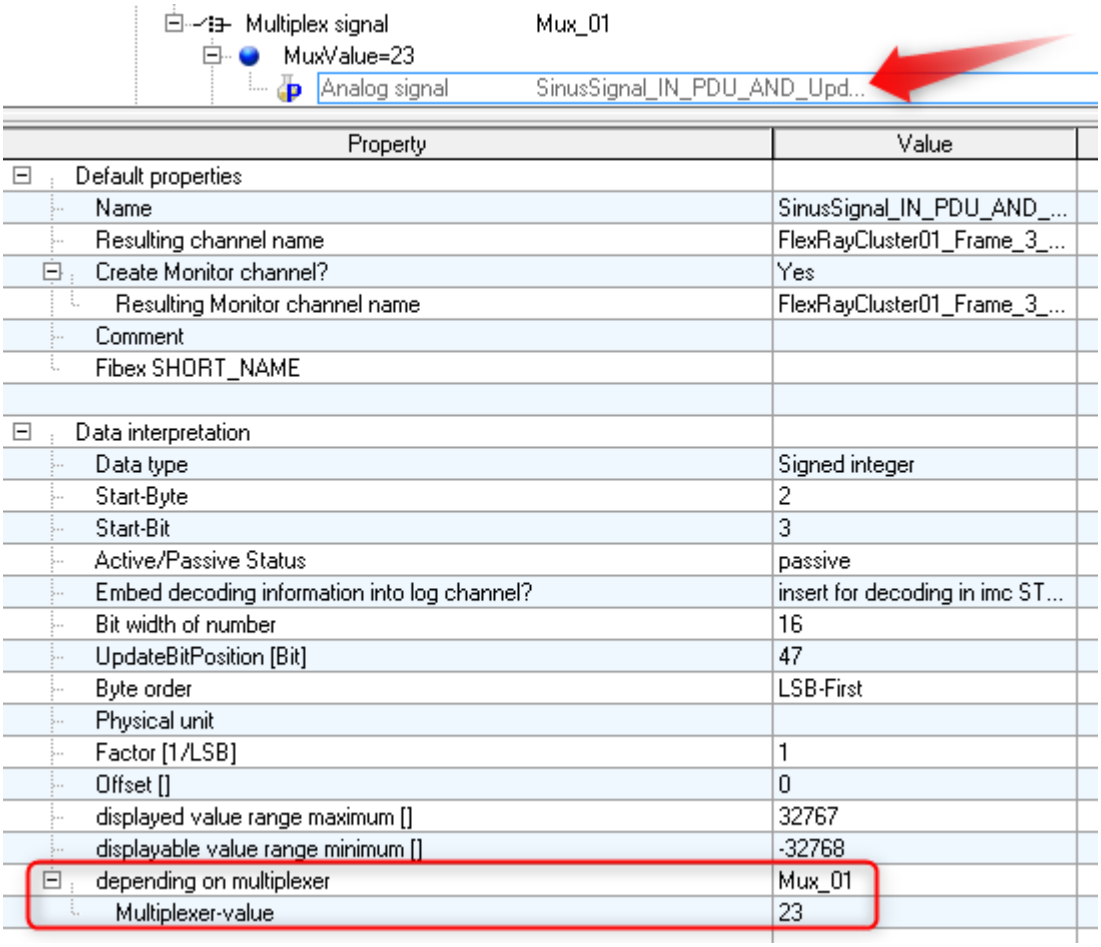
If the data type of a signal is set to "*Multiplexer*", the frame can be evaluated depending on an UpdateBbit.

1. First, a signal is created whose data type is set to "*Multiplexer*":



Set "Data type" of signal to "Multiplexer"

2. Another signal is applied. In the "*Data interpretation*" branch, the previously created multiplexer is selected in the property "*depending on multiplexer*". A further line for the "*Multiplexer-value*" is then inserted, in which the value for the validity of the signal must be entered.



Select multiplexer and set "Multiplexer-value"

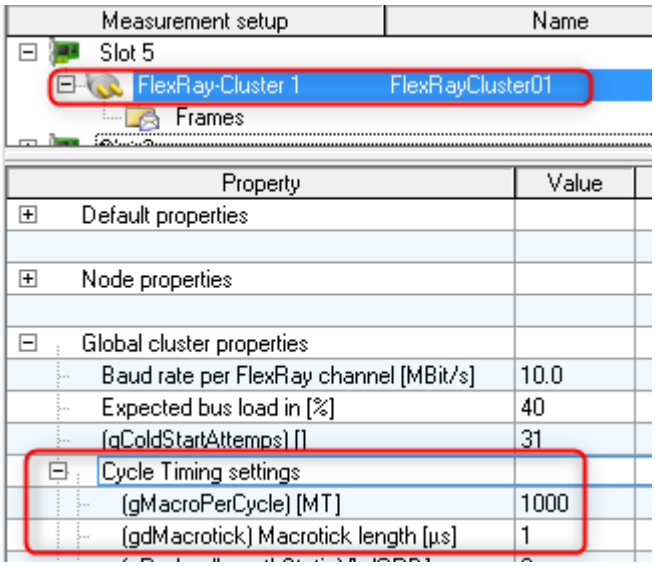
9.10.6.5.8 Blob - Decodieren with the Bus Decoder

All data associated with a FlexRay-node can be captured completely as a log channel. This is a way around the limitation of 512 channels per device instance. Along with separation into individual channels with imc FAMOS, this can be accomplished already with imc STUDIO's Bus Decoder 1101.

To do this, you make the following settings in the FlexRay-Assistant:

- The FlexRay channels to be extracted from the log channel in Bus Decoder must be set to "*insert for decoding in imc STUDIO* " in the properties under "*Embed decoding information into log channel*". With "*insert for decoding in imc STUDIO*", the channels in imc STUDIO and imc FAMOS can be extracted. With "*insert*" they are only visible in imc FAMOS.
- With FlexRay, a log channel is created with the first insertion of an incoming signal. There is no active/passive switch in the FlexRay Assistant.

Note Notes on the sampling rate (cycle time)



In FlexRay, a channel's sampling rate is not entered directly but results from a formula applied to the following parameters:

Sampling interval = *gMarcoPerCycle * gdMarcotick * CycleRepetition*

Example: 1000 * 1µs * 2 = 2ms

In this way, sampling rates can result which could not be set directly due to the specified steps as multiples of 1-,2-, and 5. Instead, the device applies the next higher permitted sampling rate.

Example:

Calculated sampling rate Bus Decoder = 333Hz

Sampling rate set in the device = 500Hz

For this reason, different sampling rates may be shown for the same signal when comparing data from the device with data from the Bus Decoder.

9.10.6.5.9 Loading FIBEX-files

Only a FIBEX file having FlexRay-Cluster description may be imported. Files with Fieldbus descriptions of other bus systems are omitted.

If multiple clusters are to be described in the FIBEX-file, then the first configuration is loaded on the Assistant's first cluster, the second configuration is loaded on the second cluster and so on.

Loading of the FIBEX files is completed when there are no more cluster configurations in the file or no more free clusters in the Assistant (device).

Import of FIBEX-Plus files

When importing FIBEX-Plus files it is also possible to load and apply PDU (Protocol Data Unit) positions. This data element determines the Byte position of a group of signals within a frame. The data element is set up in the frame triggering structure. This means that the previous incoming frames may become PDUs.

When sending frames, the property PDU-Positions belonging to the Frame-tiggerings is ignored. For this reason, only complete frames can be sent.

When importing signals, the following properties are displayed: Short-Name, sending ECU, Frame-Name, Signal- and Frame-Comment.

After loading

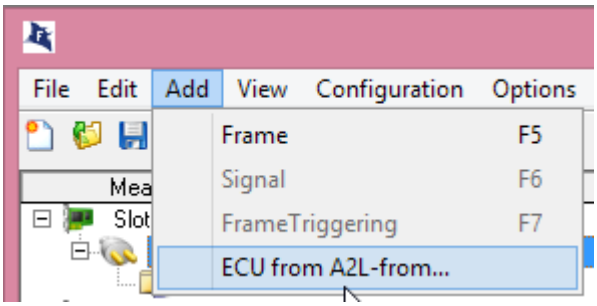
After loading a FIBEX-file, the following issues/ settings should be considered, since the imported FIBEX does not contain any information on the imc measurement system.

- Should the internal terminators be connected?
- With which channels (A and/or B) is the imc device connected?
- Should the imc device's connection work as a "Normal-", "Coldstart-" or as a "Sync-node"?
- **If** the device must work as a "Coldstart-" or as a "Sync-node", the "KeySlotID" must be specified additionally, see also FlexRay rules for the KeySlotID and node mode 634.

These settings can be found in the properties of each cluster. To make them easier to find, they appear in bold letters.

9.10.6.5.10 Loading A2L files (XCPoFlexRay)

XCP over FlexRay.



The Import Assistant for A2L files displays all the available names and their associated address and comment.

9.10.6.5.11 FlexRay rules for the KeySlotID and node mode

- At least two and at most three Coldstart-nodes must be configured in a cluster.
- Each Coldstart-node is at the same time a Sync-node.
- Each Sync-node (and thus each Coldstart-node) has **its own unique** KeySlotID.
- A KeySlotID is a SlotID located in the cycle's static segment (never in the dynamic segment).

Note

- The FlexRay Assistant is only familiar with the FlexRay nodes which exist in the imc device.
- Nodes which otherwise only exist in the connected FlexRay-cluster are not known to the Assistant.
- For this reason, they cannot be checked according to the rules stated above.

9.10.6.5.12 Procedure for connecting and recording

General Procedure

- Connect the imc device at the cluster, otherwise no recording/measurement is possible.
- Preferably **do not** be connect the imc device at the ends of the bus line, otherwise the internal terminator for the connected channel(s) (A, B) must be activated, which may lead to excessive bus terminations, since the bus may already be terminated in the connection lines used or in the other bus subscribers.

9.10.6.5.12.1 Recording (measuring) an entire cluster

Example:

- A FIBEX-file has been loaded.
- The cluster is completely present, i.e. all devices described in the FIBEX file are present and connected as described in the FIBEX-file.

Procedure:

- see General Procedure 634
- In the cluster-settings, set the Node-mode ("behavior of Cluster") to "Normal node".

9.10.6.5.12.2 Cluster with missing coldstart-node

Example:

- A FIBEX-file has been loaded.
- The cluster is only partially present. A (Coldstart-) node (device) is missing.

Procedure:

- See General procedure 634
- First, find out whether the missing device was a Coldstart-node.
- Find out how many Coldstart-nodes are still present.
  It may be that there were originally three, and that there are now still two. In that case, the bus is still operational; see FlexRay rules for the KeySlotID and node mode 634.

If there are still 2 Coldstart-nodes:

- In the Cluster-settings, set the Node-mode ("behavior of cluster") to "Normal mode".

If there is only one Coldstart-node left:

- In the cluster-settings, set the Node-mode ("behavior of cluster") to "Coldstart-node".
- Since the cluster is missing a Coldstart-node, the imc devcie must take on its function, otherwise the cluster cannot boot up (see also FlexRay rules for the KeySlotID and node mode 634).
- A Coldstart-node must have an assigned KeySlotID.
  In this KeySlotID's slot, the Coldstart node will then send a Coldstart frame. It can be difficult to select the appropriate KeySlotID without knowing the missing Coldstart-node's KeySlotID. Any Fibex file which was previously imported will at least contain descriptions of all Coldstart nodes with their KeySlotIDs.
- As an aid, a list of the KeySlotIDs specified in the Fibex file, along with their asociated Coldstart-node names, is displayed when entering the KeySlotID. You can now select the missing Coldstart-node on the basis of its name. This means that the imc device takes its place.
- Next, select the KeySlotID of the missing device. The imc device will then always send its Coldstart-frame in this slot.
- Since a Coldstart-frame is now to be sent, a corresponding send-frame must be configured.

Toward this end, there are 2 possibilities:

1. If no important signals must be sent in this KeySlot (the quick way)
   - Create a new frame
   - Set the frame's direction to "OUT (send)"
   - Go to the frame-triggering of the new frames, and set the Slot-ID here to the KeySlotID determined above.
   - Optional: find all incoming frames (frame-triggerings) having the same SlotID as the KeySlotID, and delete the frame-triggerings and any associated frames (search function: "Slot-ID=???"). Reason: Frames sent by the imc device can not be received. By removing frames, fewer channels may be registered. This increases the performance, and provides clarity in the Assistant.

2. If there are important signals to be sent in this KeySlot (the hard way)
   - Find all incoming frames (frame-triggerings) having the same SlotID as the KeySlotID, and delete all frames except the one which has defined the necessary signals. (Search function: "Slot-ID=???")
   - Set the direction of the frame to "OUT (send)"
   - Set a signal source or a constant value for each signal in the KeySlot-frame.
   - **NOTE**: It is unfortunately not possible to run cycle-multiplexing of KeySlots to be sent. For this reason, only one frame-triggering may exist for this frame. This Frame-Triggering also may only have a Cycle-Base of zero and a Cycle-Repetition of 1.

9.10.7 imc Application Module

The imc Application Module serves to **integrate measurement channels from "third party" devices or systems** into an imc CRONOS*compact* respectively imc CRONOS*flex* system via standard hardware interfaces.

Examples of possible channel sources include:

- Specialized complex sensors
- "third party" devices
- Bus systems (e.g. fieldbusses)

The standard interfaces supported include, in particular:

- Ethernet
- serial interfaces (RS-232, RS-485, RS-422)

The systems to be integrated are typically user-customized or dedicated devices by third-party manufacturers. The integration is achieved by means of a standard hardware module (APPMOD), which comes with a dedicated processor for which a custom application is programmed. This program is either created by imc on commission or can also be created and implemented by qualified partners or trained users provided with specialized development tools.

This user-specific hardware and software expansion is supported by the device software (imc STUDIO). A special version of the device software is not necessary.

Characteristics:

- encapsulated, custom hardware + software solution, embedded in a standard system
- Standard system with complete software support
- Flexible support by unaltered standard operating software
- Standard hardware component
- Stand-alone, autonomous system environment

Reference Manual

The imc Application Module describe the use and initiation of the module, such as the RS-232, RS-485, RS-422.

9.10.8 LIN-bus interface

What is the LIN-Bus?

- Affordable automobile-bus for low-speed open- and closed loop control (e.g. air conditioning, automatic window, outside mirrors, etc.)
- Single-wire bus (return line is vehicle mass (chassis))
- Data transfer rates: 1kBit/s to 20kBit/s
- Master-Slave access control
- A master device serially polls the Slave devices for the frames to be transferred.
- Related to CAN-Bus, thus affordable gateways
- Uses frame-identifier similarly to CAN
- Up to 8 data bytes can be transferred in one frame.

Note

Also observe the general notes on file buses in regard to control philosophy 474 and the sampling rate either as fixed sampling rate 752 or time stamp 752.

Reference Literature and Standards

[1]	LIN specification Revision 1.3 and 2.0, 2.1
[2]	"LIN-Bus", Andreas Grzemba, Hans-Christian von der Wense, Franzis Verlag, 2005, Poing (a German language book)
[3]	ISO / OSI reference model

9.10.8.1 DSUB-9 Pin configuration

The LIN terminals on the device are 9-pin D-Sub plugs (male). Therefore, they require cables equipped with a female terminal.

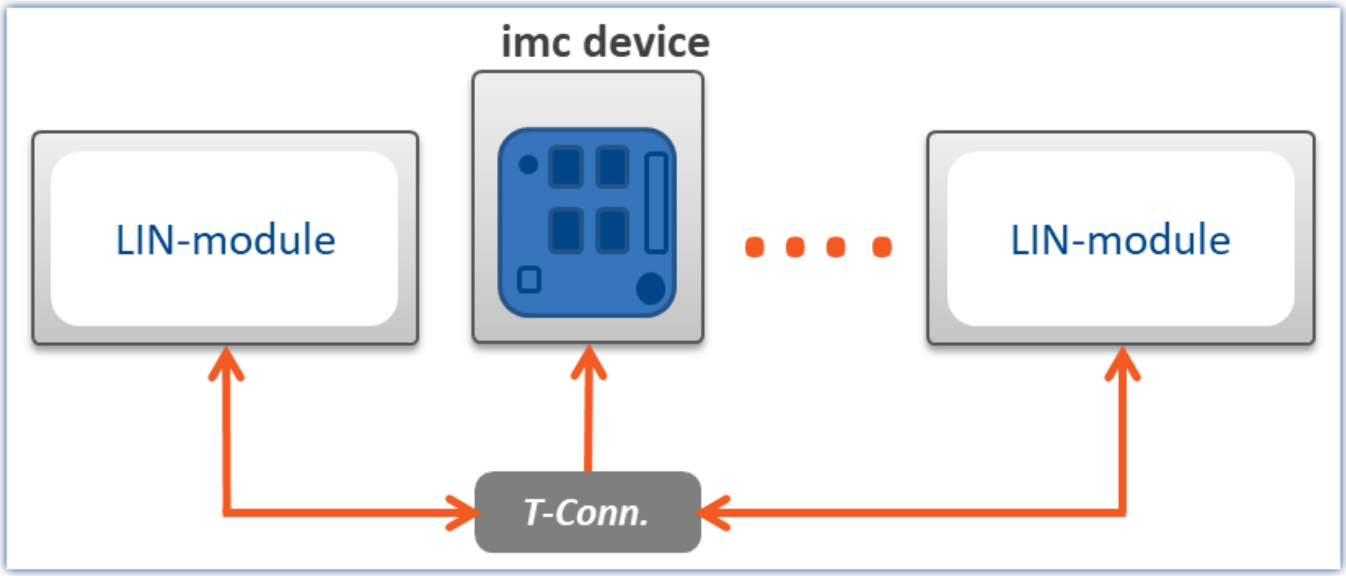
DSUB-PIN	Signal	Description
1	NC	
2	NC	
3	**LIN_GND**	**LIN Ground**
4	NC	
5	NC	
6	**LIN_GND**	**Optional LIN Ground**
7	**LIN_INPUT/OUTPUT**	**LIN bus line**
8	NC	
9	NC	

Note

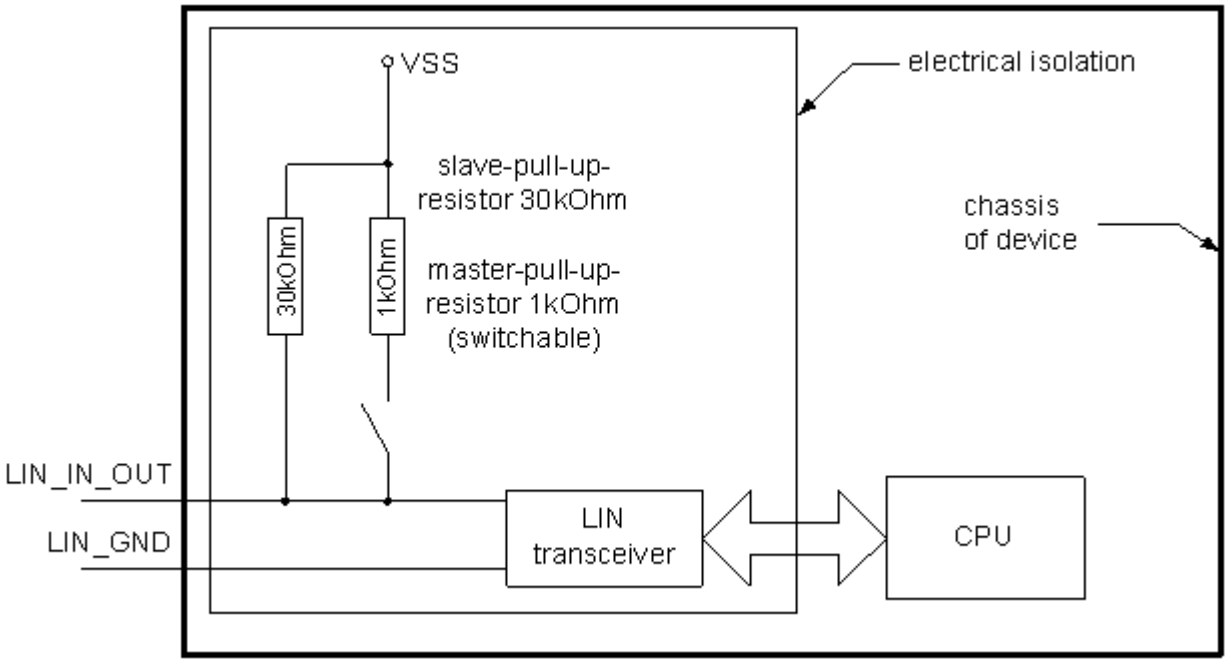
Since the Pin configuration is similar to that of the CAN-Bus, you can also use any CAN-Bus cables already available for the LIN-Bus. However CAN- and LIN-modules may not be connected together to one node!

9.10.8.2 LIN hookup and circuiting

Unlike with the CAN-Bus, no terminators are needed



LIN hookup



Circuiting and isolation for LIN-bus

The illustration shows how a LIN-bus connection is wired. The LIN transceiver with its two signal lines leading outside, LIN_GND and LIN_INPUT/OUTPUT, is electrically isolated from the rest of the imc measurement device; see LIN technical specifications.

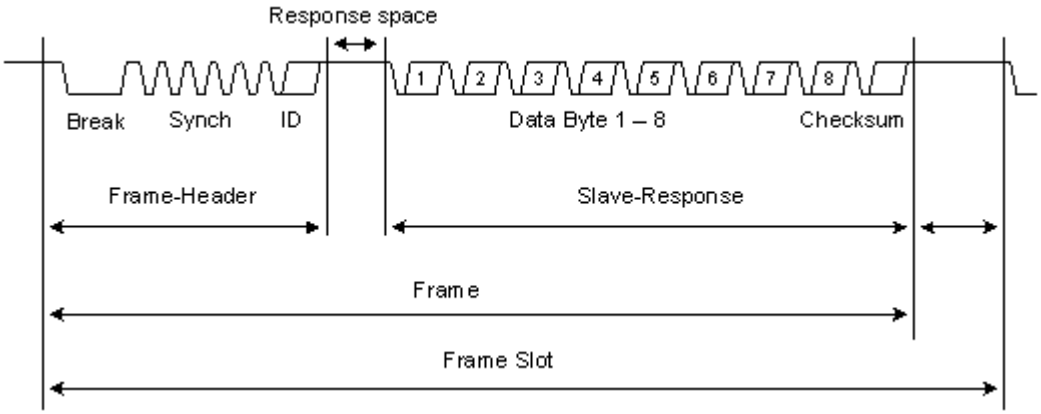
Warning

No LIN-bus should ever be connected whose potential level deviates further from the imc measurement device's potential level by more than this isolation strength.

Pull-Up resistors:

The illustration shows two pull-up resistors. According to LIN specifications, a LIN-Master device must have a 1kΩ pull-up resistor and a LIN-Slave device must have a 30kΩ pull-up resistor. Since imc's LIN devices can alternate between being Master and Slave, according to their configuration, the Master pull-up resistor is switchable.

9.10.8.3 Data transfer on the LIN-bus



Structure of a frame on the LIN-Bus

The frame is the basic element of data transfer on the LIN-Bus. A view of the frame is the best illustration of how data transfer on the LIN-Bus works.

The LIN-Bus transfers the data by means of a Master-Slave scheme. At each LIN-Bus, there is exactly one Master, which retrieves all frames to be transferred from all the Slaves. The Master generates the frame header. The frame header contains the frame identifier which determines the frame to be transferred.

The Slaves connected to the bus receive this frame header, and a certain Slave will normally append its own matching Slave-Response to this frame header. The Slave-Response consists of one or up to eight data bytes and one concluding checksum.

The Master determines when it generates the header when a frame is transferred. Thus, a Slave cannot send transmissions independently.

Note

The checksum can be computed in either of two different ways. In the calculation, Version 2.x incorporates the *Protected-Frame-Identifier* into the checksum, while Version 1.3 does not.

9.10.8.4 Sample applications and operation options

Logging (Monitoring)

The imc measurement device offers the user the ability to monitor an existing LIN-bus. This means that all frames sent by the LIN-bus can be logged as raw data and/or that interesting signals can be extracted from these frames. The signals and the log are set up in the system as measurement channels, and thus can be analyzed subsequent to the measurement.

In the process, the transferred data are not changed. The device behaves passively with respect to the LIN-bus.

Such an application requires that the LIN-bus monitored has a master, without which no data transfer can take place.

Example

Suppose you wish to analyze the data exchange concerning measurements performed on a prototype car door. The master and all slave actuators and -sensors are provided, and their functioning is not to be disturbed.

Slave

Another option which the imc measurement device offers is to act as a slave. The imc measurement device responds by sending the requested frame. The answer can consist of constants and/or variables in a variety of data formats. The variables concerned are Display variables and virtual bits, which can be computed in imc Online FAMOS, for example.

Example

You wish to analyze the data exchange concerning measurements performed on a prototype car door. The master and some slave actuators and/or -sensors are provided. In order for the LIN-bus system to be able to run, some frames of missing Slaves must be substituted. The imc measurement system can perform this task.

Master

Another possibility is to substitute a missing master. The imc measurement system itself then prompts the slaves to send data by generating frame headers for the frames to be sent, at the time intervals set.

Example

Suppose you wish to monitor a slave's data traffic, and don't have an appropriate master. In order for the slave to be able to run, particular frame headers must be generated on the bus to which the slave can respond. The imc measurement system can perform this task.

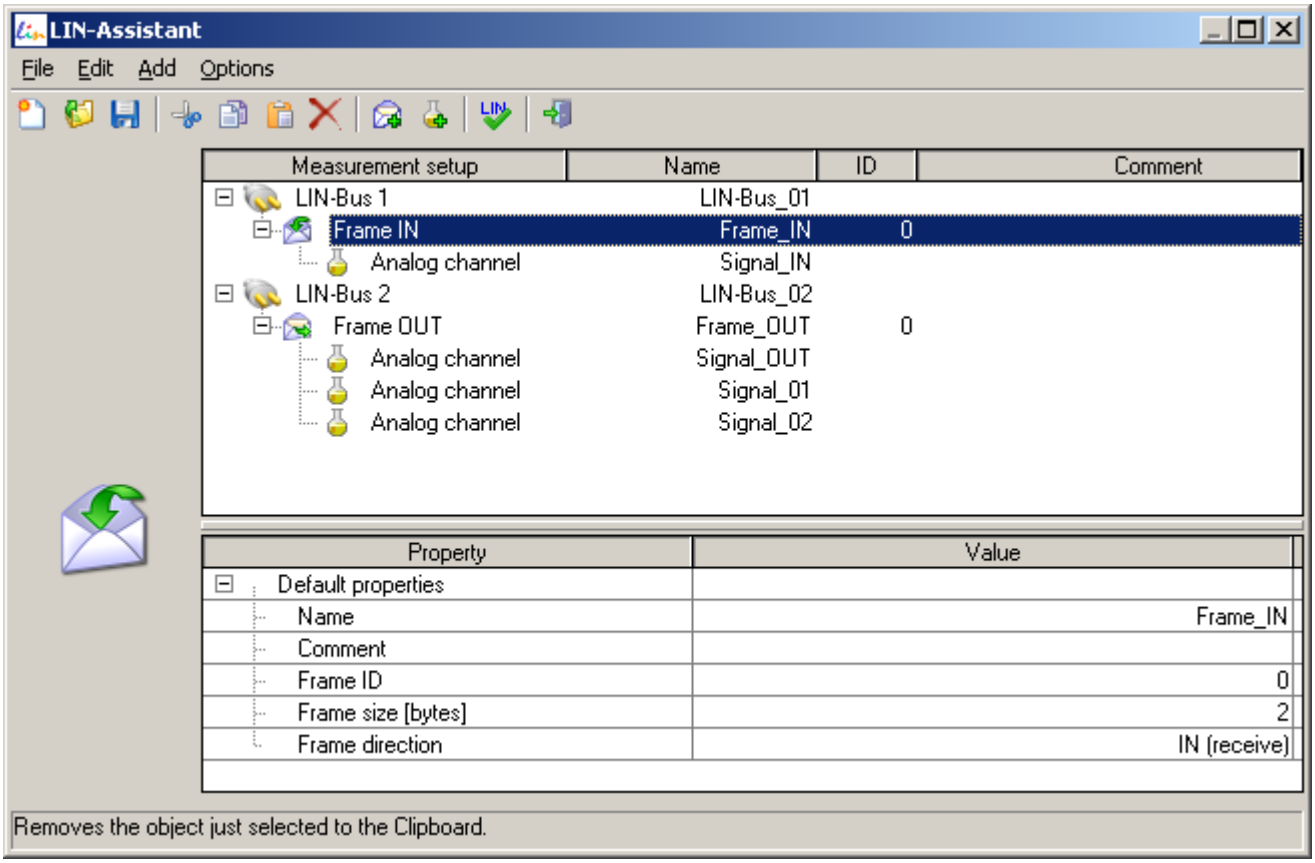
Combination

The imc measurement system is able to handle any combination of the above three scenarios. Thus, LIN-master operation can be carried out with simultaneous logging and simultaneous slave simulation.

9.10.8.5 LIN-assistant

The *LIN-Bus Assistant* is where settings for the LIN-bus terminals are made and frames and channels (signals) are defined.

Start the Assistant from within the imc STUDIO software via the Ribbon *Setup-Configuration > LIN-Assistant*.



LIN-Bus-Assistant

Short Description of the LIN-Bus-Assistant

- The LIN-Assistant is a part of the imc STUDIO software without which it is unable to operate
- It offers easy configuration setting for the data to be received or sent
- Settings for measurement channels in imc STUDIO to which received data are to be written

9.10.8.5.1 Properties of the LIN-assistant

- Simultaneous functioning as LIN-Slave and as LIN-Master (the master is limited, see Limitations of the LIN-Assistant 642)
- Configuration of the frame and of the signals
- The following data types can be received and sent:
  - Individual bits
  - Unsigned integers from 2 to 32 bits.
  - Signed integer from 2 to 32 bits
  - Floating point (real number) with 32 bits and 64 bit as per IEEE 754: Standard for Binary Floating-Point Arithmetic

- Conversion of the received LIN-bus data (integers) with factor and offset (linear transformation), floating point numbers cannot be transformed
- Conversion of the data to be sent (integers) with factor and offset, before they are written to the bus (linear transformation), floating point numbers are not transformed.
- Data to be sent can be:
  - Constant values stored in the LIN-Assistant
  - Values which are stored as a Display variable, Process-Vector-variables or virtual bit at the moment of the Master request. These variables are calculated imc Online FAMOS and written to this variable. The values can also be entered directly into the Display variables with the help of a Display, if one is connected.
- Exchange of LIN-bus configurations between different experiments: Saving and loading of the configurations on the hard drive as a *.LCF file
- Import of *.LDF – files possible, (not all the LDF-features are supported)
- Configurations can be created without a mouse (using keyboard only)
- A bus protocol is possible: all frames along with their ID and data bytes are listed with a time-resolution of 100µs
- All measurement devices in the experiment which are equipped with LIN-nodes are displayed simultaneously with their LIN-boards and connectors
- A comment can be entered for each frame
- Visual display of the signal positions in the frame (familiar from the CAN-Assistant)
- Check of configuration settings for feasibility (no duplicate assigned FrameIDs, no invalid signal positions within the frame, etc.)
- Error handling
  - Timeout error > replacement values can be inserted

9.10.8.5.2 Limitations of the LIN-assistant

Only incomplete LIN 1.3 transformation

- No nodes can be defined, only signals in the frames
- No diagnostic signals definable
- No diagnostic frames definable
- No event-triggered frames definable
- No freely definable schedule-tables can be defined for the normal operation as Master. A frame polling sequence is automatically created from the configured frames.

Only incomplete LIN 2.0, 2.1 transformation

- No nodes can be defined, only signals in the frames
- No diagnostic signals definable
- No diagnostic frames definable
- No event-triggered frames definable

When acting as Slave:

- No acceptance of or response to AssignNAD, AssignFrameID ...etc. possible

When acting as Master:

- No automatic configuration of all connected nodes possible when starting with AssignNAD, AssignFrameID ...etc.
- No freely definable schedule-tables definable for normal operation. A frame polling sequence is automatically created from the configured frames.
- No diagnostic schedule tables
- No configuration schedule tables
- No sleep-schedule tables

9.10.8.5.3 Overview window

The Overview window is the upper window in the dialog. In it, the structure of the measurement setup as it relates to the LIN-configuration is displayed schematically. You are shown the measurement devices ( ) connected with the experiment, and its name. If there is only one LIN-capable device in the experiment anyway, there is no extra icon for it.

The LIN-modules ( ) (slots) installed are indicated under each measurement device, along with the two LIN-connections ( ) (LIN-busses, LIN-nodes) which these have.

For each LIN-bus, the frames ( ) already defined with its associated signals ( ) (channels) are indicated.

9.10.8.5.4 Properties window

The Properties window is visible in the lower portion. It displays the editable properties and the settings options for the object currently selected in the Overview window. If multiple objects are selected at the same time, the Properties window shows a selection of suitable properties.

To change a property's value, click on the desired value.

Depending on the property, either a pop-down list or a text box is used.

9.10.8.5.5 Items in the main menu

Menu item		Description
	New	Removes all previously created frames and channels and creates an empty LIN-configuration.
	Load	Enables import of an entire LIN-configuration from a file. The file extension for such a file is *.LCF. The files are written in ASCII format and can be viewed in a text editor. However, it is not recommended to edit the files by hand. The second file extension is *.LDF. This file format has been defined by the LIN Consortium. The Revision 2.0 and 2.1 is supported. In an LDF file, only one LIN-Bus is described, and whenever an LDF file is loaded afterwards, this description is on the first LIN-Bus in the overall system. All other LIN-busses are deleted. Since the LIN-Assistant does not support all the capabilities and properties demanded by the LDF files, only the capabilities and properties which are supported are loaded. The following data are not loaded but ignored: • Nodes (assignment of nodes to frames discarded.) • Node_attributes (only LIN-Version 1.3 or 2.0, 2.1 is loaded, and the corresponding frames are used) • Node_composition, init_value and published_By for signals • Byte Arrays are converted to individual byte signals • Diagnostic_signals, Diagnostic_frames, Signal_goups, dynamic_frames, Sporadic_frames, Event_triggered_frames, schedule_tables • With the signal_encoding_types, only one physical_value and no logical_values, bcd_values and no ascii-values can be loaded.
	Loading a LIN-Bus	Enables import of a LIN-Bus (node) from an LCF- or LDF file. To do this, select the LIN-Bus to load and click on the menu item "*File*" > "*Load a LIN-Bus*". The system prompts you for the file to load, and the selected LIN-Bus is filled with the data from the file. For LCF files, the following applies: only the first LIN-Bus found in the specified file is used for the substitution.
	Save as	Saves the complete LIN-configuration in an LCF-file. (rf. Load...)

9.10.8.5.6 Further menus

Edit - menu

Menu item	Description
Copy Paste Cut Delete	These are the familiar Windows Clipboard commands.

Add - menu

Menu item	Description
Add a frame	To add a frame, select in the Overview window either • a LIN-bus , • an already existing frame or • an already existing signal , and click on "*Add*" > "*Frame*". The new frame is set up under the selected LIN-bus below all previously existing frames.
Add a signal	To add signals, select in the Overview window either • an already existing frame or • an already existing signal and click on "*Add*" > "*Signal*". The new signal is set up under the selected LIN-bus below all previously existing signals.

Options - menu

Menu item	Description
Testing the LIN configuration	When this command is carried out, the entire LIN-configuration is checked for errors. Any error is written in the status bar in red typeface. In most cases, the focus shifts to the defective object in the Overview window, to enable quicker debugging.

9.10.8.5.7 LIN-bus (connection) properties

Property	Value
Default properties	
Name	LIN-Bus_01
Comment	
Baud-rate [BPS]	19200
Master settings	
Master type	Master
Time for one polling cycle [ms]	500
INFO: fastest possible polling cycle [ms]	5
Sleep-WakeUp-Command-Display-Vari...	
Duration of the Master-Break [Bit-times]	13
Duration of the Master-Break-Delimiter [...	1
Duration of a Wakeup-Break [us]	1000
Log settings	
Log all frames on the bus	Yes
Channel name for the log	LIN_FrameProtocol01

LIN bus properties

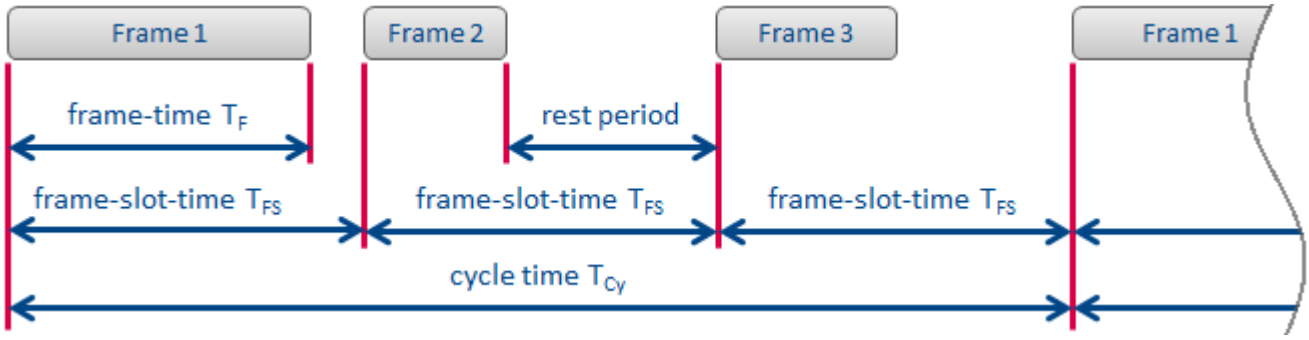
Default properties

Default properties	Description
Name	Here you can enter an arbitrary name for the LIN-bus, which will appear in the Overview window's Name column. This property serves the purpose of providing a brief description.
Comment	The comment property can contain a detailed description of the LIN-Bus.
Baud rate	This property specifies the LIN-bus's bit transfer rate. There is a pop-down list containing pre-defined standard values, but arbitrary values can also be entered. According to LIN-specifications, Baud rates of 1000 to 20000 bits per second are permitted.

Master settings

Master settings	Description
Master Typ	Here you can set whether the imc measurement device is to be the master on the LIN-bus. If Master mode is set, then as soon as the measurement is prepared, the imc measurement device polls all frames defined in the LIN-bus at user-set intervals. See Sample applications and operation options 640. The following parameters only appear if "*Master type*" is set to "*Master*".

Time for one polling cycle [ms]



Setting the polling cycle time $\mathbf{T_{Cy}}$ determines how much time is taken for polling all frames defined. The number of frames to poll is referred to here as **n**. The cycle time consists of **n** equally large time chunks which to which we refer here as frame-slot-time. Exactly one frame is transferred in each frame-slot. Since not all frames are of the same size, after transfer of one frame, a pause of varying length occurs.

The longest frame's required transfer time $\mathbf{T_{FMax}}$ determines the minimum frame-slot-time $\mathbf{T_{FSMin}}$. Taking into account the frame count **n**, the smallest possible cycle time $\mathbf{T_{CyMin}}$ can be found according to the following equation:

Where:

$$T_{CyMin} = n \cdot T_{FSMin}$$
$$T_{FSMin} = T_{FSMax}$$

The transfer time $\mathbf{T_F}$ for one frame is calculated according to the equation below:

$$T_F = 1.4 \cdot \frac{34\ \text{Bits} + 10 \cdot \frac{\text{Bits}}{\text{Bytes}} \cdot (n_{DataBytes} + 1\ \text{Byte})}{v_{Bit}}$$

Key to variables and numbers:

1.4	Safety factor as per LIN specifications
$34\ \text{Bits}$	The frame header requires approx. 34 bit times
$10 \cdot \frac{\text{Bits}}{\text{Bytes}}$	10 bits per byte (8 bits + Start- and Stop-bit)
$n_{DataBytes}$	Number of data bytes to be transferred in the frame *[Byte]*
$1\ \text{Byte}$	One byte needed for the checksum
v_{Bit}	Baud rate of the LIN-bus *[Bits/s]*

One can transform this equation to obtain this formula for the frame transfer time:

$$T_F = \frac{14 \cdot n_{DataBytes} + 61.6}{v_{Bit}}$$

The units' quantities must be stated as above.

Master settings	Description
INFO: fastest possible polling cycle [ms]	This table row indicates the smallest polling interval currently possible. This time interval changes if frames are added or removed. It also changes if you change the number of bytes in the longest frame, see "*Polling cycle time [ms]*".
Sleep-WakeUp-Command-DisplayVariable	A Display variable selected in the LIN-Assistant can be used to activate the Sleep- or WakeUp-mode. See "*LIN-PowerManagementCommands* 655"
Duration of a Wakeup-Break (µs)	250µs – 9000µs Specifies the duration of the Wakeup-break in microseconds. This break is sent either by the Master or a Slave during a LIN-wakeup. Sending a WakeUp-break is possible by writing certain values to the Sleep/WakeUp-command Display variable.
Duration of the Master-Break (bit times)	13- 15 Bit Specifies the duration of the MasterRequest's break. Unit: bit-times (see also LIN-Secification-Package Rev2.1 Section 2.3.1.1 Break Field).
Duration of the Master-Break Delimiter (bit times)	1-3 Bit Specifies the duration of the Master-Request's Break-Delimiters in bit-times (see also LIN-Secification-Package Rev2.1 Section 2.3.1.1 Break Field).

Note Possible combinations

	Break=13 BT	Break=14 BT	Break=15 BT
Delimuter= 1BT	x		
Delimuter= 3BT	x	x	x

Log settings

Log settings	Description
Log all frames on the bus	If this is set to Yes, a log channel is created which records all messages on the LIN-bus, as long as the checksum is correct upon reception. If a frame-header is received on the LIN-bus, to which no slave responds, then this is also entered into the protocol so that the corresponding Frame-ID can be determined.
Channel name for the log	This property sets the name for the log channel.

9.10.8.5.8 Frame (message) properties

Property	Value
Default properties	
Name	Frame_OUT
Comment	
Frame ID	0
Frame size [bytes]	2
Frame direction	OUT (send)
Checksum version	LIN 1.3

Frame properties

Default properties	Description
Name	Here, an arbitrary name can be entered for the frame. It will also be displayed in the Overview window under the Name column. This property represents a brief description.
Comment	The comment is for providing a more detailed description of the frame.
Frame ID	The Frame-Identifier is an integer from 0 to 63. A Frame-ID can appear only once in the group of incoming and once in the group of outgoing frames. The same Frame-IDs can be defined in different LIN-busses, however. Either a decimal or, if "0x" is prefixed, a hexadecimal number must be entered. IDs are always displayed as numbers in decimal format.
Frame size	This property sets the number of data bytes the frame has. This must be between 0 and 8. The specification V1.3 applies, according to which the **Frame-ID** and the **frame size** can be selected independently of each other. • If more bytes are set for the definition of the incoming frame than which are actually received for the frame via the LIN-bus, the missing bytes are filled with zeroes at the end of the frame. • If fewer bytes are set for the definition of the incoming frame than which are actually received for the frame via the LIN-bus, then the surplus bytes are ignored.
Frame direction	This property determines whether the frame is to be received and interpreted by the measurement device (incoming, receiving, IN) or whether this frame is to be assembled and sent (outgoing, sending, OUT).
Checksum version	For the case that the frame is to be sent or received, it is possible to use this property to set the type of checksum calculation. If the computation of the checksum is to be calculated as per the LIN-specification V1.3, then set the value to **LIN 1.3**. For the calculation as per the LIN-specification V2.0/2.1, this value is to be set to **LIN-2.0**. For received frames, the value **Either** is possible: If a frame with the ID of the selected frame is received, it must conform to the checksum rules according LIN-Spec V1.3 OR LIN-Spec V2.0 in order to be recognized as valid. As the default setting, this mode is also referred to as the compatibility mode.

9.10.8.5.9 Signal (channel) properties

Property	Value
Default properties	
Name	Signal_IN
Comment	
Data interpetation	
Data type	Signed integer
Start-Byte	0
Start-Bit	0
Bit width of number	16
INFO: Byte order	Intel
Physical unit	
Factor [1/LSB]	1
Offset []	0
Error handling	
Handle timeout error	No

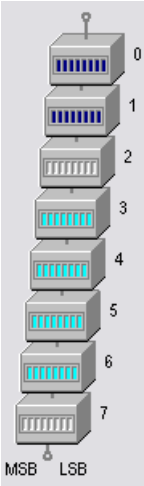
Signal properties

Default properties	Description
Name	Sets the channel's name, with which one works in imc STUDIO. This name must conform to the rules and the format applicable to a imc STUDIO channel name.
Comment	Specifies the comment for this channel.

Data integration	Description
Data type	This sets the signals type, or rather, how the data on the bus are interpreted. The following data types are possible:

Data type	Description
Signed Integer	The bit widths available for this data type are 2 to 32 bits.
Unsigned Integer	The bit widths available for this data type are 2 to 32 bits.
Floating point	Either 32-bit or IEEE 64-bit floating point format are available.
Bit	A single bit as a digital signal.

Start-Byte and Start-Bit



These two properties determine at which position within the frame the signal begins.

To simplify the positioning of the signals, a signal position representation is displayed as with the CAN-Assistant. This display is always shown whenever you select a signal in the Overview window. It shows the frame and its individual bytes, which in turn contain the bits.

- The bits appearing in dark blue represent the bits which are in use by the signal currently selected.
- The bits appearing in light blue represent the bits which are already in use by other signals belonging to the frame.
- If any signal is so configured that it would overlap the end of the frame, that signal appears red in the display.

Outgoing signals may not occupy the same bits in the frame; for incoming signals this is not a problem.

Data integration	Description
Bit-width of number	This property sets the signal's length. Different maximum and minimum values can be entered, depending on the data type.
Byte order	Per the LIN-specification, only data conforming to the INTEL byte order should be transmitted on the LIN-bus. For that reason, this property cannot be changed and is only intended for the user's information.
Physical unit	Here you can enter a physical unit or its abbreviation, e.g. "V" for Volt. Use SI-units wherever possible. It is not advised to enter any scaling prefix such as m for milli-. Instead, adjust the factor correspondingly. The imc STUDIO curve windows automatically add such prefixes to correspond to the measured values.
Factor and Offset	If the signal currently selected happens to be of the integer data type, signed or unsigned, then both the properties **Factor** and **Offset** are displayed. These values are parameters for linear transformation, which are computed by the following formulas. **Incoming signal**: $y = f \cdot x + a$ **Outgoing signal**: $x = \frac{y}{f} - \frac{a}{f}$ Variable definition y - physical value x - integer on the Fieldbus f - factor a - offset An incoming signal is a 16-bit unsigned integer. Thus, the integer's numerical range is 0 LSB to 65535 LSB. The physical quantity which it represents is a voltage between 10 V and 20 V. The factor is thus given by the equation $f = \frac{20\ \mathrm{V} - 10\ \mathrm{V}}{65535\ \mathrm{LSB}} = 1.526 \cdot 10^{-4} \frac{\mathrm{V}}{\mathrm{LSB}}$ and the offset is $a = 10\ \mathrm{V}$ Supposing the following integer were imported from the bus, $x = 13483\ \mathrm{LSB}$ it would correspond to a voltage of $y = f \cdot x + a = 12.057\ \mathrm{V}$

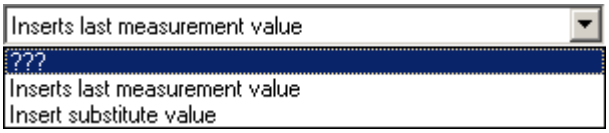
Handle timeout error

The term timeout-error refers to a case where no measured value is received from the fieldbus after a certain period; error handling can be set to deal with such an occurrence.

Parameter	Description
Handle timeout error	This settings option determines whether error handling of any timeout error would be performed.
Timeout Interval [s]	This is where to set the time limit after which a timeout-error is considered to have occurred.

Error handling action

This setting determines the action to be performed in case the timeout time limit elapses.



For this purpose, the following options are available, whose effects depend on whether a channel is sampled equidistantly or recorded with a time stamp.

Error handling	Equidistant sampling	Samples with time stamp
Last value	The last valid value is returned until a new value arrives.	After elapse of the timeout time, a new sample is generated with the last value.
Substitute value	The substitute value is returned until a new value arrives.	After elapse of the timeout time, a new sample is generated with the substitute value.

9.10.8.5.9.1 Special üroperties of the bit data type

Property	Value
Default properties	
Name	Signal_IN
Comment	
Data interpetation	
Data type	Bit
Start-Byte	0
Start-Bit	0
Name of the 16-bit port	LIN_BitPort01
Error handling	
Handle timeout error	Yes
Timeout-interval [s]	1.0
Error handling action	Insert substitute value
Substitute value	0

Special properties of the bit data type

Name of the 16-Bit-Port

If you have set the data type for an **incoming** signal to Bit, this editable property is displayed. It is the name of the digital 16-bit port into which the individual bit is sorted. With this data type, 16 individual bits are bundled for data reduction reasons. In the trigger machine, it is always only possible to trigger the entire port at once.

An editable drop-down list box is used to change the setting; all bit ports defined for this LIN-bus appear in the list.

A variety of actions can be taken here:

Action	Description
Renaming the bit-port in which the selected bits are located	The 16 bit-port assigned to this bit signal can be renamed by entering a new name. In that case, the bit-port is renamed for all signals which it contains.
Assigning the selected bits to another, already existing bit-port	By selecting a bit port already indicated in the list, the selected bits in this bit-port are moved. If the target bit-port is already full, then an error notification is posted in red in the status bar.
Assigning the selected bits to a new bit-port	By selecting the list item "*New...*", the selected bits are moved to a new bit-port. Their name is automatically generated and can subsequently be renamed by clicking once again on this property.

Note

Empty bit-ports are automatically deleted. In such cases, the settings made in the imc STUDIO: Setup are deleted along with them. This means that settings such as the sampling rate, time stamp and triggering are lost even if a new bit port with the same name is created immediately afterward, before exiting the LIN-Assistant.

9.10.8.5.9.2 Special properties for sending

Signal souce for the signal to be sent	Constant value
Constant value []	5.372

Signal source for the signal to be sent

If you have defined signals within a frame to be sent, then the signal source is set by means of this property.

There are different options available depending on the data type.

Data type	Available selection
Integers Floating point number	All of the measurement device's **Display variables** in which the signal was created; and constant value.
Integers, Bits Floating point number	Process vector variables (pv-variables). See also the note on data type conversion further below 654.
Bits	All of the measurement device's virtual bits in which the signal was created; and constant value.

By means of a particular frame-header, the Master determines the point in time at which the data are to be transmitted. The values to be send extracted from the Display variables and/or virtual bits upon reception of a frame-header and joined with the constant values into a frame. Then the data bytes are appended to the frame-header and sent along with them.

An imc Online FAMOS command such as `CAN_Send(...)` is not sensible for this bus, since the slave isn't able to send independently.

Note

- The values in the Display variable are expected to be expressed in **physical units**, because if integers are to be sent, they are processed with a factor and offset and afterwards transmitted.
- With pv-variables, there are limitations due to the conversion between data types. This is necessary because the source and target each can use different data type.
  In the data acquisition device, pv-variables with the following data types can be administered: Bit16, INT16, UINT16, INT32, UINT32, FLOAT and Bit1. By contrast, the LIN receiving device can only process SINT, UINT, FLOAT and BIT. In the conversion, it can occur that the numerical range of the sent data type is either not, or only partially, covered in the receiving data type. Such cases are indicated by the software with a warning message. For a detailed description of this topic, please contact our tech support.

Constant value

If for the property "**Signal source for the signal to be sent**" the value "**Constant value**" was selected, then this property is visible. The constant value to be sent can be entered here.

Note

The constant value is expected to be expressed in **physical units**, because if integers are to be sent, they are processed with a factor and offset and afterwards transmitted.

9.10.8.5.10 LIN-PowerManagementCommands

Issuing Sleep-/ WakeUp-commands via a Display variable

General notes

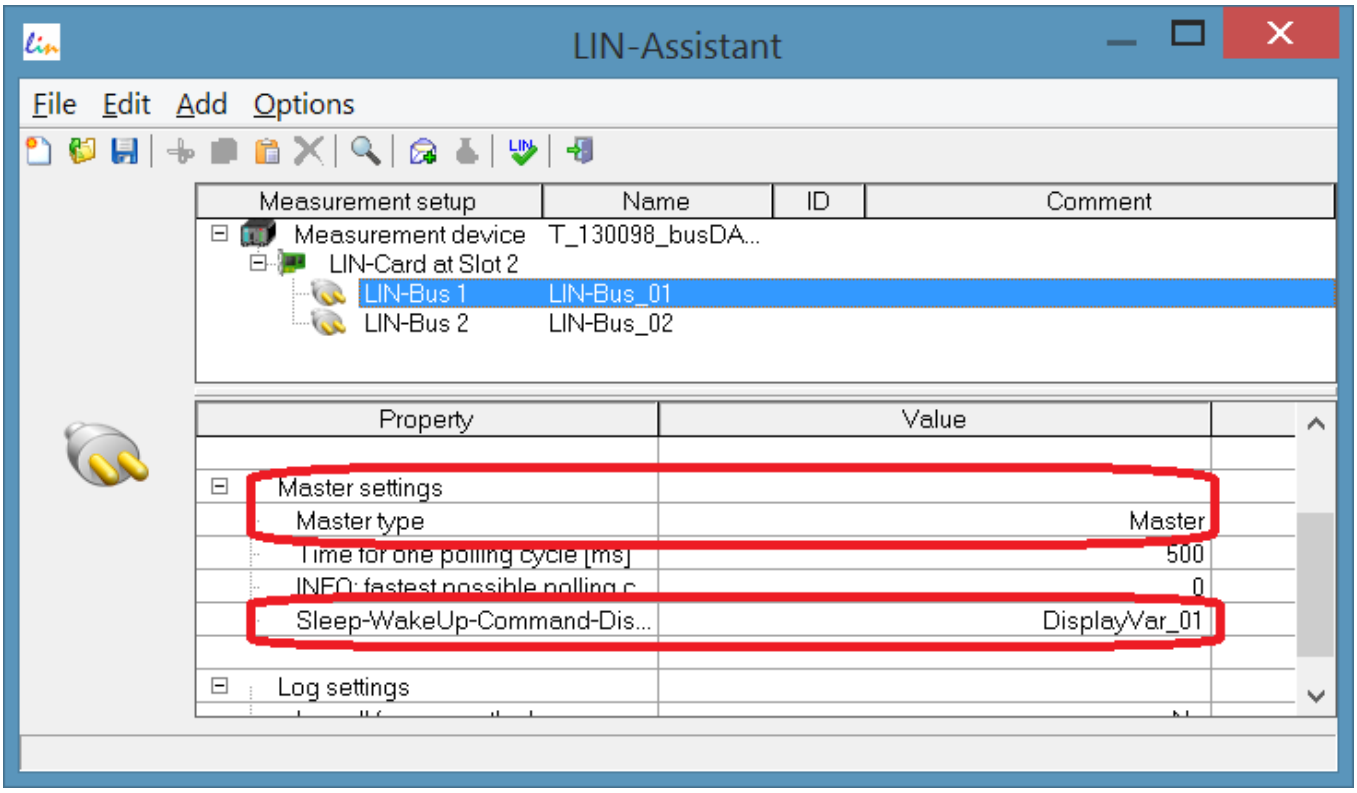
The capabilities described here pertain to the section "2.6 NETWORK MANAGEMENT" of the LIN-specifications "LIN Specification Package Revision 2.1". That is where the functioning of "Wake up" and "Go to sleep" are defined.

Note that *Sleep/WakeUp-mode* here pertains to the LIN-Bus and should not be confused with the *Sleep/Resume*-function of an imc BUSDAQ unit.

A Display variable selected in the LIN-Assistant can be used to activate the Sleep- or WakeUp-mode. This function is only available if the imc LIN interface (port) is configured as the LIN-Master. To change the Sleep/WakeUp-mode, a 16-bit word is written to the Display variable. If only one LIN-interface (port) is addressed, then once the command has been applied, the value of the Display variable reverts to zero.

Executing the command

The Display Variable is polled cyclically. As soon as its value becomes nonzero, the command is executed. It is only necessary to set the *Master Type* to *Master* and to select a Display Variable. Assignment of the Display Variable's value is performed by means of imc Online FAMOS or by setting a variable in imc STUDIO, as examples.



Structure of the 16-bit word:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
4 Bit `Status`				4 Bit `Slot`				4 Bit `Port`				4 Bit `ommand`			

Description of the bitfields:

Bits	Description
0-3	Command; available values: 0 -no function 1 -Sleep 2 -Wake Up 4 -Status query (only possible for individual ports, Slot may not be 15) 5 -Status response
4-7	Port; available values: 0 -no function 1 -Port1 (LIN-interface 1 of the LIN-Bus-card(n) selected as Slot) 2 -Port2 (LIN-interface 2 of the LIN-Bus-card(n) selected as Slot) 15 all ports of the LIN-Bus-card(n) selected as Slot
8-11	Slot; specifies the Fieldbus-slot of the LIN-Fieldbus card addressed. Other fieldbuses are counted. 0 -no function 1 -Slot1 ... N -SlotN ... 8 -Slot8 15-all slots in the device with LIN-Fieldbus cards
12-15	Status; returns the port's status. Available values: 0 -no function 1 -The port enters Sleep mode. 2 -The port is in Sleep mode. 3 -The port is in operational mode.

Example

The commands for the example in the image are then:

0x0211: Sleep

0x0212: Wake up

9.10.9 MVB-bus interface

What is the MVB-Bus?

MVB is an abbreviation for "Multipurpose Vehicle Bus". This bus is a serial communication bus for rail vehicles. It interconnects control devices with each other as well as with simple sensors and actuators.

MVB conforms to the standards of the International Electrotechnical Commission (IEC) and the German Deutsches Institut für Normung (DIN).

Reference Literature and Standards

[1] DIN EN 61375-3-1: Elektronische Betriebsmittel für Bahnen - Zug-Kommunikations-Netzwerk -Teil 3-1: MVB - Multipurpose Vehicle Bus (IEC 9/1276/CDV:2009)

English version: FprEN 61375-3-1:2009

The official English original of the international document IEC 9/1276/CDV:2009 "Electronic Railway Equipment - Train Communication Network - Part 3-1: MVB - Multipurpose Vehicle Bus" (draft in the survey) has been adopted in this design draft unchanged.

Prerequisites

Prerequisites	
Hardware	An imc device with a MVB-Bus interface.
Software	The system requirements are described in the data sheet of our fieldbus connection options.
Password	Normally not required. Only when customer-specific files are to be imported, the possibility of password protection 661 is available. When in doubt, contact our tech support 8.

9.10.9.1 Definitions

Term	Description
MVB	Multipurpose Vehicle Bus
Controller	A controller can be connected to exactly one MVB. To date, one controller is attached per MVB interface.
Frame	A collection of certain individual signal values in a single data block. A frame also has a Frame-Identifier (FrameID) which enables inferences about the data (signals) it contains. A frame is often also referred to as a message.
FrameID	Frame-Identifier. This is a 16-bit value subdivided into the (4-bit) FCode and the (12-bit) Port Address. See also [1] "DIN EN 61375-3-1"
F-Code	Describes the frame's type. There are 16 different possible (4-bit) values. For example, the frame's length is encoded in the F-code for process data-frames. See also [1] "DIN EN 61375-3-1"
Port Address	This is the lower 12 bits of the FrameID. These 12 bits, however, are only the Port Address if the F-Code is 0 through 4. Otherwise, these bits mean something else.
Process data-Frames	Frames with FCodes 0 through 4. The frame's "payload" is divided among individual signals. See also [1] "DIN EN 61375-3-1"

Term	Description
"Special-Frames"	These are frames with FCodes 5 through 15: Mastership-Transfer-Frames, General-Event-Frames, Message-Data-Frames, Group-Event-Frames and Device-Status-Frames. See also [1] "DIN EN 61375-3-1"
Fieldbus card (Interface)	This is a circuit board produced by imc which can be installed in an imc device. This card serves to connect the associated Fieldbus.
Channel	A channel is assigned to a signal which is registered with the system. The channel is the timed sequence of individual signal values received via the bus.
Signal	A signal is a part of a frame.

9.10.9.2 Capabilities and limitations

- Logging of process data frames from the bus. (I.e. frames with the FCodes 0 through 4)
- Disassembly of process data frames received into signals (measurement channels)
- No reception of "Special-Frames" having FCodes from 5 through 15. Thus, no Mastership-Transfer-Frames, General-Event-Frames, Message-Data-Frames, Group-Event-Frames or Device-Status-Frames are received.
- No sending of frames and data possible
- Import of customized bus configuration files (*.xml)
- Measurement of up to 512 signals per device possible (including all other channels)
- The aggregate sampling rate per MVB interface is about 40000 samples/second

9.10.9.3 General description of MVB

The **M**ultipurpose **V**ehicle **B**us (MVB) is a "multi-purpose"-Fieldbus for use within a rail vehicle. In the 1990s, this Fieldbus was developed as a diagnostics system.

The MVB links up programmable systems with each other and provides a direct connection to simple sensors. It conforms to the *standard IEC, TC9, WG 22 in the standard Electric Traction Equipment - Train Communication Network, Parts 1 through 3.*

It is a true data bus by means of which multiple bus nodes can be connected to one line. At any one time, only one of these nodes may send data. These data are then received by all other nodes. The data are sent using the time multiplex procedure at a gross transfer rate of 1.5 MBit/s. The signal levels on the line are the RS485 levels.

There are three different transmission media:

- EMD (**E**lectrical **M**iddle **D**istance bus): Twisted wire pair with insulation of the bus interfaces by means of transducers.
- ESD (**E**lectrical **S**hort **D**istance bus): Twisted wire pair with additional uninsulated ground line or insulated by optical coupler. Due to the optical coupler's limited input voltage range, a compensation line is needed for bridging the grounds of the optical coupler components.
- OGF (**O**ptical **G**lass **F**ibre): Fiberglass line. Point-to-point connections with so-called star couplers.

EMD is the simplest bus.

Practically all rail vehicles by the manufacturers Siemens and AEG/ABB/Adtranz/Bombardier since around 1995 are equipped with MVB as the Fieldbus. In such cases, the MVB connects the most important components of the guidance engineering. This includes the drive control, train protection, driver cabin indicators, central control units, decentralized input/outputs, automatic door control, air conditioning, auxiliary converters, as well as others.

The MVB's role is to transmit as its "process data" the control and status commands for the train's automation (such as the time or the speed), which are short but whose timing is absolutely critical, in real time (cycles of typically 16 ms to 512 ms). It is also used for transmission of diagnostics messages, etc.

Requirements and features

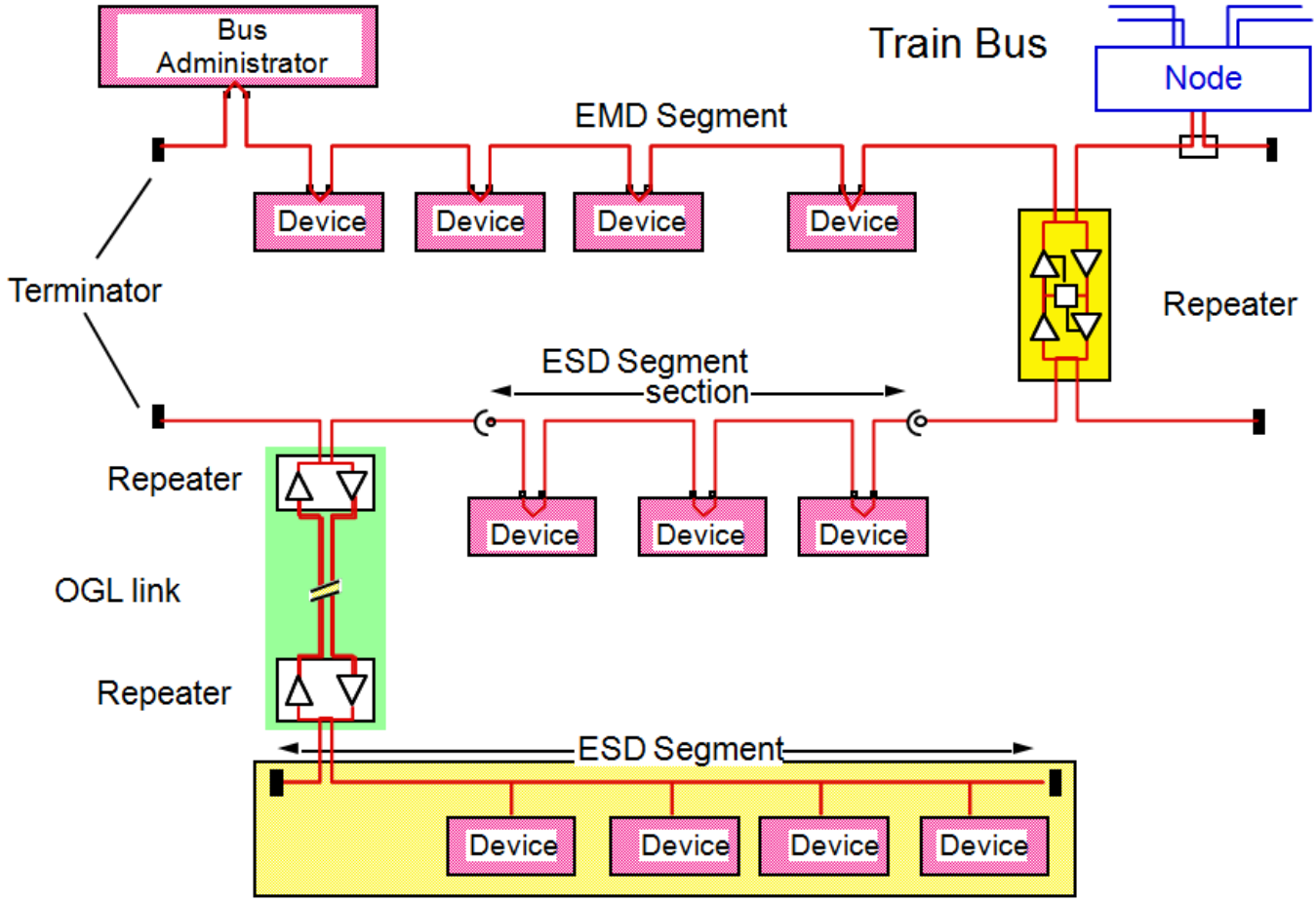
Rail vehicles present electronic equipment with difficult conditions. For this reason, the MVB bus has properties not possessed by normal computer systems' busses.

- Protection against strong **magnetic fields**
- Expanded **operating temperature range**
- High **mechanical stability** for controllers, cables and sensors
- For short line lengths, copper lines are used; optical fiber lines for long lines. With copper wire, it is possible to conduct the supply in the same line.
- **Redundant layout** and cabling
- The MVB uses a medium-independent signal transmission with **Manchester encoding**.
- The Bus management can be be switched among different positions by means of a **Token Frame**.
- The data can be sent either cyclically or sporadically.

Summary of properties

Property	Characteristics	
Transmission medium	Copper: twisted pair, RS485	Optic lines
Topography	Bus	Star
Line length	30 m with up to 32 subscribers	2000 m
Redundancy	Duplication: messages sent on both lines, received on only one. Redundancy by means of alternating master	
Gross data rate	1.5 MBit/s	
Reaction time	typically 4 µs, max. 43 µs	
Address room	4095 physical devices, 4095 logical ports, 8-bit station addresses for messages	
Frame size	16,32, 64, 128 and 256 Bit	
Access control	Central master. Control can be exchanged between different masters.	
Operation mode	Cyclical for process data, sporadic for message data with arbitration	

9.10.9.3.1 MVB topology



MVB uses one baudrate for the complete system.
The segments are connected via repeater.

9.10.9.3.2 Connection and activation

- The MVB-Bus may only be connected with the measurement device once it has been completely wired up.
- Only then may the device be activated.
- During operation, the MVB connectors must not be unplugged.

9.10.9.4 MVB-assistant

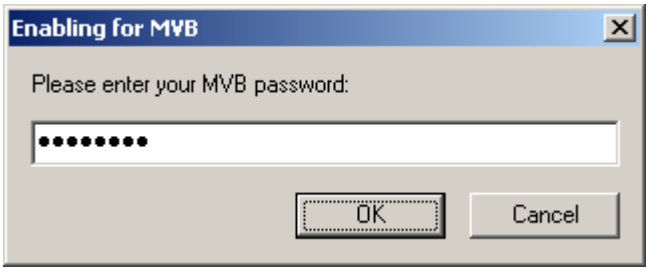
The MVB-Assistant offers the ability to make settings for the imc device's MVB terminals. This includes definitions for the frames and channels (signals) to be received.

The entire MVB-configurations of all controller in the experiment or even only the configuration of a controller can be saved/loaded to a file (*.mvbaf).

Start the Assistant from within the imc STUDIO software via the Ribbon *Setup-Configuration > MVB-Assistant*.

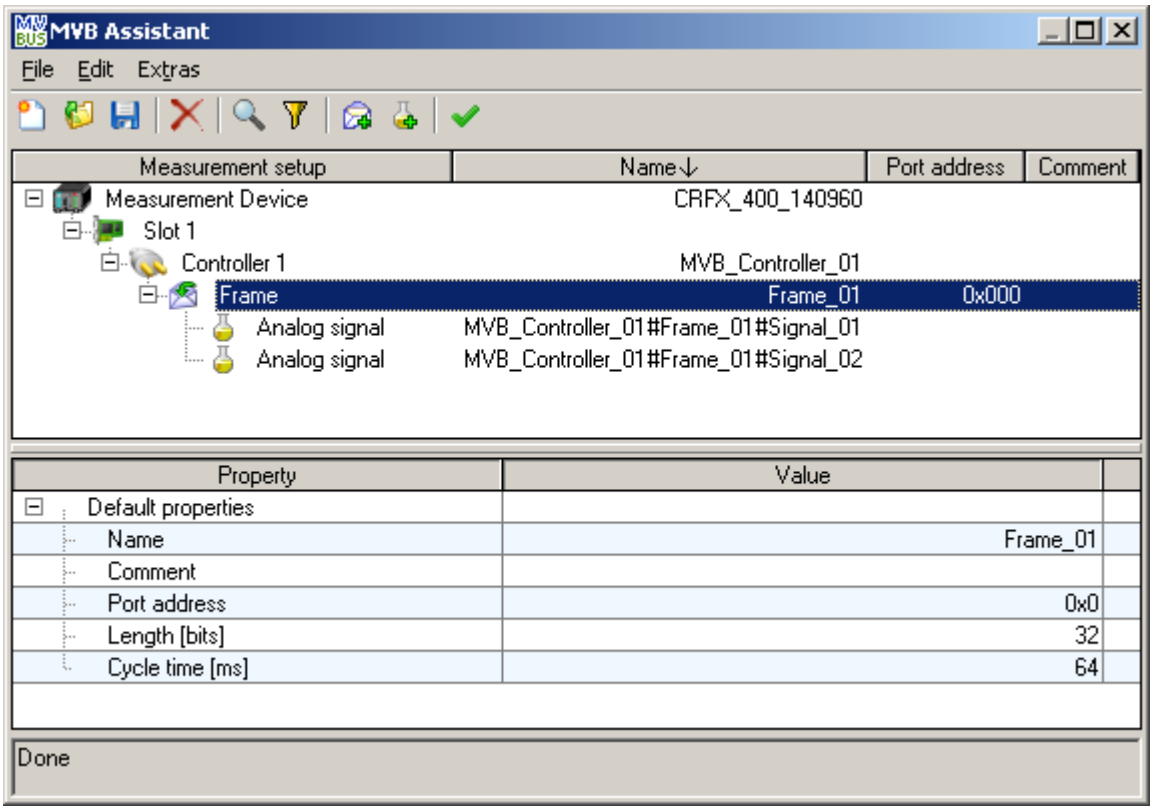
Password

Further, a controller's configuration can be imported from customized files. In order to enable import of customer-specific formats, passwords varying for respective customers are required. In that case a password is demanded, the first time the Assistant is called. When in doubt, contact our tech support 8.



Entering the password

9.10.9.4.1 Overview window



In the Assistant's upper region, the overview is displayed. In it, the structure of the measurement setup as it relates to the MVB-configuration is displayed schematically. You are shown the measurement devices ( ) connected with the experiment, and its name.

The MVB interfaces ( ) (slots) installed are indicated under each measurement device, along with their respective MVB-controller ( ) which these have.

For each MVB-cluster, the frames ( ) already defined with its associated signals ( ) (channels) are indicated.

9.10.9.4.2 Properties window

The Properties window is visible in the lower portion. It displays the editable properties and the settings options for the object currently selected in the Overview window. If multiple objects are selected at the same time, the Properties window shows a selection of suitable properties.

To change a property's value, click on the desired value.

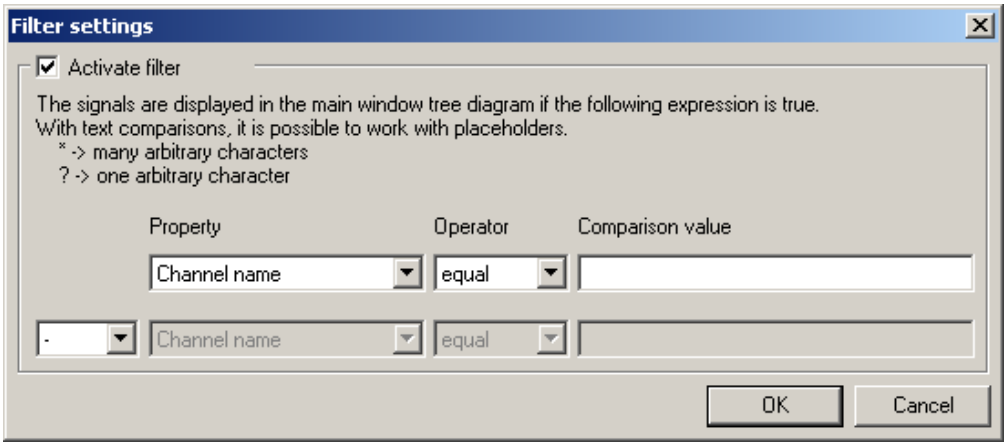
Depending on the property, either a pop-down list or a text box is used.

9.10.9.4.3 Menu

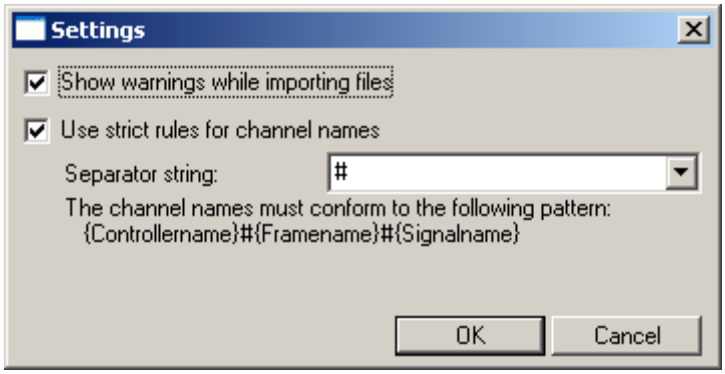
9.10.9.4.3.1 File

Menu item		Description
	New	Removes all previously created frames and signals and creates an empty MVB-configuration.
	Load All	The entire MVB-configurations of all devices and MVB-controllers in the experiment are loaded from a file. The file extension for this is MVBAF (*.mvbaf) "MVB-Assistant-File".
	Save All As	This is the counterpart to "Load All...". The entire MVB-configurations of all devices and MVB-controllers in the experiment are written to a file. The file extension for this is MVBAF (*.mvbaf) "MVB-Assistant-File"
	Load/import the selected controller	Imports the configuration of the controller currently selected from a file. The other controllers in the experiment are not affected. The following file formats are available for selection. **File format** / **Remarks** MVB-Assistant-File (*.mvbaf) / If the file contains multiple controller-configurations (if these were generated by means of the "Save All As..."-command), the file's first controller is loaded to the currently selected controller. The file's other controllers are ignored.
	Save the selected controller	Saves the selected controller in MVB-Assistant-File (*.mvbaf) –format to a file. In this way, it is possible to subsequently load the configuration to a different controller.

9.10.9.4.3.2 Edit

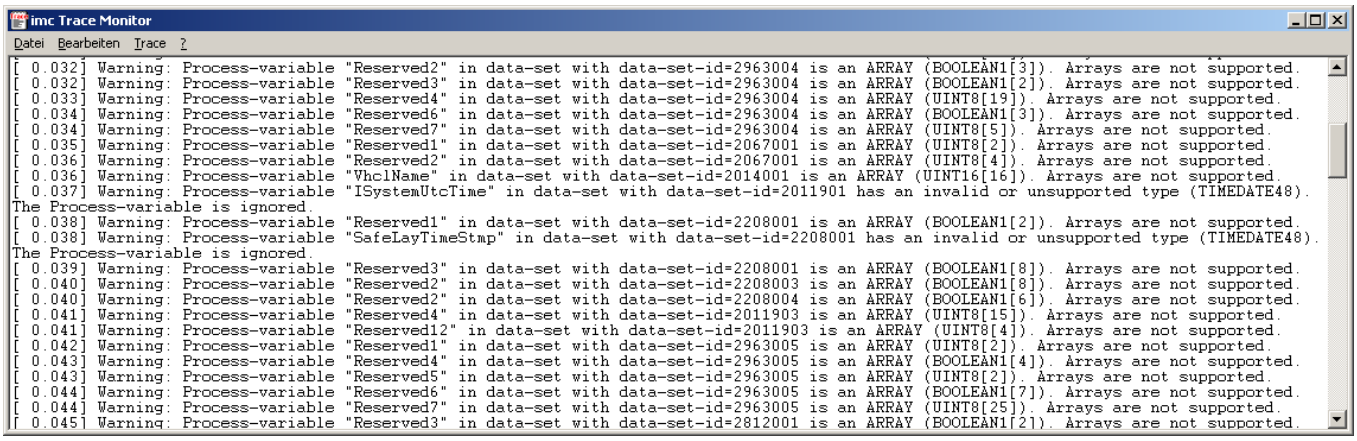
Menu item		Description
	Delete	Removes the selected entry
	Add new frame	Adds an additional frame to the selected cluster.
	Add new signal	With this menu item, you can add an additional signal to the selected frame.
	Find	It is possible to search for words or word fragments in the overview window. When many frames/signals have been defined, it is helpful for searching for them by name.
	Find Next	Searches for the next instance of the text entered.
	Filter	 The signals are displayed in the main window's tree diagram if a logical expression's condition is met.

9.10.9.4.3.3 Extras

Menu item	Description
✔ Test configuration	Checks whether the settings are valid as they are made. If not, the cursor skips to the faulty object in the Overview window and a message is posted in the status bar.
Settings	 *Dialog: Settings*

Settings: Show warnings while importing files

If the option "Show warnings while importing files" is active, then upon import of customized files, a list of messages indicating problems found is displayed.



imc Trace Monitor

9.10.9.4.3.4 Tree diagram context menu

Right-clicking the mouse over the tree calls a context menu. Some of the commands listed here are also available in other menus. For this reason, only unique menu items will be discussed here.

Menu item	Description
Expand all frames	Opens all frames in the tree.
Collapse all frames	Closes all frames in the tree.
Select all subordinate controllers	Selects all controllers belonging to the currently selected measurement device.
Select all subordinate frames	Selects all controllers belonging to the currently selected measurement device.
Select all subordinate signals	Selects all signals belonging to the currently selected measurement device, controller or even only those of the frame.

9.10.9.4.4 Properties of controllers

Property	Value
Default properties	
Name	MVB01
Comment	
Construction rule for the message-log-channel	{CTRL}_{SIG}
Construction rule for Signal- and Bit-Port- names	{CTRL}_abc_{FRM}_{SIG}_123
Log settings	
Log all data received on the MVB?	Yes
Base name of the message-log-channel	Log01
Resulting name of the message-log-channel	MVB01_Log01

Properties of a Controller

Default properties	Description
Name	Freely definable name for the Controller.
Comment	Comment on the Controller
Construction rule for names of log channels and signal channels	The channel names can be composed from the name entries with placeholders: {CTRL} for Controller 665, [FRM} for Frame 667 and {SIG} for Signal 668. The placeholders must always be spelled in capital letters. Example: Name of the controller = "*MVB01*", name of the frame= "*Frame01*" and name of the signal = "*Sig01*" from *{CTRL}_abc_{FRM}_{SIG}_123,* the resulting name is: "*MVB01_abc_Frame01_Sig01_123*"

Log settings	Description
Logging of all configured and received data	Enables recording of a bus log of all configured and received frames with this controller's FCodes 0...4. In this log, not only the frame's payload data are recorded but also the time received and the port address. The option allows the data stream to be unpacked as individual channels in imc STUDIO by the Bus Decoder 1101 or via imc FAMOS. MVB_Controller_01#Protokoll01 t [s] | MVB_Controller_01#Protokoll01 0.00561 | 0000: 98 12 01 0.01409 | 0001: b0 f9 0.02449 | 0002: 00 00 00 a0 14 15 8f c0 0.03349 | 0003: 00 00 00 40 03 db 7b 40 0.04160 | 0000: 98 12 01 0.05009 | 0001: b0 f9 0.06049 | 0002: 00 00 00 a0 14 15 8f c0 0.06942 | 0003: 00 00 00 40 03 db 7b 40 0.07759 | 0000: 98 12 01 0.08607 | 0001: b0 f9 0.09647 | 0002: 00 00 00 a0 14 15 8f c0 *Bus log example*
Base name of the message-log-channel	Freely definable channel name of the bus log channel.
Resulting name of log channel	The log name resulting from the Construction rule 665.

9.10.9.4.5 Properties of frames

Property	Value
Default properties	
Name	Frame_01
Comment	
Port address	0x0
Length [bits]	32
Cycle time [ms]	64

Properties of a Frame

Standard Eigenschaften	Beschreibung
Name	Freely definable name for the frame.
Comment	This is a text box for saving comments on the frame.
Port address	The Port address identifies the frame on the bus. This can take values of 0 through 4095.
Length [bits]	This specifies the frame's length in bits. The following values are possible: 16, 32, 64, 128 and 256
Cycle time [ms]	This is the planned time interval between two successive frames having this port address. This value is read from the files to be imported. This value is used if one of this frame's channels is registered for the first time. In this case, it is used to calculate the channel's sample time, which must always be less than or equal to the cycle time specified here. Example: The cycle time is 64ms. The system then sets the next lower sampling rate of 50ms.

9.10.9.4.6 Properties of signals

Property	Value
Default properties	
Name	Sig01
Resulting channel name	MVB01_abc_Frame01_Sig01_123
Comment	
Data interpretation	
Active/Passive Status	**active**
Data type	UINT
Start-Bit	0
Bit size	32
Physical unit	
Factor [1/LSB]	1
Offset []	0
Error handling	
Handle timeout error	Yes
Timeout-interval [s]	1
Error handling action	Inserts last measurement value

Propertie of a signal

Default properties	Description
Name	Freely definable name for the signal.
Resulting channel name	The channel name which results from the construction rule 665 in the Controller properties.
Comment	This is a text box for saving a comment on the signal. This comment may have a maximum of 255 characters. It is the same comment which will be visible with the registered channel.

Data integration	Description
Active/Passive Status	Allows signals to be hidden The configuration files to be imported often contain multiple thousands of signals. Up to 990 Fieldbus channels can be registered for each imc device. There is a limit of 512 active channels per device. Once the count of signals belonging to a device has been exceeded, it is necessary to select which ones are important for the measurement.

Active/Passive Status	Remarks
active	The signal is set as an active channel and recorded.
passive	The signal is registered in the system as a channel, but set to passive. No measured data are recorded.
Do not register with administration	The signal is not registered in the system. This signal only exists in the Assistant. No measured data are recorded.

Data integration	Description
Data type	This sets the signals type, or rather, how the data on the bus are interpreted. The following data types are possible:

Data type	Remarks
SINT	Signed Integer, negative numbers in two's complement
UINT	Unsigned Integer
BOOL	Single bit - If this data type is set, there is another signal property to set: the name of the 16-bit port channel
FLOAT	Floating point number with 32 bit conforming to "IEEE 754 (ANSI/IEEE Std 754-1985; IEC-60559:1989 - International version)"

Start-Bit	Specifies the bit-offset within the frame before the start of the current signal.
Bit size	Specifies how many bits from the start-bit on belong to the signal. What value range is possible for this property depends on the data type.

Data type	Possible value range for "Bit size"
SINT, UINT	Minimum 2 bits; Maximum 32
BOOL	Property not available, since only 1 bit.
FLOAT	always 32-Bit

Physical unit	Here you can enter a physical unit or its abbreviation, e.g. "V" for Volt. Use SI-units wherever possible. It is not advised to enter any scaling prefix such as m for milli-. Instead, adjust the factor correspondingly. The imc STUDIO curve windows automatically add such prefixes to correspond to the measured values.
Factor and Offset	If the signal currently selected happens to be of the integer data type, signed or unsigned, then both the properties **Factor** and **Offset** are displayed. These values are parameters for linear transformation, which are computed by the following formulas.

Incoming signal	Outgoing signal
$y = f \cdot x + a$	$x = \frac{y}{f} - \frac{a}{f}$

Variable definition

y = physical value	f = factor
x = integer on the Fieldbus	a = offset

Data integration	Description
	An incoming signal is a 16-bit unsigned integer. Thus, the integer's numerical range is 0 LSB to 65535 LSB. The physical quantity which it represents is a voltage between 10 V and 20 V. The factor is thus given by the equation $f = \frac{20\ \mathrm{V} - 10\ \mathrm{V}}{65535\ \mathrm{LSB}} = 1.526 \cdot 10^{-4} \frac{\mathrm{V}}{\mathrm{LSB}}$ and the offset is $a = 10\ \mathrm{V}$ Supposing the following integer were imported from the bus, $x = 13483\ \mathrm{LSB}$ it would correspond to a voltage of $y = f \cdot x + a = 12.057\ \mathrm{V}$

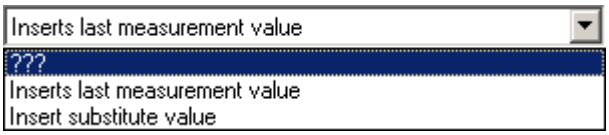
Handle timeout error

The term timeout-error refers to a case where no measured value is received from the fieldbus after a certain period; error handling can be set to deal with such an occurrence.

Parameter	Description
Handle timeout error	This settings option determines whether error handling of any timeout error would be performed.
Timeout Interval [s]	This is where to set the time limit after which a timeout-error is considered to have occurred.

Error handling action

This setting determines the action to be performed in case the timeout time limit elapses.



For this purpose, the following options are available, whose effects depend on whether a channel is sampled equidistantly or recorded with a time stamp.

Error handling	Equidistant sampling	Samples with time stamp
Last value	The last valid value is returned until a new value arrives.	After elapse of the timeout time, a new sample is generated with the last value.
Substitute value	The substitute value is returned until a new value arrives.	After elapse of the timeout time, a new sample is generated with the substitute value.

9.10.10 Modbus Interface

The **Modbus** protocol was developed in 1979 by Gould-Modicon for communication with programmable controllers. Modbus is an open protocol whose version Modbus/TCP is described in the IEC 61158 standard.

The imc Modbus interface is a fieldbus module that can be used to equip imc measurement devices.

For configuration the complete layout "*Application module*" is used, which enables the necessary configuration steps. The interface is able to receive signals.

Prerequisite:
Which devices can be equipped with the interface and from which firmware this is supported, please refer to the technical data sheet of the imc Modbus interface.

9.10.10.1 Preparation

Integration in the experiment

In order for imc STUDIO to be able to load the following Assistant, the device hardware configuration must be declared. imc STUDIO automatically loads the module configuration for Modbus and enables the configuration by its assistant.

The **assistant** for configuring the Modbus interface is **opened** via the page "*Application module*".

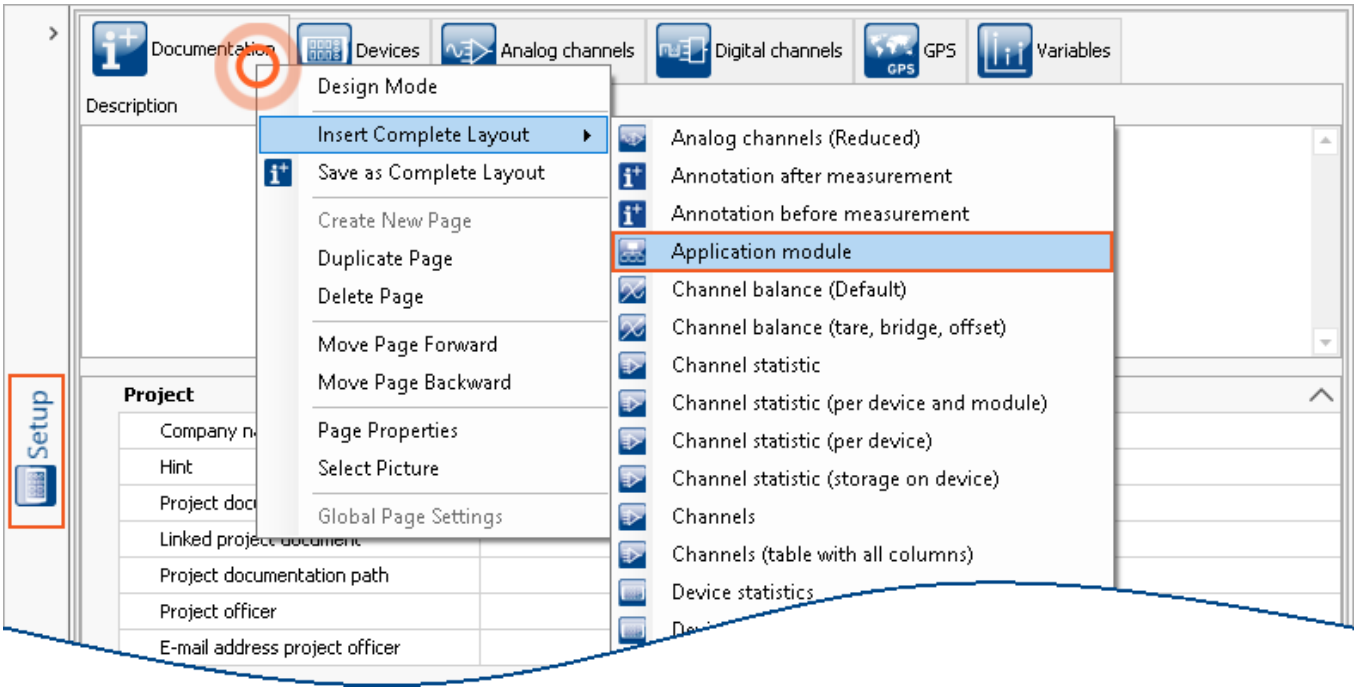
By default, the Setup page "*Application module*" is not displayed.

Displaying a Setup page

To add a page to the user interface,

- open the context menu over a page-tab or over the empty space to the right of one.
- Click on the corresponding page entry in the list under "*Insert complete layout*".

In consequence, the page is inserted to the right of the selected locations.



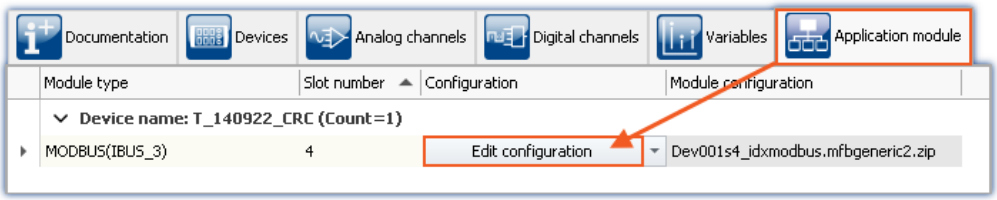
Displaying a Setup page

Note Saving views

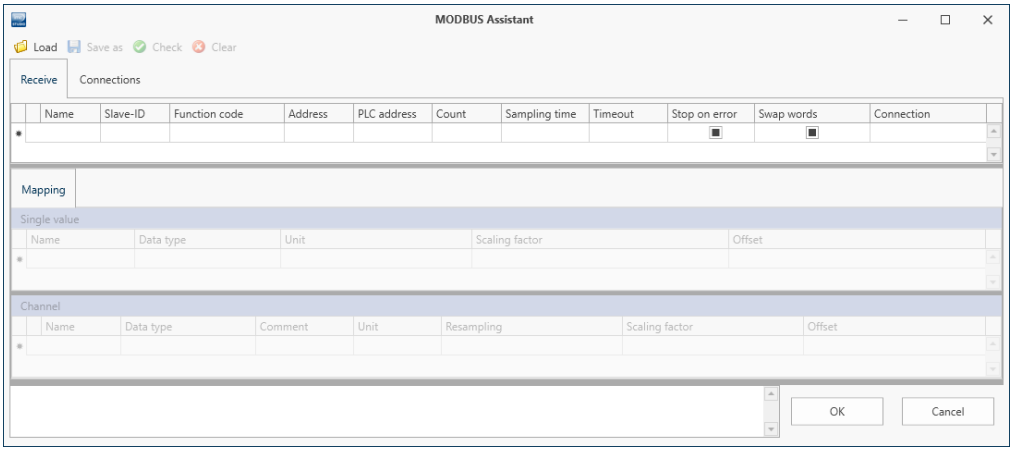
Changes to the view, such as: "*Insert Setup page*", must be saved with the view in order to be visible the next time the program is started.
See the manual "*imc STUDIO (general)*" > "*Views* 155".

9.10.10.2 Modbus assistant

In order to start the assistant from imc STUDIO, click on "*Edit configuration*" under "*Setup-Application module*".



Edit configuration opens Modbus assistant



Empty Modbus assistant

9.10.10.2.1 Menu

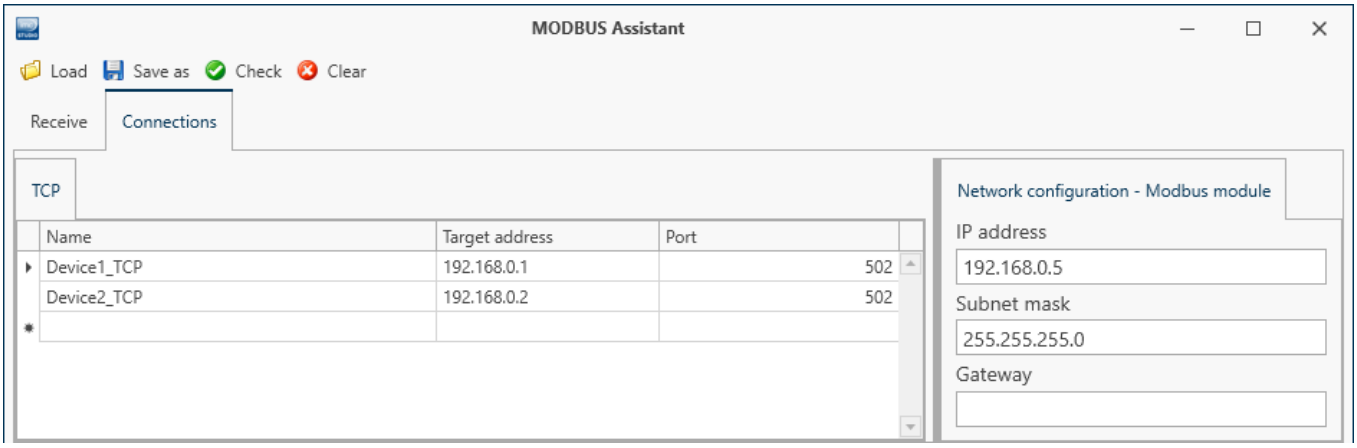
Modbus Assistant menu

Menu item	Description
Load	Loads a **Modbus** configuration having the file extension *.**dcz**.
Save as	Saves the current **Modbus** configuration having the file extension *.**dcz**.
Check	Check the current **Modbus** configuration for duplicate or faulty names, and for plausibility.
Clear	Resets the **Modbus** configuration.

9.10.10.2.2 Connections

On the page under the tab "*Connections*", you can add the serial or TCP/IP addresses of the devices whose data are to be received.

TCP/IP



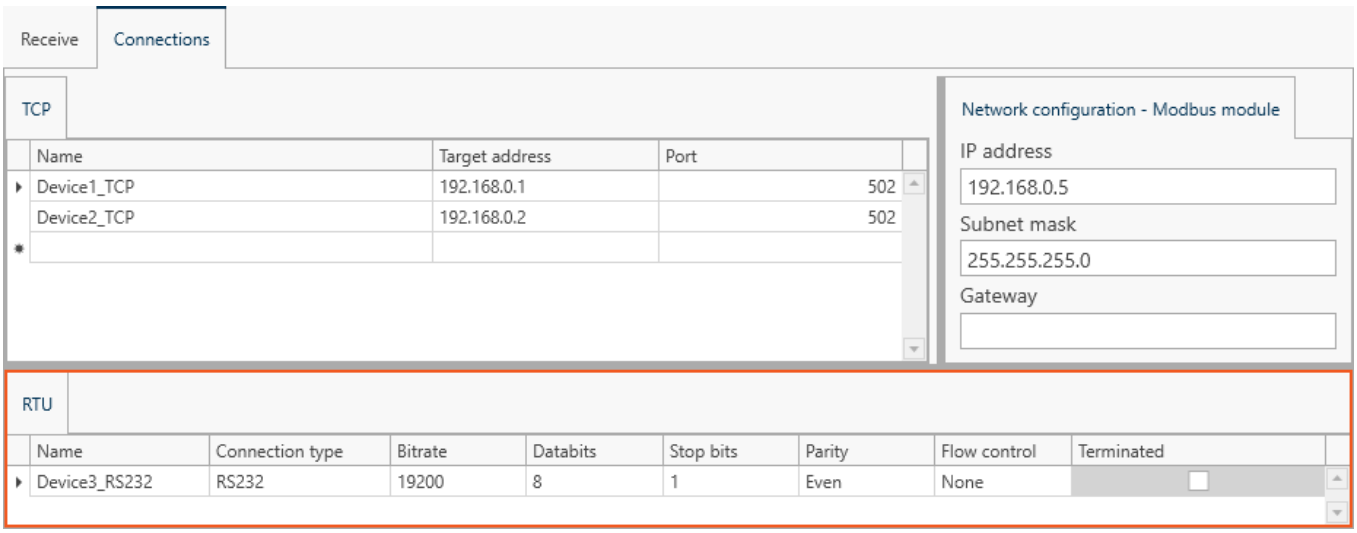
Modbus Connections

Parameter	Description
Name	Freely assignable designation of the source
Target host	TCP/IP address of the source, e.g. 192.168.1.1.
Port	502

Network configuration Modbus module

TCP/IP address for the Modbus card; it will be applied upon performing the Preparation procedure.

RTU - serial connection



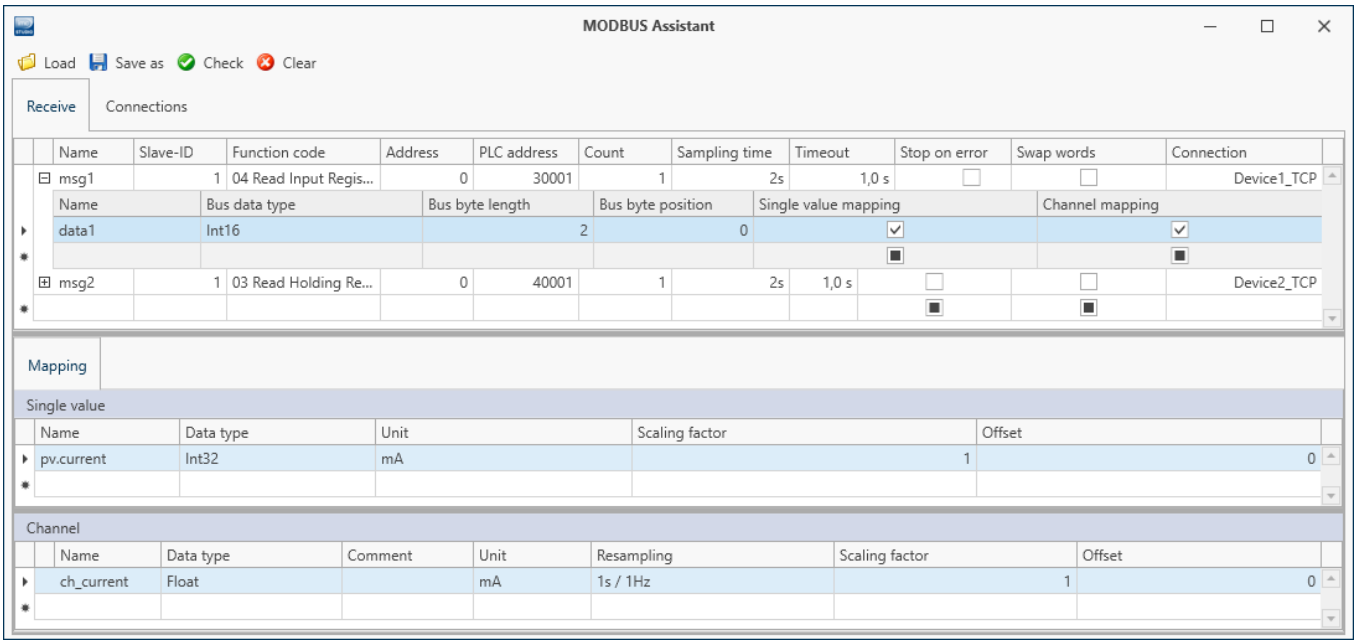
Modbus serial connections

Parameter	Description
Name	Freely assignable designation of the source
Connection type	RS232, RS232 (half-duplex), RS232 (full-duplex), depending on the source.

Additional serial interface settings: Bitrate, Databits, Stopbits, Parity, Flow control and Terminated.

9.10.10.2.3 Receive

The page "*Receive*" is where packets and their sources, sampling rates, etc., as well as the signals they contain are defined.



Modbus Receive dialog

Within a data packet, Modbus receives up to 2000 coils or 125 registers (up to 250 Bytes). The signals are addressed by the Byte position of the first Byte and the count of Bytes in the signal.

The data received are assigned in the imc system to channels or single values (process vector variables).

Selected rows can be deleted using the **Del**-key.

Packet definition

Parameter	Description
Name	Designation of the request
Slave-ID	Identifier of the Modbus device
Function code	01 Read Coils 02 Read Discrete Inputs 03 Read Holding Registers 04 Read Input Registers
Address	Base address of request (0-based)
PLC address	Device address for the request (1-based). Specified by selection of the function code
Count	Count of requested registers/coils
Samling time	Interval at which the request is sent
Timeout	Maximum wait time between request and response
Stop on error	When active, , for instance upon elapse of the timeout, the request is **not** repeated.
Swap words	Changes the order of data words within data types which are larger than 16 bits.
Connection	Selection of the source which was defined under Connections 673.

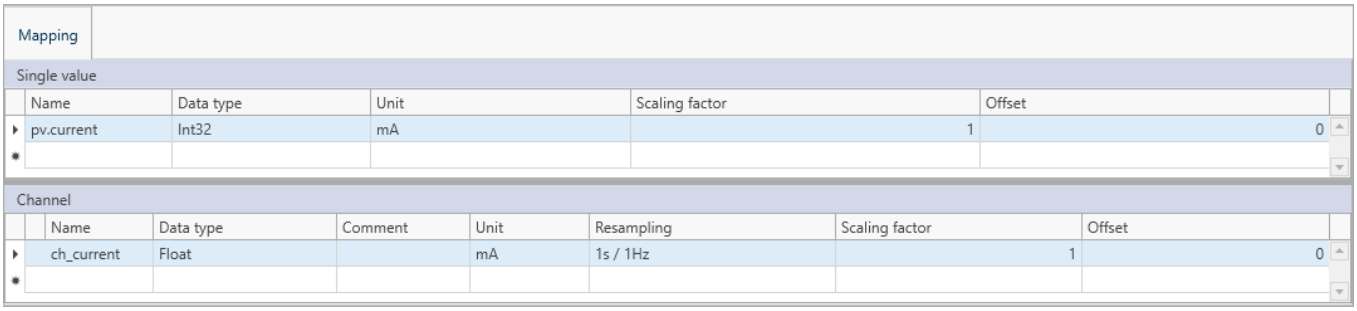
Signals on the bus

Parameter	Description
Name	Designation of the packet
Bus-data type	Selection of the data format expected for the signal **Data type** / **Variants** / **Bit count** Integer / Int8, Int16, Int32 / with 8, 16 or 32 Bit Unsigned Integer / UInt8, UInt16, UInt32 / with 8, 16 or 32 bits Floating point numbers / Float, Double / 32-bit and 64-bit
Bus-Byte length	Byte length of the signal's first byte. Wird automatisch mit dem Bus-data typ gesetzt.
Bus-Byte position (zero-based)	Start position of the signal's first Byte.
Single value-/channel-mapping	When this option is activated, a process vector variable and/or a channel is created, for which you can make settings under "*Mapping* 676".

9.10.10.2.3.1 Mapping

Depending on whether the associated option is activated in the column **Single value-/channel mapping** 675, the selected signal is listed under "*Mapping*". A Signal can be made available as a channel, a process vector variable, or as both.

In the lower region under "*Mapping*", the signals from the data packet are defined for the imc system.



Mapping Receive-dialog

Parameter	Description
Name	The signal's designation as a channel or process vector variable in the imc system.
Data type	Selects the data format to be used in the imc system **Data type** / **Variants** / **Bit count** Integer / Int32 / 32 bits Floating point numbers / Float / 32 bits
Offset/Scaling factor/Unit	The offset and scaling factor for integer formats. These are to be adopted from the specs of the sending device, and the unit as well.
Comment for Channel	A comment to be saved as a property with the channel. Freely defined text having up to 255 characters.
Resampling for Channel	Output rate. No interpolation performed. The signal is updated at the "*Bus-cycle time*" and outputted at the rate specified under "*Resampling*". If *Resampling* > *Bus-cycle time*, the same value is used repeatedly. If *Resampling* < *Bus-cycle time*, not all values received from the Profinet bus are processed.

9.10.11 Profinet interface

Profinet stands for ***Proc***ess ***Fi***le ***Net***work-***I***sochronous-***R***eal-***T***ime. This technology, developed by member companies of the Profibus user organization in conjunction with Siemens is based on Ethernet-TCP/IP and UDP/IP and conforms to the standards IEC 61158 and IEC 61784.

Integration in imc CRONOS*compact* is provided by means of the interface module **imc PROFINET-IRT**. For configuration purposes, the complete-layout "*Application module*" is used, which enables the necessary configuration steps.

9.10.11.1 Preparation

Integration in the experiment

In order for imc STUDIO to be able to load the following Assistant, the device hardware configuration must be declared. imc STUDIO automatically loads the module configuration for Profinet-IRT and enables the configuration by its assistant.

The **assistant** for configuring the Profinet interface is **opened** via the page "*Application module*".

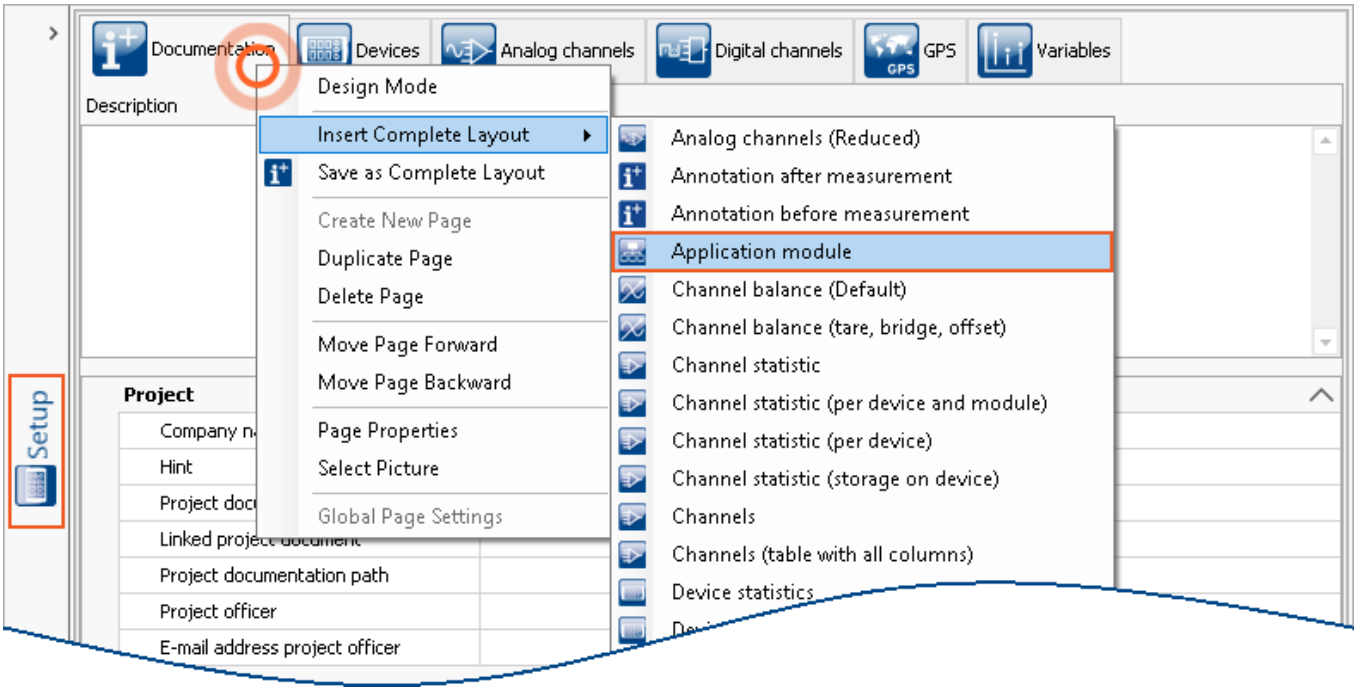
By default, the Setup page "*Application module*" is not displayed.

Displaying a Setup page

To add a page to the user interface,

- open the context menu over a page-tab or over the empty space to the right of one.
- Click on the corresponding page entry in the list under "*Insert complete layout*".

In consequence, the page is inserted to the right of the selected locations.



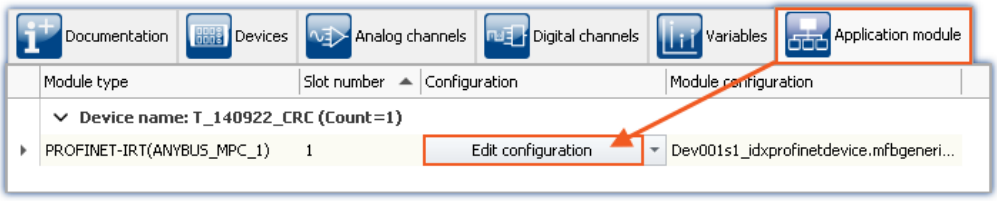
Displaying a Setup page

Note Saving views

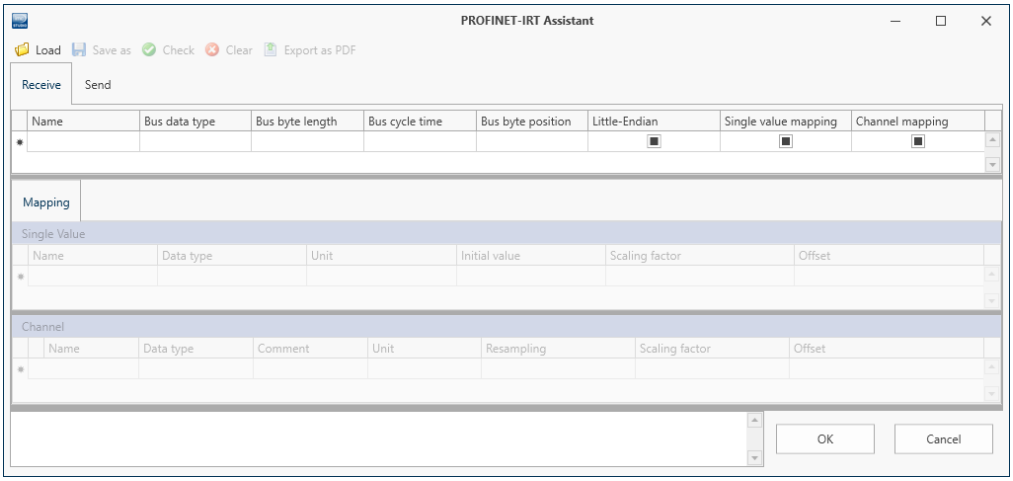
Changes to the view, such as: "*Insert Setup page*", must be saved with the view in order to be visible the next time the program is started.
See the manual "*imc STUDIO (general)*" > "*Views* 155".

9.10.11.2 PROFINET-IRT assistant

In order to start the assistant from imc STUDIO, click on "*Edit configuration*" under "*Setup-Application module*".



Edit configuration opens PROFINET-IRT Assistant



Empty PROFINET-IRT Assistant

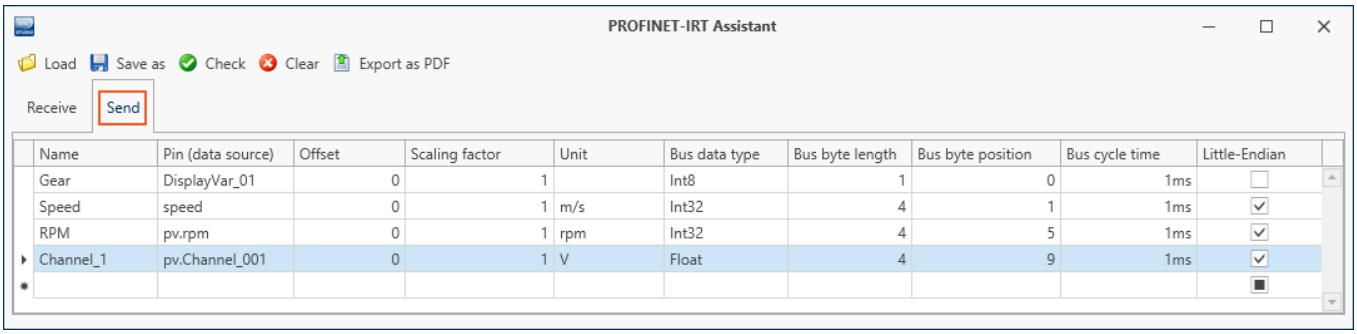
9.10.11.2.1 Menu

PROFINET-IRT assistant menu

Menu item	Description
Load	Here, a Profinet configuration having the extension *.**dcz** can be loaded.
Save as	Saves the current Profinet configuration with the file extension *.**dcz**.
Check	Checks the current Profinet configuration for duplicate or faulty names, and plausibility.
Clear	Resets the Profinet configuration.
Export as PDF	Creates a PDF with listed parameters.

9.10.11.2.2 Send

Under the page *Send*, selected channels captured by the imc system are sent to the PROFINET-IRT receiver. From the perspective of the SPS, this is referred to as the **input-measurement data space**.



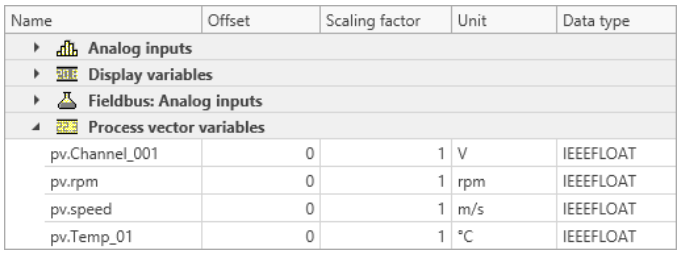
Name	Pin (data source)	Offset	Scaling factor	Unit	Bus data type	Bus byte length	Bus byte position	Bus cycle time	Little-Endian
Gear	DisplayVar_01	0	1		Int8	1	0	1ms	☐
Speed	speed	0	1	m/s	Int32	4	1	1ms	☑
RPM	pv.rpm	0	1	rpm	Int32	4	5	1ms	☑
Channel_1	pv.Channel_001	0	1	V	Float	4	9	1ms	☑

PROFINET-IRT Send-dialog

PROFINET-IRT uses the Ethernet-network and sends up to 20 blocks having 64 Bytes each (for a maximum of 1280 Bytes) in a data package. The data are addressed by the position of the first Byte and the count of Bytes in the signal.

It is possible to send channels, process vector variables and Display variables. Digital channels can not be sent bit-by-bit, but only as Bytes.

Selected lines can be deleted by means of the Delete-key. Caution: "*Clear*" deletes not only the selected line but the complete list!

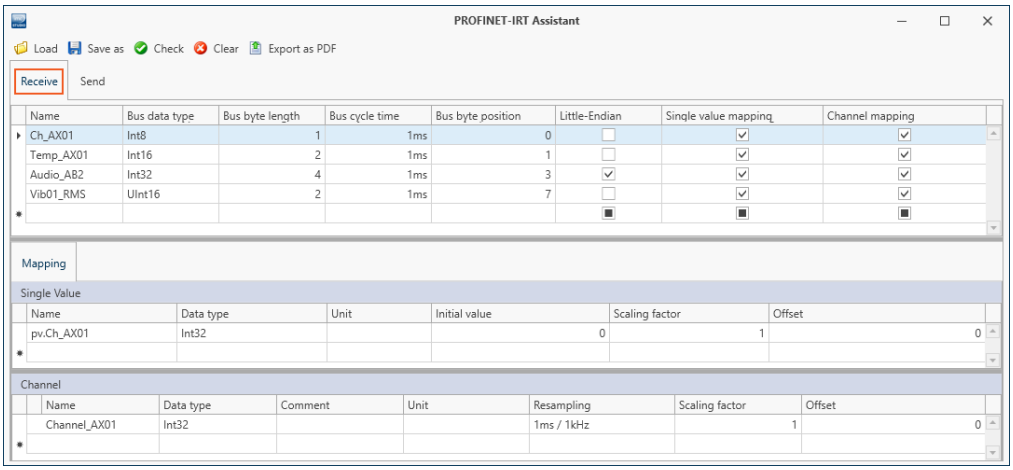
Parameter	Description
Name	Designation of the signal as information for the receiving device.
Pin (data source)	A click in the cell of the column *Pin* opens a sub-dialog with all signals which can be sent by the imc system.  *PROFINET-IRT Pin-dialog* Channels, process vector variables and Display variables can be sent.
Offset/Scaling factor/Unit	These parameters can not be edited. Once the signal has been selected in the Pin-column, the offset, scaling factor and unit are entered by the system. For signals of the data type Float, the offset and the scaling factor are already reflected in the value and therefore Offset= 0 and Scaling factor = 1 always applies. For integer values, there is a resulting offset and scaling, which is entered by the system.
Bus-data type	Data type: Int16, Int32, Float etc. This is set in the Pin-column upon selection of the signal.
Bus-Byte length	This is set in the Pin-column upon selection of the signal. For Int16, = 2 Byte; for Int32, = 4 Byte; etc.
Bus-Byte position	Position of the signal's first Byte. The system consecutively adds the signals to the column as they are selected. The position can be re-assigned later if desired.

Name	Offset	Scaling factor	Unit	Data type
Analog inputs				
Display variables				
Fieldbus: Analog inputs				
Process vector variables				
pv.Channel_001	0	1	V	IEEEFLOAT
pv.rpm	0	1	rpm	IEEEFLOAT
pv.speed	0	1	m/s	IEEEFLOAT
pv.Temp_01	0	1	°C	IEEEFLOAT

Parameter	Description
Bus cycle time	The signal's updating interval. This determines the frequency at which the signal is read for PROFINET-IRT. In actual fact, the signal is updated at intervals of half of the cycle time, in order to avoid losing any update. Different bus cycle times may be entered for signals belonging to the same data packet. **Remarks**: Since all signals are sent as a complete packet, signals having relatively long cycle times are not always actually updated, meaning that the most recent value is sent repeatedly for a few intervals. The receiver is aware of the interval through the configuration file (*.dcz), so that the value there is only read whenever the cycle time associated with the signal has elapsed.
Little Endian	You must specify whether the order of Bytes is sent according to Little-Endian format (activated), or in Big-Endian format (inactive). Little-Endian is the order in which the "Little End" (lowest value in the Byte sequence) is saved first. **Example**: The hexadecimal number 3F56hex is saved at the memory addresses 1000 and 1001: Little Endian system: 56 at the address 1000 and 3F at the address 1001 Big Endian system: 56 at the address 1001 and 3F at the address 1000

9.10.11.2.3 Receive

On the page "*Receive*", signals are extracted from the PROFINET-IRT data packet and made available for the imc system as either channels or single values. From the perspective of the SPS, this is referred to as **output-measurement data space**.



PROFINET-IRT Receive-dialog

PROFINET-IRT uses the Ethernet-network and sends up to 20 blocks having 64 Bytes each (for a maximum of 1280 Bytes) in a data package. The signals are addressed by the position of the first Byte and the count of Bytes in the signal.

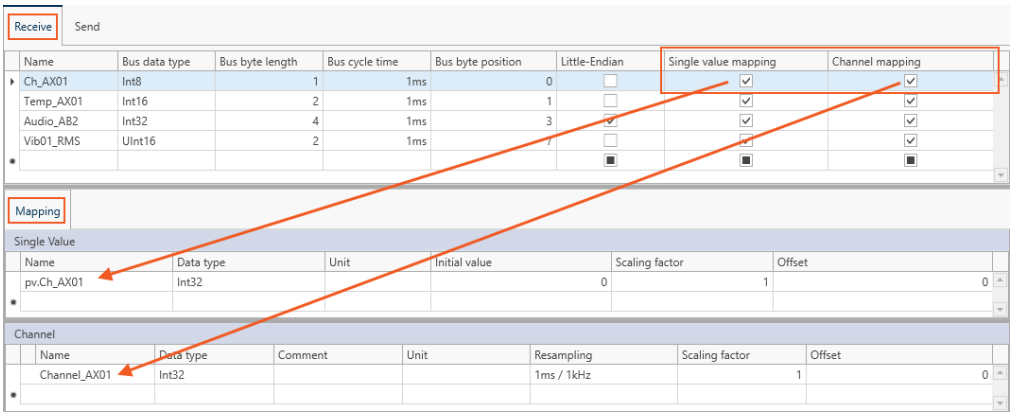
The data received are mapped in the imc system to channels or single values (process vector variables). It is not possible to write to Display variables directly. Digital signals can only be mapped to 2-Byte channels or 32-bit process vector variables.

Selected lines can be deleted by means of the Delete-key. Caution: *Clear* deletes not only the selected line but the complete list!

Parameter	Description
Name	The designation of a signal as it is registered with the sender . This entry is only used for documentation purposes. The name of the channel or process vector variable which is mapped and visible in the imc software is set under "*Mapping* 682".
Bus-data type	Selection of the data format expected for the signal **Data type** / **Variants** / **Bit count** Integer / Int8, Int16, Int32 / with 8, 16 or 32 bits Unsigned Integer / UInt8, UInt16, UInt32 / with 8, 16 or 32 bits Floating point numbers / Float, Double / 32-bit and 64-bit
Bus-Byte length	Byte length of the signal.
Bus-Byte position	Start position of the signal's first Byte.
Bus-cycle time	The signal's updating interval. This determines the frequency at which the signal is read for Profinet. In actual fact, the signal is updated at intervals of half of the cycle time, in order to avoid losing any update. **Remarks**: Different bus cycle times may be entered for signals belonging to the same data packet. Since all signals are sent as a complete packet, signals having relatively long cycle times are not always actually updated, meaning that the most recent value is sent repeatedly for a few intervals.
Little Endian	You must specify whether the order of Bytes is received according to Little-Endian format (activated), or in Big-Endian format (inactive). See also "*Send* 680".
Single value-/channel-mapping	When this option is activated, a process vector variable and/or a channel is created, for which you can make settings under "*Mapping* 682".

9.10.11.2.3.1 Mapping

Depending on whether the respective option is activated in the column **Single Value-** or **Channel-mapping**, the signal selected under "*Mapping*" is listed as *Single Value* and/or *Channel*. The signal can be provided as a channel, a process vector variable or both.



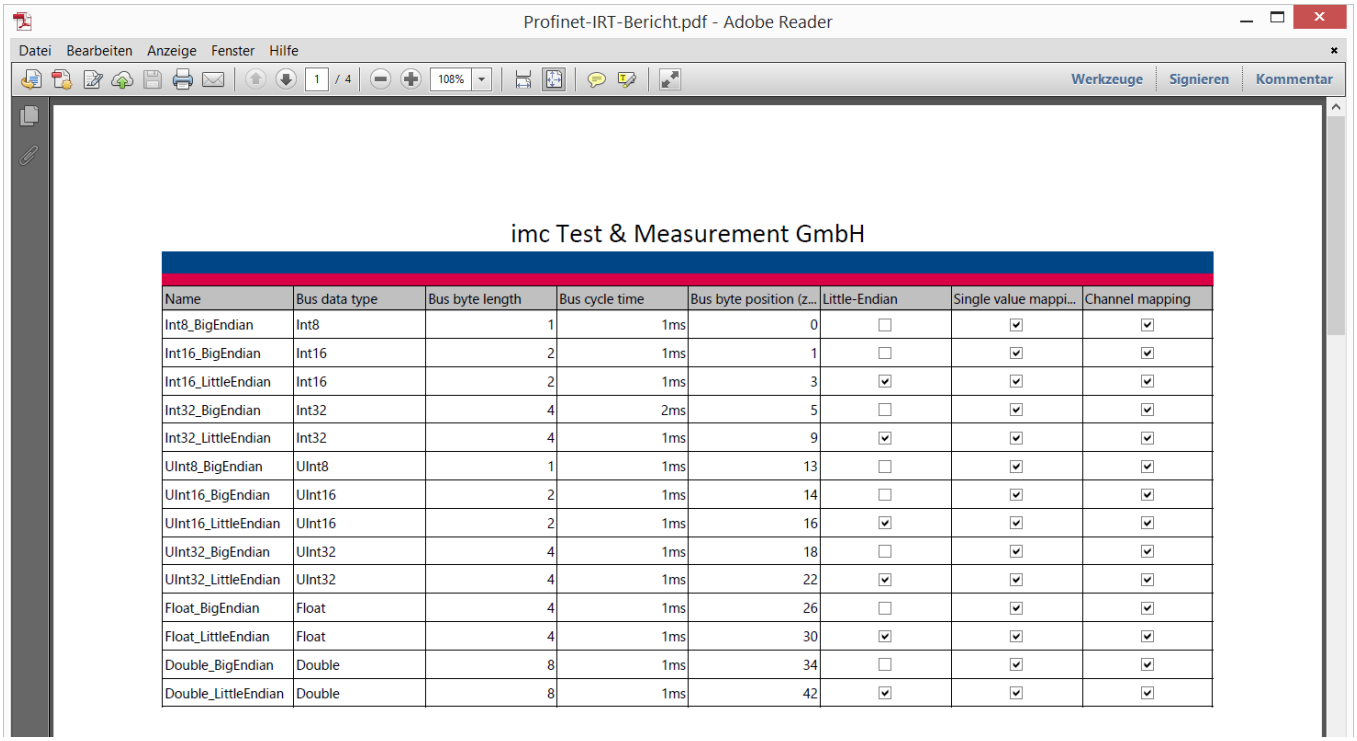
PROFINET-IRT Receive dialog

In the lower area under "*Mapping*" the signals of the PROFINET-IRT data packet are defined for the imc system.

Parameter	Description
Name	The signal's designation as a channel or process vector variable in the imc system.
Bus-data type	Selects the data format to be used in the imc system **Data type** / **Variants** / **Bit count** Integer / Int32 / 32 bits Floating point numbers / Float / 32 bits
Offset/Scaling factor/Unit	The offset and scaling factor for integer formats. These are to be adopted from the specs of the sending device, and the unit as well.
Initial value for single values	Preset which applies until arrival of the first data packet.
Comment for Channel	A comment to be saved as a property with the channel. Freely defined text having up to 255 characters.
Resampling for Channel	Output rate. No interpolation performed. The signal is updated at the "*Bus-cycle time*" and outputted at the rate specified under "*Resampling*". If *Resampling* > *Bus-cycle time*, the same value is used repeatedly. If *Resampling* < *Bus-cycle time*, not all values received from the Profinet bus are processed.

9.10.11.2.4 Export as PDF

The table is exported as a PDF file.



imc Test & Measurement GmbH

Name	Bus data type	Bus byte length	Bus cycle time	Bus byte position (z...	Little-Endian	Single value mappi...	Channel mapping
Int8_BigEndian	Int8	1	1ms	0	☐	☑	☑
Int16_BigEndian	Int16	2	1ms	1	☐	☑	☑
Int16_LittleEndian	Int16	2	1ms	3	☑	☑	☑
Int32_BigEndian	Int32	4	2ms	5	☐	☑	☑
Int32_LittleEndian	Int32	4	1ms	9	☑	☑	☑
UInt8_BigEndian	UInt8	1	1ms	13	☐	☑	☑
UInt16_BigEndian	UInt16	2	1ms	14	☐	☑	☑
UInt16_LittleEndian	UInt16	2	1ms	16	☑	☑	☑
UInt32_BigEndian	UInt32	4	1ms	18	☐	☑	☑
UInt32_LittleEndian	UInt32	4	1ms	22	☑	☑	☑
Float_BigEndian	Float	4	1ms	26	☐	☑	☑
Float_LittleEndian	Float	4	1ms	30	☑	☑	☑
Double_BigEndian	Double	8	1ms	34	☐	☑	☑
Double_LittleEndian	Double	8	1ms	42	☑	☑	☑

9.10.12 PROFIBUS interface

Using the PROFIBUS Configurator 684, you can create a configuration for the Assistant in imc STUDIO

With PROFIBUS Assistant 692 you setup the fieldbus interface for the imc devices accordingly.

Note

PROFIBUS configurator and assistant **has not been available in English** at the time of printing this manual.

9.10.12.1 PROFIBUS configurator

Unfortunately, there is no universal standard for the exchange of PROFIBUS configurations.

However, there are standardized module descriptions: GSD (General Station Description) files. GSD files contain so-called basic device data which characterize a PROFIBUS subscriber's properties. The PROFIBUS Configurator joins the modules present with the GSD files in the configuration.

The resulting configuration is exported from the PROFIBUS Configurator in csv format and then re-imported into imc STUDIO by the PROFIBUS Assistant.

Installation

This program is not automatically installed with the rest of the package. It is on the imc STUDIO installations medium in the folder: Products\imc DEVICES\Tools\imcPROFIBUSConfigurator.

Start the setup and follow the instructions which appear. The program can then be installed at an arbitrary location on your hard drive.

9.10.12.1.1 First start

When you start the software, you are prompted to select a GSD folder. This folder contains the GSD files of the system you wish to configure. Alternatively, the GSD folder can contain a complete collection of your system's GSD files and serve as a library.

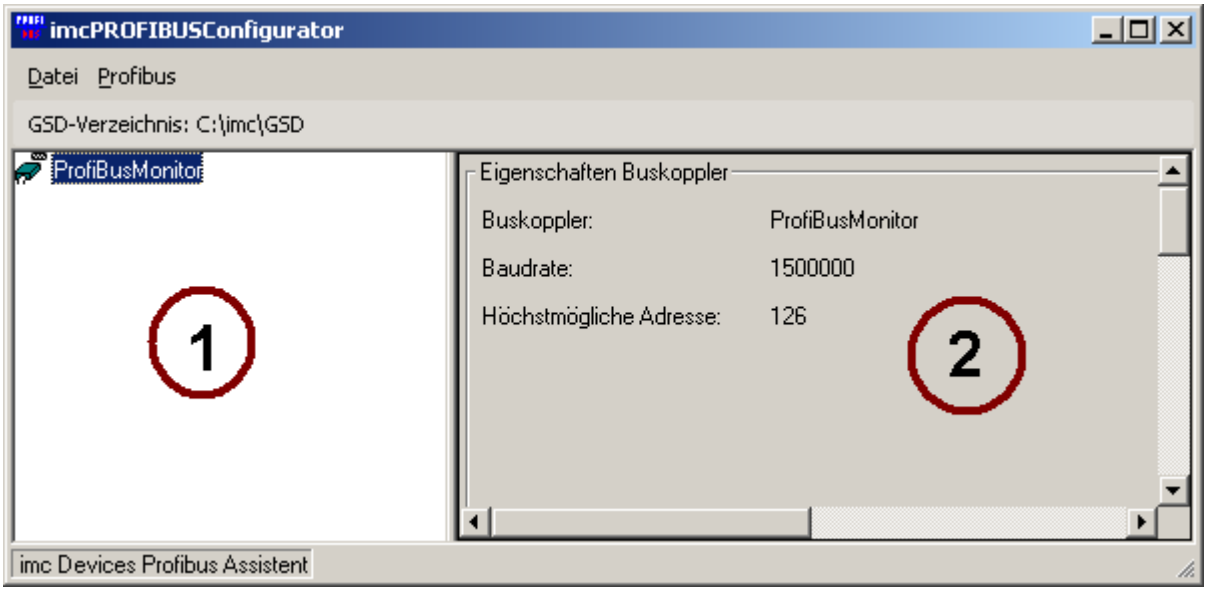
If necessary, create such a folder on your PC and copy your module's files to there.

Selecting the GSD folder

9.10.12.1.2 Main window

The main menu consists of two parts:

- **Part 1** maps the bus structure. Right-clicking the mouse over one of its elements opens the editable options belonging to it. For example, the element properties or adding subordinate elements.
- **Part 2** shows the properties of the respective element, while not allowing any changes.



Main PROFIBUS Configurator window

File menu

Menu item	Description
Open	Loads a previously stored configuration in XML format.
Save configuration	Saves the configuration as a XML file
Export	Exports the configuration as a CSV file. In this format the configuration is subsequently imported in PROFIBUS Assistant 691.
GSD Verzeichnis wählen	Selects the GSD (General Station Description) 684 files.

Profibus menu

Menu item	Description
Add Master	Opens the dialog for selecting the Master 686.
Add Slave	Opens the dialog for selecting the modules 686.
Monitor settings	Settings dialog for the Bus monitor 690.

9.10.12.1.3 Adding a Master / Slave

The menu items *Profibus > Add Master* and *Add Slave* each cause a settings dialog to appear.

The dialog lists all master or slave modules located in the GSD folder. If the GSD folder is not set 685 or not contain any GSD files, the list remains empty.

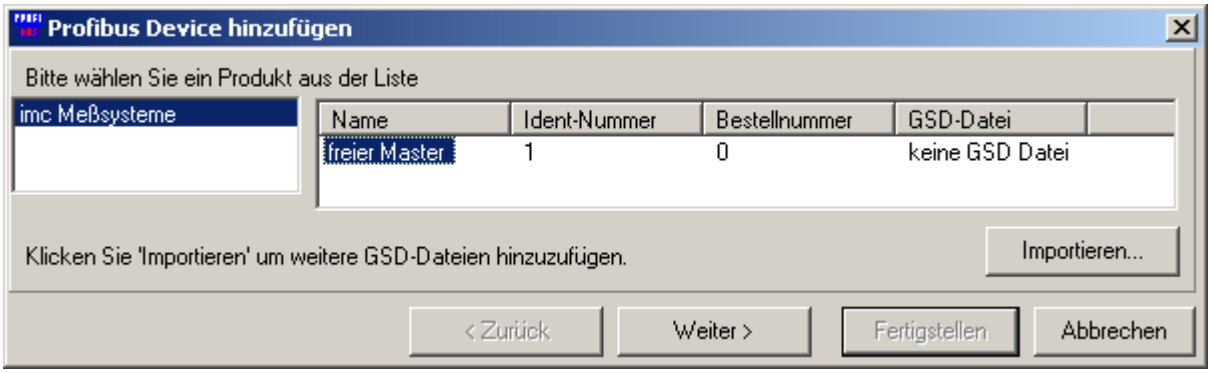
Click on the desired system and then on *Next*. In the next window, you can set any desired properties, such as the name or PROFIBUS addresses. Using the *Import* button, you can add additional GSD files.

Note

Notice that only such elements are listed whose GSD files are located in the GSD folder.

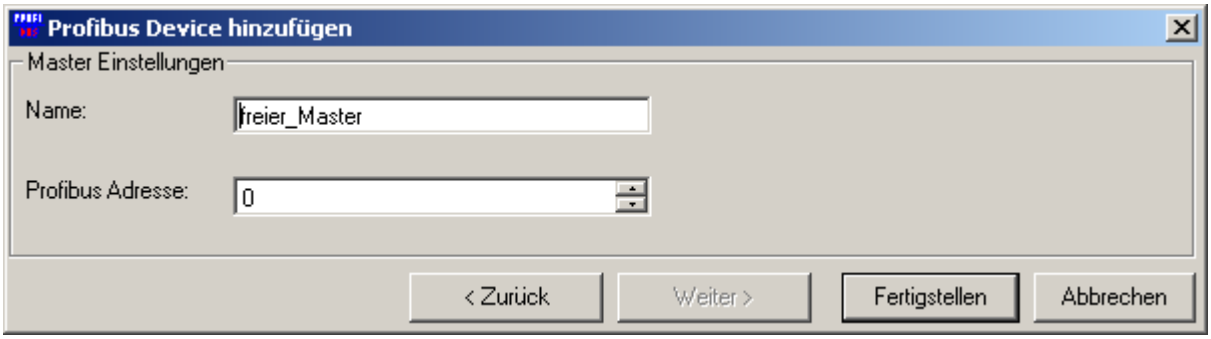
Configuring the Master

If you wish to add a new Master, you don't necessarily need a GSD file. The software recognizes a *free Master* which is able to function without a GSD file.



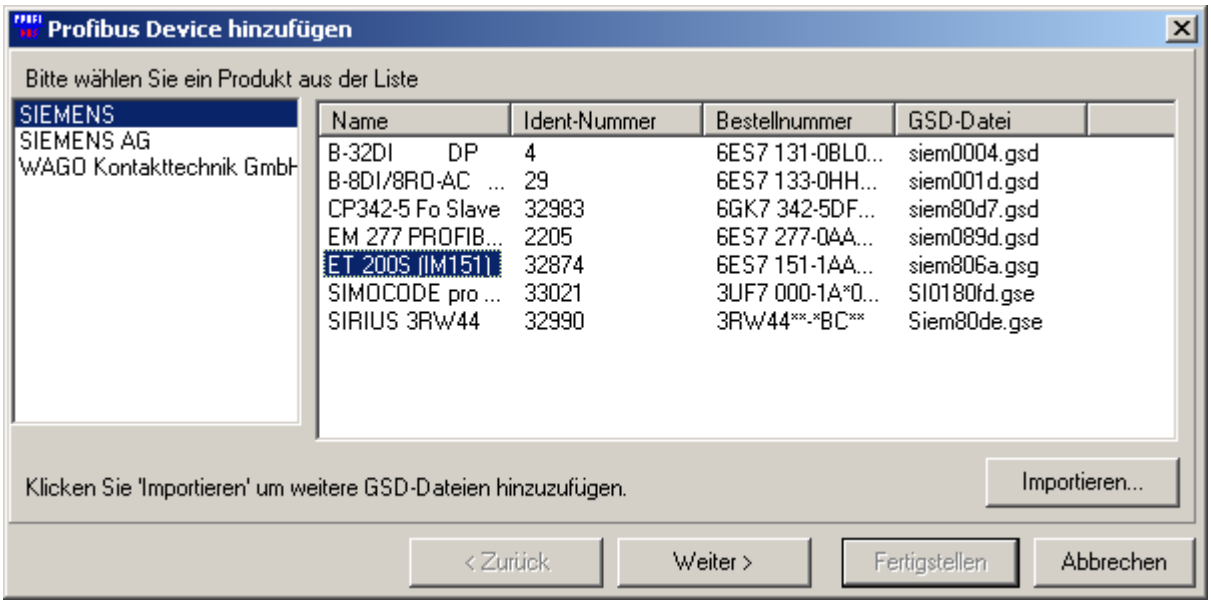
Configuring the Master

Click on *Next >* and set the name and ID.



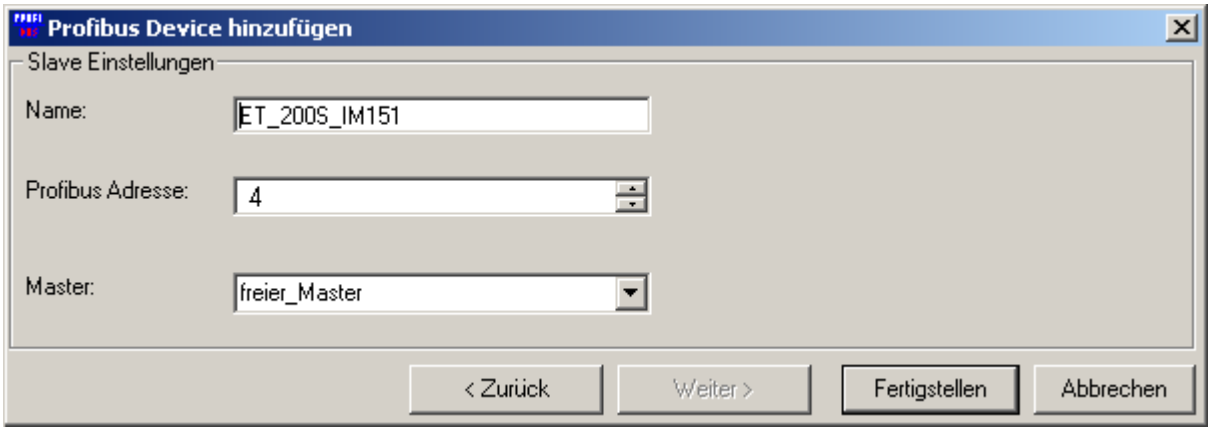
Setting the Master ID

Configuring the Slave



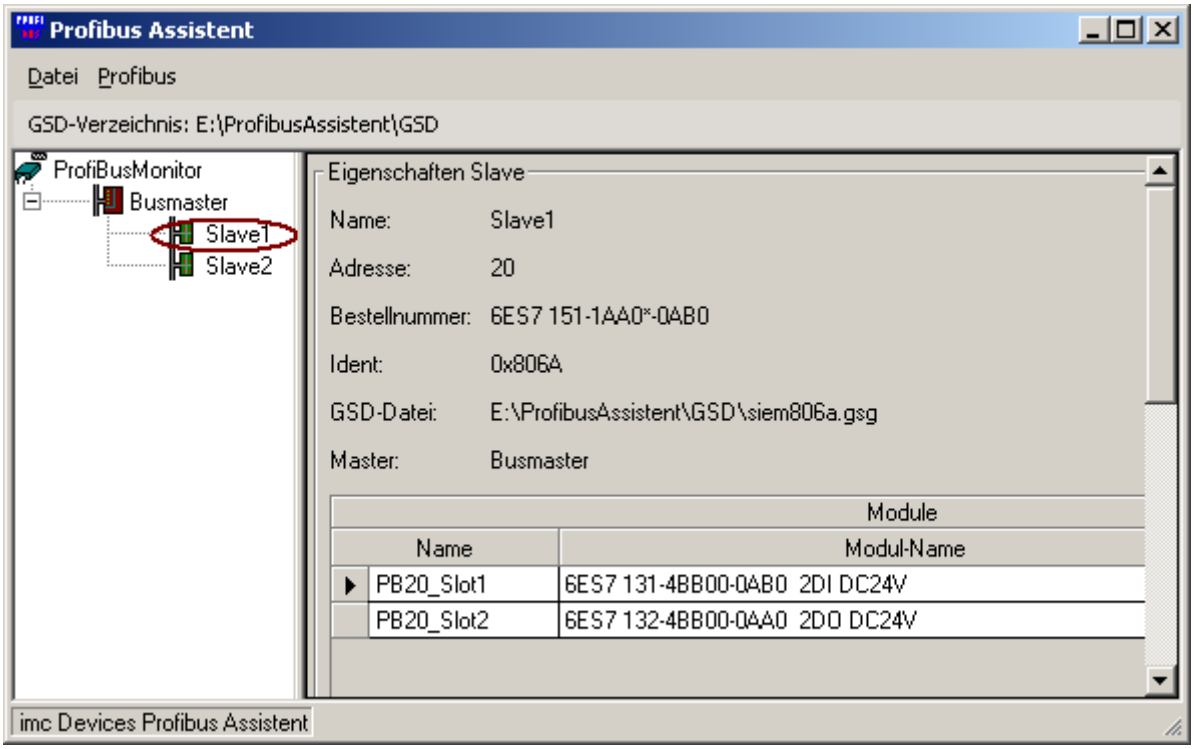
Adding elements

Select the connection terminal and click on *Next>*.



Set the Profibus address and the designation here

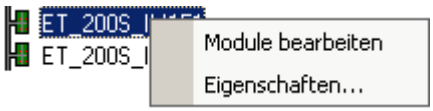
The modules entered now appear in the main window:



Modules in Profibus assistant

With modular Slaves, you need to enter their configuration manually. This concept assumes that there are usually **Bus terminals** as PROFIBUS-Slave subscribers available. These are equipped with a variety of **input**- or **output terminals**.

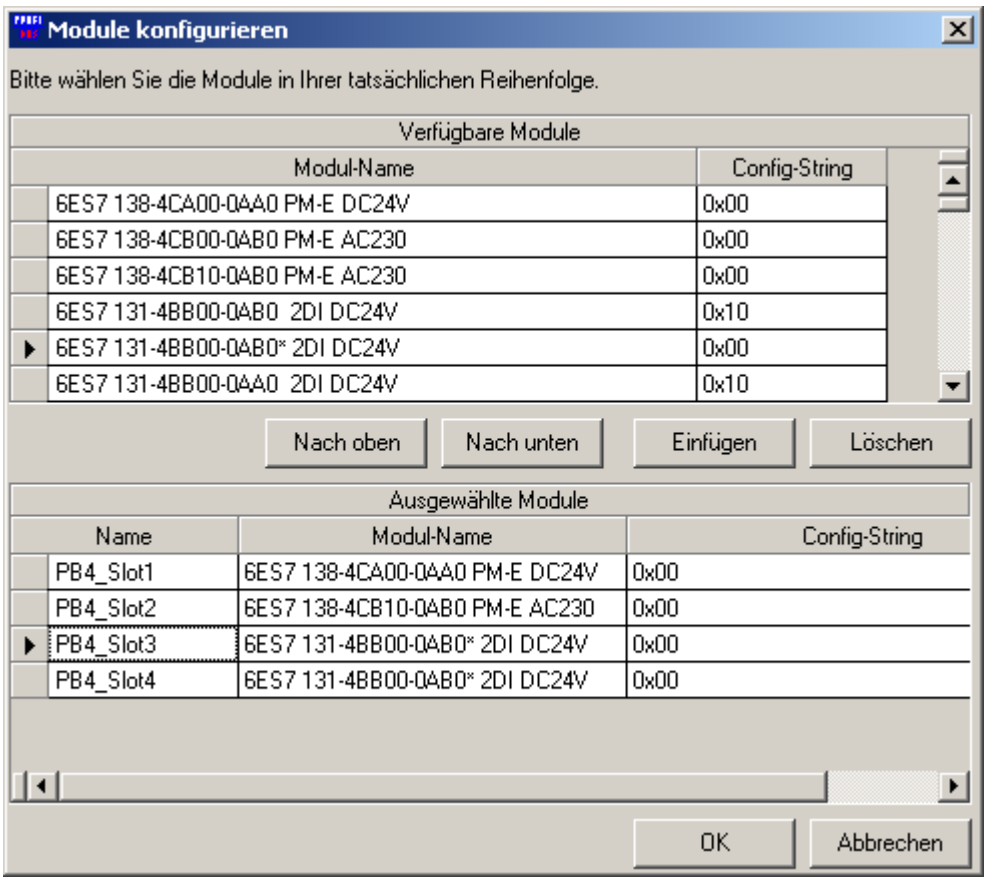
Right-clicking over the Master or the Slave opens the context menu:



Properties enables direct editing of the ID and name.

To change individual elements' properties separately, select **Module bearbeiten.**

The dialog for configuring the module appears.



Module configuration

Use the *Add* button to add the desired modules. Change the order using the buttons *Nach oben* and *Nach unten*.

Note

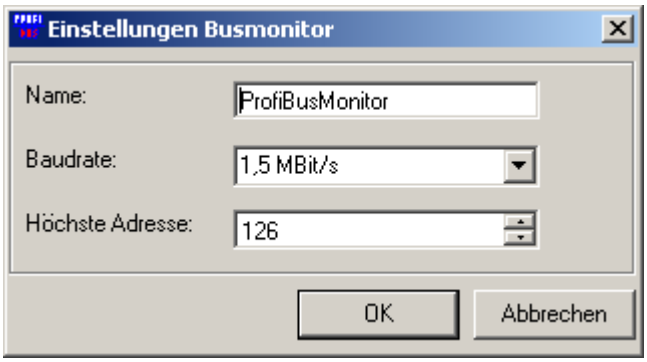
Note that it is absolutely necessary to order the modules in their actual order.

The selected modules are provided with names automatically; this naming can be altered by the user and it later reflected in the imc STUDIO channel names.

9.10.12.1.4 Monitor settings

Using the menu item Profibus/Monitor Settings, you can set the name, Baud rate and highest address.

The Baud rate set here and the name appear in the PROFIBUS Assistant in the connection settings 692.



Monitor settings

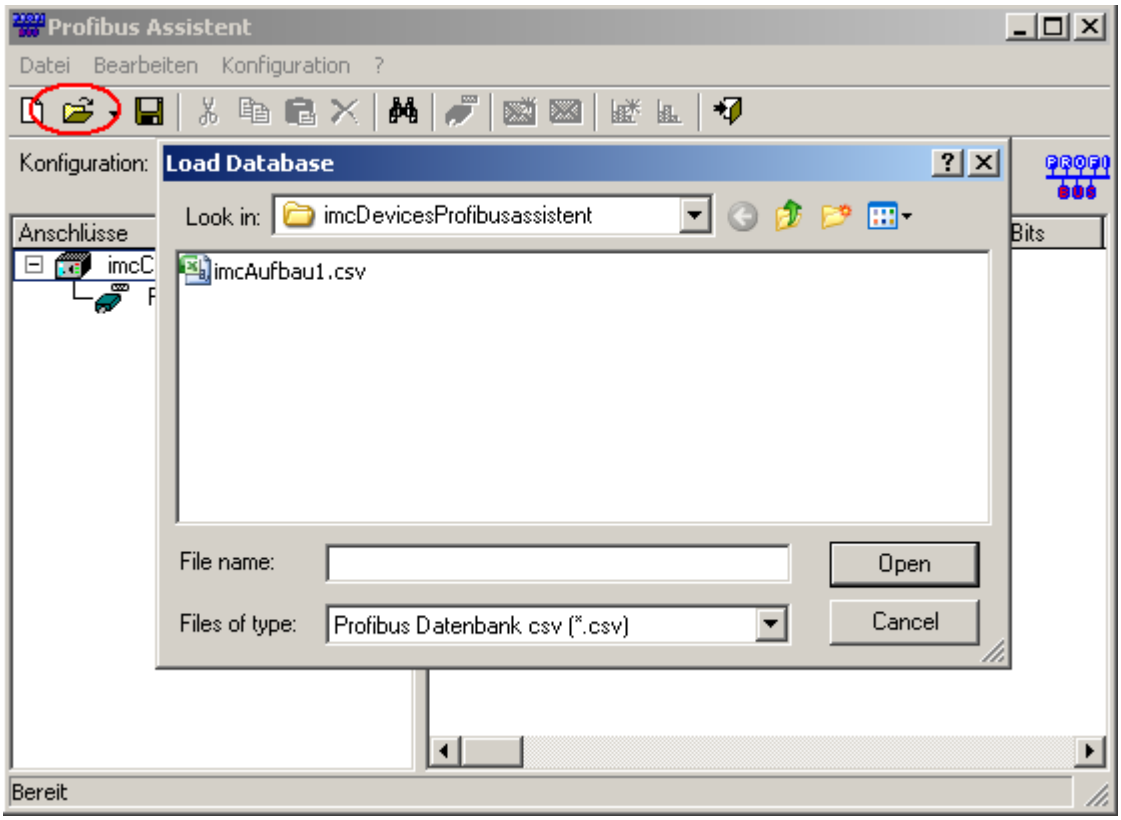
Settings	Description
Baud rate	Transfer speed in bits per second. The Baud rate used by the bus must match the one used by the device. Otherwise, the bus is disturbed and unable to read data.
Name	Designation of the respective PROFIBUS port. This is mainly for the purpose of distinguishing multiple PROFIBUS-monitor connections.
Highest address	The highest address refers to the highest possible subscriber address in the Bus. Normally, it is set to 126. If you have fewer subscribers, you can reduce the highest address. When you try to access a higher address, an error message appears. We recommend leaving this entry at 126, since there are no advantages to reducing it.

9.10.12.2 Transferring the configuration to imc STUDIO

Once you have modeled your system with the PROFIBUS Configurator, add the configuration to an imc STUDIO experiment.

To do this, proceed as follows:

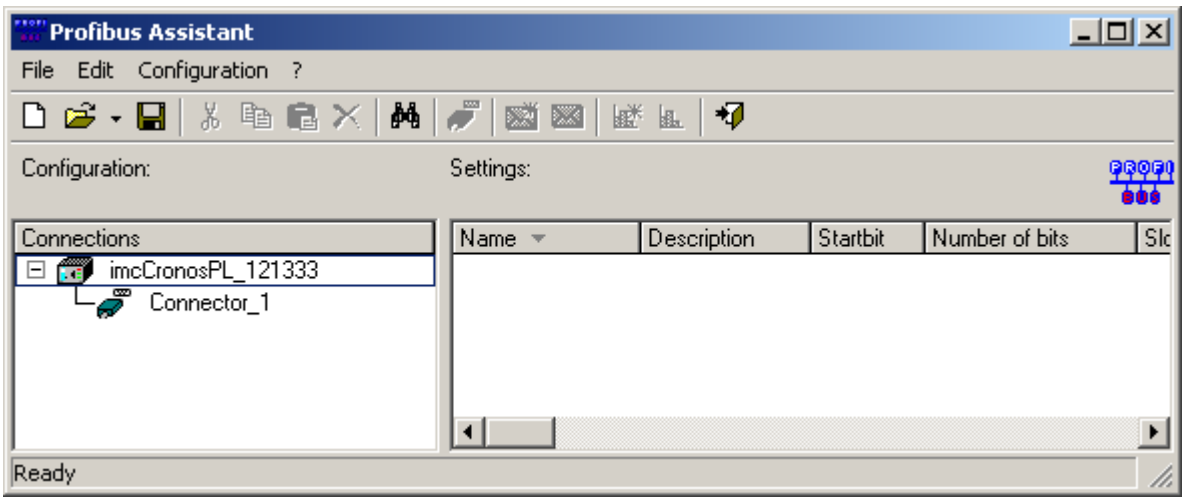
1. In the main PROFIBUS Configurator window, select the menu item *File > Export*
2. Save the file wherever desired
3. Start imc STUDIO
4. Start the PROFIBUS Assistant 692
5. Open the file saved in step 2 above



Loading the configuration

9.10.12.3 PROFIBUS assistant

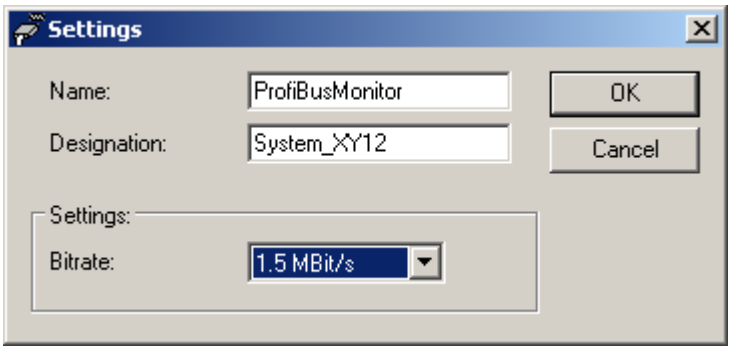
Start the Assistant from within the imc STUDIO software via the Ribbon *Setup-Configuration > Profibus-Assistant*.



PROFIBUS Assistant

Menu	Description
File menu	Here you will find the usual items for opening and saving a configuration in the CSV format. The configuration is saved in XML format. However, the loading dialog is preset to "CSV" and hides XML files. To display the XML files, select "*.*".
Edit menu	Along with copy and paste, functions for searching and for selecting list entries are here.
Configuration menu	contains the functions for creating a PROFIBUS configuration. The functions are described further below.

9.10.12.3.1 Connection settings

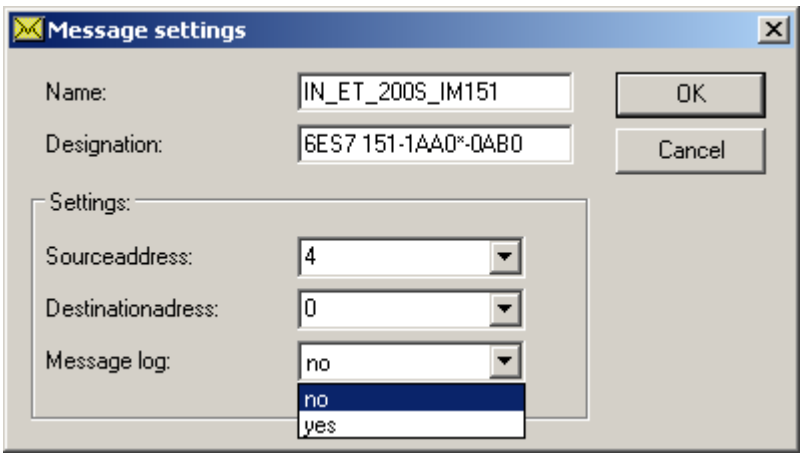


Connection settings

Name, Designation and Bitrate of the connection. The settings 690 created in the PROFIBUS Configurator appear here automatically.

9.10.12.3.2 Message settings

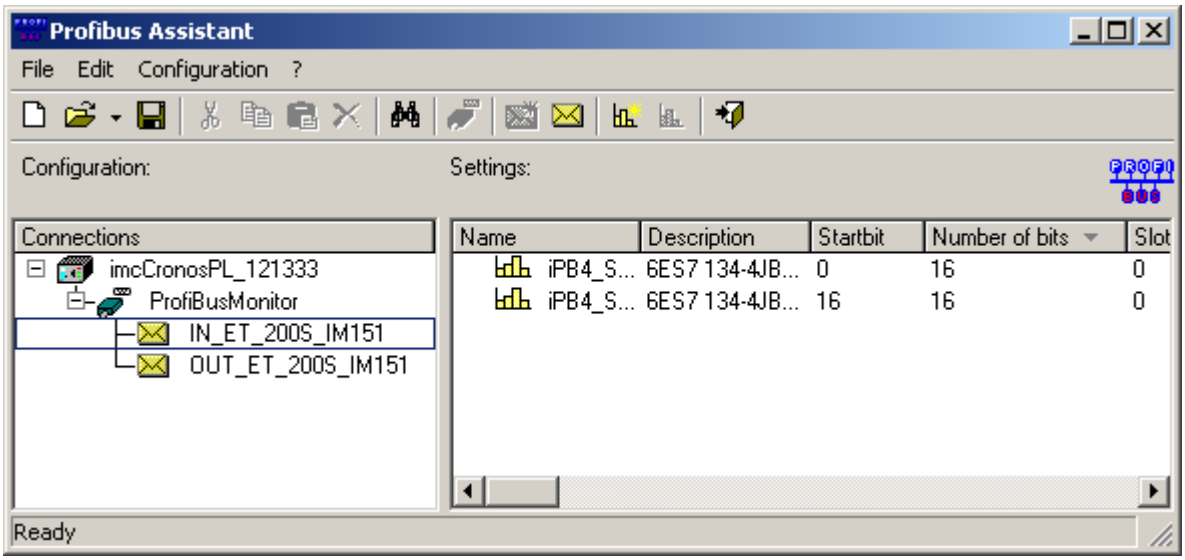
The configuration file created with the PROFIBUS Configurator 684 already contains definitions for all available messages in the system.



Message settings

If you wish to view all of a message's raw data, activate the control Message log in the message settings. The log channel is then available as a separate channel in imc STUDIO.

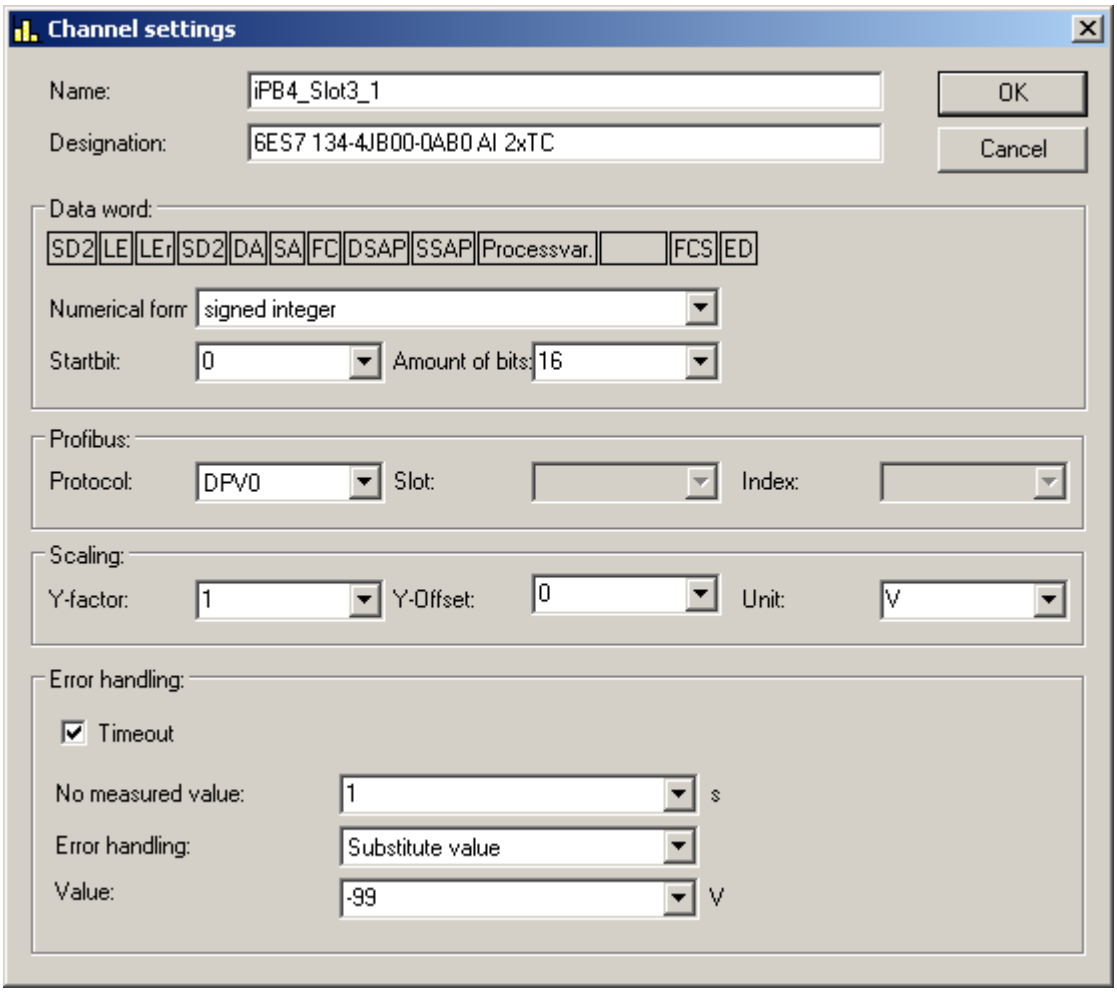
9.10.12.3.3 Channel settings for imc STUDIO



Profibus Assistant

Messages are shown on the left side. The messages are actually the PROFIBUS telegrams extracted from the recorded data stream. They are identified by their address data (source / target address).

Each message contains multiple channels, which are displayed on the right side. By right-clicking on the respective channel, you open the channel's context menu. This menu contains the item *Channel properties...* By clicking on this menu item you access the settings dialog below.



Channel settings

With this dialog you set the scale for the channel's data and name the units. You also filter individual bits out of the channel by changing the entries for the start bit and stop bit values. This procedure is especially sensible for binary data.

Once you have parameterized all the channels desired, exit the PROFIBUS Assistant. The parameterized channels then are available in imc STUDIO in the usual way.

Setting	Description
Name	Channel name for imc STUDIO. The usual constraints on variable names apply: no special characters, the first character must not be a number, no spaces etc.
Designation	Additional description appearing in imc STUDIO as a comment. Maximum of 256 characters, otherwise no constraints.
Data word	Message which is uniquely identified by the name. • Numerical format: Signed or unsigned integer, 32 or 64 bit or digital. • Startbit: Position of the channel's first bit, beginning with 0. • Amount of bits: Selection by means of list or by direct numerical entry.
Profibus	DPV0 or DPV1 protocol. With DPV1, entry of the slot- and index is accessible.
Scaling	Entry of the factor, offset and unit: Resulting value = number in the data word x Factor + Offset [unit]
Error handling	Only taken into account of timeout is activated. • No measured value since: Duration since when a missing value was regarded as an error • Error handling: Last value as a number, substitute value or ignore message • Value: Substitute value if this was the setting under Error handling

9.10.13 RoaDyn interface

This RoaDyn Interface provides the interface between the wheel force transducer system from Kistler with an imc measurement system, which is equipped with this RoaDyn Interface. The RoaDyn 2000 as well as the KiRoad Performance System are supported. The compatible Kistler unit versions are 4.01a, 4.01b and c with DSP-Type_ VC33.

Along with the RoaDyn system's (3x force, 3x torque, angle, angular velocity, temperature, supply voltage) main channels, all additional service channels (single force components, errors etc.) are available.

9.10.13.1 What is the RoaDyn system?

The RoaDyn-On-board electronics system 2000 is a "RACK"-based series of devices for connection to RoaDyn wheel force transducers.

The wheel force transducers are mainly used in endurance strength tests and for in-vehicle measurements of dynamic road traction. This typically includes measurements of performance profiles, friction coefficients, and air resistance. The collaboration between the RoaDyn 2000 system and the RoaDyn wheel force transducers in test drive settings provide insight into the performance of both the engine and the power train.

KiRoad Performance

The KiRoad Performance provides the power supply for all RoaDyn wheel force transducers. It also processes the raw signals from the load cells with crosstalk and lever-arm compensation. The control unit delivers data in digital and analog form to maximize the accuracy of measurement results. See the manufacturer's documentation for notes on operation of the system.

imc Online FAMOS as an enhancement of the imc measurement device comes with a large variety of real-time functions for the purpose of analysis and evaluation of measurement results. The imc Online FAMOS Class-Counting Kit provides functions for calculating endurance strength. imc Online FAMOS makes it possible to immediately transform incoming measured data into results, and to accomplish this without a PC directly in the measurement device. With synchronization across multiple channels, imc Online FAMOS is able to perform comparisons and other mathematical operations on incoming data and compile statistics. Configuration settings for RoaDyn can be made by Remote control. See the manufacturer's documentation for notes on operation of the system.

9.10.13.2 Prerequisites

imc Hardware (imc measurement device)

- device with RoaDyn Interface

Signal interconnections

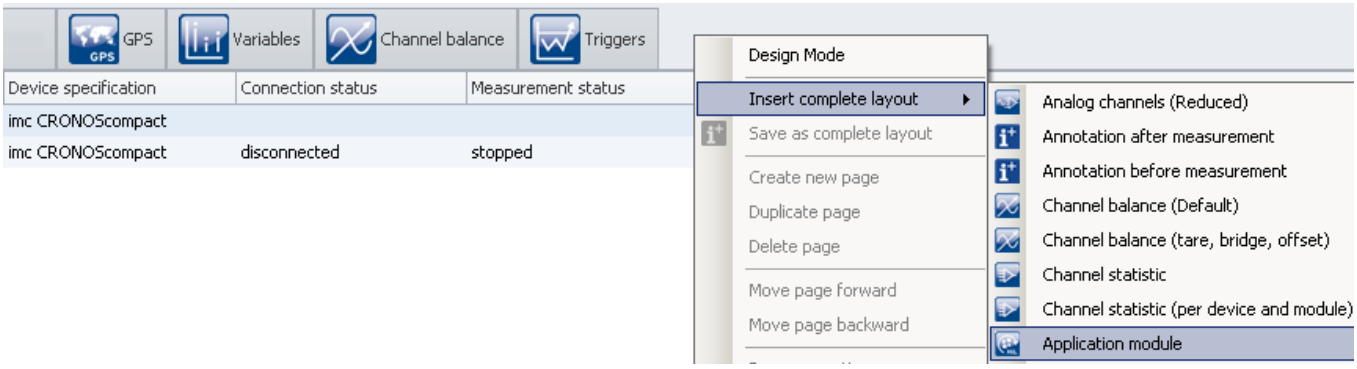
- RoaDyn 2000's BNC terminals Clock and Trigger (CLK and TRG) are to be connected to the Application module's respective BNC terminals of the same name CLK and TRG.
- The CRFX/RoaDyn Application module's network jack (RoaDyn) is to be connected to the RoaDyn-On-board electronics System 2000's Ethernet interface (ETH).

Note

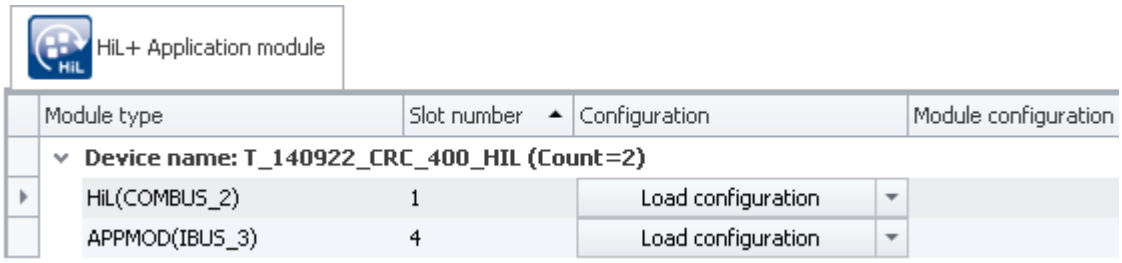
Carefully check the signal connections. The Application module's Ethernet socket must not be confused neither with the imc measurement device's standard terminal (LAN) nor with EtherCAT OUT.

How to open the assistant with the desired application?

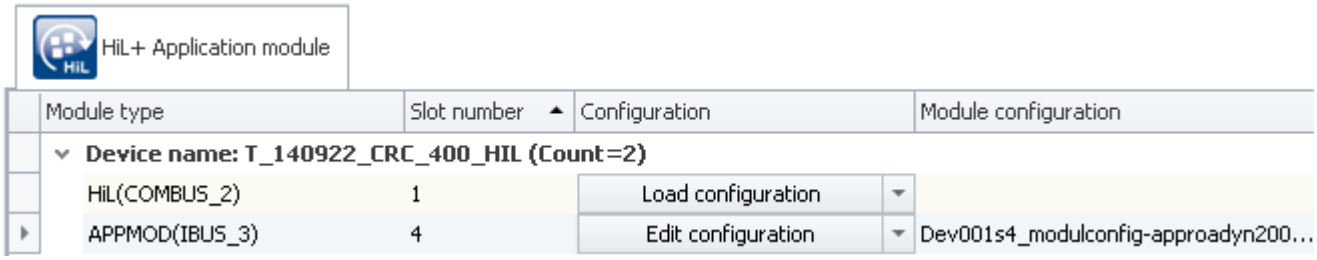
First of all it is necessary to add an additional pre-configured setup page. In order to add the needed page to your user interface open the context menu over a page-tab or over the empty space to the right. Select the following page entry in the list under *Insert complete layout*: *Application module*.



In consequence, the page is inserted to the right of the selected locations:



Now you can load the APPMOD configuration, please select the file ***ModulConfig-AppRoaDyn2000.appmod.zip*** that contains the application and is located on your data carrier in the folder: ..\Products\imc DEVICES\Tools\imcAppMod\Applications.

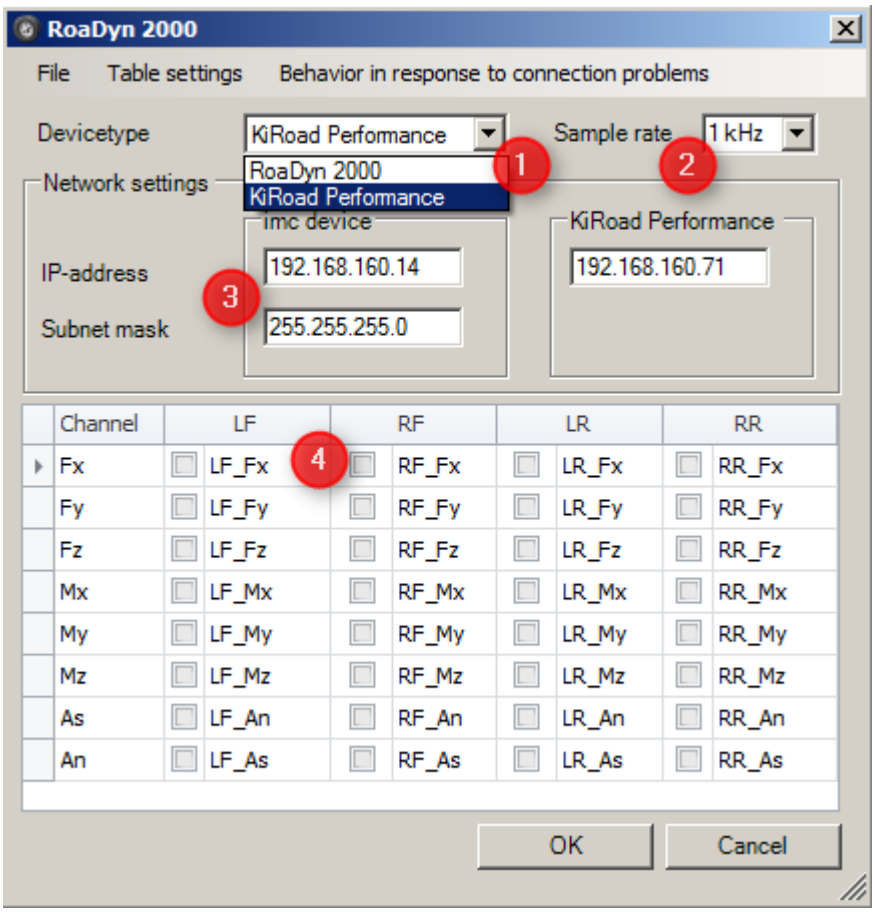


Now you can open the assistant by clicking the button "*Edit configuration*".

9.10.13.3 Configuration mode

In Configuration mode of the AppRoaDyn2000 application, the following settings are available:

[1] Devicetype (RoaDyn 2000 or KiRoad Performance)
[2] Sample rate
[3] Network settings
[4] Selection of wheel signal channels



[1]: Please select your device:
RoaDyn 2000 or
KiRoad Performance

The wheel signal channels for the individual wheels are arranged in columns **[4]**:

LF: left front wheel, RF: right front wheel,
LR: left rear wheel, RR: right rear wheel

The channel names are assigned according to the RoaDyn 2000 designations. In the displayed dialog (see above), the x-direction force on each wheel force transducer is selected. The channels appear as fieldbus channels and are available for the purposes of triggering and for imc Online FAMOS calculations, etc.

At this time, the following **sample rates [2]** are available as settings for any channel: 1 ms (1 kHz), 2 ms (500 Hz), 5 ms (200 Hz), 10 ms (100 Hz) and 20 ms (50 Hz).

Note

Changing the sampling interval during measurement leads to malfunctions.

When configuring the network **[3]**, the Application module's and the RoaDyn 2000's or KiRoad Performance system's IP addresses must be entered.

Note

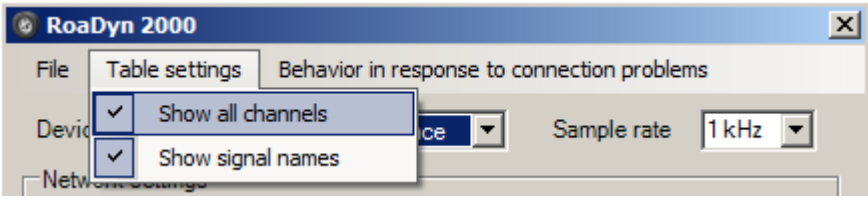
In the Configuration dialog (see above), the imc measurement device appears above the box in which you must enter the Application module's IP address.

In addition to the IP address, you must enter the network mask for the Application module, e.g. 255.255.255.0. The RoaDyn System 2000's or KiRoad Performance system's IP address can not be changed using the application "AppRoaDyn2000". This address is preconfigured and can be imported by means of the RoaDyn Remote control.

Note

The Application module and the RoaDyn System 2000 or KiRoad Performance must be located in the same net mask with different IP addresses.

For **special tasks**, such as accessing individual hub channels (in cases of servicing the wheel force transducers), all channels can be activated for the AppRaoDyn2000 Assistant.



9.10.13.3.1 Editing a configuration procedure and simulation mode

The following module parameter: "***Operation Mode"*** can be configured with three values. This module parameter is preconfigured with the value "0". This value is also used whether a valid module parameter could not be found.

0	If the RoaDyn 2000 respectively the KiRoad Performance could be found, possible configuration errors of the RoaDyn 2000 device will be announced during measurement preparation. If the RoaDyn 2000 respectively the KiRoad Performance could not be found, the simulation mode will be activated and will commit alternate values ("0.0") into the channels.
1	In this mode there will be no configuration problems with the RoaDyn 2000 respectively the KiRoad Performance announced during measurement preparation. The first configuration problem with the RoaDyn 2000 respectively the KiRoad Performance will activate the simulation mode.
2	Each configuration problem / communication problem with RoaDyn 2000 respectively the KiRoad Performance will abort the measurement preparation.

Once the simulation mode is activated the configuration circle will terminate without errors. The activation of the simulation mode will be reported with application specific error code #15 (-5447) when the measurement starts.

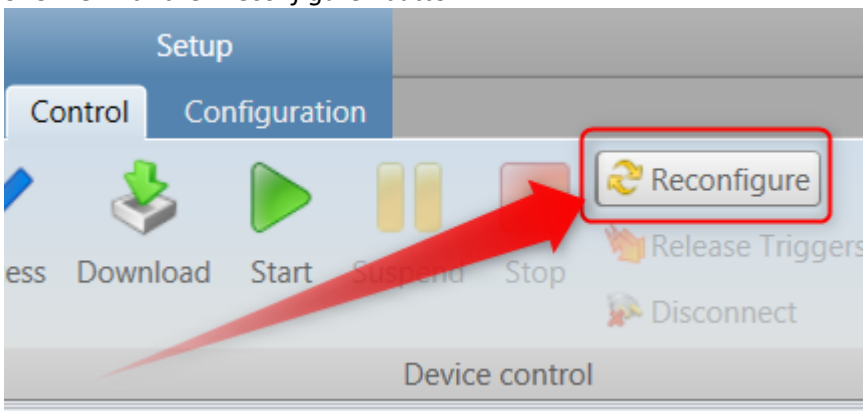
Note Note on the wiring of *KiRoad Performance*

If the "Synchronization Cable 55135189" from Kistler is used, the BNC line "1" must be used for clock and the BNC line "2".

9.10.13.3.2 Prerequisites for transferring the configuration

If the RoaDyn is in measurement mode, a changed configuration cannot be transferred directly. The following steps must be observed here:

1. Activate the configuration mode *KiRoad Performance* 697.
2. It is recommended to establish the Ethernet connection to the KiRoad Performance control unit via a switch so that the IP configuration is safely retained.
3. To enable the transfer of the configuration, the data transfer must be stopped. To do this, execute the imc STUDIO with the "*Reconfigure*" button.



Now the external configuration of the sensor is possible.

9.10.13.4 Application update

In the case of operation on a standard Application module, an application update is performed as described in the instructions for the Application module assistant. In the version of the Application module defined for the operation of the AppRoaDyn2000, the application is updated during the normal software update of the device. If adjustments to the configuration have to be made, this is done by the assistant.

9.10.13.5 Troubleshooting

Possible messages appearing during a measurement procedure

The following errors affecting the Kistler- System 2000 can cause an error message to be posted:

Message	Description
(-)5408	The configuration is not in order; —> Check the configuration using the Assistant. This message is generated for KiRaod Performance and RoaDyn 2000 if the simulation mode is not desired in such a case.
(-)5410	Initialization of RoaDyn® 2000 is not possible; —> Check the RoaDyn® 2000's network settings and the configuration, restart RoaDyn® 2000 This message is generated for KiRaod Performance and RoaDyn 2000 if the simulation mode is not desired in such a case.
(-)5434	Communication problem with the STOP message; Unable to stop the running RoaDyn® 2000 measurement. —> Check the RoaDyn® 2000's network settings and the configuration.
(-)5435(3)	Internal software problem. —> Switch the devices off and then on again.
(-)5436(4)	Communication problem with the device: The HELLO message was not answered or no connection to the device could be established (Socket connections fail). —> Check the RoaDyn® 2000 configuration; switch RoaDyn® 2000 off and on again. This message is generated for KiRaod Performance and RoaDyn 2000 if the simulation mode is not desired in such a case.
(-)5437(5)	Communication problem with the GETMAC message: Query of the RoaDyn 2000's MAC-address failed. —> Check the RoaDyn® 2000 configuration; switch RoaDyn® 2000 off and on again.
(-)5438(6)	Communication problem with the AUTO message: RoaDyn® 2000 unable to activate the channel setting. —> Check the RoaDyn® 2000 configuration; switch RoaDyn® 2000 off and on again.
(-)5439(7)	Communication problem with the DCC message: RoaDyn® 2000 has not processed the channel selection. —> Check the RoaDyn® 2000 configuration; switch RoaDyn® 2000 off and on again.

Message	Description
(-)5440(8)	Communication problem with the AUTO message (after preparation of DCC message): —> Check the RoaDyn® 2000 configuration; switch RoaDyn® 2000 off and on again.
(-)5441(9)	Communication problem with the SetMode message. RoaDyn® 2000 unable to set the measurement mode. —> Check the RoaDyn® 2000 configuration; switch RoaDyn® 2000 off and on again.
(-)5442(10)	Communication problem with the SetSampleRate message: RoaDyn® 2000 unable to set the sampling rate. —> Check the RoaDyn® 2000 configuration; switch RoaDyn® 2000 off and on again.
(-)5443(11)	Communication problem with the SetClock message: RoaDyn® 2000 did not apply the Clock-setting. —> Check the RoaDyn® 2000 configuration; switch RoaDyn® 2000 off and on again.
(-)5444(12)	Communication problem with the SetClock message: RoaDyn® 2000 did not accept the Clock-setting query. —> Check the RoaDyn® 2000 configuration; switch RoaDyn® 2000 off and on again.
(-)5445(13)	Problem with setting the Clock-Mode. RoaDyn® 2000 did not accept the Clock-setting. —> Check the RoaDyn® 2000 configuration, check the Trigger- and Clock-lines. During the measurement, this error code is reported if KiRoad Performance has stopped the "Data Stream" at TCP level.
(-)5446(14)	During measurement preparation, RoaDyn® 2000 device either DID NOT respond to the channel poll or reported an error. This is generally due to the RoaDyn® 2000's operating terminal not being in the basic screen. This must first be resolved and then the measurement preparation is to be repeated after a waiting time of at least 15 seconds.An error occurred while saving the configuration in KiRoad (Run: <store></store> has failed). During the measurement, this error code is reported if errors occur when reading the "Data Stream" from KiRoad Performance.
(-)5447(15)	RoaDyn® 2000 device was not configured correctly, the application is in simulator mode and reports with the configured sample rate a "0"-value as alternative. Note: This message will only be generated after a measurement initialization.

Note Important notes for KiRoad performance

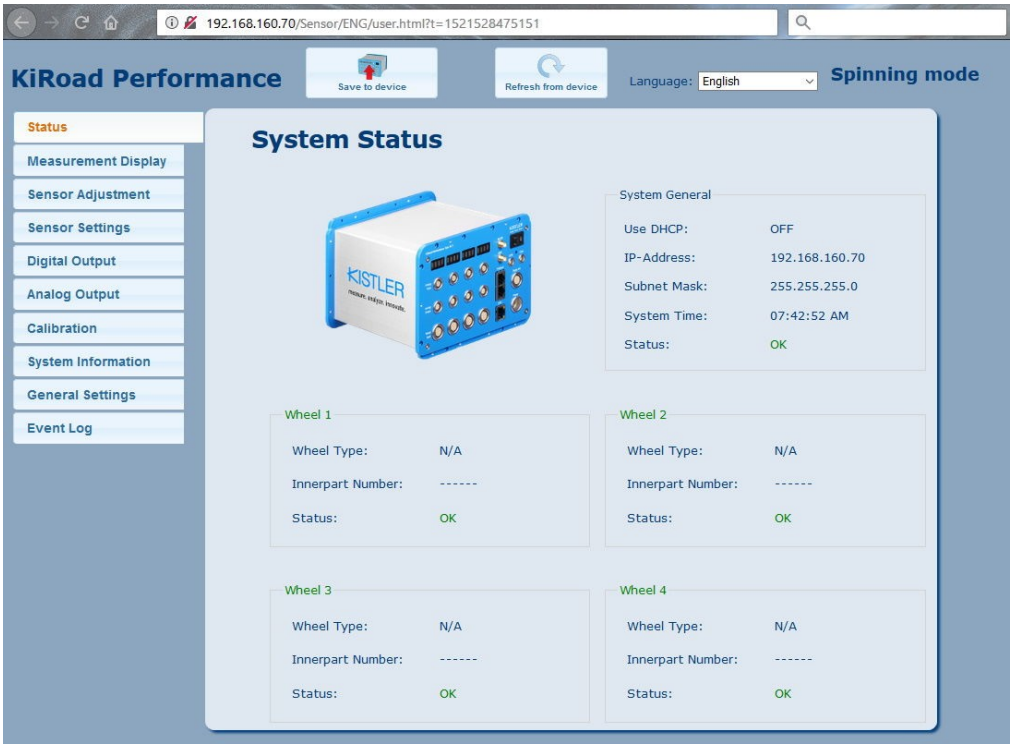
Operation with KiCenter

While working with KiRoad Performance, problems can occur if the KiCenter is open. In this case, terminate all active KiCenter instances in the KiRaod Performance network.

WEB interface

KiRoad Performance can be configured with a user interface in the web browser. With this you can read and download the EventLog of the KiRoad Performance. In the EventLog you can also find notes on malfunctions, the meaning of which can be requested from Kistler.

Example of the KiRoad Performance web interface:



Example of the KiRoad Performance web interface

9.10.14 SPI-bus interface

What is the SPI-bus?

The Serial Peripheral Interface (SPI) is a bus-system for a synchronized serial data bus, by means of which digital circuitry is connected according to the Master-Slave principle.

The SPI-module's firmware is supported by FPGAs in processing the command lists. This includes configuring the chip selects and generating the correct bit sequence for the frames on the MOSI signal line. The MISO frames coming from the sensor are captured, time-stamped and written to a log channel along with an identifier. One log channel is generated for every chip select on each SPI. If multiple sensors are addressed via different chip selects on the same SPI interface, one log channel is generated per sensor. The focus is on achieving good performance even when there are many sensors and a high clock rate, so that the rated maximum data rates on the log channel can be assured over the long term.

Reference

The description of the hardware and the pinouts of the terminals are presented in the special documentation of your SPI. These specs will be amended to future editions of the device manuals.

What settings are needed?

- Hardware parameters
- Assignment of the SPI lines to the connectors (which signal to which pin?)
- SPI clock
- ECLK clock rate
- Adjustability of polarity and phase of the SPI-communication
- Signal conditioning settings on the board, e.g. High signal level and switch
- "In Frame" and "Out of Frame" communication
- Channel name and sample interval (100 ms to 1 s) for the frequency counter (measured frequency at the WDGI input); one frequency counter per SPI is possible.
- Linkage with a Display variable for specifying the internal VDDA and one additional specification of VDDDD. You can use the same Display variable for multiple SPI interfaces.
- Common sampling interval (100 ms to 1 s) for the four analog channels and their channel designations
- Definition of frames
  - Frames based on MOSI and on MISO are defined pair-by pair
- Bit width (8 to 128 Bit) MOSI and MISO

Note

The Bus Decoder 1101 is opened via the page: Data Processing 1098.

Prerequisites

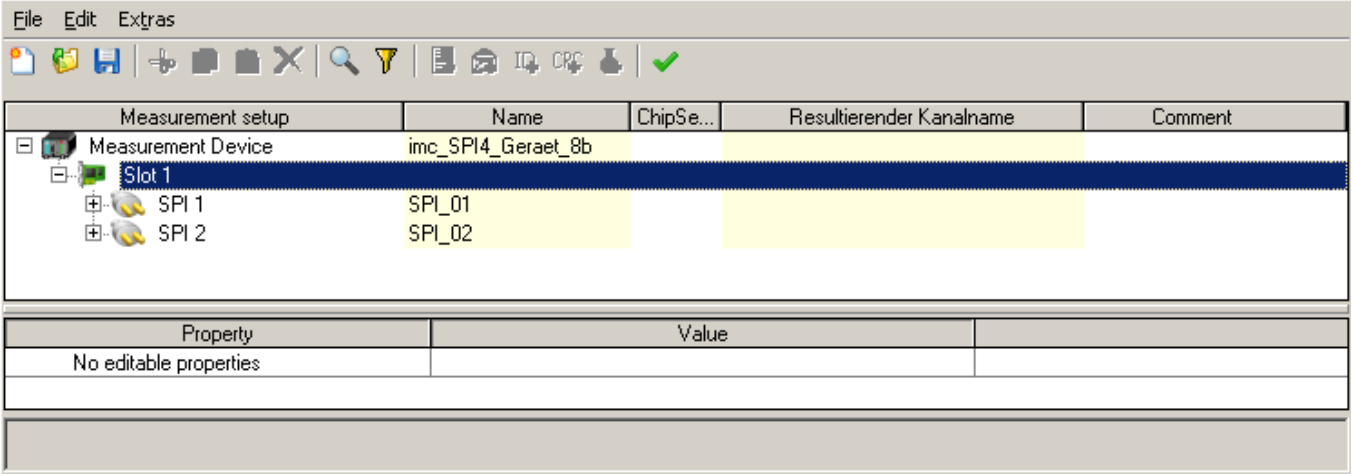
Prerequisites	
Hardware	An imc device with a SPI-Bus interface.
Software	The system requirements are described in the data sheet of our fieldbus connection options.

9.10.14.1 Definition of terms

Term	Description
Bus Decoder	The Bus Decoder shows the channels created upon selection of the data log channel.

9.10.14.2 SPI assistant

For the purpose of creating your configuration, the SPI Assistant displays all SPI interfaces which the experiment contains. These individual base units are selected by navigating in the Assistant.

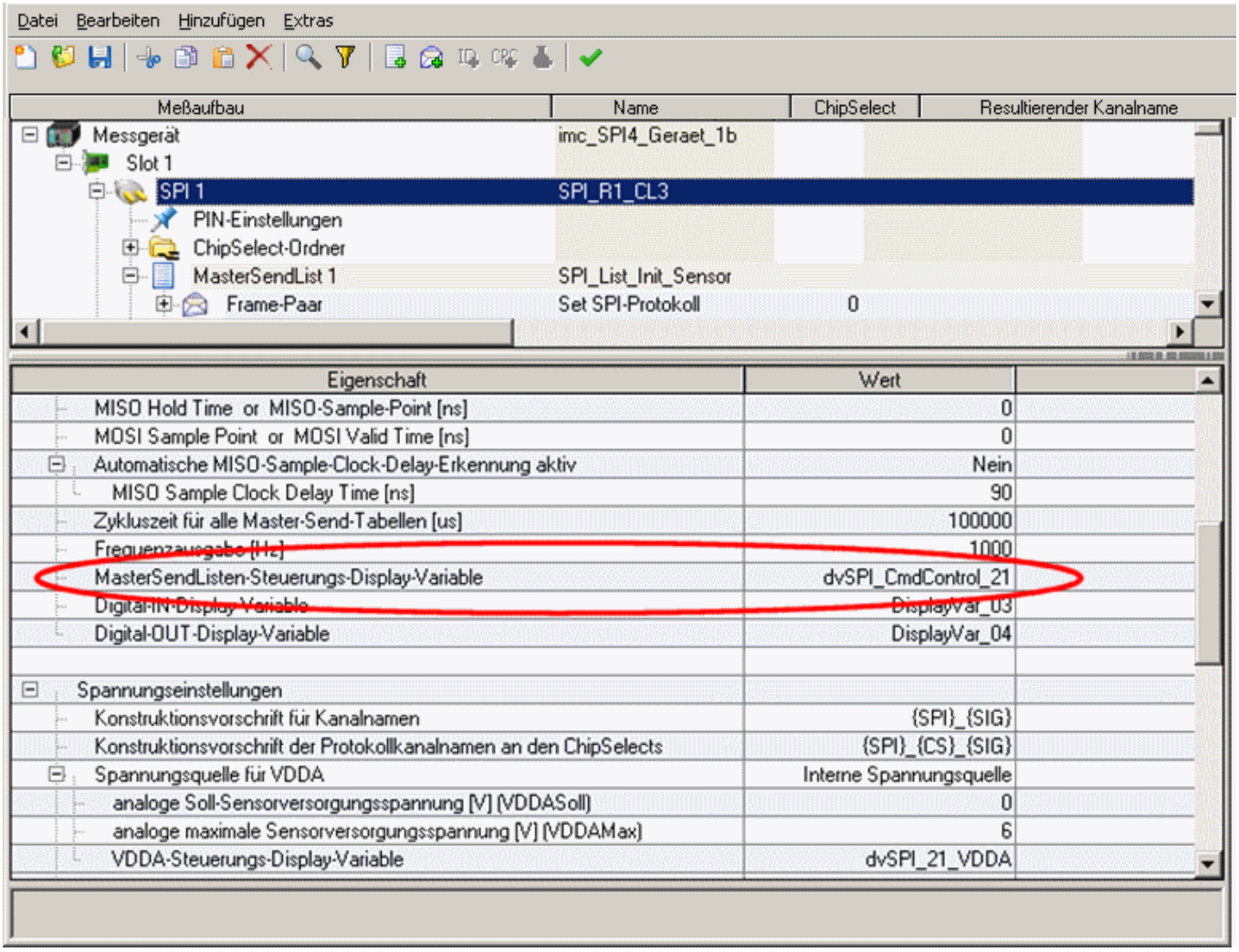


The Assistant is structured in tables, in order to present a large number of parameters in a clear way and to make them available for editing. The Assistant graphically displays the structure of the frame currently being edited.

9.10.14.2.1 Activating and deactivating the Master-send-lists via display variable

General

In the SPI-Assistant, you can select a Display variable at each SPI-interface. This can be used to activate/deactivate already configured Master-Send-lists.



During a running measurement, it is possible to write activation/deactivation commands to these Display variables. To do this, a 16-bit word is written to the Display variable. Once the command has been successfully adopted by the SPI system, the value of the Display variable is reset to zero.

Structure of the 16-bit word

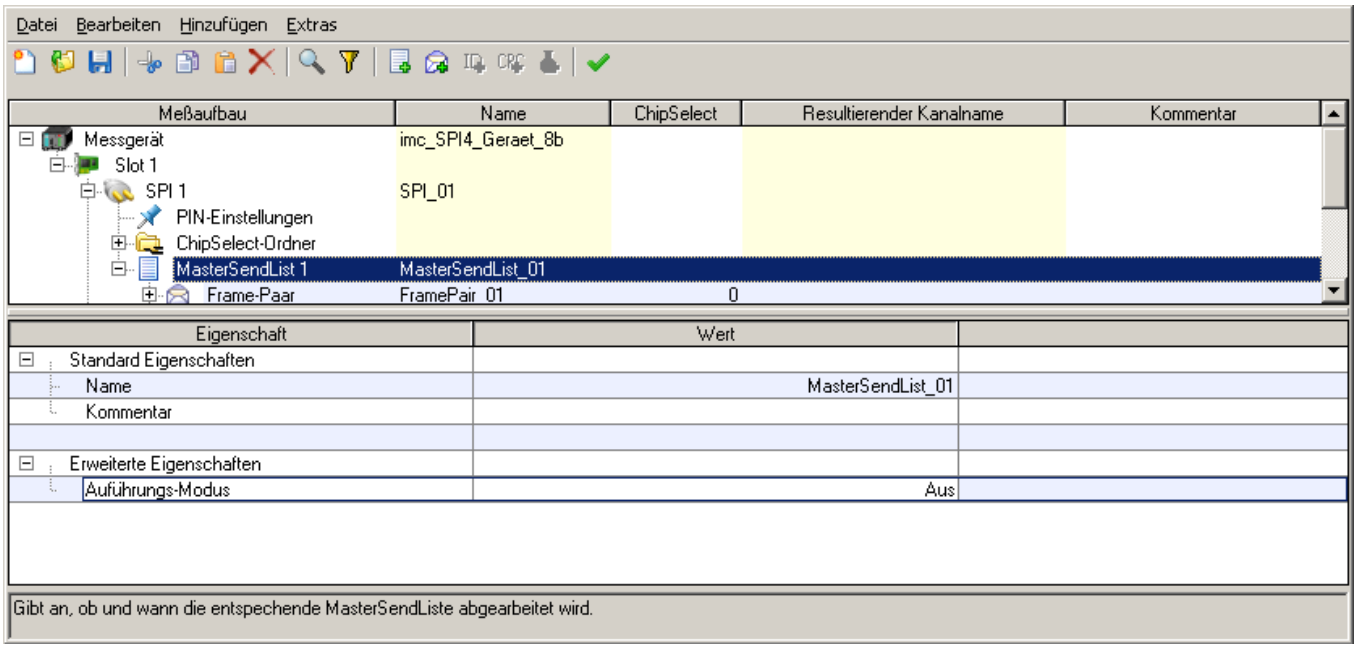
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
4-bit Port				**4-bit** command				**8-bit** list index							

Description of the bitfield

Bits	Description
0-7	List index States the index of the list. The first list has the index 0. Valid values include: 1 through 16, or 0 for all lists
8-11	Command States the command. Valid values include: 0 no function 1 activate Master-Send list as specified for "Execution-mode upon Start/Initialize" (off/once/cyclical) (like command 5)* 2 deactivate Master-Send-list (like command 6)* 3 activate Master-Send-list once 4 activate Master-Send-list cyclically 5 activate Master-Send-list as specified for "Execution-mode upon Start/Initialize" (off/once/cyclically) (like command 1)* 6 deactivate specifies (like command 2)* 7 activate Master-Send-list as specified for "Execution-mode Preset1" (off/once/cyclical) 8 activate Master-Send-list as specified for "Execution-mode Preset2" (off/once/cyclical) 9 activate Master-Send-list as specified for "Execution-mode Preset3" (off/once/cyclical) 10 activate Master-Send-list as specified for "Execution-mode Preset4" (off/once/cyclical)
12-15	Port Specifies the index of the SPI-interface. Valid values include: 1-2, or 15 for all ports

Notes

In the SPI-Assistant, up to 4 presets can be configured for each Master-Send-list. The commands 7 through 10 can be used to switch back and forth among these presets. A single command can replace laboriously switching from list to list each time. This makes the imc Online FAMOS programs smaller and independent of the SPI-Assistant's configuration.



If the voltages VDDD and/or VDDA are also controlled by imc Online FAMOS for the sensors, it makes sense to set the Execution mode to "OFF" at the start of all Master-Send-lists. Otherwise, SPI-data are sent to the sensors, even though no voltage supply is activated, or while the sensors are still booting.

Examples of complete 16-bit commands

Example	Command	previously
Switch off all lists in all ports:	0xF200 or 0xF600	only 0xF600
Switch off all lists in all ports as with prepare:	0xF100 or 0xF500	only 0xF500
Switch on List 1 in Port 1 as with prepare:	0x1101 or 0x1501	only 0x1101
Switch on List 1 in Port 2 as with prepare:	0x2101 or 0x2601	only 0x2101
Switch on List 1 in Port 2 cyclically:	0x2401	0x2401
Switch all lists in all ports to Preset 4:	0xFA00	
Switch List 5 in Port 2 to Preset 1:	0x2705	

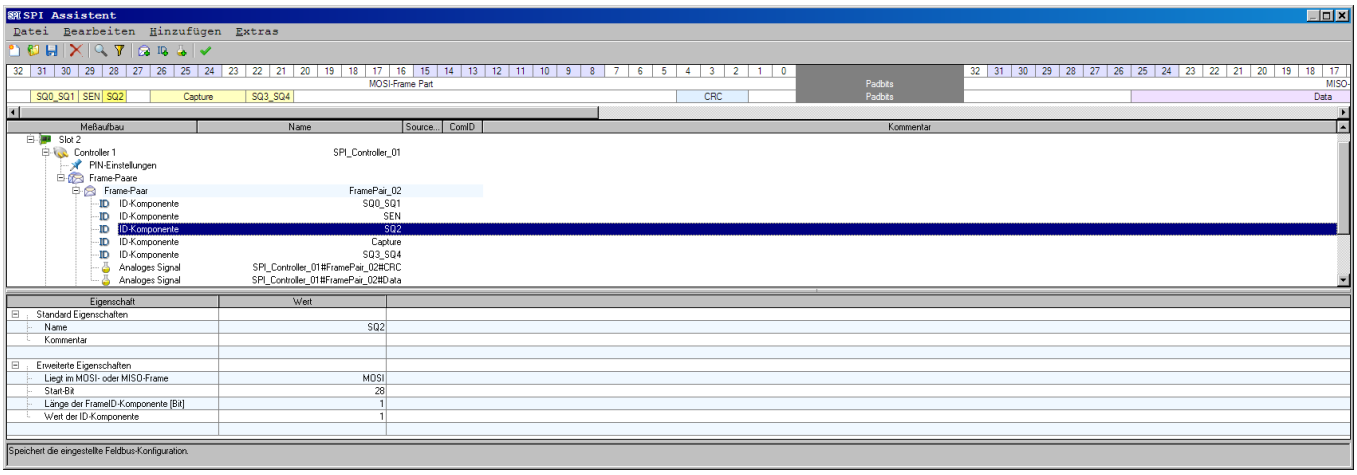
9.10.14.2.2 Chip select settings

Signal output (constant output on MOSI).

Here, you can configure the MOSI frame to be sent, e.g. "SEN" = 1, "SQ_3_4" = 2. These selected value uniquely identify the frame.

- Start bit
- Bit width
- MSB first/last
- Default value or in case of ID: reference value

Multiple such default values can be defined for each frame. The individual defaults may not overlap each other.



9.10.14.2.3 SPI assistant error handling

Start/Stop log: imc STUDIO logs the measurement's starts/stops.

Lost Ethernet connection: imc STUDIO logs whether the Ethernet connection to the device is still intact.

Error in signal:

Checksum and Parity are not logged as separate channels. Instead, they are joined into a common error channel. If for special applications' purposes, the content of the checksum and parity need to be available for the analysis, they are both additionally extracted from the frame like a normal signal.

If a sensor does not respond as expected, a NO_DATA error is reported like for a CRC-error. This error is recognized when a certain bit combination is not present in the sensor's response.

9.10.14.2.4 Special features

Special provisions for conveniently performing the typical editing when there is a large number of sensors which are (almost) the same, without losing the flexibility of connecting multiple different sensors.

- Multi-selection among many signals/frames
- Filter for designation of signals/frames.
- Placeholders for numbering in the name. This provides fast assignment of channel names when there are a number of similar sensors where the channel names only differ by a sequential number.
  Example: for an SMB460 in the frame RD_SENSOR_DATA, you assign to channel 1 the name RD_SENSOR_DATA_CH1_#. Using multi-selection for all sensors in the Assistant, you obtain the names RD_SENSOR_DATA_CH1_1, RD_SENSOR_DATA_CH1_2, RD_SENSOR_DATA_CH1_3.

- SPI-Timeout of each channel
  If no more SPI-data (MISO/MOSI frames) are received within a defined time interval (**Timeout-time**), the channel's value can automatically jump to a pre-set error replacement value. A Timeout occurs, for example, if the channel is deactivated.

9.10.15 XCPoE-bus interface

The XCP Specification documents describe an improved and generalized version of CCP. The generalized protocol definition serves as standard for a protocol family and is called "XCP" (Universal Measurement and Calibration Protocol).

The "X" generalizes the "various" transportation layers that are used by the members of the protocol family e.g "XCP on CAN", "XCP on TCP/IP", "XCP on UDP/IP", "XCP on USB" and so on.

Before continuing reading, it is advisable to have at least skimmed the technical literature named below. This documentation is oriented towards readers possessing some knowledge of the field.

Since "*XCP on Ethernet*" is not truly a standard defined in ASAM, consider the phrase "*XCP on Ethernet*" to mean the conjunction of "*XCP on TCP/IP*" and "*XCP on UDP/IP*".

The imc measurement device can implement either "*XCP on TCP/IP*" or "*XCP on UDP/IP*". The choice can be amended using the Assistant.

Reference Literature and Standards

[1]	"XCP -Part 1- Overview"; Ver. 1.0; ASAM e.V.
[2]	"XCP -Part 2- Protocol Layer Specification" Ver. 1.0; ASAM e.V.
[3]	"XCP -Part 3- Transport Layer Specification XCP on Ethernet (TCP_IP and UDP_IP)"; Ver. 1.0; ASAM e.V.
[4]	"XCP -Part 4- Interface Specification"; Ver. 1.0; ASAM e.V.
[5]	"XCP -Part 5- Example Communication Sequences"; Ver. 1.0; ASAM e.V.
[6]	"Interface ASAP2 Detailed Specification"; "Applications Systems Standardization Working Group"; "Interface Specifications"; "Interface 2"; Ver. 1.40 of 03/31/2000

Prerequisites

Prerequisites	
Hardware	• XCPoE Interface Slave (for sending imc channels via XCP) • XCPoE Interface Master(for receiving XCP with the imc device)

9.10.15.1 Definitions

Term	Description
XCP	Derived from CCP where "X" stands for the different transport layers ("XCP on CAN", "XCP on TCP/IP", "XCP on UDP/IP")
CCP	Can Calibration Protocol
XCPoE	XCP on Ethernet

9.10.15.2 XCPoE slave

Sends channels which are **captured by the imc device** and **sent via XCPoE**.

Operating in "*Slave*" mode specifies that the communication is essentially governed by the external XCP-master and not by the imc-system. However, Slave mode does not actually determine in which direction (in/out) this communication flows. In principle, not only the the outputting of variables is defined for an XCP-slave, but also the receiving (input, stimulus). However, the latter is not supported by the imc Slave-Module at this time!

9.10.15.2.1 Transfer of variables

The Assistant for the XCPoE-Slave Module, offered by imc STUDIO, provide a global option 711 with which all of the imc system's "variables" are activated for transfer. These are embedded in an automatically generated *.A2L 711 file, which can then be transferred to the XCP-Master for configuration.

- All **measurement channels** (FIFO-channels, not process vector variables) of the system are made available:
  - analog channels
  - digital channels
  - Fieldbus-channels
  - Virtual channels from imc Online FAMOS (OFA)
- Channels to be transmitted via XCP **must be assigned to "BaseTrigger"**! This means that they may not be linked to any trigger event or begin upon the measurement's "START".
- The channels are outputted **at their specified sampling rate *as far as possible***, even on XCP:
  - For the OFA channels, the rule is: Output = **sampling rate of the OFA-output**, which allows more than 2 data rates in the system.
  - **Max. 5 different output rates** are supported! This limit is determined by the XCP-protocol.
  - **Max. 100 kHz per channel**: The limit is determined by the XCP-Module's performance.
- The **XCP-messages** are assigned a **time stamp**, which refers to their acquisition date in the imc-system. ***All channels*** in the XCP-message are mutually in ***perfect synchronicity***.
- The moment of output, meaning of the data transport via Ethernet, can vary (latency). All of the system's channels (incl. OFA) must be valid in order that the data can be outputted via XPC consistently with each other at this moment in time.

Note

When OFA channels are transferred to the Master, imc Online FAMOS should not change their sampling intervals. Any sampling intervals changed by imc Online FAMOS will **not** be treated separately.

When the measurement is ended, the following error message may appear: "*Overflow of measurement data!*"

The Master should no longer be connected when the measurement is stopped. In all cases observed where this applied, **no overflow** occurred while the Master was connected.

9.10.15.2.1.1 Update from imc STUDIO before 5.0R1

The design of XCPoE under versions of imc STUDIO prior to 5.0R1 was based on process vector variables. With this design, the channels could be transmitted independently of the trigger. Triggered channels belonging to these configurations would be adopted in the A2L-file and the interface-configuration, but would cause a blockage (wait for the trigger) there.

Note

If there are any existing configurations, based on older versions, having **triggered channels**, then these need to be modified on the occasion of the software update and be **converted to** completely **untriggered measurements**.

9.10.15.2.2 Dynamic creation of the A2L-file

Every XCP-device will usually have an **A2L-file** stating

- how to address the device,
- which data can be queried via XCP,
- and how the data are to be interpreted.

Since the number of channels and their settings such as the factor and offset can be very different from experiment to experiment with imc, the A2L-file is generated only once the data are certain.

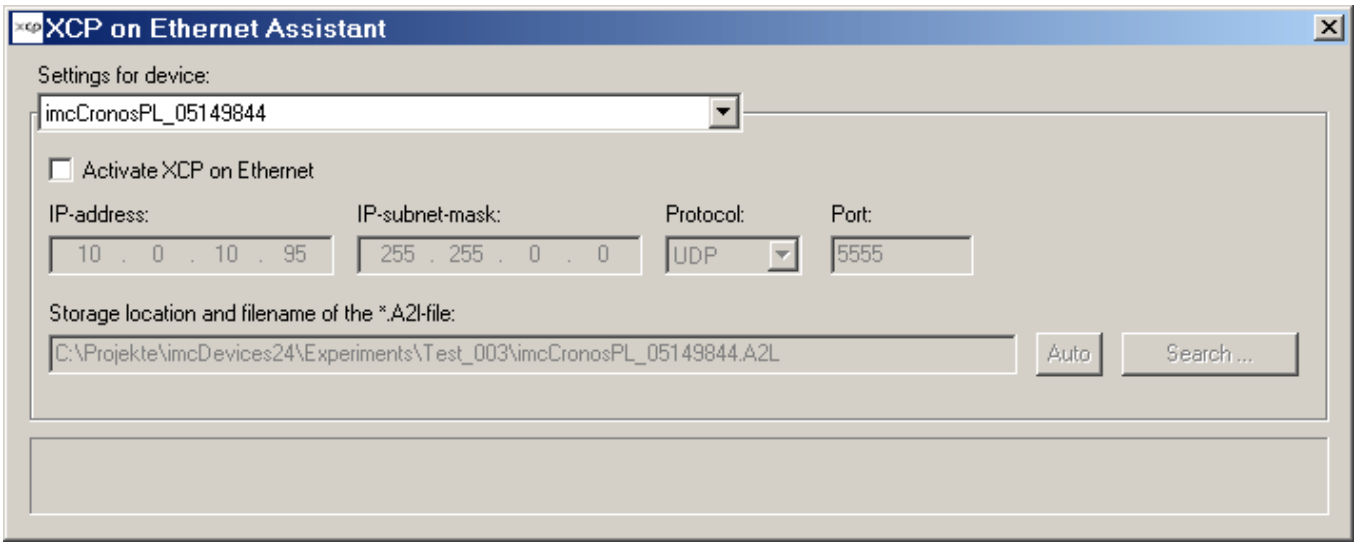
This is the case when the experiment is prepared. In the preparation process, one A2L-file per XCPoE-capable device used in the experiment is written to the storage location specified in the XCPoE-Assistant.

There, the XCP-receiver can retrieve the file.

9.10.15.2.3 XCP on Ethernet assistant - slave

Start the Assistant from within the imc STUDIO software via the Ribbon *Setup-Configuration > XCPoE-Slave*.

Then the XPCoE Assistant opens:



XCPoE Assistant

Setting	Description
Settings for device	Selection of the device if there are multiple devices in the experiment.
Activate XCP on Ethernet	Using this checkbox, the XCPoE can be activated/deactivated at the next preparation procedure. For instance, XCPoE must be deactivated if you wish to assign the IP address set in the connected network to another device. This is because when the XCPoE is deactivated, the IP address which the device may be using is released for use. Also, when XCPoE is deactivated, no A2L-file is written during the preparation process. If XCPoE is activated for an imc measurement device, then the following types of channels can **not** be queried via XCPoE: • TimeStampASCII channels (e.g. Fieldbus-logs)
IP address & IP subnet mask	Here, the IP address and the subnet mask which the device is to use for XCPoE are set.
Protocol	This list box control determines whether "*XCP on TCP/IP*" or "*XCP on UDP/IP*" is to be used.
Port	Depending on the protocol setting, the TCP- or respectively UDP-port via which the device is to be addressed is set.
Storage location and filename of the *.A2L-file	Since the A2L-file is generated during the preparation process, it's possible to set this file's storage location here. **Button: Auto** This button resets the storage location back to the current experiment folder and the A2L file's name back to the device's name, with a A2L file extension. **Button: Search** Enables the selection of a different storage location and a different name for the A2L-file.

9.10.15.3 XCPoE master

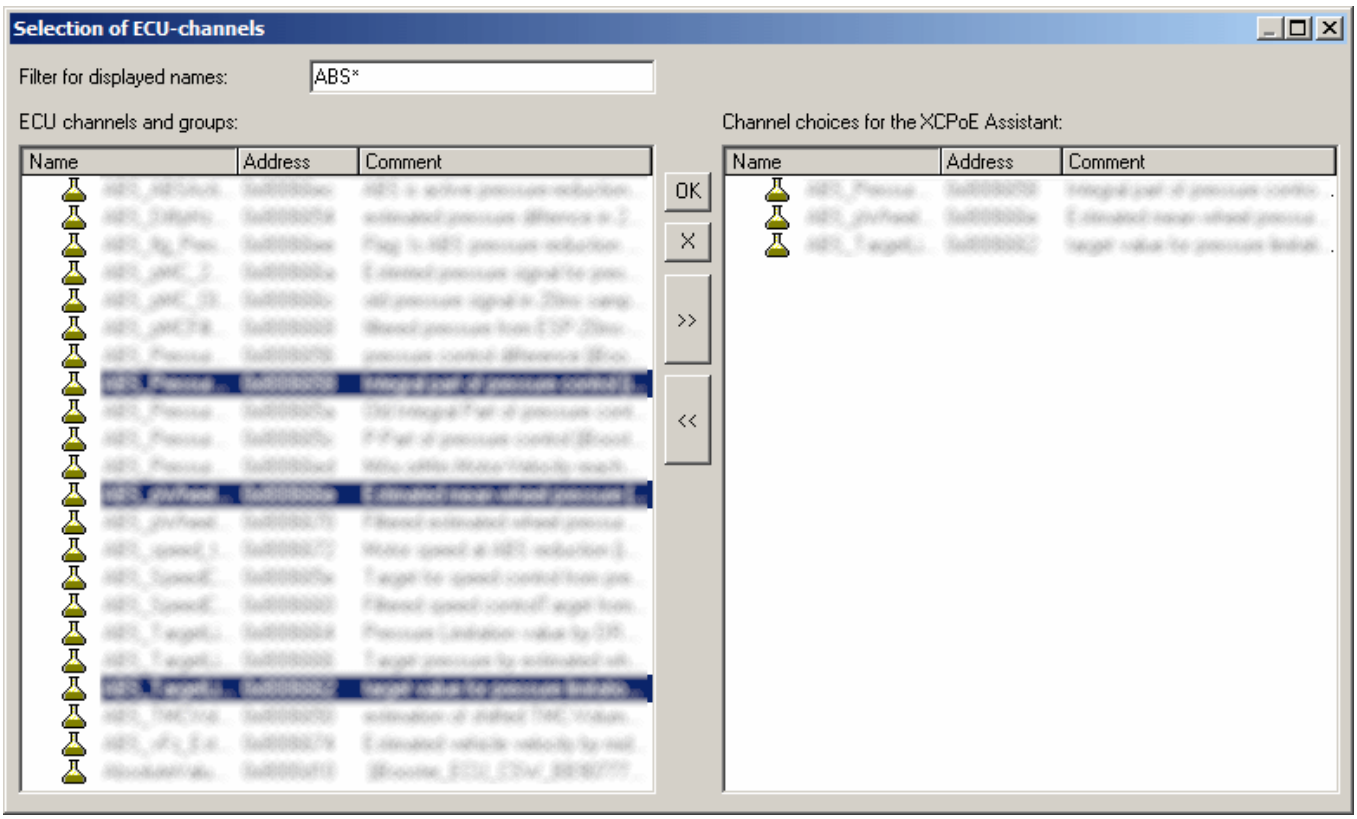
Capture of fieldbus channels via XCP

9.10.15.3.1 XCP on Ethernet assistant - master

Start the Assistant from within the imc STUDIO software via the Ribbon *Setup-Configuration > XCPoE-Master*.

9.10.15.3.2 Loading an A2L-file

Select the A2L file by selecting *File > Import*. Find the channels desired by means of the Filer. Select the channels by moving them to the right side.

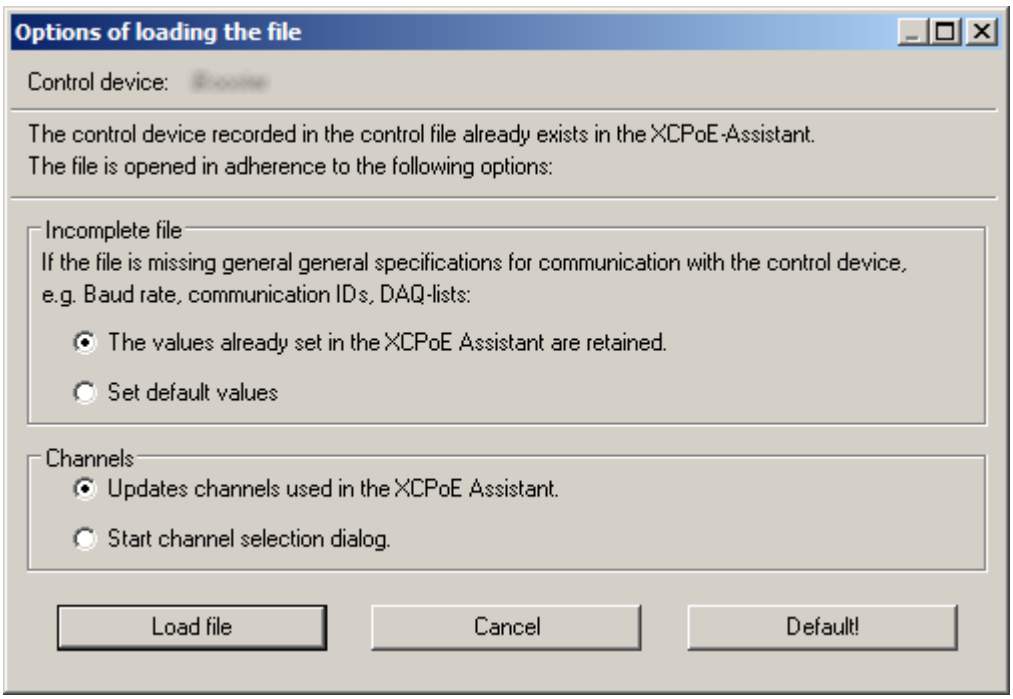


Channels selection

Note XCPplus

The extended format XCPplus is supported.

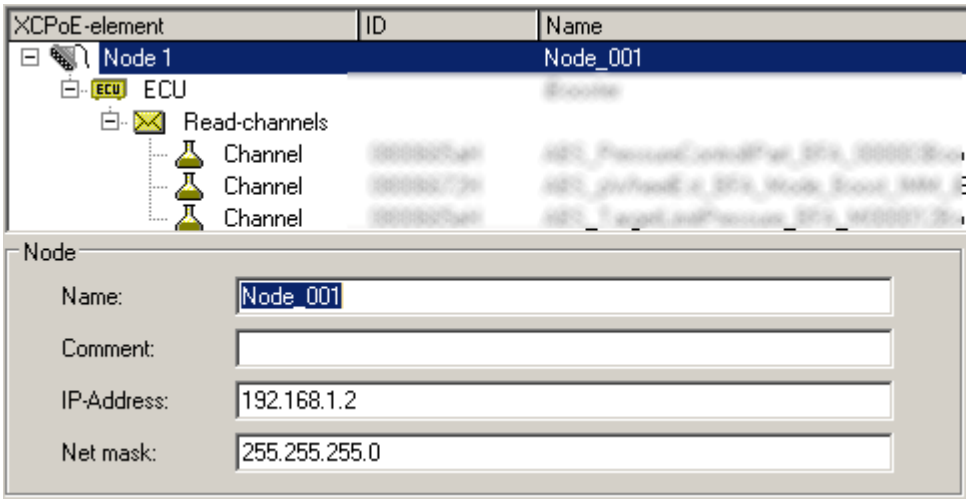
If an ECU branch exists already, an additional dialog appears initially:



Import dialog, if another ECU is already configured

9.10.15.3.3 Node settings

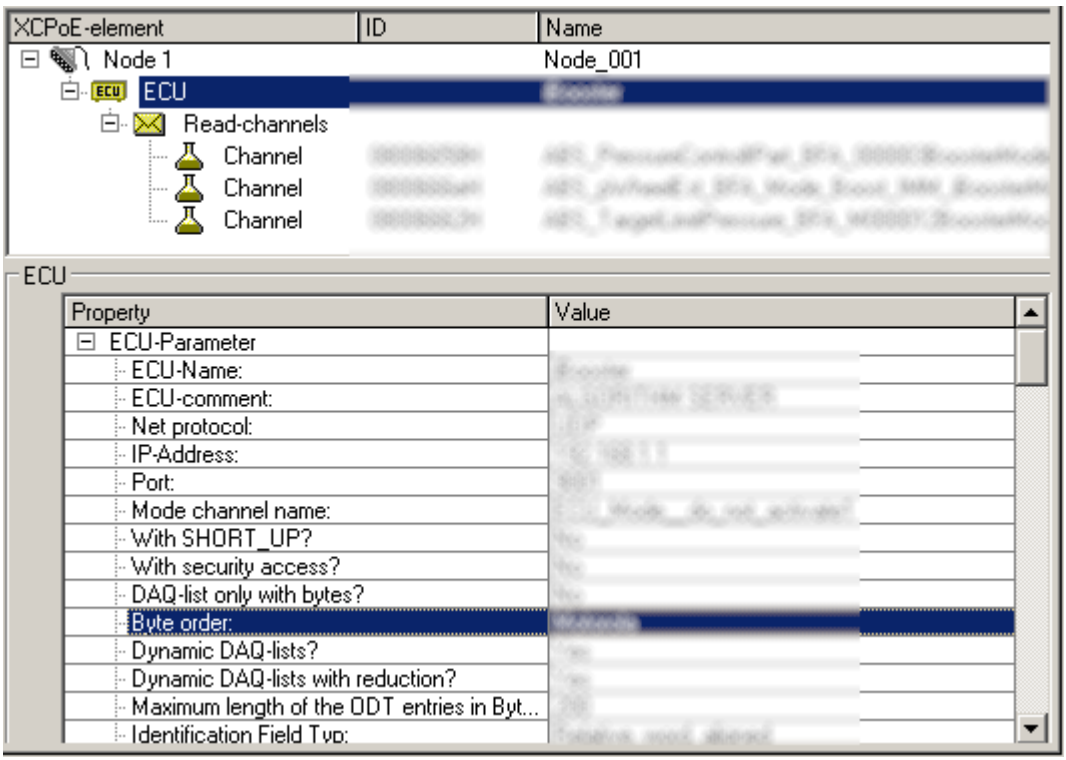
At the node, set the IP address and the subnet mask.



XCPoE settings: node

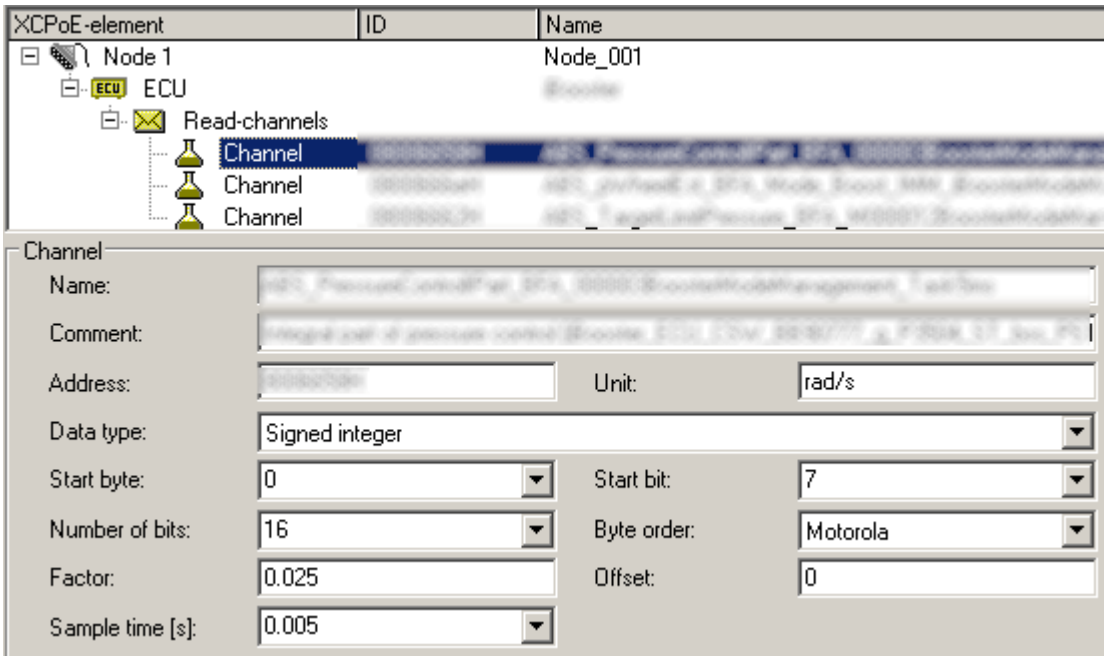
9.10.15.3.4 ECU-settings

Upon selecting the ECU branch, an editable table of the ECU's settings appears.



ECU-settings

9.10.15.3.5 Channel settings



Channel settings

9.10.16 GPS

Reference

See chapter: "*Setup pages - Configuring device*" > "*Configuring Channels and Variables*" > ... > "*GPS* 1146"

9.11 Special modules and Third party devices

9.11.1 WFT module

Motor vehicle developers use 6-component wheel force transducers (WFTs – Wheel Force Transducer) to measure and record the various stresses on the vehicle's wheels during actual test drives – 3 forces (Fx, Fy, Fz) and 3 torques (Mx, My, Mz). These measurement results serve as the basis of simulation computations, or can serve as the input parameters for test bench runs.

The WFT-system is operated via a user interface belonging to the imc CRONOS family of devices. This makes a comprehensive software solution possible in conjunction with imc CRONOS and imc STUDIO.

This document provides a description of the parameters for configuring the WFT-module.

Reference WFT-documentaion

A separate documentation describes setup and operation of the measurement wheel.

Notes

Importing parameters from the sensor

Upon first connecting with the sensor, all relevant settings are imported from the sensor. For instance, the measurement ranges are set appropriately so that they will no longer need modifying. If any change is detected at the sensor (for instance, a different serial number), then the parameters are re-imported upon the next established connection, and the existing configuration is overwritten.

Joint channel settings

The same sampling rate and filter settings apply for all channels belonging to the same connector.

9.11.1.1 WFT-parameters

In addition to the standard configuration choices, the WFT-module has special parameters and combination constraints. The module has two connectors. Internally, various output signals are computed from the signals, which are available as analog channels.

Names of analog channels are structured as follows: ChannelType_Position(l: left or r: right)_Address. Example: Fx_l_00, Revs_l_00, Status_l_00

When you select the WFT-channels from the channel list on the Setup-page "*Analog Channels*", an additional dialog appears: "*WFT*", in which you can make settings.

The following parameters are important for measurement with the WFT-module.

Dialog: WFT - Sensor information (read only)

Parameter - WFT Sensor information	Description		
	long name	*short name*	*column ID*
Sensor type	Sensor type	*Type*	*eSensorMaterial*
	Returns information on the sensor's material: aluminum, titanium or steel		
Sensor serial number	*Sensor serial number*	*Sensor number*	*eSensorSN*
	Serial number of the WFT		
Sensor firmware	*Sensor firmware*	*Firmware*	*eSensorFirmware*
	Version of the WFT firmware		
Sensor info	*Sensor information*	*Sensor info*	*eSensorInfo*

Dialog: Measurement mode

Parameter - Measurement mode	Description		
	long name	*short name*	*column ID*
Mode	*Measurement mode*	*Mode*	*eChannelMode*

Basic setting denoting the kind of measurement

Mode	Description
mobile (rotating)	mobile deployment with rotating zero balancing
mobile (stationary)	mobile deployment with stationary zero balancing
stationary	stationary deployment; fixed angle of WFT

Coupling	*Coupling*		*eCoupling*

Depending on the channel selected, various usage options are listed.

Coupling	Description
TTI	Transducer Telemetric Interface - Default value which can not be changed except for the angle-channel.
Spirit level	Only possible for the angle-channel, in order to determine the "*angle error*" by means of the internal bubble ("spirit") level. After the preparation procedure, the LED flashes at a constant rate. In the balancing process, the angle error is determined 719. Next, set the coupling back to "TTI" and run zero adjustment for the other channels (using the angle error found).

Parameter - Measurement mode	Description *long name*	*short name*	*column ID*
Wiring	*Channel wiring*	*Wiring*	*eWiring*
	The input circuiting: "*Telemetry*". Not possible to change.		
Input range	*Input range*		*eRange*
	The measurement ranges are imported from the sensor.		

Input range	Description
Fx, Fz, Fy, Mx, Mz, My	Should not be changed
Aux.	Quantization: 32mV/V / 32768

Input range	Unit	Range	Default
Fx/Fz	kN		
Fy	kN		
Mx/Mz	Nm		
My	Nm		
Aux	mV/V	[0.5; 32]	32 mV/V

Dialog: WFT - Transformation

Parameter - WFT Transformation	Description *long name*	*short name*	*column ID*
Axial displacement	*Axial displacement*		*eAxialDisplacement*
	Used for transforming the coordinate system to "*TTI*"		

Unit	Range	Default
mm	[-999.00; 999.00]	0 mm

Radial displacement	*Radial displacement*		*eWheelRadius*
	Used for transforming the coordinate system to "*TTI*"		

Unit	Range	Default
mm	[0.00; 999.00]	0 mm

Dialog: WFT - Balance

Parameter - WFT Balance	Description *long name*	*short name*	*column ID*
Balance interval	*Balance interval*		*eBalanceInterval*

In revolutions (for rotating balance) or seconds (for stationary balancing)

Balancing	Unit	Range	Default
rotating balancing	rotations	[2; 1000] rotations	4 rotations
stationary balancing	s	[0.2; 100.0] s	4 s

Dialog: Miscellaneous

Parameter Miscellaneous	Description *long name*	*short name*	*column ID*
Angle error	Tab: Channel balance		

Quantization: 180° / 32768; balanced by spirit level or manual entry

Unit	Range	Default
°	[-120; 120°]	0°

Filter characteristic	Tab: Analog channels > Dialog: Filter		
	Filter characteristics	*Characteristic*	*eFilterCharacteristic*

Here, you select the filter characteristic for the channel selected:

Low-pass (8th order): Butterworth or Bessel

Filter cut-off frequency 1	Tab: Analog Channels > Dialog: Filter		
	Filter cut-off frequency 1	*Cut-off frequency 1*	*eFilterCutoff1*

Unit	Range	Default
Hz	Depending on the characteristics	20 Hz

Unit system (WFT)	Tab: Devices > Dialog: Measurement options		
	Unit system (WFT)	*Unit system*	*eUnitSystem*

Here you can choose the system of units: "*SI*" or "*US*". You must additionally change the parameter "*Temperature unit °F instead of °C*" to change the temperature unit.

Channels with associated parameters

Channels transmitted by the sensor (one each for the left and right wheels)

Channels	associated parameter
Status; Status-Flags 722 **Channel name** / **Unit** Status_*	---
Fx **Channel name** / **Unit** Fx_* / kN	Input range Fx/Fz
Mx **Channel name** / **Unit** Mx_* / Nm	Input range Mx/Mz
Fy **Channel name** / **Unit** Fy_* / kN	Input range Fy Offset **Info**: determined by "*TTI*" upon balancing **Action**: Offset-balance
My **Channel name** / **Unit** My_* / Nm	Input range My Offset **Info**: determined by "*TTI*" upon balancing **Action**: Offset-balance
Fz **Channel name** / **Unit** Fz_* / kN	Input range Fx/Fz
Mz **Channel name** / **Unit** Mz_* / Nm	Input range Mx/Mz
Revolutions **Channel name** / **Unit** Revs*	Input range **Range**: [-32768; 32737] **Info**: number of revolutions by the WFT

Channels	associated parameter
RPM	Input range

Channel name	Unit
RPM*	RPM

Range	Info
[-3061.224; 3061.131]	current rotation speed of the WFT; derived from internal calculations of the TTI

Temperature

Channel name	Unit
Temp_*	°C

Input range

Range	Info
[-128.0000; 127.9961]	temperature of the WFT

Auxiliary - extra input

Channel name	Unit
Aux_*	mV/V

Input range Aux

Offset

Info	Action
determined by "*TTI*" upon balancing	Offset-balance

Fx (rotating coordinate system)

Channel name	Unit
rot_fx_*	kN

Input range Fx/Fz

Offset

Info	Action
determined by "*TTI*" upon balancing	Offset-balance

Mx (rotating coordinate system)

Channel name	Unit
rot_mx_*	Nm

Input range Mx/Mz

Offset

Info	Action
determined by "*TTI*" upon balancing	Offset-balance

Fz (rotating coordinate system)

Channel name	Unit
rot_fz_*	kN

Input range Fx/Fz

Offset

Info	Action
determined by "*TTI*" upon balancing	Offset-balance

Mz (rotating coordinate system)

Channel name	Unit
rot_mz_*	Nm

Input range Mx/Mz

Offset

Info	Action
determined by "*TTI*" upon balancing	Offset-balance

Channels	associated parameter
Angle	Input range
Channel name / **Unit**: Angle_* / °	**Range** / **Default** / **Action**: [-180.0000°; 179.9945°] / 0 / spirit ("bubble") level
	Angle error
Angle, Sine	Input range
Channel name / **Unit**: Sin_*	**Range** / **Default**: [-1.000000; 0.999969] / 0
Angle, Cosine	Input range
Channel name / **Unit**: Cos_*	**Range** / **Default**: [-1.000000; 0.999969] / 0.999969

Status-Flags

Bit	State-Code	Description (bit active/passive)
2^0	1	WFT detected / not detected
2^1	2	not yet activated; always 1
2^2	4	Offset-balancing running / not running
2^3	8	Shunt active/passive
2^4	16	Remote active/passive
2^5	32	Bubble level mode active/passive
2^{15}	32768	Error / no error

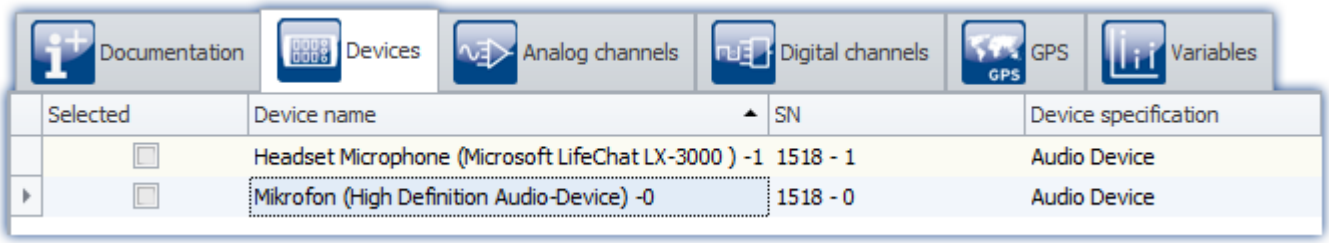
9.11.2 Third party devices

The following complete third-party devices are available:

- AudioDevice 723
- ChannelLoader 724
- fos4x 725
- FunctionSimulator 727

9.11.2.1 AudioDevice

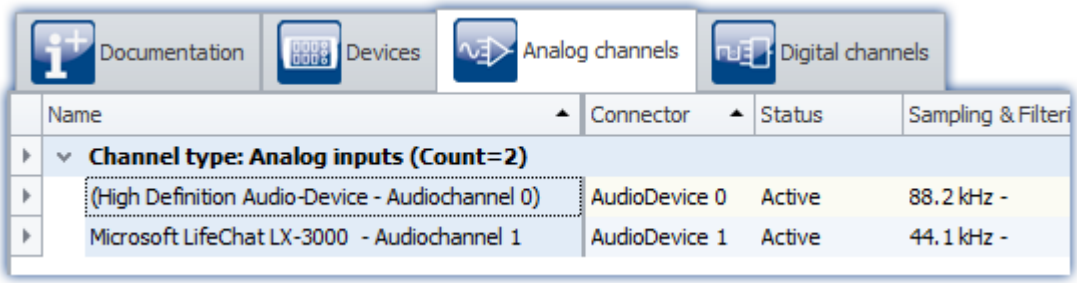
It is possible to add the audio devices of your computer to imc STUDIO. Therefore open the 3rd Party Device Management and add the AudioDevice. Afterward all audio devices are listed in the device list.



Device list: AudioDevice (Example)

On the **analog channel** page the audio channels are listed and also can be saved to the PC.

If multiple channels are available for a device, it is only possible to set one sample frequency for each channel of the device.



Channel list: AudioDevice (Example)

Note

- If the audio device is disconnected during the measurement, a error for each audio channel will appear in the logbook, showing that the data transfer is interrupted for this channel. To continue the measurement of this channel, start the audio device by the device control action in the Device table on the Setup page.
- If the experiment is opened on another computer, the corresponding hardware (USB device, sound card) must be available.
- If necessary, check your firewall for settings or rules: the process "imc.Studio.exe" (Program folder) needs the rights to access to the audio device.

Note Remarks to certain devices

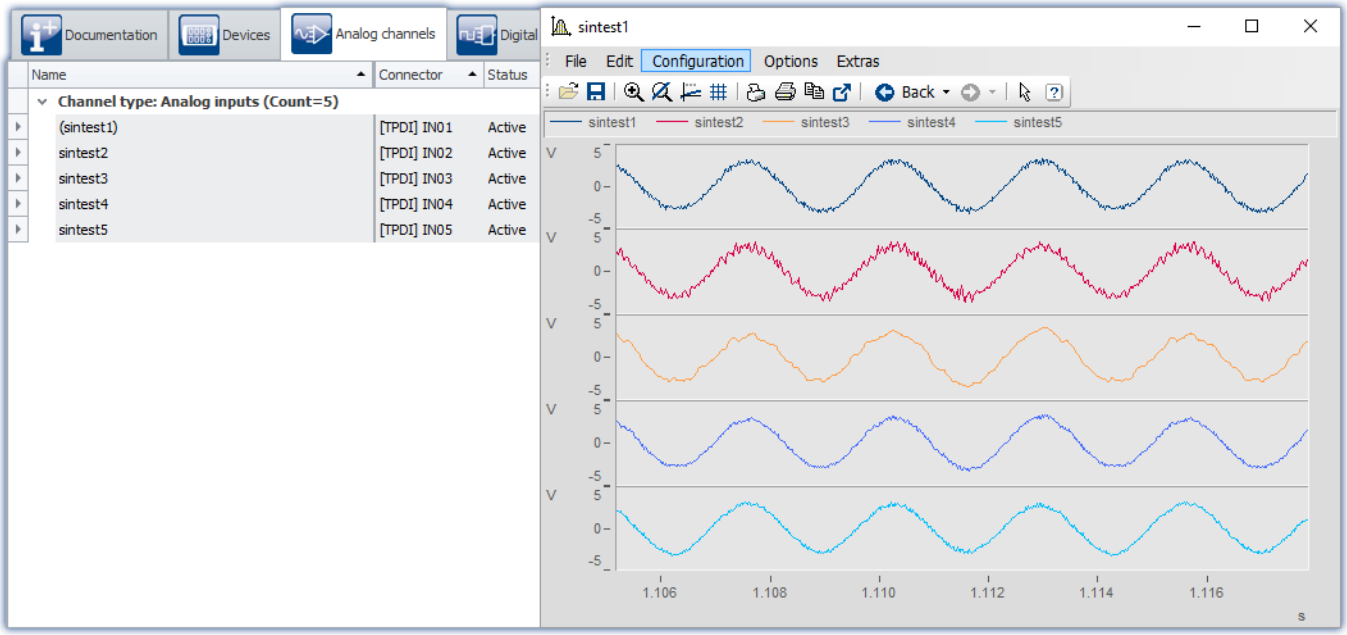
- With the webcam **Logitech HD Pro C920**, it was noticed that it was only delivering one audio channel, although according to the Windows settings there should be two channels.
  This may also apply to other audio devices.
- With the Headset **Microsoft LifeChat LX-3000**, it was noticed that after disconnecting and reconnecting the USB cable, a new measurement is only possible after the *second* start.

9.11.2.2 ChannelLoader

The ChannelLoader is for simulating or replaying data (e.g. recorded measurement data or imc FAMOS files. It can be added using the 3rd Party Device Management.

After selecting the ChannelLoader a file-open-dialog appears where one or more files can be selected. After the preparation the files are listed as channels on the analog channels pages.

After starting the device the signals are played periodically as input signal.



ChannelLoader: sintest1 - sintest5

Missing data

If the file to be played back is missing, an error message appears which provides the expected path and filename of the missing file.

The 3rd-party device uses experiment-specific files

When a file is selected which is located below the Experiment-folder, the location is saved in "relative" terms with respect to the experiment. If the location of the experiment changes, the file will still be found as long as it is also in the new experiment. The file from the old experiment-folder is not us.

Example: You create a dat-file in the "*Meta*"-folder. If you now export the experiment and import it to a different device or under a new name, the "*Meta*"-folder is also present there (together with the dat-file). By means of the relative path, the copied dat-file is found and used.

Note: Other folder names can also be used. See the advantage of the "*Meta* 113" folder.

Whether the path is saved in relative or absolute terms depends on the relationship to the Experiment-folder. If the file is below the Experiment-folder, the path is "relative". If the file is outside of the Experiment-folder, the path is "absolute".

9.11.2.3 fos4x

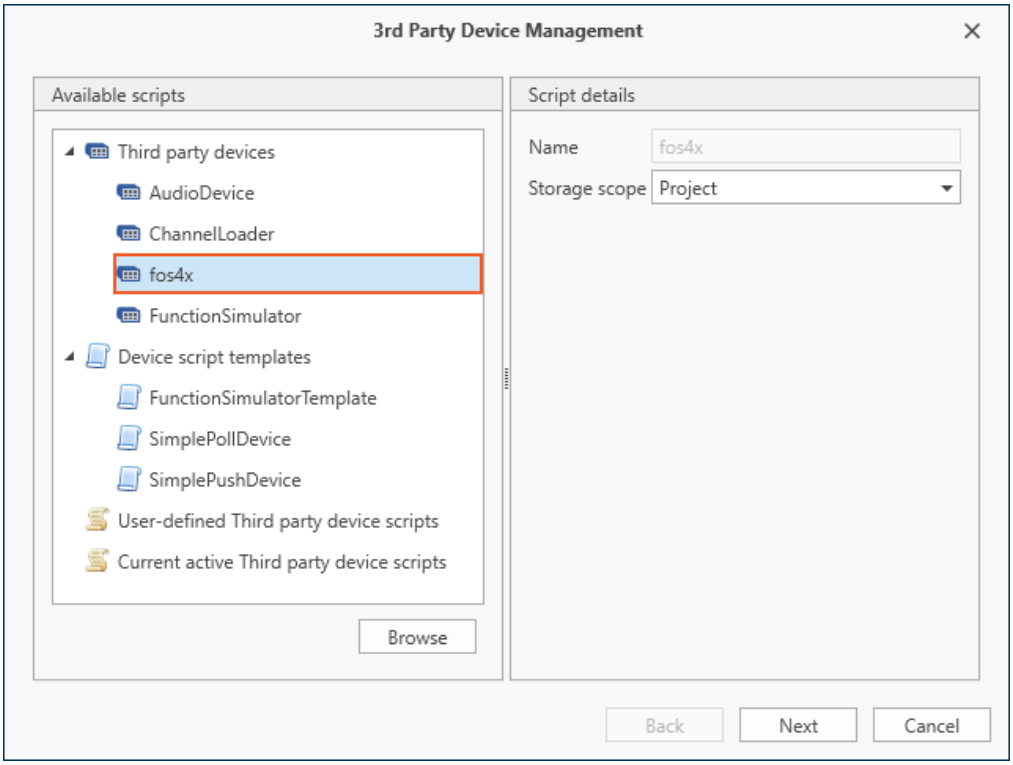
It is possible to use **fos4x** devices in imc STUDIO as a 3rd-Party-Device. These devices can be configured and controlled via the imc STUDIO.

Note License activation

To use a fos4x-Device the license imc STUDIO 3PDI-fos4x is required.

Adding a device - Integrating a fos4x-Device

Go to the Setup page. In the menu ribbon, click on *Setup Control > 3rd Party Device Management* and select *fos4x*.

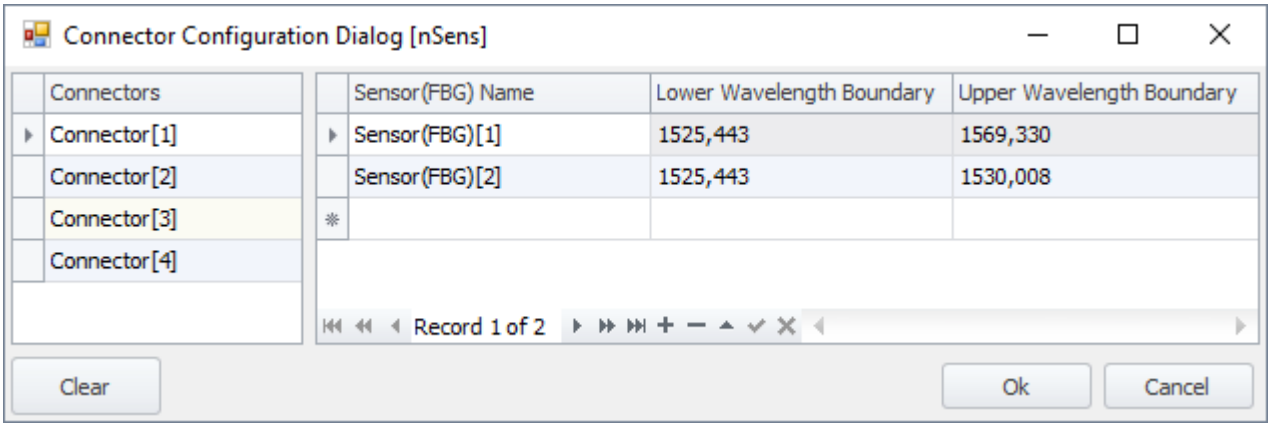


Third-Party-Device Management

9.11.2.3.1 Device settings

After the device has been added to the Devices list by means of the *3rd Party Device Management*, the device can be configured.

If a **fox4x nSens** is selected, the following dialog will be opened:



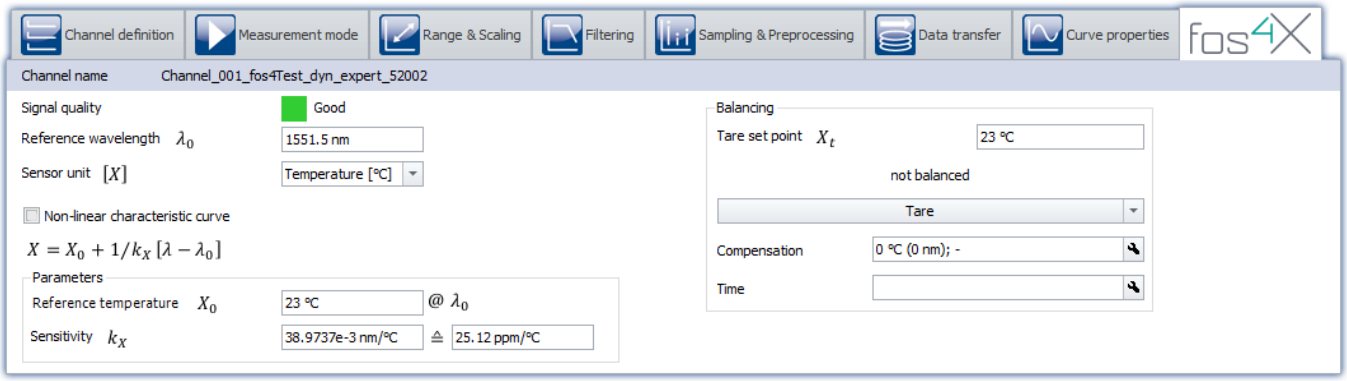
Configuration dialog (fos4x nSens)

Note Verification of the measurement range

If the expected measurement value is either at the measurement range boundary or even outside of the range possible, the resulting bandwidth will not be practicable because imc STUDIO automatically applies a headroom of approx. 5 nm. Therefore, overwrite the settings later, in order to ensure an adequately large bandwidth. If this isn't possible, change the location of the measurement value on the fos4X page.Seite.

9.11.2.3.2 Channel settings

On the "*Analog channels*" page there is a new dialog "***fos4x***".



Analog channels (fos4x)

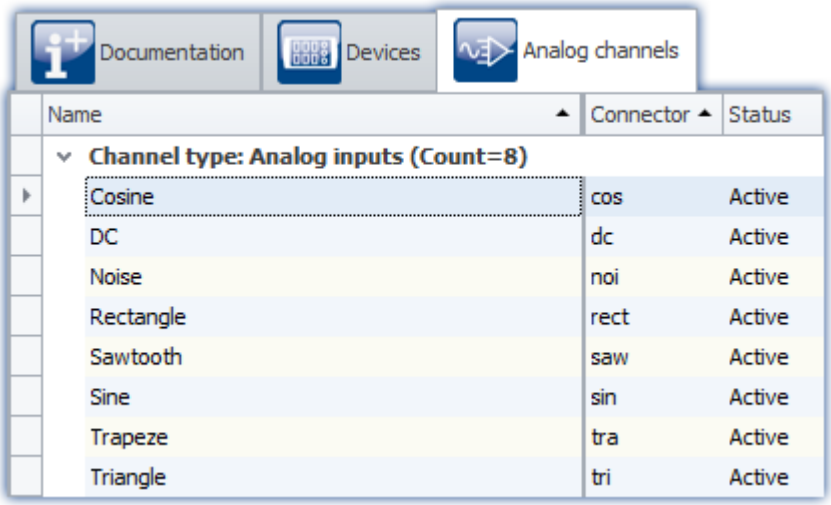
Reference

For further information, see **fos4x** data sheet or **fos4x** manual.

9.11.2.4 FunctionSimulator

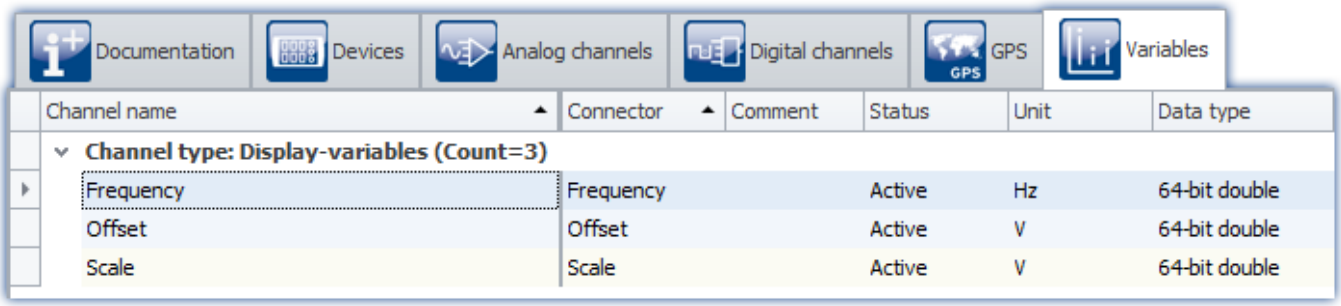
The FunctionSimulator is device which simulates various kinds of mathematics functions, including sine, cosine, triangle, sawtooth and squarewave.

The analog channels then appear on the Setup-page under "*Analog channels*".



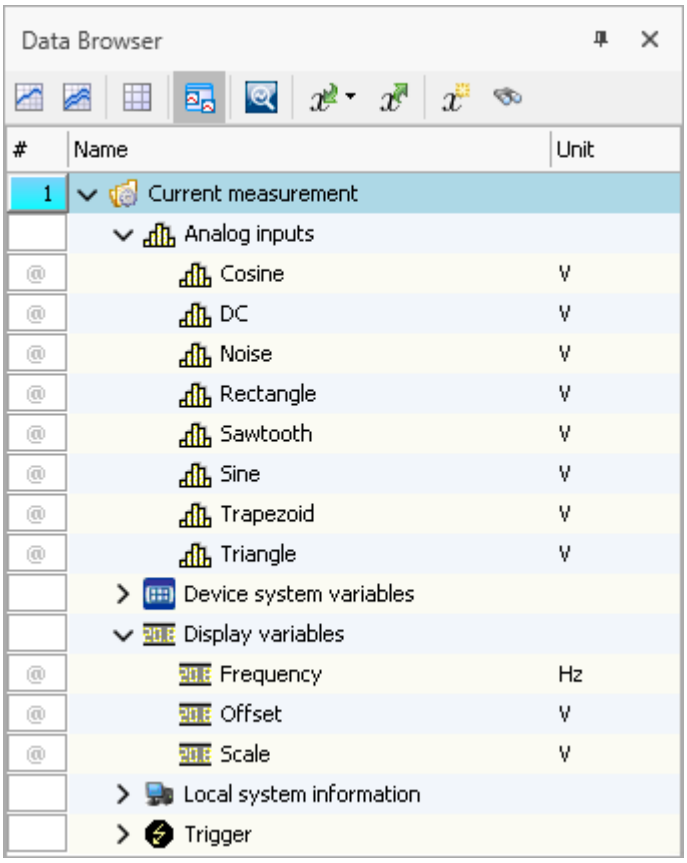
Setup: FunctionSimulator (Analog Channels)

It is also possible to adjust the frequency, the scaling and the offset. The Display variables appear on the Setup page under "*Variables*".



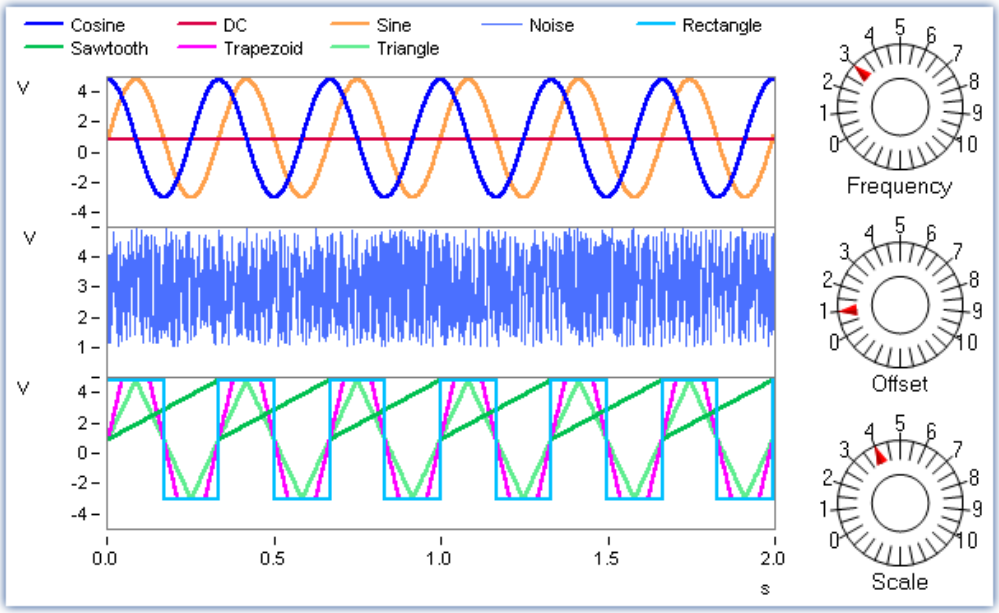
Setup: FunctionSimulator (Variables)

After selection and configuration of the device, the channels and Display variable are listed in the Data Browser.



Data Browser: FunctionSimulator

The signals can be displayed in the curve window.



Curve window: FunctionSimulator

By changing the Display variables, it is possible to alter the frequency, amplitude and offset during runtime.

Note

- The FunctionSimulator can be used as trigger target action of an imc device.
- Important: The time zone in the imc device must be corresponding to the PC's time zone.

9.12 Sensors, characteristic curves and TEDS - for imc DEVICES devices

Supported imc measurement device groups

A device belonging to the firmware group A 187 (imc DEVICES) - e.g. CRONOS Familie, C-SERIE, SPARTAN, BUSDAQ.

Definition of terms

Term	Description
Sensor	There are a variety of sensor types. They differ not only in shape, size, in the physical principles on which they are based, as well as in terms of their conversion between variables, but also in terms of properties, spec sheets and in how they are connected with measurement devices.
imc SENSORS	imc STUDIO supports export of sensor spec-sheet data from a sensor database and the application of the data in configuring channels. For this purpose, the program imc SENSORS must be installed. imc SENSORS is a tool for managing and editing information on sensors. This applies in particular to the specs in the spec sheets and the calibration values. For a detailed description of the database itself, see the user's manual for **imc SENSORS**.
User-defined characteristic curve or imc STUDIO characteristic curve	A sensor which is set up in imc STUDIO. This one is a sensor characteristic curve having multiple points. Its functional scope is limited in comparison with the sensors from imc SENSORS. Not all amplifiers support this type of characteristic curve. See "*List of amplifiers which support sensors having characteristic curves* 740"
TEDS	imc STUDIO supports readout of sensor spec-sheet data from a sensor TEDS and the application of this information in configuring channels. On TEDS you can save sensor information 733 created using imc SENSORS.
Connected sensors	The list of sensors in the experiment. This list provides a quick overview. Additionally, you can also use the sensors for other channels.
Filter 735	A filter is a grouping of multiple sensors. E.g. Thermocouple, Strain gauge, Piezo-electric, ... You can create additional filters in imc SENSORS and import them into the tool window "*Sensors*".

Tool Window "*Sensors*"

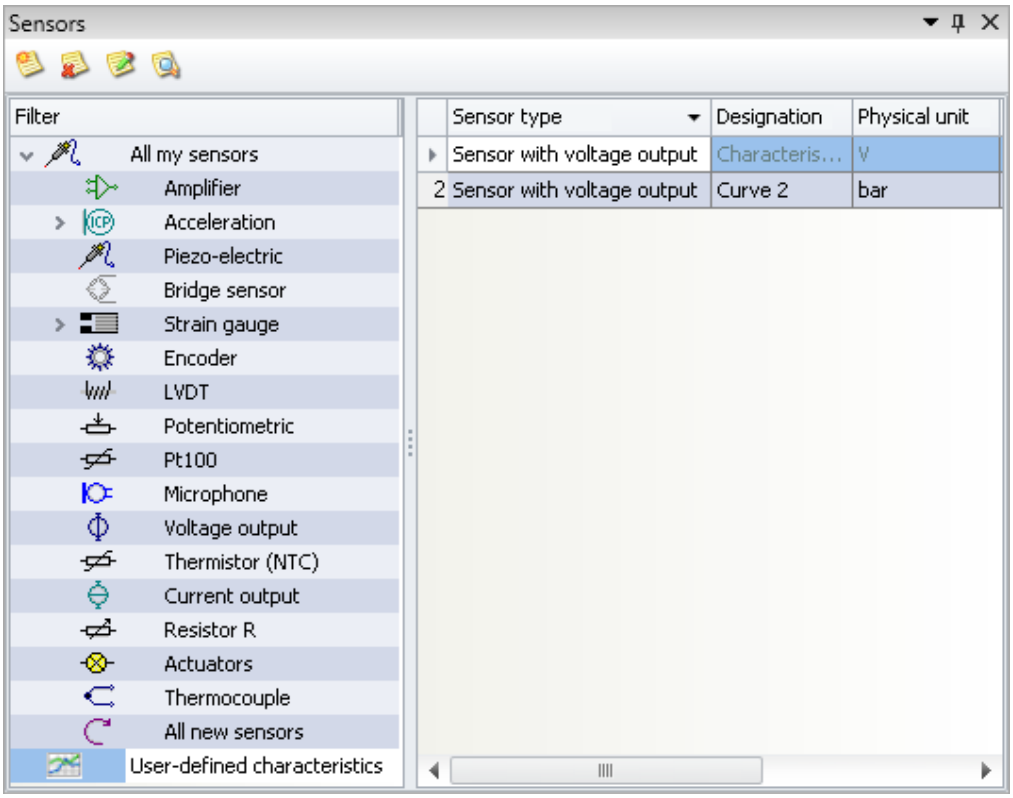
The tool window "*Sensors*" presents all information on existing sensors and characteristic curves.

Here is where you find information on the sensors used. Additionally, you can create and edit "*user-defined characteristic curve*". If imc SENSORS is installed, the sensors from this database are shown and are available for use.

Note "*Sensors*" - The tool window is not available

By default, this tool window is not displayed. To create it, use the dialog: "*Tool Window Chooser* 141":

Ribbon	View
View > Choose Tool Windows ( )	Complete
Extra > Choose Tool Windows ( )	Standard



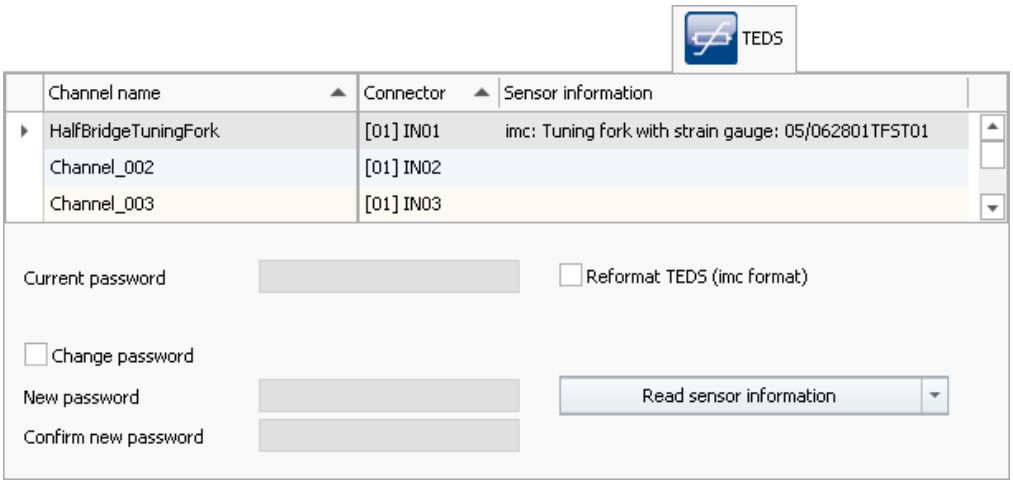
"Sensors" tool window in the plug-in "imc STUDIO Setup " (example)

The pages "*TEDS*" and "*Analog channels*" > "*Channel definition*"

On the Setup pages "TEDS" and "*Analog channels*" > "*Channel definition* 344", you can use sensor information from TEDS to configure channels.

Note "*TEDS*" - The page is not available

By default, this page is not displayed. To display it, please follow the instructions in the chapter: "*Additional Pages* 407".



This page is subdivided into two areas: the "*Channel table*" and a single "*Dialog*". (For information on the operation and configuration of these: "*Tabular Display* 231" and "*Dialogs* 238")

Setup page "TEDS"
The page is not required for reading and resetting the sensor information.

9.12.1 What are TEDS

The sensor TEDS are serial ROMs. They are connected to an amplifier channel via a digital signal line (*1-wire-PROM*). Each sensor TEDS has a unique ROM-ID (*Sensor-ID*, *silicon serial number*). A sensor TEDS can also contain additional information on a sensor. Electrically erasable and re-writable sensor TEDS are referred to as *Sensor-EEPROMs*.

Note

Some amplifiers support *TC-TEDS* (see the respective device manual for more information).

- Therefore the amplifier has to be equipped with a TC-TEDS-adapter. Instead of the standard DSUB-plugs, SMMI-thermocouples for type-K can be connected.
- By using suitable SMMI-thermocouple plugs (TC-TEDS) sensor information for thermocouples can be imported.

9.12.2 Writing sensor information to the channel

Sensor information can be **imported from TEDS**, or you can use the sensor information **from the "*Sensors*" tool window**.

Prerequisites:

- The amplifier and imc STUDIO must both support the sensor. The sensor information is only applied if the channel also supports the properties entered.
- A list of modules supported is presented in the chapter: "*List of supported sensors (TEDS/imc SENSORS)* 740 ".
- A list of amplifiers/devices which support sensors having characteristic curves is presented in the chapter "*List of amplifiers which support sensors having characteristic curves* 740".

Note Saving imported sensor information

The sensor information once imported and linked to a channel are **saved with the experiment**. If the experiment is later transferred (e.g. copied to another PC), this sensor information isn't lost!

TEDS - Read sensor information

imc STUDIO must be connected with the device.

Open the Setup page: "*TEDS*" or "*Analog channels*" > "*Channel definition* 344".

Read sensor information

- Select the channel desired, to which the TEDS is connected (multi-selection is possible if a TEDS is connected to each channel).
- From the list, select: "*Read sensor information*"
- Click on the button

The sensor information is read out and used to configure the channel. All changes resulting from import and application of sensor info are indicated in the "*Sensors*" > "*Connected sensors* 734" tool window.

Note Sampling rate

If a **sampling rate** is defined in the TEDS, **it is ignored**. Since more than two different sampling rates per device are not possible, this would in most cases lead to an interruption of the reading process.

Using sensor information from the "Sensors" tool window

In order to write a sensor's information and configuration to a channel, use Drag&Drop to move the sensor to the channel desired. The configurations stored in the sensor are applied in the channel configuration.

9.12.3 Reset channel's sensor information

- Open the Setup page: "*TEDS*" or "*Analog channels*" > "*Channel definition* 344".
- Select the channel desired (multi-selection is possible)
- From the list, select: "*Reset channel's sensor information*"
- Click on the button

The sensor information is deleted.

Note The configuration remains intact

The channel configuration is not reset to its condition before importing. The current configuration remains intact as long as the sensor did not set any parameters which were not possible without sensor information. In that case, the default value is restored.

9.12.4 Writing sensor information to the TEDS

In order to write to and configure TEDS, **use the software imc SENSORS**.

In imc STUDIO, only very **limited** capabilities for this purpose are provided. In imc STUDIO, the only sensor information you can write to the connected TEDS is the info which was linked with the channel from the imc SENSORS database.

This means that if the channel is connected with a sensor from imc SENSORS, then only that exact sensor can be recorded in the TEDS. Info on the sensor is provided in the "*Sensors*" tool window and in "*Connected sensors* 734"

Note

No changes to the channel properties made in imc STUDIO are written to a sensor TEDS. This also goes for sensor information which can be edited by imc STUDIO. The only data which can be written to the sensor, then, are whatever is displayed in the *Sensors* tool window, not anything that is subsequently changed using imc STUDIO!

Example: The input range was set to ±5 V using the sensor database or Sensor-TEDS. Later, the input range was changed to ±2 V. This setting is not adopted in the Sensors tool window. The previously imported value ±5 V remains intact in the sensor TEDS.

- Open the Setup page: "*TEDS*".
- Select the desired channel, whose channel settings are to be written to the TEDS (multi-selection is possible if a TEDS is connected to each channel)
- Select from the list: "*Write sensor information*"
- Click on the button

If the TEDS is password protected, the password must be entered.

Here, it is also possible to edit the password.

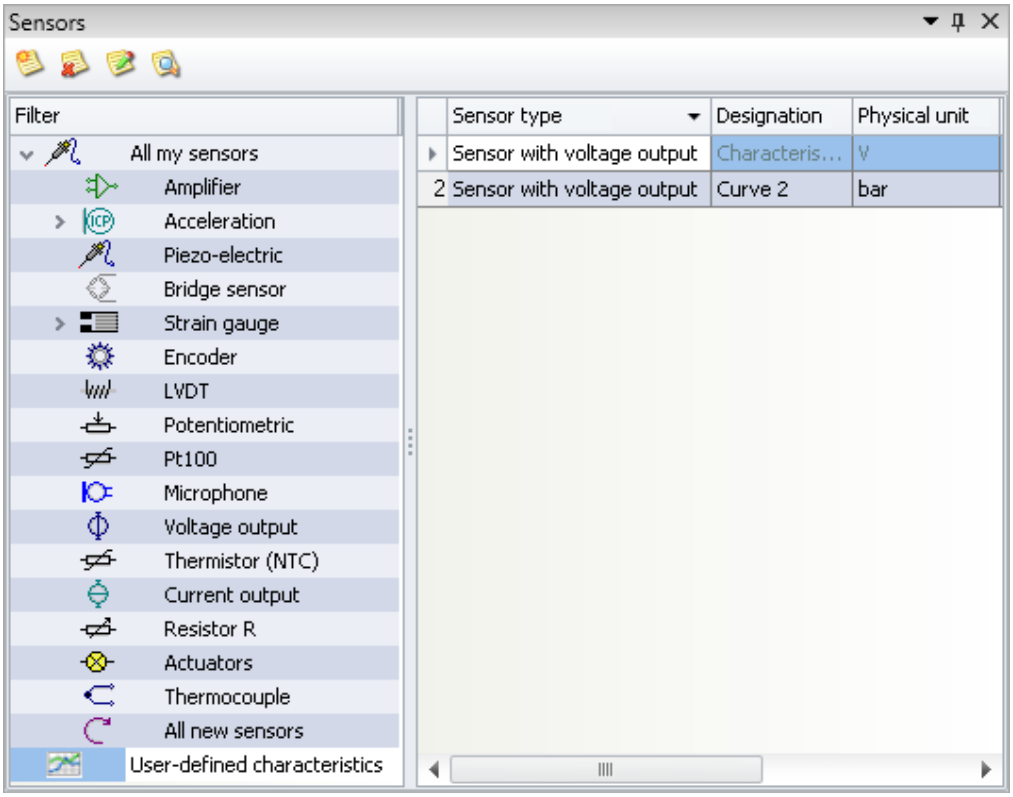
Note Password protection

- In 1-Wire technology, sensor-EEPROMs do not have any password protection of the hardware or write-protection.
- Using the imc proprietary format (*TEDS format (imc-Format)*), all imc applications observe a software password protection: This means that only upon entry of the correct password, writing information to the TEDS can proceed. The password is a 32-bit integer.

9.12.5 Structure and functions of the "Sensors" tool window

The window is split in two parts. At the left, there is a structured list of all sensor groups. At the right you see the sensors belonging to the selected group and their respective properties.

Groups	Description
All my sensors	The list of sensors from imc SENSORS.
User-defined characteristics	The list of user-defined characteristic curves. These characteristic curves can be created and edited here.
Connected sensors	The list of sensors in the experiment. This list provides a quick overview. Additionally, you can also use the sensors for other channels. **Adding associated imc STUDIO characteristic curves to the group heading "User-defined characteristic curves".** User-defined characteristic curves (imc STUDIO characteristic curves) displayed here can be copied to the group heading "User-defined characteristic curves" by means of Drag&Drop. Thus, these characteristic curves are known to all experiments and available for use by them.



"Sensors" tool window in the plug-in "imc STUDIO Setup " (example)

Toolbar and context menu

Menu item	Description
Create a new characteristic curve 736	Opens the dialog for creating/editing user-defined characteristic curves.
Delete characteristic curve	Deletes the user-defined characteristic curve selected.
Edit characteristic curve	The selected user-defined characteristic curve can be modified. The dialog for creating/editing user-defined characteristic curves is opened.
Open imc SENSORS	The program imc SENSORS is opened in a separate program window.
Find	Here you can search through the list of sensors either according to the "Filter"-name or the "Sensor"-parameters.
Add to imc SENSORS 741	Adds an existing sensor to the imc SENSORS database.
Layout	
Standard layout	Restores the original layout. E.g., imported filters are deleted and moved or deleted filters are set to back as they were earlier.
Import Filter 735	Here, a filter created in imc SENSORS can be imported in order to enable a user-defined grouping.
Delete Filter	Deletes the filter selected. Note: Only the filter is deleted, not the sensors indicated. Normally, only the selected filter is deleted. However, if the selected filter contains subordinate filters, then the user is prompted to specify whether to also delete these subordinate filters. Subordinate filters are all filters, located below the selected filter and are indented at least 1 position further right than the selected filter.
<< (level higher)	In the tree diagram, the filter is indented one position to the left. This raises it up one level in the hierarchy.
>> (indentation)	The filter is indented one position to the right. In this way it can be made to be a subordinate filter to the one above it, for example.
Filter list hidden	The filter list is hidden and appears as a bar in the lower portion of the tool window. There, the currently selected filter name is displayed. By clicking the mouse over this region, the list is opened in a separate window and the filter can be switched. Clicking the mouse again over the tool window hides the list again.

User-defined grouping - importing filters

A filter is a grouping of multiple sensors. E.g. Thermocouple, Strain gauge, Piezo-electric, ...

You can create additional filters in imc SENSORS. These are not automatically displayed in imc STUDIO. Import the filter via the context menu in the Filters list "*Layout*" > "*Import Filter*".

Reference	Filter configuration
An exact description of the filter configuration is presented in the separate manual on imc SENSORS.	

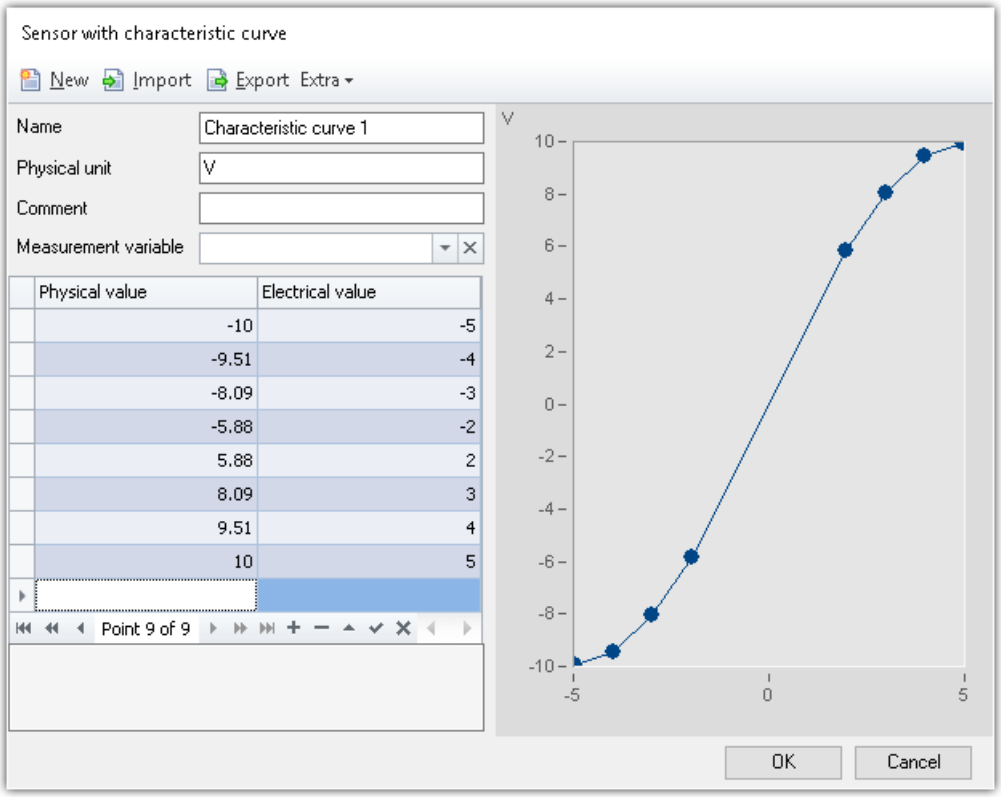
9.12.6 Creating a new sensor characteristic curve

Sensor characteristic curve provide the ability to convert measured electrical readings to physical values by reference to a plot of the relationship between those two variables. The characteristic curved can be generated for two different purposes:

Purpose	Description
Direct conversion of data at the input channel	The characteristic curve is created **in the Sensors-tool window** and can be dragged to the channel by means of Drag&Drop 732. In this way, it functions similarly to an imc SENSORS-characteristic curve. Not all amplifiers support this type of characteristic curve. See "*List of amplifiers which support sensors having characteristic curves* 740"
Subsequent conversion, e.g. via imc Online FAMOS	Using the **command "*Measure a user-defined characteristic curve*"**, a characteristic curve is saved in a dat-file. This file can subsequently be imported as a supplemental file for imc Online FAMOS, for example.

In both cases, the same dialog is used for entering the characteristic curve's points. There, the values can be entered in different ways:

Entry technique	Description
Manual entry	The characteristic curve's values are entered point-by-point.
Point measurement	The electrical values are imported directly from the sensor. Please note that the signal's stability is neither monitored or controlled, so for this reason it only makes sense to apply this functionality to very stable constant signals.
Import	This lets you import previously generated sensors. These must be saved in the format *.dat* and can be generated using the imc software imc FAMOS, for example.



Example of a user-defined characteristic curve

Parameters of a characteristic curve

Parameter	Description
Name	Name of the characteristic curve to be created
Physical unit	Unit of the corrected value
Comment	
Measurement variable	Optional: Entry of values by means of the measured variable The electrical values can be imported from the sensor directly. Select the sensor's relevant pv-variable. The electrical values can now no longer be entered manually, but are instead measured directly. The measured values are entered.
Adjusting variable	- no function
Delay [s]	Optional: The current value is imported once the time entered has elapsed since the button is pressed.

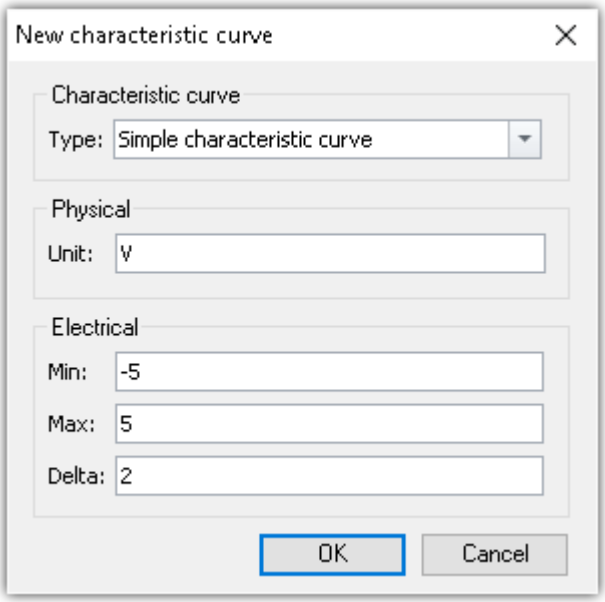
Configuration via the Sensors-tool window

To create a new sensor, open the context menu and select "*Characteristic curve*" > "*New*" or click on the corresponding button ( ). The dialog for creating/editing a user-defined characteristic curve opens.

Menu item	Description
New	Opens an Assistant capable of setting up a simple sensor. You can also enter the points in the table.
Import	This lets you import previously generated sensors. These must be saved in the format *.dat* and can be generated using the imc software imc FAMOS, for example.
Export	You can export sensors once generated, in order to save them separately, or for use in a different operation.
Extra	Here you can access the options for the characteristic curve dialog. For instance, measurement and adjusting variables, which enable additional functions for creating characteristic curves, can be activated.

Quickly generating multiple points for subsequent editing

To quickly generate multiple points at fixed intervals, click on "*New*" ( ). A window appears, which provides a template for generating a characteristic curve.



In this example, a straight line is generated between -5 and +5 with a point-to-point interval of 2. In this way, you are able to generate many points quickly, which can later be easily modified.

Configuration by means of the command "*Measure a user-defined characteristic curve*"

In this command, the settings for the characteristic curve dialog are pre-defined. Parameters can be pre-defined and hidden, or editable. Additionally, a target file needs to be defined:

Parameter	Description
Filename	Storage location of the characteristic curve file in dat-format

Running the command

When the command is run, the dialog for capturing the characteristic curve's individual points appears. This is preconfigured according to the settings saved for the command. Once the data points have been captured and the dialog closed, the file is saved on the hard drive. Then it can be imported with a subsequent command for imc Online FAMOS, for example.

Entering/measuring points for the characteristic curve

A characteristic curve always consists of at least three points. In the table, the values for the characteristic curve are entered/measured. The characteristic curve is displayed accordingly in the curve window.

Values for the characteristic curve:

- Electrical value: the reading measured by the sensor
- Physical value: the value to be indicated in response to the corresponding electrical reading.

Manual entry

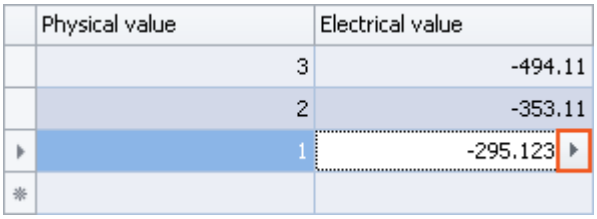
The values of the characteristic curve are entered point-by-point.

Measuring the values by means of the measured variable

The electrical values can be read from the sensor directly. Please note that the signal's stability is neither monitored or controlled, so for this reason it only makes sense to apply this functionality to very stable constant signals.

Select the sensor's associated pv-variable in the "*Measurement variable*" field. Now the electrical values can no longer be entered manually, but are measured directly. The measured value is automatically entered.

To read out a value, select the associated cell and click on the button at the cell's right edge:



Acquiring a new electrical value via the measurement variable

When a "*Delay*" is defined, the value is only imported once the specified time has elapsed since the button was pushed.

9.12.7 List of amplifiers which support sensors having characteristic curves

Support of sensor characteristic curves is enabled for following devices:

Amplifier	Device				
	CRPL/SL	CRC	CRFX	CRXT	CANSAS
ICPU-8	●	---	---	---	---
DCB-8	●	---	---	---	---
LV2-8	●	---	---	---	---
UNI-8	●	---	---	---	●
ISO2-8	●	●	●	● as of 2.13 R1	---
ISOF-8	---	---	●	● as of 2.13 R1	---
UNI-4	●	●	●	● as of 2.13 R1	---
SC2-32	●	●	---	---	---
ICPU2-8	Ø	●	●	● as of 2.13 R1	---
UNI2-8	Ø	●	●	● as of 2.13 R1	---
DCB2-8	Ø	●	●	● as of 2.13 R1	---
B-8	Ø	●	●	● as of 2.13 R1	---
LV3-8	Ø	●	●	● as of 2.13 R1	---

Device	Firmware
Cx-41xx-N	●
Cx-41xx-FD	●
SPAR-N	●

●: Feature supported imc STUDIO 5.2 R10 included firmware 2.13 R1

Ø: Feature currently not supported

---: Amplifier not available for this device series

9.12.8 List of sensor supported (TEDS/imc SENSORS)

Implemented sensor information	
Accelerometer	Based on the ICP principle: current-fed; voltage measured with AC coupling
Bridge	Common bridge, bridge sensor, e.g. for force, no strain gauge
LVDT	LVDT sensors, differential coils. Supplied with AC voltage.
Microphone	Microphone with built-in amplifier
PT100	Pt100 and related, e.g. Pt1000
StrainGauge	Strain-gauge bridge
Thermocouple	Thermocouple
Voltage	General voltage sensor
Current	Sensors with amperage output, e.g. 0..20mA or 4..20mA
Amplifier	Conditioner, amplifier with voltage output
Potentiometric	Potentiometric sensor, operated in half-bridge configuration, for example

Channel types that do not support TEDS or measurement modes that are not transferable from imc SENSORS to imc STUDIO	
Encoder	Incremental encoder sensors
DigitalIn	Digital input of the measurement device, e.g. for querying a switch
ActorDigital	Digital output of the measuring device, e.g. relay for switching / control
Resistance	Resistor, usually current-fed and whose voltage drop is measured to derive the resistance, which is often proportional to another physical quantity.
Thermistor	Thermistor, resistance thermometer with non-linear characteristic curve
PiezoElectric	Piezo-electric (acceleration/force/sound), requiring a charge amplifier
ActorVoltage	Analog voltage output
ActorCurrent	Analog current output
ActorPulse	Pulse output, e.g. PWM (pulse width modulator)

9.12.9 Sensor database - imc SENSORS

imc STUDIO supports export of sensor spec-sheet data from a sensor database and the application of the data in configuring channels.

For this purpose, the program imc SENSORS must be installed. The sensor database imc SENSORS administers such information for sensors. In this program, it's possible to set up, edit and administer entries for sensors.

Open imc SENSORS

If imc SENSORS is installed, the sensor database can be started via the "*Sensors*" tool window. To do this, open the context menu and select "*Open imc SENSORS*" or click on the corresponding button ( ).

Importing sensor data from the sensor database

In order to transfer sensor information to a channel, it is not necessary for imc SENSORS to be opened. The "*Sensors* 732" tool window lists all sensors in the database.

Adopting sensor information in the sensor database

To transfer sensor information from a sensor TEDS to the sensor database, open the context menu of the "*Sensors*" tool window under "*Connected sensors* 734" and select "*Add to imc SENSORS*". The channels settings remain unaffected.

9.13 imc SIMPLEX - Sensors for imc DEVICEcore devices

9.13.1 Overview - Setting up

imc SIMPLEX enables extremely fast configuration of imc ARGUS*fit*-channels by means of validated sensor settings prepared in a cloud. This massively reduces the risk of errors in configurating a measurement system.

Application examples

- When you scan a sensor with your smart phone, the sensor is displayed in imc STUDIO. By means of Drag&Drop it can be dragged to the target channel. The channel adopts all of the sensor's information and is ready for measurement.
- When you have a pile of cable terminals in front of you, how do you know which one is the right one? Simple: all cable terminals are equipped with RFID. The app lets you scan the cable terminals and obtain the info indicating what terminal you are holding in your hand.
- You have a cabinet full of sensors in their packaging. The packaging is equipped with RFIDs. Using the app, you can scan a package to find out what sensor it contains and whether it is right for your measurement, and even whether it is already appropriately calibrated.

Product overview

imc SIMPLEX consists of three components: the imc STUDIO plug-in, the mobile app and the website.

Using the website interface it is possible to set up and configure sensors. This determines which imc measurement modules can measure with the sensors. The finished sensor configuration is made available in a database. If the sensor is equipped with RFID, a barcode or QR-code, it can be scanned with a smart phone. imc STUDIO then provides the appropriate sensor configuration to apply to the channel. It is also possible to enter specific measuring point definitions including harmonized signal names, or generic sensor types.

Supported imc measurement device groups

A device belonging to the firmware group B - e.g. imc ARGUS*fit*, imc EOS.

System requirements

- Browser: an up-to-date browser
- Smart phone operating system: Android 5 or higher, or IOS 10.3 or higher (find details in your respective app store)
- Smart phone: NFC-enabled or barcode/QR code enabled, depending on application
- imc STUDIO 2023 R3 or higher
- Connection to the internet with access to the imc SIMPLEX server

Licensing

Licensing is independent of imc STUDIO and the imc LICENSE Manager.

- **Separate licenses** are required for imc SIMPLEX, which cover the **use of the cloud services** and the imc SIMPLEX Assistant.
- The license model is a **subscription** that is activated or renewed by **paying an annual fee**.
- Licensing involves creating user accounts on the imc SIMPLEX server. Upon order, the number of allowable user accounts is specified. Later extensions are supported.

- A user account includes the authorization to log in to the imc SIMPLEX server via the mobile app, the website and via imc STUDIO. In addition, depending on the configured user role, it allows the configuration and/or readout of sensor information. A user account is not bound to specific devices.
- The installation of the mobile app and the imc STUDIO plug-in are license-free. However, their meaningful use requires log-in with a licensed user account.

9.13.1.1 Administration

Along with access to an imc SIMPLEX-database, you obtain a user account having administrator rights.

Initial setup of the database and of the administrator

If the database had already been used previously, the following steps are not required.

To begin with, setup of the database is performed. For this purpose, a license card ships with the system.

License card

The license card contains a QR-code for registration. Open the link behind the QR-code and enter the info about the database and the Admin-user.

The database is created using the info entered and can now be used.

The following actions are performed in the imc SIMPLEX Assistant. To do this, open the imc SIMPLEX homepage 745 in the browser: https://simplex-assistant.imc-tm.com

Login as the Admin-user.

In the role of this user, you can set up additional users.

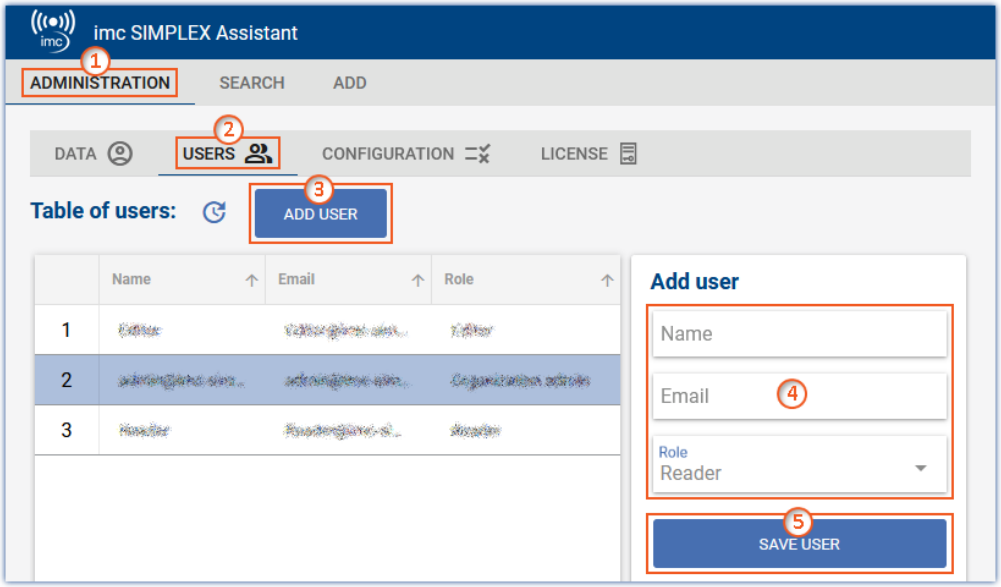
User roles

In order to use imc SIMPLEX, each user requires a user account. For this purpose, various user roles having various privileges are available:

Function	Reader	Editor	Admin
Sensor reading	•	•	•
Sensor assignment to channel	•	•	•
RFID scanning	•	•	•
Sensor setup		•	•
Sensor editing		•	•
Sensor deletion		•	•
User setup			•
User role editing			•

Creating a user

Open "*Administration*" > "*Users*" > "*Add user*"

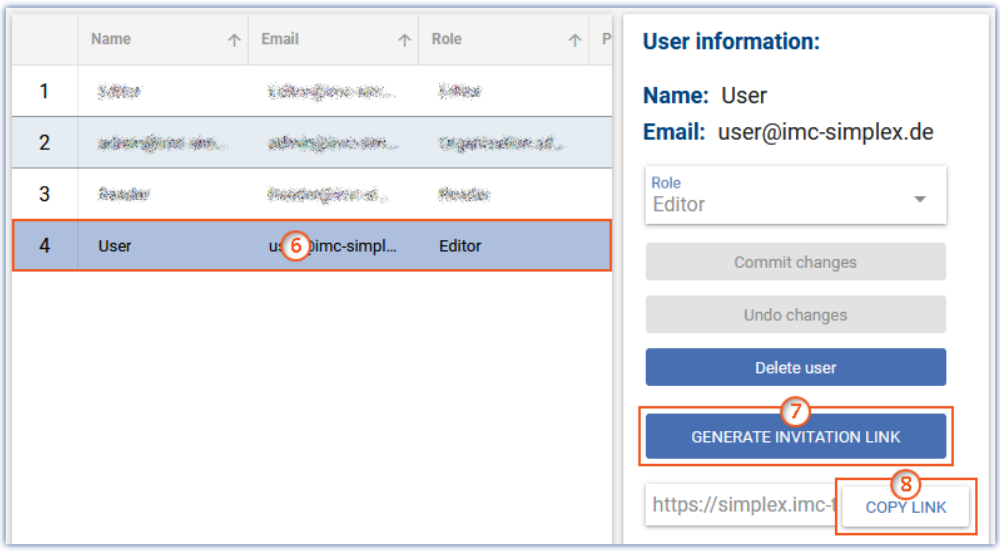


Enter the new user's name and e-mail address and specify their role ④. You can also change the role subsequently.

By means of the e-mail address, the user can later log in to imc SIMPLEX.

Click on "*Save user*" ⑤ to create the new user.

The new user then appears in the list. Select that user ⑥.



Click on the button "*Generate invitation link*" ⑦.

Next, click on the button "*Copy link*" ⑧. This copies the invitation link to the Clipboard.

By means of the link, the user can assign a password if desired. The link is valid for 4 weeks, after which a new link must be generated.

Send this link to the user.

License

Open "*Administration*" > "*License*".

Here, important information on your database and on the status of your license is displayed.

Term	Description
Organization name	Name of the database
License ID	Internal number of the license
Status	Active: The database can be used at this time. The status is set to passive once the expiration date has been reached. Please arrange to renew your license in a timely manner.
Licensed users	The maximum user count. This number depends on how many you order. You can increase the number ordered at any time.
Expiration date	Date on which the database is deactivated. Please arrange to renew your license in a timely manner.

9.13.1.2 Opening the Web- and App-Assistant

The imc SIMPLEX-Assistant can be opened via a web browser. Or, you can use the app on your smart phone.

Browser

Use the following link:

https://simplex-assistant.imc-tm.com

Or open the tool window imc SIMPLEX 746 in imc STUDIO. From there, you can go to the Options by clicking the button . In the Options, you will find the button "*Create and edit sensors*". Also use this to access the website.

Log on with your e-mail address and password.

App

Search in the respective app-store for imc SIMPLEX. Then install the app.

imc SIMPLEX
imc Test & Measurement

Android
Google Play Store

iOS
Apple App Store

Log on using your e-mail-address and password.

9.13.1.3 Activation under imc STUDIO

Activating the component

Start by activating imc SIMPLEX by means of the Product configurator 29, in order to make it appear in the software.

Logging on the user

Open the imc STUDIO Options.

Ribbon	View
Extra > Options ( )	all

Navigate in the Options to "*Setup*" > "*SIMPLEX options*".

Log on using your e-mail address and password.

Close the Options and apply the changes.

Showing the tool window

In the main Setup window, the tool window "*imc SIMPLEX*" will now appear in the lower region next to the logbook.

If you use an existing database from an older version of imc STUDIO, it is possible that the tool window will not appear automatically. In order to make it appear, use the dialog: "*Choose tool windows* 141":

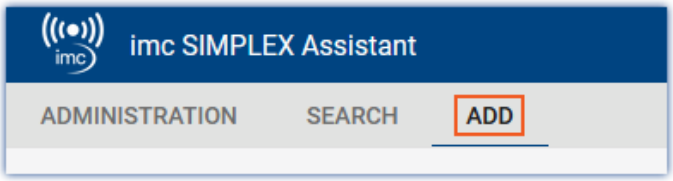
Ribbon	View
View > Choose Tool Windows ( )	Complete
Extra > Choose Tool Windows ( )	Standard

9.13.2 Setting up and configuring a sensor

The following actions are performed in the imc SIMPLEX Assistant. To do this, open the imc SIMPLEX homepage 745 in the browser: https://simplex-assistant.imc-tm.com

Setting up a sensor

On the web interface, open the tab: "*Add*".



Use the buttons having their various symbols to page through the configuration, beginning with the Measurement Type. The next symbol-buttons only become enabled once the area currently open has been configured completely.

Each measurement type has different configuration possibilities. Thus under "*Sensor*" for example, different settings are available for a temperature measurement than for a pressure measurement.

Compatible measurement modules & systems

In the notification box "*Compatible Modules & Systems*", a list of all devices and modules which support the current configuration is presented.

Configuring the sensor

Symbol	Description
Measurement	The sensor can be completely configured using the symbols "*Measurement*", "*Sensor*", "*Calibration*" and "*Digitization*". These settings directly affect the channel configuration and thus the measurement results. Be aware that the settings must be made for each of these areas proceeding from left to right. If any changes are made, all settings in areas located further to the right are discarded. Example: If you make any changes under "*Sensor*", all settings under "*Calibration*" and "*Digitization*" are discarded.
Sensor	
Calibration	
Digitization	

"General" and "Identification"

Symbol	Description
General - Test specimen	• **Signal name**: takes effect on "*Channel Name*" • **Comment 1, 2, 3**: takes effect on "*Channel Comment*". If this is filled out, the texts appearing in the channel comments are written in succession, divided by the character ".". • **Image**: not applied by the channel
General - Sensor	• **Last calibration date**: date when the sensor was last calibrated • **Calibration interval**: the interval at which repeat calibrations must be performed. The Sensors overview presents an indication of whether the calibration is still valid and for how long.
Identification - Sensor	• **Sensor supplier**: information for identifying the sensor • **Sensor name**: information for identifying the sensor • **Sensor range lower limit**: affects "*Curve Properties*" > "Min" • **Sensor range upper limit**: affects "*Curve Properties*" > "Max" • **Sensor order no.**: if present - for purposes of simplified reordering • **Sensor serial no.**: information for identifying the sensor
Identification - RFID/QR	• **imc SIMPLEX Tag I, II, III**: RFID-tag - You can join existing RFID-tags to the sensor, or write your own numbers to an RFID by means of an appropriate smart phone app. If you have multiple sensors of the same kind, you can assign the same tag to all of the RFIDs. For instance, use an appropriate smart phone app to import the tag from the RFID and enter the number here. See also: "*Create/dissolve link between RFID-tag with the imc SIMPLEX-app* 749". • **TEDS ID**: is currently not in use • **imc ID**: automatically assigned unique ID. Is required for describing RFIDs.

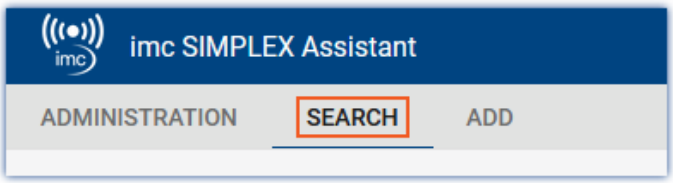
Checking the sensor and adopting it into the database

Symbol	Description
Check	Once the sensor has been completely configured, the button "*Check*" causes an overview to be displayed. Additionally, the data are checked. If the entries are correct, configuration can be concluded by clicking on the button "*Save data in imc SIMPLEX DB*" and the sensor is then adopted into the database.

The sensor is now available to all users of the database.

Editing a sensor

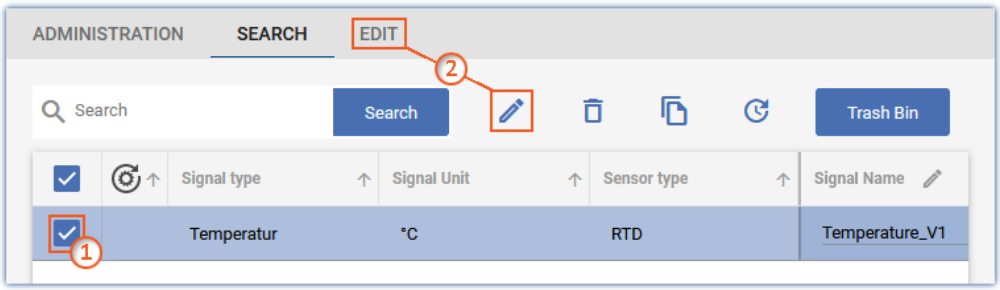
Open the "*Search*" tab on the web interface:



A list of all sensors appears.

In the table, select the sensor ① which you wish to edit.

Click on the pencil icon, or the "*Edit*" button ②.



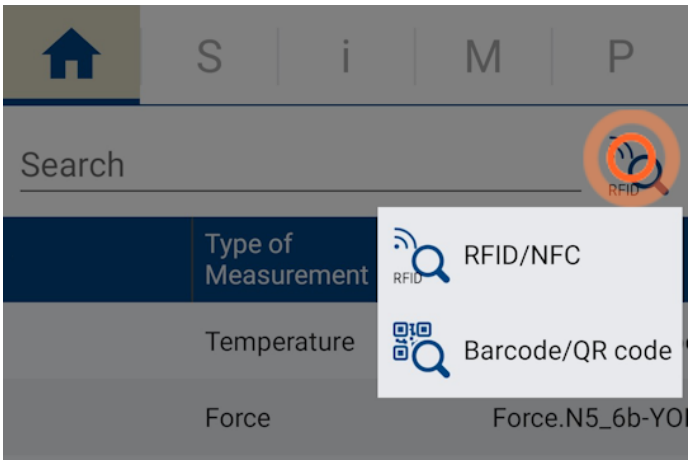
You can now change any parameters present.

9.13.2.1 RFID/NFC or barcode/QR code - link/release with the app

Prerequisites:

- An NFC-enabled smart phone or barcode/QR code enabled
- The imc SIMPLEX app 745 app is running on the smart phone.
- You are logged into the system with appropriate user roles 743.

Toggling between modes: "RFID/NFC" and "Barcode/QR code"



Toggling between modes
Click and hold the Sensor-Scan icon

- Click and hold the Sensor-Scan icon.
- A selection menu appears in which you can select the scan-method:
  - RFID/NFC
  - Barcode/QR code
- Select the scan-method desired.

Link with RFID tag

- Ensure that "*NFC*" is activated on the smart phone.
- Select the target sensor in the app.
- Click on the Sensor-Scan symbol
- Scan the RFID chip.

If the RFID tag has not yet been assigned to a sensor, you will be asked if you want the "*tag*" to be assigned to the last sensor you selected.

If confirmed, the RFID tag will be linked to the sensor.

Link with barcode/QR code tag

- Select the target sensor in the app.
- Click on the Sensor-Scan symbol
- Scan the code with the smart phone's camera.

If the tag has not yet been assigned to a sensor, you will be asked if you want the "*tag*" to be assigned to the last sensor you selected.

If confirmed, the tag will be linked to the sensor.

Removing the Tag-assignment

If the tag was incorrectly assigned, it can be removed again using the app. To do this, **hold the green tag** on the "*S*" **side for a little longer** and confirm removal when prompted.

9.13.3 Applying the sensor configuration to the channel

In the "*imc SIMPLEX*" tool window, all information on the sensors present is displayed.

In order to assign a sensor's information and configurations to a channel, use Drag&Drop to drag the sensor to the appropriate channel. The configurations saved in the sensor are applied to the channel configuration.

If the target channel does not support the sensor, the mouse cursor symbol becomes the circle-backslash "prohibited" symbol. In the column "*Compatible modules and systems*", all modules are listed which are able to use the respective sensor.

9.13.4 Scan sensor via smart phone

Scan sensor via cell phone

If the sensor is equipped with an RFID, bar code or QR-code, and this code is entered as a tag, it can be scanned using the app. imc STUDIO (tool window: "*imc SIMPLEX*") automatically filters for this "*Tag*". Only the sensor that was scanned last is displayed. You can then easily move it by means of Drag&Drop to the channel which is to measure with that sensor.

The filtering can be removed at any time.

The user who is logged on sets up the connection between imc STUDIO and the app. The same user must be logged on in both devices.

9.13.5 Information and tips

FAQ

Question: can multiple sensors be linked with an imc SIMPLEX-sensor?

Answer: Yes. There are two different ways.

- Using different tags: It is possible to linked up to 3 tags with an imc SIMPLEX-sensor.
- Using one tag with multiple sensors: It is possible to link one and the same tag with multiple sensors.
  - This is accomplished relatively easily with Barcodes/QR codes by means of multiple clicks.
  - When using RFID-chips, an app for writing to RFIDs is required, as well as writable RFIDs. In this way, it is possible to link any number of chips with the same tag.
    Search the app-store for "nfc tag writer". There you will find a variety of apps, including some for free, by means of which you can write to RFID-tags.

9.14 Information and tips

9.14.1 Initial value for variables - Beginning the measurement - Jumps at the output

FAQ

Question: What happens when multiple "writers" each try to set their own value during the preparation? For instance, if a value is set for the DAC via a Widget, and also in imc Online FAMOS in the control command: "OnInitAll".

Answer: To prevent signal jumps on a DAC-output channel, then during initialization, the system checks whether a DAC output channel is initialized in the "OnInitAll" in the imc Online FAMOS-code. If so, this value is used and any previously set value from the data pool (e.g. via a Widget) is ignored.

When "*preparing*" (reconfiguring) imc Online FAMOS wins and the value in the imc STUDIO variable is overwritten.

9.14.2 Data recording with sampling or with time stamp

Data recording with sampling

Equidistant sampling is performed according to the following algorithm: each time the sampling interval elapses, one new value is recorded.

- e.g. with an analog input: the instantaneous reading is recorded
- e.g. with a Fieldbus: the most recent reading transmitted on the fieldbus is recorded

The files are saved compactly **without any time stamp**, as a pure series of readings. The individual values' time references are derivable from the parameters "*Measurement start time*", "*Sampling interval*" and "*Count of preceding values*".

Note — When to use equidistant sampling

- If the sensor transmits at a more or less regular rate, then sampling is the most compact means of storing data and the quickest way of viewing it.
- If the is going to be processed with imc Online FAMOS. For many processing calculations, such as all digital filtering, FFTs, etc., it is important to obtain equidistantly sampled data.

Data recording with time stamp

To each reading, a time stamp is assigned, which reflects the point in time when the reading was measured. This time stamp occupies 6 Bytes. Thus, an individual time-stamped sample occupies 6 Bytes more memory than a sample having a fixed sampling rate.

Note — When does it make sense to capture time-stamped readings?

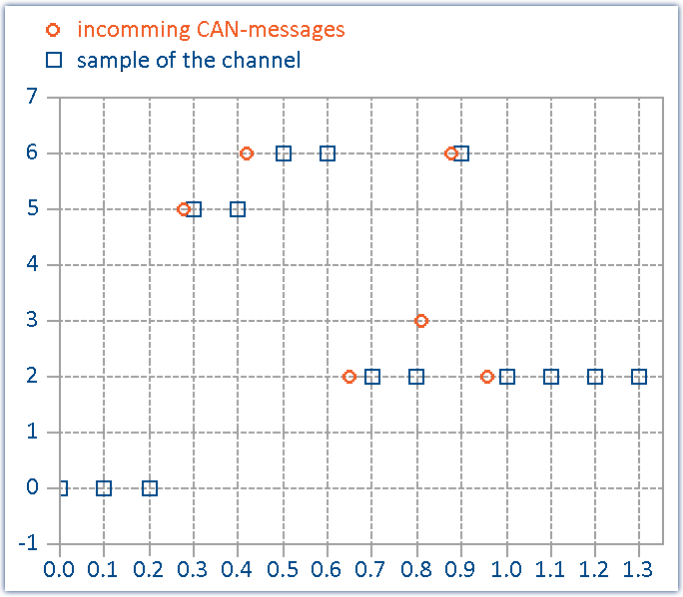
- When you intend to record the exact moment in time when a reading is captured.
- When the sensor is transmitting values at extremely irregular intervals, so that equidistant sampling would unnecessarily increase the data volume.

Disadvantages of time-stamp data acquisition

- It is not possible to subject the data to processing calculations, such as by using imc Online FAMOS.
- If a channel sends high volume of data, the memory requirements must be taken into account. In particular, the device's circular buffer memory requires correspondingly more memory for the same buffer duration than equidistant measurement.

9.14.2.1 Capturing fieldbus input data

Fieldbus messages arrive at more or less irregular intervals, i.e. they cannot be captured at a constant sampling rate. Nonetheless, in many cases it is desirable to record data at regular time intervals. In fact, many fieldbus sensors do attempt to send their messages at regular intervals. Even if they do not succeed at doing this, it is often acceptable to assume this to be the case.



The last reading captured via the Fiedbus is always recorded

This method has important consequences:

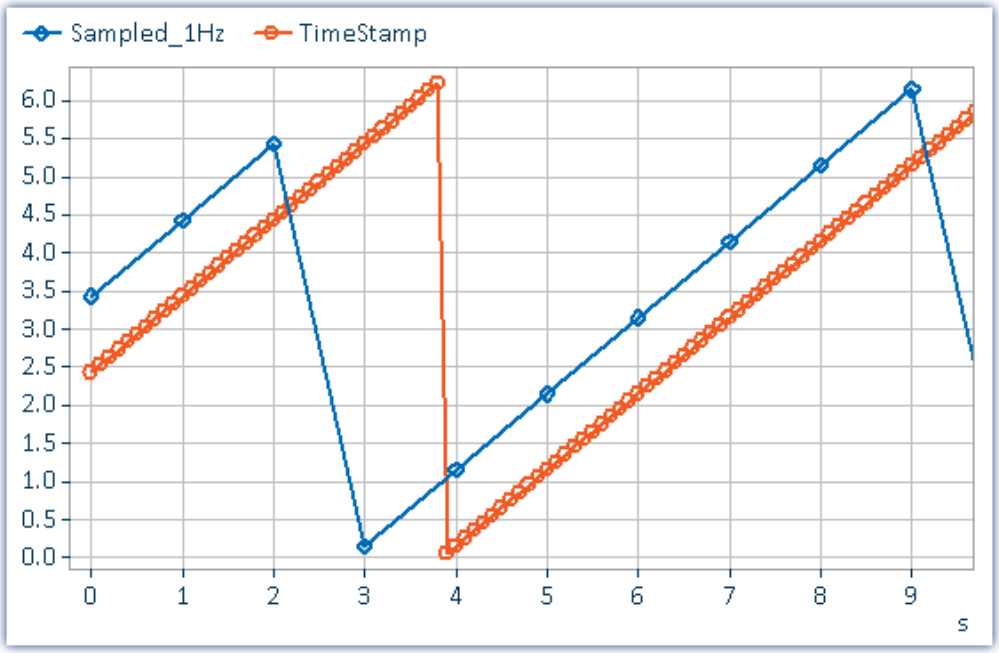
- As long as no messages have been received, the fieldbus channel is filled with zeroes. This can only occur at the beginning of an untriggered measurement before the first fieldbus messages have been received. When a measurement is stopped and restarted without being re-initialized, there are usually previous messages which are being used. In general, zeroes are not created when messages are being sent every 200 ms or less. If the zeroes at the beginning are not desired, we recommend performing a triggered measurement. When you start a triggered measurement, the system is armed and waits until the specified trigger condition (e.g. message value $\neq 0$) occurs.
- If the sampling rate is set significantly more rapid than the intervals at which the sensor transmits, you obtain unnecessarily many data, where successive data points have the same value, thus wasting memory and computation resources. If the sampling rate is significantly slower than the intervals at which the sensor transmits data, then the sparse sampling loses data values.
- It is recommended to select as far as possible a sampling interval matching the messages sending interval.
- If the sensor no longer transmits, the last value received is entered in the channel as the current sampling value.

Notes regarding the display in the curve window

When comparing time-stamped data with equidistantly sampled results from the same fieldbus channel, the equidistantly sampled data appear to be associated with earlier sampling times than the time-stamped data.

This would seem to contradict the previously described functioning principle of the system. What is actually happening is that the equidistant channel adopts the value measured at the end of the sampling interval, but places it at the start of that interval.

In the example below, a channel's messages are recorded with a time stamp (red: TimeStamp). The messages are sent every 100 ms. The same channel is additionally sampled at a fixed sampling rate 1 Hz. As the diagram illustrates, both channels begin at 0 s. The first value of the 1 Hz channel corresponds to the last message in the first sampling interval. The channel thus appears to be running ahead of schedule:



Why is the value placed at the beginning of the interval and not at its end?

With equidistant sampling there is always one sampling interval's worth of uncertainty in the time. For technical reasons, the physically captured value is delayed by the analog input filter, the digitalization, and the process of transfer to the fieldbus, so it is better to apply at least approximate compensation.

9.14.3 Pulse spikes in the signal

If the data capture is performed without using an analog low-pass filter, then disturbed signals may show outliers, peaks or spikes.

Tip: Nonlinear filters

Use the imc Online FAMOS functions `Median3`, `Median5` and `SlopeClip` to cut out such disturbances. If low-pass filters are used (whether analog or digital), the peaks are only "mashed flat" and cause a local offset error.

9.14.4 Why are channels of one device not synchronized?

Answer 1: You are using different filter settings. Also the setting "*AAF*" for different sampling rates result in different low-pass filter settings. To sample two channels absolutely synchronous, the filter and sampling rate rate settings must be the same.

Answer 2: The RAM-buffer time is too big. In certain cases, the RAM-buffer 429 size could result in sample shift. (Only applies to devices belonging to the firmware group A 187 (imc DEVICES))

Answer 3: Channels are assigned to different triggers. Only with curve window setting "*date/time absolute*" the channels are displayed correctly in respect of time. If the curve window is set to "*linear*" (s) the time of the triggers will be displayed together.

Reference

Other scenarios are described in the section "*Synchronization* 327".

9.14.5 Behavior upon data overflow

If data overflow occurs when retrieving the measured data, a gap in the data will result. How the system handles this depends on the data storage settings.

With interval data saving

If a data overflow occurs when the measurement data are retrieved, resulting in a data gap larger than the sampling interval, a new file is created rather than an empty file for the missing data.

With and without interval data saving

If data overflow occurs **within an interval**, or in the **absence of interval data saving**, there are multiple approaches:

Data storage	Behavior
Data storage in the device	No gap in the measured data will be generated. The new samples continue to be appended to the existing measured data, no matter how many samples are lost. For this reason, they are assigned an incorrect time stamp (the samples have a time offset). As soon as a new file is started, the time stamp is again accurate (e.g. interval boundary). For devices belonging to the firmware group A 187 (imc DEVICES): In the "SysLog"-file 417, it is exactly specified when a data overflow occurred and how many data points are missing. Thus, the file can be corrected.
Data storage on the PC	The channel becomes an event-based data set. The new samples are saved in a new event. They thus have a correct time stamp. All data from before and after the overflow are present. The time stamp is correct.

Reference See also

- "*Behavior upon outage of the network connection* 428" (RAM-buffer)
- "*Avoidance of Data Overflow* 755"

9.14.6 Avoidance of data overflow

Very many factors can contribute to a data overflow. There is a concrete correlation to the following factors:

1. Data Rate, Channel Count and Formatting 756
2. Occupied Memory, Measurement Duration and Folder Count 757
3. Manufacturer and Age of the Storage Medium 757
4. RAM buffer duration of the channels 427

See also:

- Set long-term measurement correctly 757
- More tips 758

The accumulating data volume can **not** be exactly calculated for the following settings:

- Triggers, especially ones with long pre-trigger times
- imc Online FAMOS functions
- CAN log channels

Note

- In order to avoid data loss, it is urgently recommended to **test** whether the anticipated accumulating data volume within a specified time frame can reliably be saved on the desired storage medium!
- During a **running measurement** having a high sampling rate, you should **never** try to **access the storage medium in the device** using the Windows Explorer shell. Otherwise, this additional burden could cause a data overflow.

Reference See also

"*Behavior upon outage of the network connection* 428"

Data Rate, Channel Count and Formatting

Every storage medium has a maximum data rate at which data can be continuously written to the medium. The manufacturers commonly state maximum values which cannot be achieved in practise.

Every storage medium has a maximum data rate at which data are continuously written to the medium. The maximum data rate is normally stated in kByte/s. Note that the size of a measured value (sample) can vary from 2 Bytes (some signal conditioner channels) to 4 Bytes (e.g. virtual channels) all the way to 10 Bytes (e.g. Float values with time stamp). With an aggregate sampling rate of 400 kSample/s (only signal conditioner channels), this results in a data rate of 800kByte/s for the storage medium! Older flashcard memory models are not able to go that high; flashcards "age" over the duration of their use.

The size and number of assignment units (clusters) and thus also the file system 452 used have a substantial effect on the storage medium's speed! Small clusters can dramatically reduce the speed! If high data rates are required, it is normally recommended to have a size of 8 kB/cluster.

The optimum size of the clusters must be determined for each storage medium separately. For all of them, the following applies:

- **Few channels having a high data rate**

  If a few channels having a high data rate are being recorded, then **large clusters** on the data carrier provide better advantage. Formatting with FAT32 on the PC and drive sizes < 8 GB creates disadvantageously small clusters, which in conjunction with the full aggregate sampling rate can lead to a data overflow.

  Whenever using cards of up to 8 GB, always use the formatting by the device.

  In the device, cards larger than 512 MB are formatted with 8 kByte clusters and cards larger than 4 GB are formatted with 16 kByte clusters. Cards of up to 1 GB can alternatively be formatted by the PC with FAT16. With cards of 16 GB onward, there is no difference whether the formatting is done in the PC or in the device.
- **Very many channels with a low data rate**

  If hundreds of channels having a low data rate (e.g. CAN channels) are saved, the exact opposite is rue. Here, **small clusters** are an advantage. This means that drives with up to 8 GB should in such cases be formatted in the PC with FAT32.

Occupied Memory, Measurement Duration and Folder Count

The larger the data volume and the more folders on the storage medium, the longer it takes to find free clusters. In order to ensure high data rates, it is recommended that the storage medium be deleted from the measurement or be re-formatted. This also applies to large card volumes such as 16 or 32 GByte.

Make an estimate of the prospective number of folders. Particularly in the modes "*Interval data saving* 423" and "*Trigger events in separate folders* 434", a great many folders can be created in a short time.

Note

Creating more than 1000 folders is to be avoided, since this significantly slows down the file system.

Manufacturer and Age of the Storage Medium

In recent years, manufacturers have succeeded in raising the maximum data rate per second. Be certain to check your card before using it. The cards offered by imc are tested for their data rate and functionality.

Note

- imc has no way to affect the quality of the removable storage media provided by the various manufacturers.
- Storage media which come with newly purchased devices have been inspected in the framework of quality assurance and have passed the relevant tests.
- We expressly declare that the use of removable storage media is at the user's own risk.
- imc and its resellers are only liable within the framework of the guarantee and only to the extent of providing a substitute.
- imc expressly declines any liability for any damages resulting from loss of data.

Set long-term measurement correctly

Say, for instance, you wish to make a really long measurement. You set the channel sampling rate to 100 kHz, the measurement duration to "*undefined*" and the circular buffer time for the display to "*unlimited*". Then you set the channel data to be displayed in the curve window and start the measurement. Here's what happens:

1. The measurement runs smoothly, at first.
2. After a while, there seem to be pauses for several seconds. The software seems to run more slowly.
3. Later, a "*Data overflow*"-message appears.

Diagnosis: As measurement data are captured by the measurement device, the operating system requests memory. Also, the relocation of memory space can take considerable time. If this time is too long, the device is no longer able to stash the data, the PC doesn't read the data out in time, data are lost, a data overflow occurs.

→ Tip: Circular buffer memory

- For the purpose of displaying the channel data from indefinitely long measurement runs, use the circular buffer memory. This circular buffer displays the data for a specified duration, e.g. 30 minutes. This memory is demanded only once at the start of the measurement and is cyclically overwritten. It is then possible to show up to the last 30 minutes in the curve window. The system continues to work at constant speed, for as long as needed.
- Data can also be saved without the circular buffer memory. This will not require any of the operating system's RAM.

→ Tip: Scroll-mode

When the curve window is used to display data, it should be used in "*Scroll-mode* 1414". In Scroll-mode, the curve window always displays the same amount of data, so it always keeps the same speed. If you don't set the X-axis to Automatic (or Unzoom), the amount of data displayed constantly increases and plotting takes longer and longer. Scroll-mode actually cannot prevent data overflow in the PC-RAM! But it is a helpful technique for keeping the displayed time window constant.

More tips

If the error message "*Data overflow*" appears during a measurement, it means that some measured data have been lost. The device buffers captured data for a specified time interval. If the PC is too busy to read in data for longer than this time, the device memory will run out of capacity and any newly captured data will overwrite the previous data.

→ Tip: Increase buffer time

For devices belonging to the firmware group A 187 (imc DEVICES): Use the controls on the page: "*Analog/Digital channels*" in the dialog: "*Data transfer*" to set a greater buffer time 429 for each channel. "*Auto*" is min. 10 s; it is often possible to set a much longer time. Set the same buffer time for all channels transferring data to the PC, and make it as long as possible. Channels which are only processed by the device don't need extensive buffering; 2 s are usually enough.

For devices belonging to the firmware group B 187 (imc DEVICEcore): Here, no modifications are necessary. The buffer duration is optimally allocated automatically.

→ Tip: Reduce data rate

If possible, reduce the data rate by means of the data reduction functions in imc Online FAMOS.

→ Tip: PC processing time, Scroll-mode

Avoid performing actions with the PC which demand much processing time. E.g., working with other applications, using Unzoom or automatic X-axis in the curve window when there are huge amounts of data, using complicated curve window display options like Waterfall, Color map, many events, plotting with big dots, ...

→ Tip: Monitor the system variable "*Critical channel fill level* 1139"

Proportional (in percent) fill level of the "critical channel's" FIFO. The "critical channel" is the channel with the highest fill level. The fill level is allowed to rise temporarily and have high peaks. However, it may never reach 100%, or a data overflow occurs in consequence. If the fill level rises not temporarily but constantly, this indicates an overflow will occur later.

9.14.7 Parameterizing bridges

Question: How can I parameterize a bridge in imc STUDIO, if my scaling factor is unknown?

Answer: Condition: The bridge can be unbalanced with a defined load.

We demonstrate the issue using the example of torque measurement, in which a weight is used to apply a defined torque of 4.17 Nm to the shaft.

1. Calculating the actual scaling factor: Activate the channel on the page: **Analog/Digital channels** in the dialog: **Channel definition**. Set the desired sampling rate (Dialog: **Sampling & Preprocessing**). Leave the scaling factor (Factor) (Dialog: **Range & Scaling**) at the default value *1 "mV/V" / "mV/V"*.
2. Select the desired bridge type (half-, full-, quarter bridge in the Voltage mode) in the dialog: **Measurement mode** and set the necessary power supply. Initially, the maximum possible input range is selected.
3. Save the experiment and then on the page: **Channel balance** in the dialog: **Balance** select the desired channel from the channel list. Activate bridge balancing in the column *Balance*.
4. Once the bridge has been balanced, the shaft is stressed with the defined torque and a measurement is performed. The measured value is, for instance, 0.5mV/V (bridge diagonal voltage). Naturally, this value is relatively imprecise, since the input range is 500mV/V.
5. Stop the measurement. On the page: **Analog/Digital channels** open the dialog: **Measurement mode** and then select the next bigger input range from 0.5mV/V, in other words 1mV/V and then repeat the measurement. Now you obtain a more exact measured value of, for example, 0.541mV/V.
6. Using this value, we can calculate the scaling factor: 4.17 Nm/0.541 mV/V = 7,708 "Nm" /mV/V. This value can now be entered in the dialog: **Range & Scaling** for the scaling factor Factor.
7. You now have a choice of input ranges in physical units in the dialog: **Measurement mode**. Here, select the maximum expected number of measured values as the desired input range.
8. If the input range was changed, it is necessary to repeat the bridge balancing!

Note

- **Caution: Every time the input is changed, it is necessary to perform bridge balancing again before starting the measurement!**
- The balancing values are stored with the experiment and imported by the device upon preparation of the measurement. This means that after activating or deactivating the measurement device, the balancing values are not available in the device! Exception: Autostart / Diskstart.

9.14.8 Scaling and input range

imc STUDIO returns measured data correctly scaled in the appropriate physical unit. However, there are a number of different settings possibilities which may cause confusion.

Basically the input range is derived from the selected electrical input range on the page: **Analog/Digital channels** in the dialog: **Measurement mode** and the scaling factor (Factor) on the page: **Analog/Digital channels** in the dialog: **Range & Scaling**.

The following possibilities are available:

- Scaling factor without offset 760
- Scaling factor with offset 760
- Calculated two-point scaling 760
- Two-point scaling by means of the amplifier adjustment dialog 761

Scaling Factor without Offset

If the expected electrical input range is not known, ignore step 1 below.

In the example, a full bridge is described which returns a maximum electrical value of 35 mV/V at a scaling factor of 356.65 N per mV/V.

Step	Description
1. Input range	On the page: **Analog/Digital channels** in the dialog: **Range & Scaling**, select the appropriate input range so as to make the maximum expected value fall in the upper third of the input range. For an expected maximum value of 35 mV/V, the appropriate input range is 50 mV/V.
2. Scaling factor	Enter the scaling factor as the Factor; in this example 356.65 N/mV/V.
3. Input range	Note that afterwards the input range is also indicated together with the physical unit.

The full bridge used in the example normally causes an offset in its rest state. For measurement bridges, this offset should always be compensated using on the page: **Channel balance** the dialog: **Balance**.

Scaling Factor with Offset

The example presents a displacement sensor which delivers 0 V at -11.55 mm. The scaling is 0.2597 V/mm. The maximum value is 25 mm.

Step	Description
1. Input range	On the page: **Analog/Digital channels** in the dialog: **Measurement mode**, select the appropriate voltage range. The total displacement is 25-(-11.55) mm= 36.55 mm. This means that the sensor delivers a max. value of 36.55 mm · 0.2597 V/mm = 9.492 V. If necessary, set the coupling to DC and then set an input range of 10 V.
2. Scaling factor	Go to the dialog: **Range & Scaling**. The scaling factor must be converted to physical units per volts: Y-factor= 1/ 0.2597 V/mm= 3.8506 mm/V Note that the value must be expressed in the SI-unit "m" and not "mm".
3. Offset	Enter the offset if -11.55 mm.
4. Input range	The offset entered causes the input range to shift.

Caluculated Two-point Scaling

The example describes a pressure sensor delivering a current of 4 to 2 0mA. The spec sheet indicates that 4 mA represents applied pressure of 0bar and 17.5 mA represents 10 bar.

Step	Description
1. Input range and Coupling	On the page: **Analog/Digital channels** in the dialog: **Measurement mode**, select the appropriate current range. The UNI-8 used in our example must first be set for DC-coupling. Next select current measurement and then an input range of 20 mA.
2. Range & Scaling	Go to the dialog: **Range & Scaling**.
3. enter the values	Next enter the two value pairs from the spec sheet.

Two-point Scaling by means of the Amplifier Balance Dialog

Along with calculated two-point scaling which is performed by the measurement system's motherboard, it is possible to have the scaling performed already by the amplifier module. In this case, it is no longer necessary to deal with the offset and factor on the page: **Analog/Digital channels**. However, this method requires measurements of both reference points which means a connection is needed to the device.

With this technique, you need to know the expected electrical measurement value in order to be able to select the appropriate input range. This may no longer be changed later, since that would make the adjustment values obsolete.

In the example below, a power sensor is to measure the load on a crane. A standard weight of 100 kg is available. The sensor delivers a maximum of 8 V.

Step	Description
1. Input range and Coupling	On the page: **Analog/Digital channels** in the dialog: **Measurement mode**, select the appropriate voltage input range. If necessary, set the coupling to DC and then select an input range of 10 V.
2. Scaling factor	Go in the dialog: **Range & Scaling** . Enter 1 N/V as the Factor.
3. Channel balance: Point 1	Activate the device and connect it with the PC. Go on the page: **Channel balance** in the dialog: **Scaling / Cable compensation without Sense line**. Connect the sensor and enter 0 N under "*Point 1 set value*". Measure the unstressed crane by clicking on the button "*Scaling point 1*".
4. Channel balance: Point 2	Apply a load to the crane with the standard weight. The force in Newtons is 100 kg · 9.861 m/s^2 = 986.1 N. Click on "*Scaling point 2*".
5. Input range	In the *Scaling* column, click on the button: *Two-point scaling* Go back to the page: **Analog/Digital channels** in the dialog: **Range & Scaling**. The resulting input range is displayed.

The method is recommended because it takes the entire measurement chain into account.

It is theoretically possible to enter an additional Y-factor, and an offset.

9.14.9 Breakdown of the start/stop and triggering behavior

How does the system behave exactly upon starting? How many clock cycles are required until the trigger is released? What are the effects of pre-processing?

Here you will find various applications examples which clarify these questions.

There is a distinction among three device types.

- fixt configuration devices belonging to the firmware group A 187 (imc DEVICES), such as C-SERIES, CRONOS-SL, CRONOS*compact*, ...
- dynamically configurable devices belonging to the firmware group A 187 (imc DEVICES), such as CRONOS*flex*, CRONOS-XT
- devices belonging to the firmware group B 187 (imc DEVICEcore), such as ARGUS*fit*, EOS

The following examples each present one device type as an illustration for their respective group. They thus represent the other devices within their group:

1: CS | 2: CRFX | 3: ARGUS*fit*

Starting with "BaseTrigger" (without extra trigger)

CS	CRFX
The first measured value appears directly. The subsequent measured values then each appear after elapse of the sampling interval specified.	The first measured value appears after a delay. Only once the second measured value is available, both values are displayed. The subsequent measured values then each appear after elapse of the sampling interval specified.
Example: sampling interval: 5 s The first measured value appears immediately. Its relative position is 0 s. The second after 5 s, …	Example: sampling interval: 5 s The first two measured values appear after 5 s. Their relative positions are 0 s and 5 s. The third appears after 5 s more, …

Re-start

CS and CRFX
The first measured value appears directly at its associated clock pulse in reference to the previous measurement. This means that if two sampling rates are used, the slower one is not synchronized to the relative position 0 s, as faster channels are.

Stopping the measurement:

CS	CRFX
The measurement stops immediately; no subsequent measured value appears.	Once the measurement is stopped, one more measured value for each channel appears after the sampling interval specified. Only after that, the measurement is really concluded. Stopping is thus delayed by a maximum of one sampling interval of the slowest channel.

Pre-processing: e.g. arith. mean

CS	CRFX
Display and sorting of the first measured value is delayed due to the duration of the averaging. The subsequent measured values appear after elapse of the specified sampling interval, multiplied by the number of reduction points.	Pre-processing not possible
Example: Sampling interval: 1 s | Function reduction points: 8 The first measured value appears after 8 s. Its relative position is at 8 s and not synchronized to the others at 0 s. The second appears after another 8 s, …	

Re-start, with pre-processing

CS	CRFX
The first measured value appears directly at its associated reduced clock pulse in reference to the previous measurement. This means that if two sampling rates are used, the slower, reduced channel is not synchronized to the relative position 0 s, as faster channels are.	Pre-processing not possible

Starting with a fast channel governed by "BaseTrigger" and a slow channel governed by "Trigger_01" | Trigger-source: Virt-Bit

CS and CRFX	ARGUS*fit*
The first measured value appears directly at its clock pulse once the trigger has released. The second value appears after elapse of the sampling interval specified.	---
Example: Sampling interval: 5 s; the trigger releases after 21 s The first measured value appears (in absolute terms) 25 s after the start.	---

Stopping "Trigger_01" (measurement continues)

CS and CRFX	ARGUS*fit*
Once the trigger is stopped, two more measured values appear	---
Example: sampling interval: 5 s; triggered measurement is concluded after 41 s At each of the following times, another measured value appears: 45 s and 50 s	---

Starting with fast channel governed by "BaseTrigger" and slow channel governed by "Trigger_01" | Trigger-source: same channel

CS	CRFX
For: Signal > Level or Positive slope The first measured value appears two clock pulses after the event. The second value appears after elapse of the sampling interval specified.	**For: Signal > Level or Positive slope** The first measured value appears directly at its clock pulse once the trigger has released. The second value appears after elapse of the sampling interval specified.
Example: Sampling interval: 5 s; the trigger releases after 21 s. The first measured value appears (in absolute terms) 30 s after the start.	Example: Sampling interval: 5 s; the trigger releases after 21 s The first measured value appears (in absolute terms) 25 s after the start.

Stopping "Trigger_01" (measurement continues)

CS	CRFX
For: Signal > Level After stopping the triggered measurement, three more measured values appear.	**For: Signal > Level or Positive slope** After stopping the triggered measurement, one more measured value appears.
Example: Sampling interval: 5 s; the triggered measurement concludes after 41 s. At the following times, more measured values appear: 45 s, 50 s and 55 s	Example: Sampling interval: 5 s; the triggered measurement concludes after 41 s. At the following time, one more measured value appears: 45 s
For: Positive slope After stopping the triggered measurement, two more measured values appear.	
Example: Sampling interval: 5 s; the triggered measurement concludes after 41 s. At the following times, more measured values appear: 45 s and 50 s	

9.14.10 Exchanging values among the devices

For exchanging or processing values across devices, there are a variety of approaches and methods. Some are specific to a device type, others are specific to the imc STUDIO components (e.g. imc Inline FAMOS).

Exchanging via digital outputs/inputs

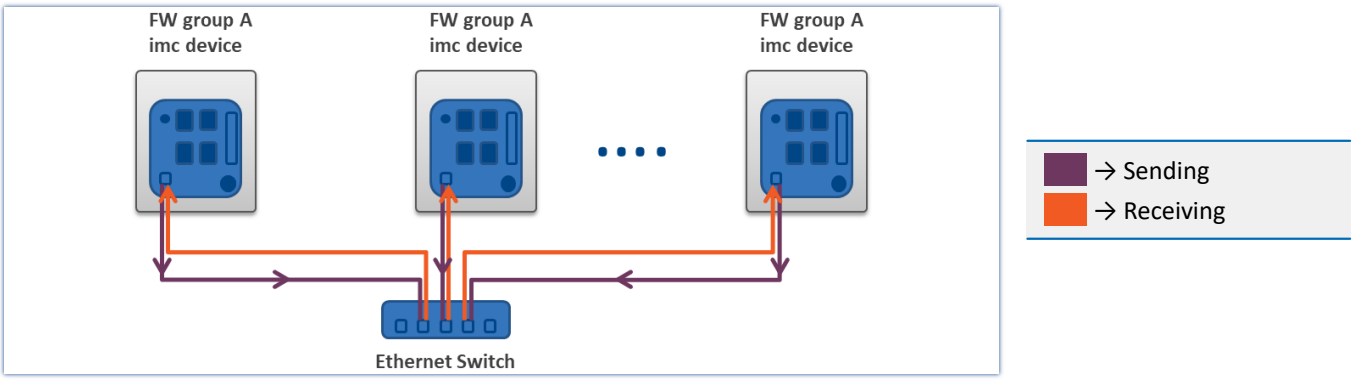
Communication via individual bits/port

- Applies to device which are equipped with the appropriate module.

Using the digital outputs, control functions can be implemented. In this way, communication between the devices is also possible via an appropriate connection line. Refer to the respective device manuals for more information on connection.

Exchanging variables across devices via Ethernet

- Applicable to devices belonging to firmware group A 187 (e.g. imc CRONOS, imc C-SERIES, ...).



Communication via individual bits - use of Ethernet bit variables

If the value of an Ethernet bit changes, this change is recognized and applied by all imc devices located in the same network. This means not only by the devices belonging to the experiment.

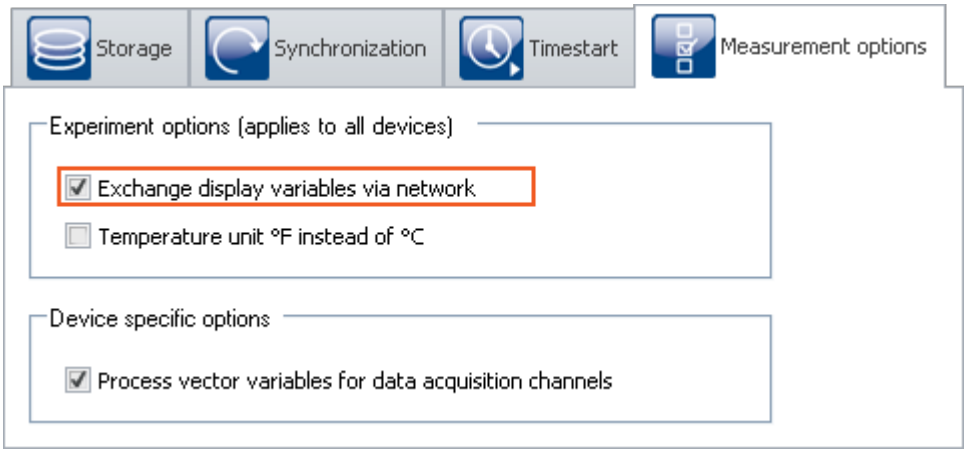
No additional settings are necessary.

Communication by means of variable - use of Display variables

If the value of Display variable changes, this change is recognized and applied by all imc devices located in the same network. This means not only by the devices belonging to the experiment.

Prerequisite:

- The measurement option "*Devices*" > "*Exchange display variables via network* 338" is activated for the devices.



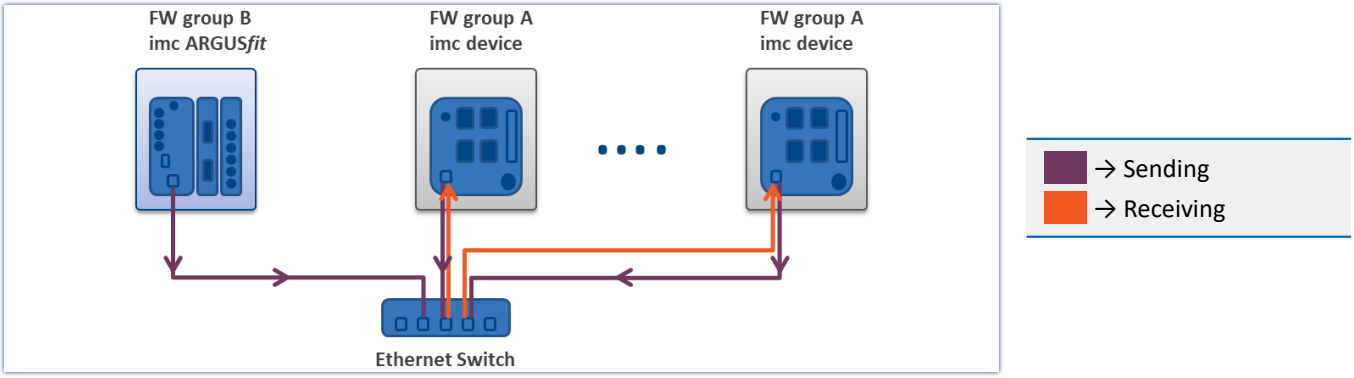
Activation for devices of firmware group A (z.B. imc CRONOS...)

Notes

- The maximum sending-/reading rate is 10 ms. For this reason, any changes occurring at a higher speed (such as on channels sampled at high-speed) can not be transmitted.
- Checking and sending of the display variables burdens the measurement device's and the Ethernet's resources. For this reason, the function should only be activated if data exchanging between two devices is intended.

Exchanging variables via the Ethernet - from imc ARGUS*fit* to devices belonging to firmware group A

imc ARGUS*fit* devices **can send values** to the Display variables of imc devices belonging to firmware group A (e.g. imc CRONOS, imc C-SERIES, ...). However, **reception of values** from other devices is **not possible**.

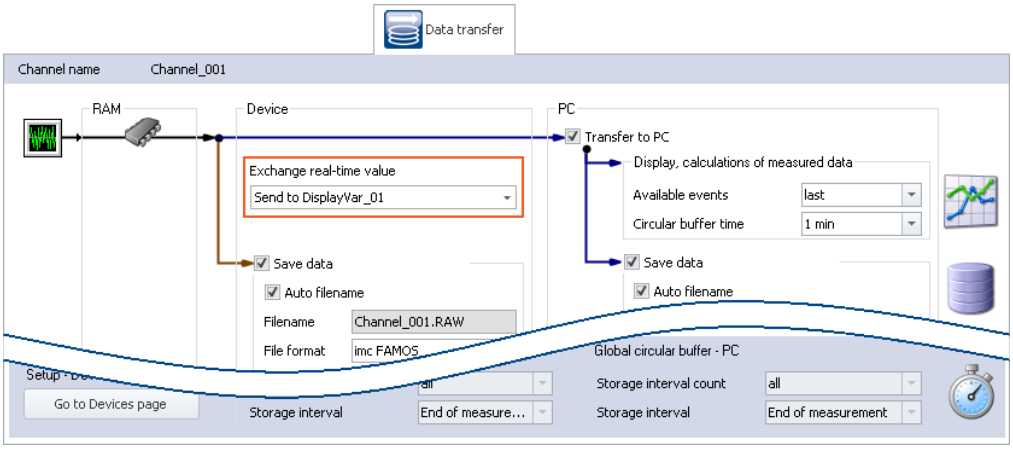


For each individual channel it is possible to select whether to send it to a Display variable.

If the value at the imc ARGUS*fit* channel changes, this change is recognized and applied by all devices of firmware group A which are located within the network and have the appropriate settings. This means not only by the devices belonging to the experiment.

Prerequisites:

- Firmware group A device: The measurement option "*Devices*" > "*Exchange display variables via network* 338 " is activated for the devices - see also "*Use of the Display variables* 766 "
- For the parameter "*Data transfer* 370 " > "*Exchange real-time value*", a Display variable is to be selected for the channels to send.



Assigning an imc ARGUSfit channel to a Display variable

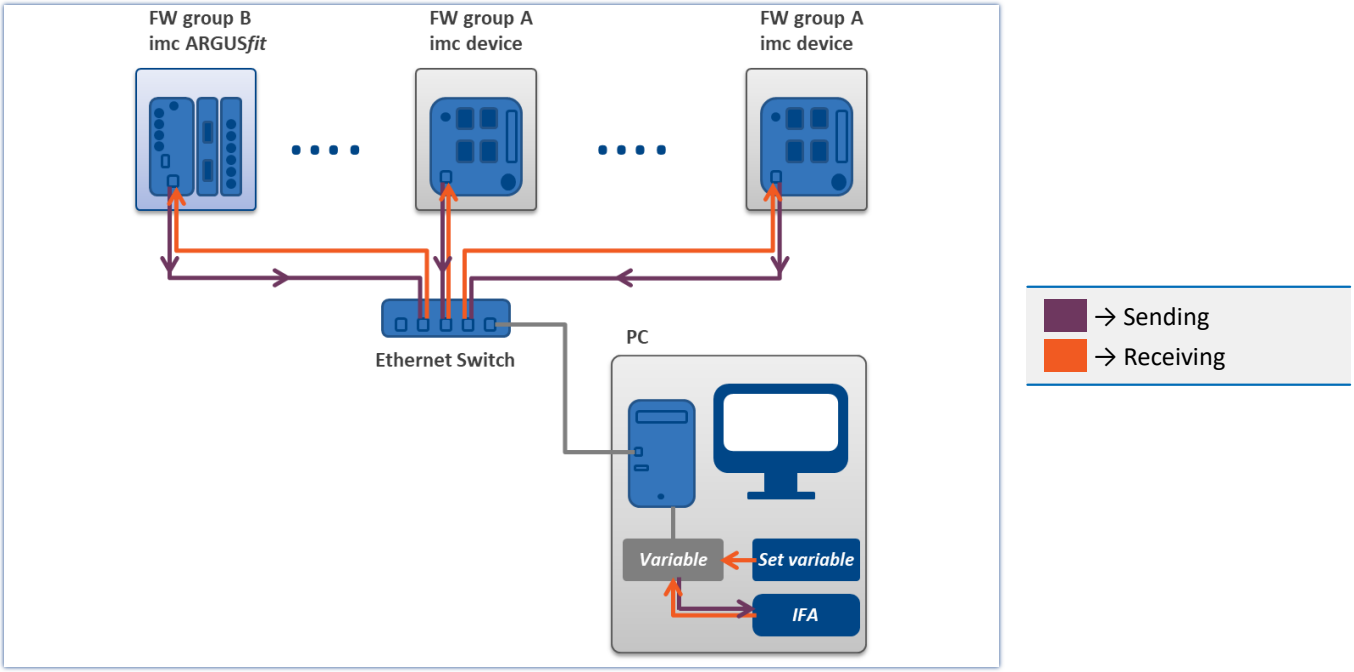
Notes

- For each channel (maximum of 32), one Display variable can be selected as the target.
- The maximum sending-/reading rate is 10 ms. For this reason, any changes occurring at a higher speed (such as on channels sampled at high-speed) can not be transmitted.
- Always the last value detected by the device is sent. If any additional values have been accumulated in the device, they are ignored.
- Immediately upon sending the values, the action "*Prepare*" begins. Even if the actual measurement procedure has not yet begun.
- Sending the values burdens the Ethernet, therefore this function should only be activated if data exchange between to devices is intended.
- For imc EOS devices, this function is not available.

Exchanging values among devices by means of processing on a PC

By means of imc STUDIO-functions, the values can be provided to and exchanged among the devices in various ways.

For example, this can be done either at a specific point in time by setting variables using commands such as "*Set variables* 1764", or continually by means of calculation operations in imc Inline FAMOS.



10 Setup - Advanced Device Functions (imc DEVICES)

Here, certain special topics regarding "*Setup*" are presented.

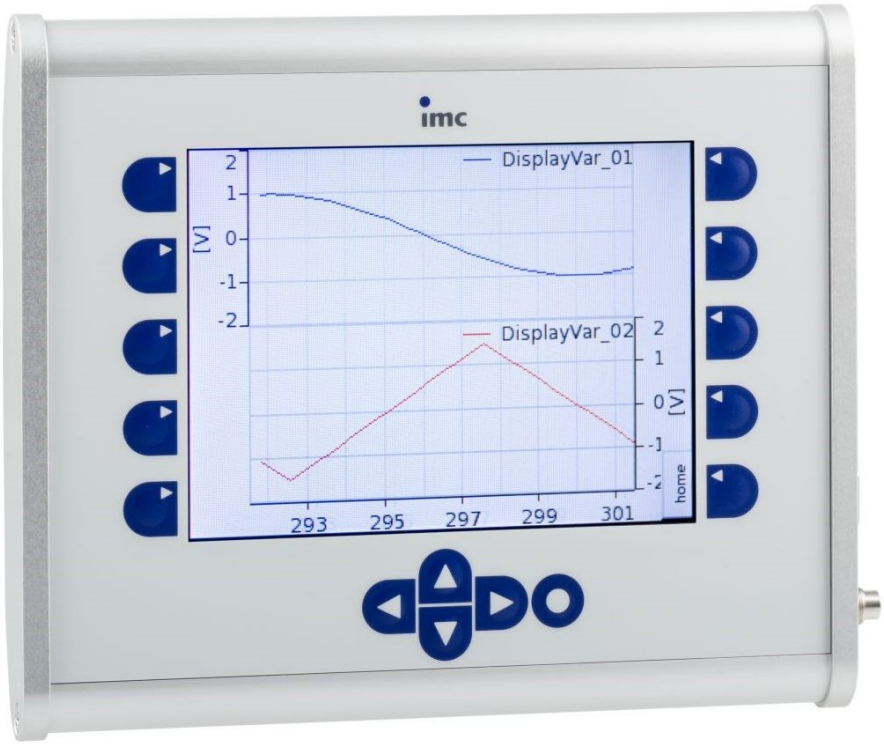
The following chapters apply to devices of the firmware group A 187 (imc DEVICES) - e.g. CRONOS family, C-SERIES, SPARTAN, BUSDAQ.

Chapter overview

Synopsis	Section
Observing and controlling measurement via the imc Display	imc Display editor 769
Sending text messages via the device	imc Messaging 786
Special topic: Transferring files via FTP to the device. Transferring configurations via FTP.	Configure via FTP 809
Operating a device via web-browser	imc REMOTE WebServer 822

10.1 imc Display editor

This chapter describes the operation of imc displays.



External display 320 x 240 pixel

The display offers, besides textual readouts, the display of such graphical objects as curve plots, bar charts, virtual gauges and digital status readouts. The resolution of display is 320 x 240 pixels.

10.1.1 Overview

The optional display screen enables interaction between the user and a running measurement process by posting read-outs of system states and allowing parameter adjustments via the membrane touch panel.

If the measurement device is prepared in such a way that it loads a certain configuration upon being activated, it's possible to run a measurement without any PC. The Display serves as a comfortable status indicator device and can replace or complement imc STUDIO when it comes to controlling the measurement. It can even be used where no PC can go, e.g. at temperatures of -20°C or +70°C.

The Display can be connected or disconnected at any time without affecting a running measurement. This makes it possible, for example, to check the status of multiple devices running simultaneously one at a time.

Interaction with the measurement device is provided by means of Display variables or bits, which can either be evaluated to obtain status indications or modified in order to influence the measurement process.

Reference

The technical data you will find in the device manual.

Note

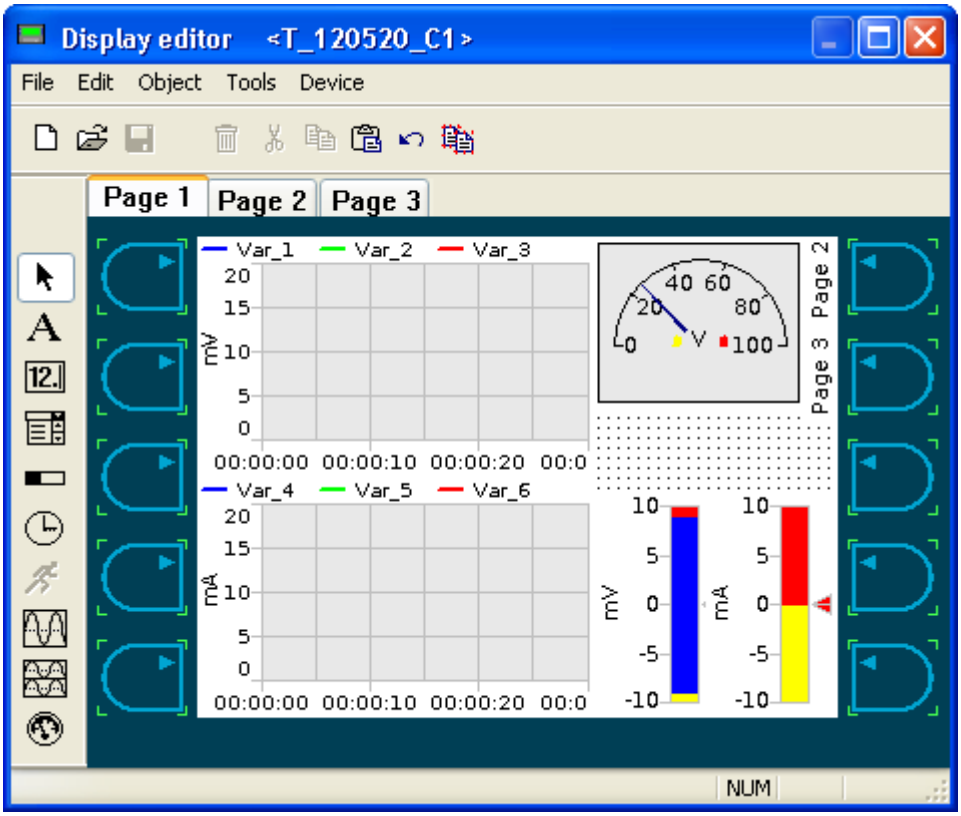
- Control of a Display is exerted via a serial connection. For external display, a normal zero-modem cable can be used.
- It is not possible to adjust the **refresh rate**, which rather depends on the measurement device's performance demands. At top performance, the data are updated 15 times per second.
- The measurement device is not able to independently recognize the Display type. Instead, it must be specified beforehand in the software. Once the Display type has been changed, the device must be rebooted. (See: "*Device Properties* 221")

Ribbon	View
Setup-Configuration > Device Properties	Complete

10.1.2 Display editor in imc STUDIO

The display is organized in multiple pages. One or more controls can be positioned on each page. The available elements for interfacing with the measurement process are the Display variables, virtual bits and Ethernet bits.

Along with the display controls, it is also possible to configure function buttons. Every key-combination command is associated with a particular page, so that one and the same key can call different functions depending on what page is currently open.



Under Edit, more Pages can be created. Each page works like a separate Display. Page selection is made using the Display's Function Keys 780 or by remote control by means of the imc Online FAMOS function `DisplaySetPage` 931.

10.1.2.1 Menu

Menu	Description
File	Import/Export: The Display's configuration can be imported and exported via the Display Editor. The configuration file is an XML document. By this means, one experiment's Display configuration can be copied for a different experiment. The item ***Save to configuration*** saves the Display's current settings with the experiment. Saving also takes place when the Editor is exited.
Edit	Along with the usual editing functions such as *Copy/ Delete/ Cut* and *Undo*, there are also functions for arranging objects in case they overlap each other. Here you will also find the important function *Add page* and *Delete page* for creating new pages in an experiment. To copy an object within a page, click on it while holding down the *Ctrl* key and drag the object to another location; a copy will automatically be created there.
Object	The object menu contains the same items which are offered in the toolbar.
Extras	The item "*Firmware update* 786" enables you to update a display to the same version as the measurement device. Normally, the *Firmware update* affects all of a device's components. However, it's not always likely for an external Display to be connected to the device during the update. This item can handle this issue retroactively.
Device	Selection of the device whose Display is to be configured.

10.1.3 System menu

If the Enter button is held down for about 3 seconds, the System menu appears. It contains the following items:

Item	Description
start display demo	Starts a **Display Demo**. • The demo encompasses multiple pages and displays generated measurement data. • The demo automatically scrolls through the pages. • The demo can also be turned off via the System menu.
backlight	Sets the **background lighting**
display mode	Choice of Mode as **graphical display** or **alphanumeric display**
page timer	Sets the time interval for the **automatic page scrolling**
exit	Exits the System menu

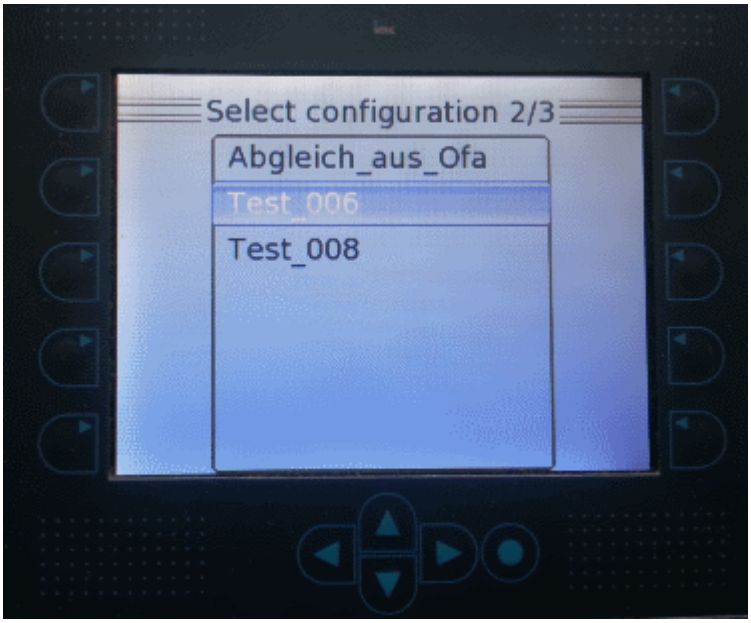
10.1.4 Keyboard

Button	Beschreibung
Enter key	(solid circle) • Starts Input mode (sets the focus to an input element) • Ends Input mode (Enter)
Arrow keys	• Up/ down to navigate through different elements (focus), or to scroll within a selection/ menu • Right/ left to scroll through pages
Function keys	• freely configurable

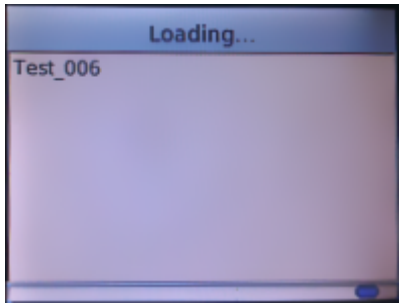
10.1.5 Diskstart

If the device memory contains ***Diskstart configurations***, they can be selected in the display. If a Diskstart configuration was saved as an ***Autostart*** configuration, the selection procedure is skipped and the experiment starts immediately.

Once the device is activated, the experiment appear in a list.



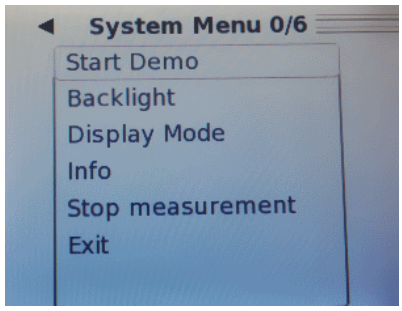
Using the UP/Down arrow keys, you can select the desired entry. Open the experiment using the round button.



Loading the experiment...

During running measurement, the round "*Enter*"-button takes you to the Display-menu.

There, you can conclude the measurement and start a different configuration.



10.1.6 Graphical display elements

For every display element there is a configuration dialog for making its settings. For the display of texts and numbers, the fonts indicated at right are available for selection.

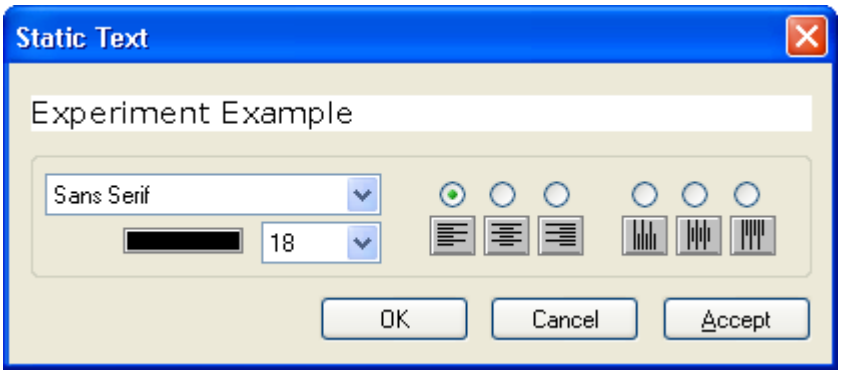
The types presented below are common to all dialogs.

Serif
Serif Bold
Sans Serif
Sans Serif Oblique
Sans Serif Bold
Sans Serif Bold Oblique
Sans Serif Mono
Sans Serif Mono Oblique
Sans Serif Mono Bold
Sans Serif Mono Bold Oblique

10.1.6.1 Static text

This element generates a static text on the Display which cannot be edited form the standard interface.

It should be used mainly for adding comments to active dialog elements.



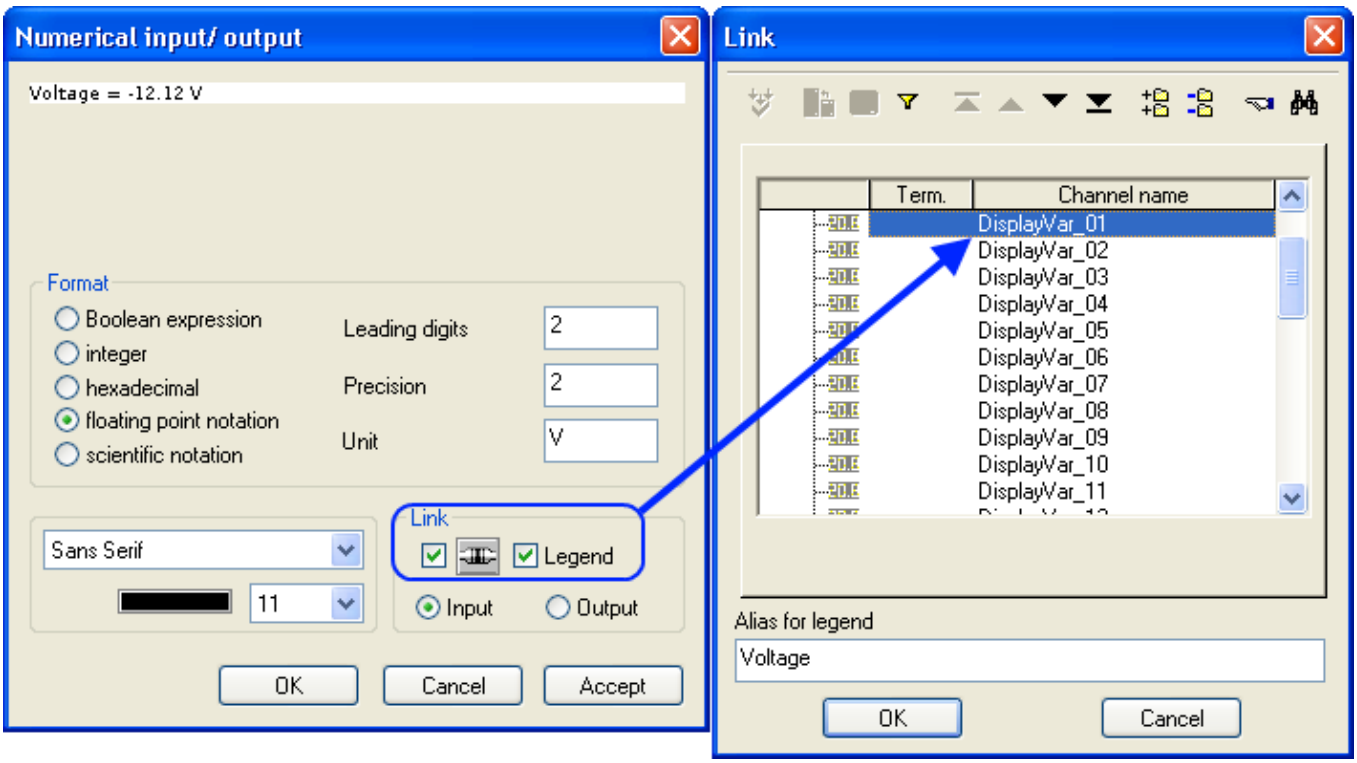
- horizontal text
- 90°-rotated text
- 3 different fonts (Serif, Sans-Serif, Sans-Serif-Mono)
- normal, cursive, bold, bold-cursive

10.1.6.2 Time

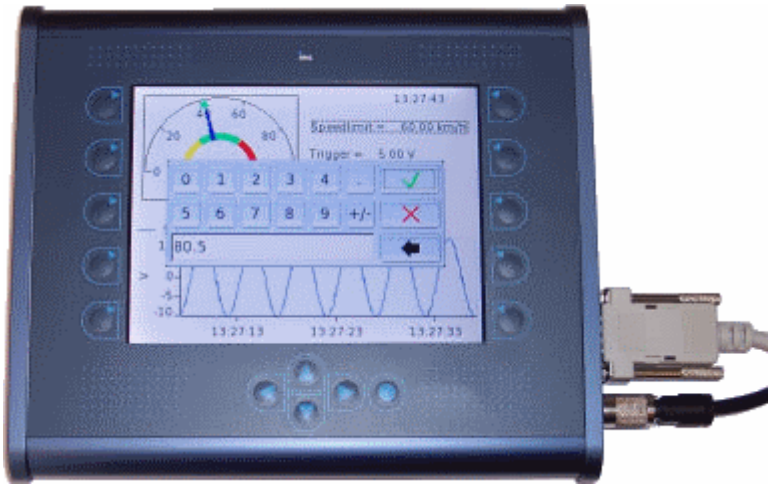
Display of the time with the selected font.

10.1.6.3 In-/Output (numerical)

12. This object, which is linked to a variable, comes with a box for display of a unit to go with the numerical value is displays. The legend states the displayed quantity's name.

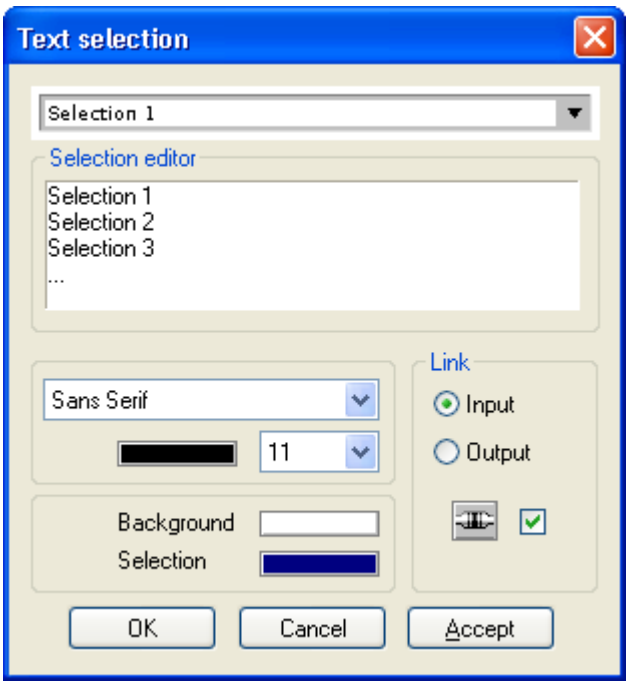


If the box is not only used for display of numerical values, but also for making inputs, the option *Input* must be activated. In the Display, the values can then be entered by means of cursor keys.



10.1.6.4 Combobox (text selection)

A combobox enables you to make a "1 from n" selection, where n is the total number of entries. The entry selected depends on the value of the linked Display variable.

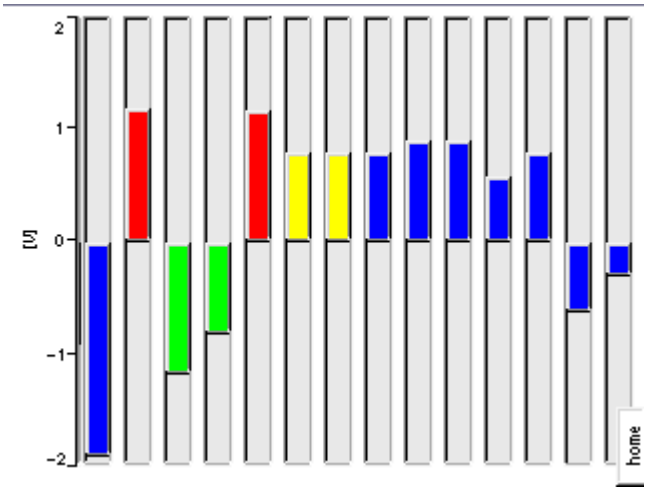


Enter one line of text for each expected state of the linked variables.

Note that the system work with indices: if the variable has the value "0", this corresponds to the first line. The value "1" corresponds to the second line, "2" corresponds to the third line, etc..

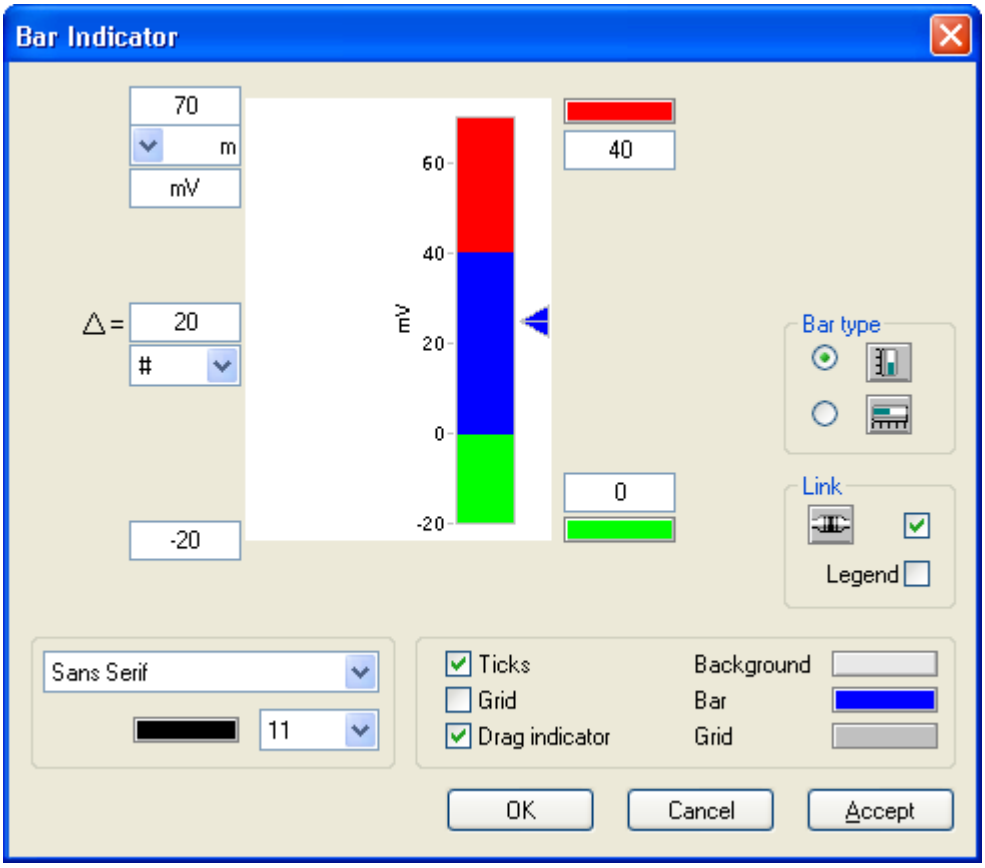
In the box Link, define the channel connection and designate the configuration either as an input or an output.

10.1.6.5 Bar meter

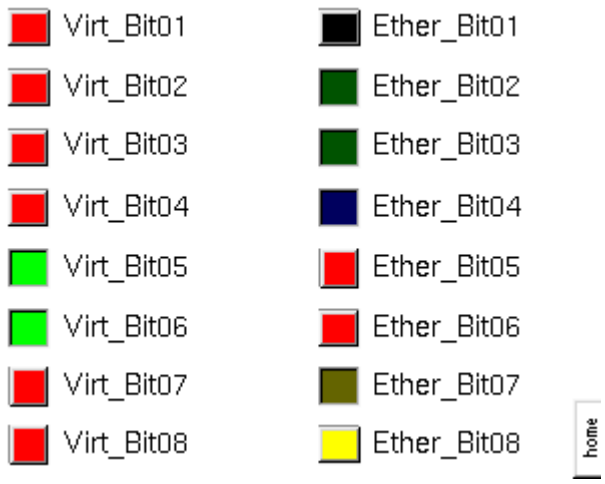


The bar meter is an output element which lets you graphically display a value which varies within a certain range (like a level indicator). The bar can run horizontally or vertically.

Specify the minimum and maximum value of the associated channel. Within this range, the variable's value will be indicated. Depending on the variable's momentary value, the bar will turn colors; the particular colors and associated value limits can be freely selected. The slave pointer's initial point and its precision can also be set.

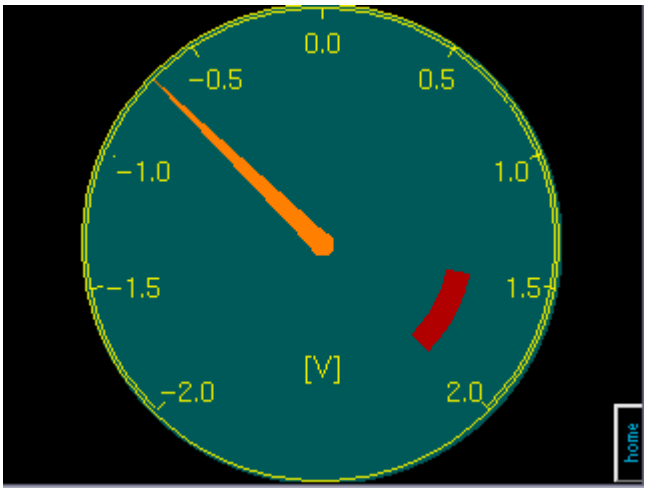


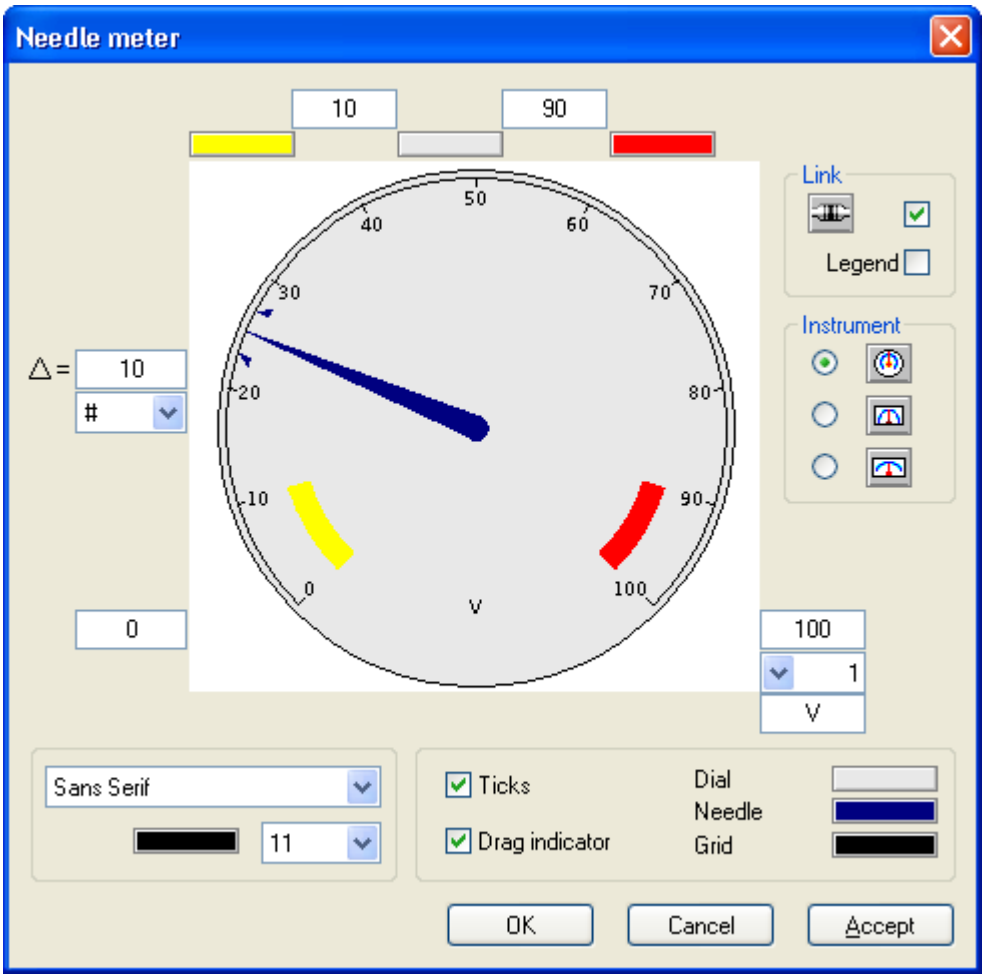
The bar meter can also be used to display digital states. In this case, there is simply no slave pointer or numerical values.



10.1.6.6 Needle meter

Needle meters are set in a similar manner to the bar meters. They additionally feature a scale; the available options are 270°, 180° and 120°-display. The color shading along the scale can be set as desired.

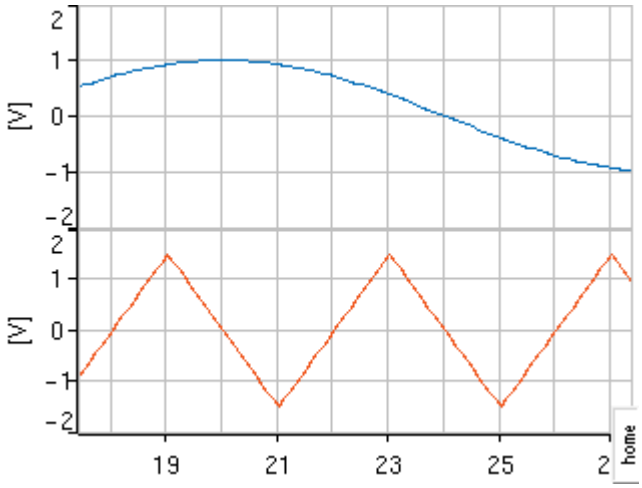




10.1.6.7 Curve window/line chart

The curve window can display the signal curves of up to eight channels. E.g., the signal course of a channel can be displayed in the curve window with the help of a Display variable.

In contrast to the curve window in imc STUDIO, here only the channel's signal value is transferred to the Display. Other information such as the name and unit must be entered in the Editor.

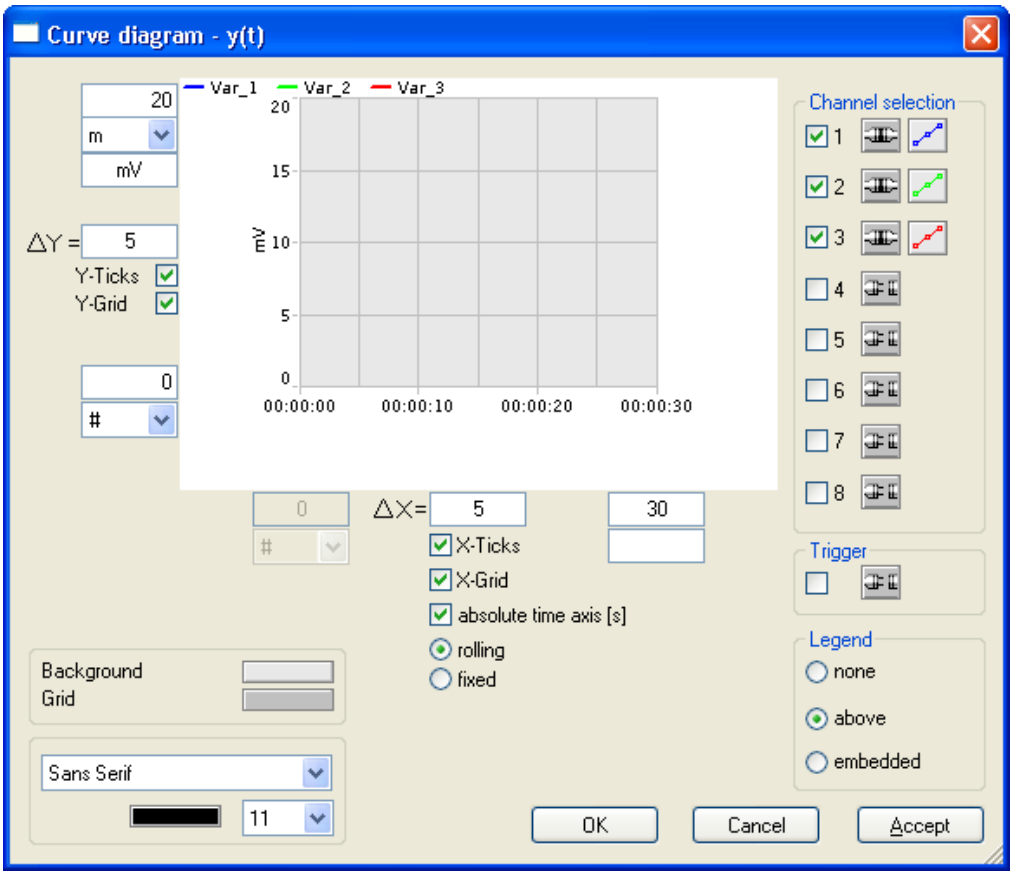


Each channel has a link to a Display variable, which can be activated or deactivated as desired. The dialog for making links has a control Legend, in which, for instance, the channel's name can be entered.

The scaling for the Y-axis is set in the editor, as well as the unit and the number of ticks.

The time axis can refer to either absolute or relative time and be set to either remain stationary or to scroll. Below the X-axis, a permanent text can be added if desired.

Both horizontal and vertical grid lines, either together or separately, can also be displayed.



Note

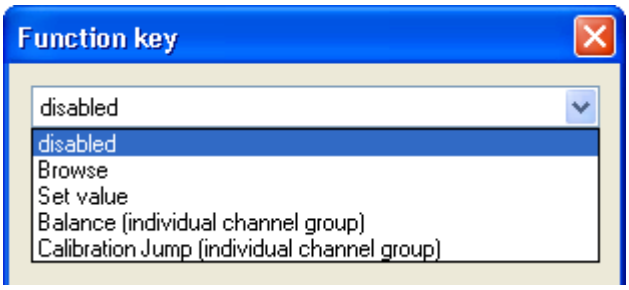
- The curve window in the display is reproduced on the PC. However, no information on the measurement process is available to the display besides the data stream which is transferred by means of display variables. This means that the display knows nothing about the unit, data structure or trigger machine.
- As soon as a trigger stops data acquisition on a channel, imc Online FAMOS stops writing to the associated display variable. If the measurement continues, because other channels haven't stopped, the display simply continues to scroll on with the last value measured.

10.1.7 Function keys on the Display

There are keys on the display's left and right edges which can be configured as desired.

The keys are disabled per default. With the listbox you determine, whether the key:

- scrolls to a certain page
- sets the value of a variable
- or performs balancing and shunt calibration for dedicated channels.

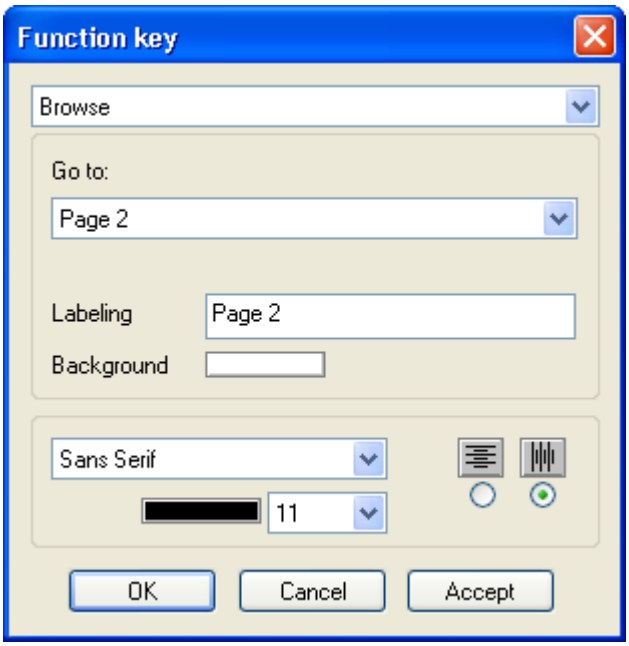


Note

Using the imc Online FAMOS function `DisplaySetButton` 931, the following functions can be executed by imc Online FAMOS as well.

10.1.7.1 Browse

If you have created several pages (Menu Edit 771), you can call those pages via function keys in mode *Browse*.

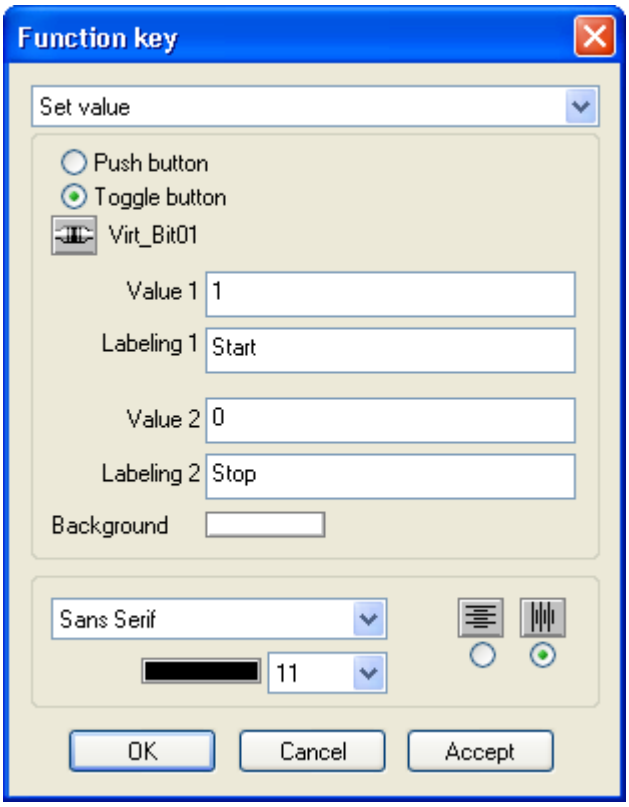


10.1.7.2 Set values

In mode *Set value*, a function key can be linked to a virtual bit or a display variable.

Toggle button:

Using the key as toggle button, e.g. a single key can start and stop a measurement.

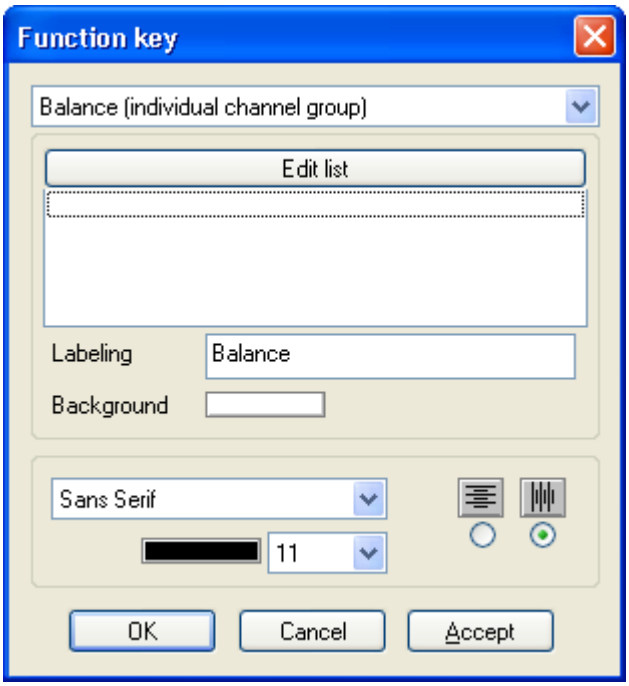


Push button:

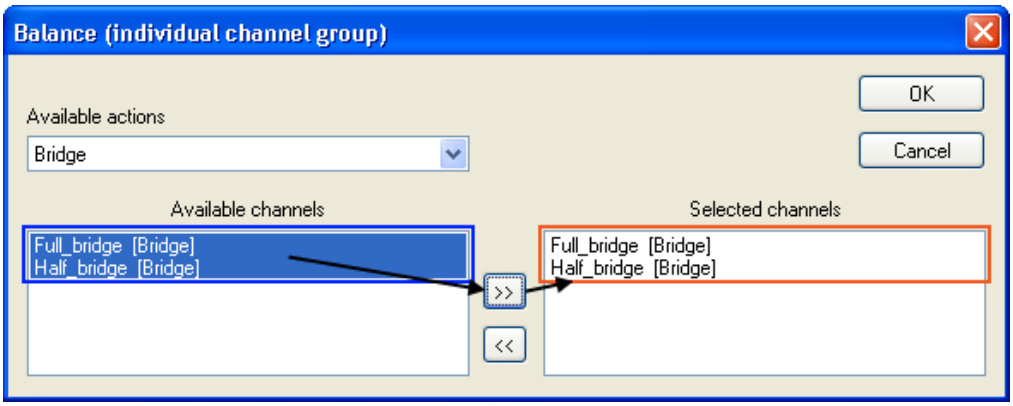
A key linked to a display variable, a certain value could be assigned to the variable that could control a process in imc Online FAMOS.

10.1.7.3 Balance (individual channel group)

The Display can be used to perform balancing of either all or selected channels. Each button can have a selection of channels assigned to it. It is also possible to make a distinction between a true bridge balancing and taring.



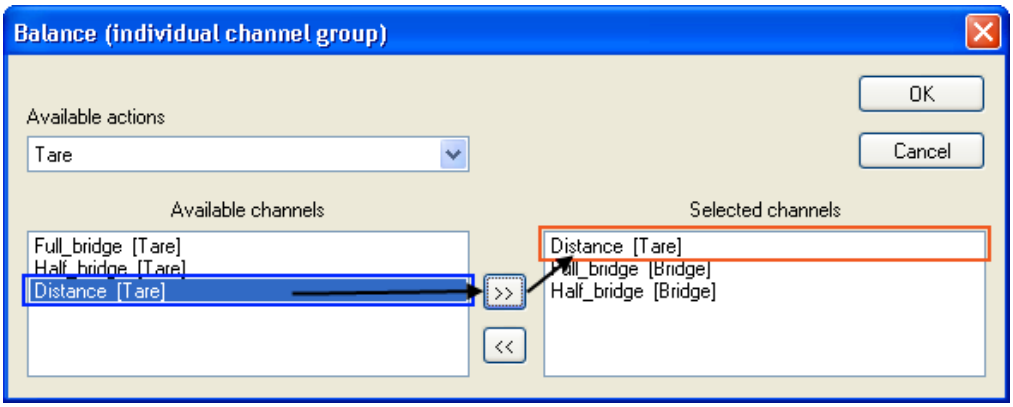
Clicking on the button "*Edit list*" opens a channel selection dialog.



Balancing can also be carried out channel-by-channel. This selection dialog sets which channels are balanced when the Display function key is pressed.

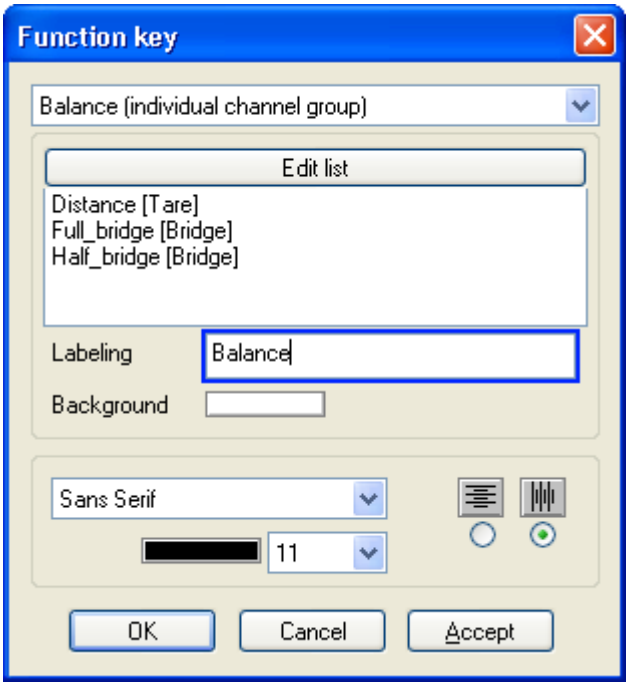
Under *Available actions*, you have the choice between *Bridge* and *Tare*. Bridge channels can be either balanced or tared. For details see the manual: Plug-in: "*Setup*" > "*Channel balance* 377".

When *Bridge* is selected, as in this example, two active bridge channels appear. Double-clicking on a channel or making a selection and subsequently clicking on the >> button makes the channel assignments.



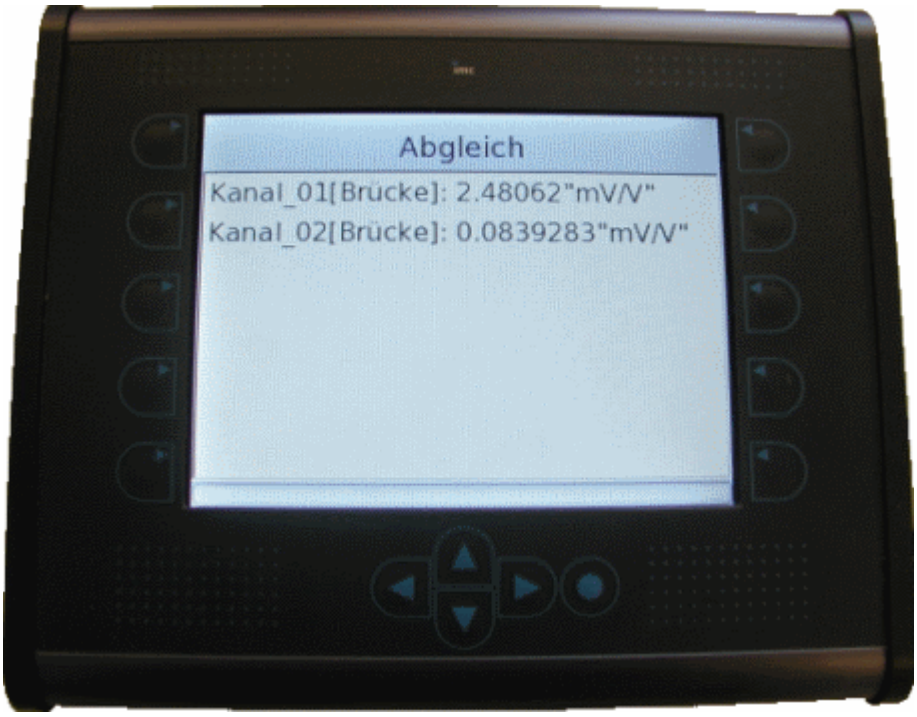
Under Available actions: *Tare*, an additional channel for capturing distance data appears.

After exiting the selection dialog with *OK*, the channels appear in the Display function key's main dialog.



Under "*Labeling*", the key has a visible caption of selected font size and background.

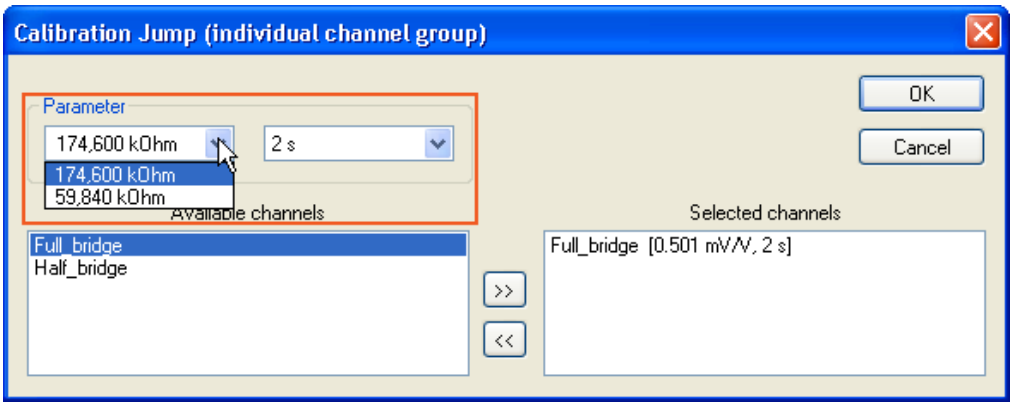
To carry out the action, the Display switches to balancing mode. Next, the compensated values are displayed:



Pressing the round button returns the screen to the normal view.

10.1.7.4 Shunt calibration (individual channel group)

To check the functioning of the bridge channels connected, a shunt calibration can be performed. Assignment of the channels to the Display button is performed in the same way as Balance 783. However, a shunt calibration 381 includes additional settings options:



Under *Parameter*, the calibration resistance value and the duration of its application are set. The calibration resistance selected, in conjunction with a known bridge resistance, produces a computable unbalance. This is the reason why in the plug-in: **Setup** on the page: **Analog channels** in the Dialog: **Measurement mode** is able to indicate the bridge impedance even for full and half bridges:

However, the actually measured unbalance usually deviates from the computed one, since the line resistances are not taken into account.

The shunt calibration is performed without any indication in the Display.

10.1.8 Firmware update

The display is running on its own operating system, that is updated with the firmware update 63. If the display has not been connected during updating all other components, the update of the display is executed with the first configuration using a display setup.

In case the display does not work properly after an update, it has to be restarted. If a measurement configuration with a display setup has been started, but the display had not been connected, the display will be ignored when connected to the running measurement.

10.1.9 Information and tips

Problem: Display is not recognized, Display-Editor is not in menu or editor shows different display type.

Answer: The display is not configured correctly.

You wish to use an external imc Display with your measurement device, which has a DSUB-9 Display terminal:

1) Start imc STUDIO, connect your device
2) Open the "*Device properties* 221"
3) Set your type of display; Apply with *OK*
4) Close imc STUDIO and deactivate the device, connect the display to the measurement device; switch on the display.
5) Switch the device on, so the display indicates a connection to the device.
6) Your display may run a firmware 786 update when preparing the measurement.

10.2 imc Messaging

This chapter describes the features of imc measurement devices for sending e-mail text messages.

10.2.1 Overview

What is imc Messaging?

Devices having the appropriate interface support sending of text messages in response to specified events. The available events are any signal changes of the virtual bits and ethernet bits. In any bit, the transitions both from 0 to 1 and from 1 to 0 can be linked with one or more message texts.

Each such message can have its own target defined for it.

Along with obligatory text components, the message texts can also contain the current values of the ethernet bits, virtual bits and Display variables from the device. Other available text components include the date, time, device name and certain imc Messaging variables, which can be filled with the currently valid values at the moment the message is generated. The information for imc Messaging is comprised of these specifications:

Information	Description
AllMessageCounter	number of all generated messages
AllMessageLosses	number of messages which were rejected because undeliverable or due to lack of resources
EventOverruns	number of events which could not be processed because they accumulated too quickly in succession
ThisMessageCounter	placeholder in message text for the number of times this message was generated
ThisMessageLosses	placeholder in message text for the number of times a generated message of this kind was lost

What can imc Messaging be used for?

Practically any event occurring during a measurement can determine the values of virtual or ethernet bits, either directly or with the help of an imc Online FAMOS program. Among other things, this makes it possible to use trigger events, digital inputs or conditions defined in imc Online FAMOS as the release for messages.

In the simplest case, this can be used to alert a service technician by means of an e-mail when a threshold is exceeded or a similar event occurs.

The ability to place the current values of the ethernet bits, virtual bits and Display variables from the device in the message makes it possible to generate simple reports as well as malfunction messages. With appropriate programming, time-controlled messages, for example, can be sent, which could contain the most important data on processes current state.

Note, however, that imc Messaging is conceived exclusively for occasional sending of messages. It should generate only a few messages per hour or even per day! For more information, refer to the section on reliability, and the technical specs.

How does imc Messaging work?

The devices cyclically examine the virtual bits and ethernet bits for changes. If a change is detected, a check is made of whether there are any message configurations defined for that event.

If there are, then in addition to the virtual bits and ethernet bits already read in, the Display variables, the date, and the time are saved. This ensures that the values are as up-to-date as possible for any later generating of messages.

The detected events and all their data are entered into an event queue. All of this queue's entries are later evaluated and all necessary message texts are generated. In the process, the variables in the message texts are replaced with the stored values.

The messages thus generated are saved in a sending queue, from where they can be sent via the respective suitable interface.

Note

- Virtual bits and ethernet bits should not change their value for durations under 50ms; better yet 200ms.
- Renamed virtual bits or ethernet bits must be entered in the MSG-file under their default name (e.g.Virt_Bit01).
- The messages may not be sent in the order they were made (see Delivery of the messages 789)!

imc Messaging begins its activities already after preparation for measurement. Be aware, therefore, that messages could be sent even before a measurement starts if a bit for which a message is configured changes. imc Messaging remains active even after stopping the measurement. In particular, it continues to try sending messages generated during the measurement! Also the counter variables of imc Messaging are not affected by the measurement's start or stopping. Only upon preparing the next measurement are the counter variables reset and any unsent messages discarded!

In order to deactivate imc Messaging, a measurement must be prepared which contains no message configurations.

How to configure imc Messaging?

The configuration of imc Messaging is accomplished by means of one ASCII-file per device, each has to be imported as supplemental file 212. The designation is "*DEV*"+"*XXX.msg*", where *XXX* stands for the device number with a preceding 0, e.g. "*Dev012.msg*".This file must contain a description for each message which contains not only the actual text but also determines the triggering event and the message's destination.

What particular specifications are needed depends on the respective message type. The syntax of the configuration file and all details on the necessary parameters are described in detail under Configuration 790.

System requirements for imc Messaging

E-mail can be sent either via ***modem***, or via a ***network interface***, if an appropriate server can be reached via the network.

10.2.2 Message type: e-mail

An e-mail can contain almost any amount of text, with the size of the configuration file, however, setting certain limits. Transmission of e-mail by the provider usually occurs after a few seconds, though there is no guarantee for this. By contrast, the e-mail-client may be set to retrieve messages only at longer intervals. Sending an e-mail doesn't require a modem if a mail server is accessible via the LAN interface.

However, to send an e-mail, an **account with a mail server** is necessary so that the device can be identified by the server by means of a login name and password.

Devices having serial numbers 16xxxx or 19xxxx support ***StartTLS encryption***, which is a necessary requirement for most e-mail providers. For devices not supporting it, it is recommended to use either a local e-mail server which transmits the messages with encryption to the e-mail provider, or one of the remaining providers who continue to allow unencrypted transmission.

A step by step tutorial can be found in the chapter: "*Sending an e-mail* 808".

10.2.3 Reliability

10.2.3.1 Recognizing events (time response)

There is a limit to how many events for which messages are configured can appear in rapid succession. While one message is being sent, any other events occurring can only be stored. However, the event queue has only a limited amount of space. Once this is used up, all subsequent events are discarded. These lost events are counted in the imc Messaging variable `EventOverruns`.

It isn't possible, however, to reconstruct afterwards which messages were lost due to discarded events!

In order for a bit's signal transition to even be recognized, the transitions must not occur too quickly in succession. otherwise the event will be lost, as if it never happened; it can't even be counted as a lost event in `EventOverruns`. Therefore, use appropriate programming or use holdoff times to ensure that any event remains intact for a long enough time.

As soon as imc Messaging is no longer occupied with sending a message, the accumulated events are translated to message texts and entered into the sending queue. This queue's volume is also limited, so that messages can be lost if they are generated more rapidly than they can be sent.

How long a message must be kept in the queue depends on the one hand on the actual sending duration (a few seconds up to approx. 5min) and on the other hand on the receiver's accessibility, since the message is repeated several times if there is an error (see Delivery of the messages 789).

If the modem is busy, then all messages will be delayed until it's available. Thus in this case, the messages could stay in the queue indefinitely!

In this way, the transmitting is delayed and the queue's available space can be exhausted, for which reason no new messages may be able to be accommodated.

All lost messages are counted in the imc Messaging variable `AllMessageLosses`. In addition, the variable `ThisMessageLosses` records for each individual message configuration how often a message stemming from this configuration was lost.

10.2.3.2 Delivery of the messages

As soon as there is a message in the queue ready to be sent, an attempt is made to send it. If an attempt fails, the message is returned to the queue for a repeat attempt later following a pause.

This process is repeated several times, each time increasing the pause between delivery attempts. Once a certain number of repeat attempts have failed, the message is considered undeliverable and deleted. The imc Messaging variable `AllMessageLosses` and the corresponding instance of `ThisMessageLosses` are updated accordingly.

If the modem is busy, all messages are be held back until it's once again available. Therefore, an occupied modem won't lead to failed delivery!

If a message can't be delivered due to configuration errors such as a wrong password, then multiple attempts to deliver it may come at (substantial) monetary cost.

Note

Therefore it is strongly recommended to test every message configuration under authentic conditions. imc is not liable for any costs resulting from the use of imc Messaging!

10.2.3.3 Storing unsent messages

If a message must be discarded because there's no more room in the sending queue or because it can't be delivered, it is stored on a data carrier in the device, if possible. Even if the device is deactivated or if messages must be discarded due to preparation of a new measurement, storage in the device should still take place if at all feasible. Towards that end, the messages are saved even upon generating the message text and only deleted from the backup folder after being successfully sent.

If the device has an internal hard drive, it will be used for storage. Otherwise the system will check the device for the presence of a removable storage drive to which to save the messages.

For storage of the messages, a folder with the name "UnsentMessages" is created in the data carrier's main directory. All unsent messages are saved in this folder under filenames which provide indication of when and how the message was generated.

The filenames take the following form:

Date	Time	Number	Event	Event count	Type
2004-07-05	09-31-46	#4	Virt_Bit03=1	#2	EMAIL

Example: `2004-07-05 09-31-46 #4 Virt_Bit03=1 #2.EMAIL`

The first number corresponds to `AllMessageCounter`. It represents the incremental counter of messages since a measurement was last prepared. The second number is the message's `ThisMessageCounter`. It states how often the event occurred.

The filename extension indicates whether the message is an e-mail.

10.2.4 Configuration

The configuration of imc Messaging is accomplished by means of one ASCII-file per device, each has to be imported as supplemental file 212. A configuration file's name consists of the prefix "DEV", the three-digit device number (under imc STUDIO always "001") and the ending ".MSG". The filename is not sensitive to upper/ lower-case spelling.

Example: `DEV001.MSG`

The next chapter may create the impression that creating a configuration is a difficult process requiring specialized knowledge. In fact, however, obtaining the necessary information is very easy. Additionally, the chapter Templates 805 lists the necessary entries for a variety of providers, making it possible to perform the configuration through simple cutting and pasting.

10.2.4.1 Basics of configuration file structure and syntax

This section may give the impression that making a configuration is quite complicated, requiring special skills. However, the necessary information is actually easy to obtain. Additionally, from p. 11-24 Templates onward, the necessary specifications for a variety of service providers are presented, making it possible to accomplish the configuration by simple copying and pasting.

For each message, a separate segment must be entered in the configuration file defining all details such as the trigger event, message type, addressee and the text. A segment describes exactly one message! Even if an single event generates multiple messages, a corresponding number of separate segments must be created.

A segment is always prefaced by the designation of a bit, written inside square brackets. All subsequent lines are in reference to this bit, until a new segment begins.

Example: `[Virt_Bit04]`

Within a single segment, each line contains a keyword followed by an 'equals'-sign and the relevant value. The keyword must stand at the beginning of the line without any preceding spaces or tabs. There may not even be any other characters between the 'equals'-sign and the value! The values are entered without quotation marks.

The order of the lines isn't relevant. All keywords within a segment can be ordered in any way desired. However, it is recommended to arrange the keywords in convenient groups.

A line beginning with the character "#" is considered a comment line. Empty lines and comments are allowed at any location in the file and are ignored.

The following specifications must be present in every segment:

Parameter	Keyword	Possible values
Trigger event	`Event`	TransitionToOne, TransitionToZero
Message type	`MessageType`	EMAIL
Message destination	`Destination`	Telephone number or e-mail address
Message text	`Message`	Any text with variables. Special characters are not supported.

The keyword "`Event`" determines what event triggers the message. The value "`TransitionToOne`" stands for a rising edge or a bit's transition from 0 to 1. Accordingly, "`TransitionToZero`" means the reverse transition triggers a message.

Example: `Event=TransitionToOne`

"`MessageType`" states whether an "`EMAIL`" is to be sent. Depending on the message type selected, various additional specifications are needed.

Example: `MessageType=EMAIL`

The specification of the message recipient is made under the keyword "`Destination`". The recipient of an e-mail may be one or more e-mail addresses, separated by commas.

Example: `Destination=Hans.Mustermann@Firma.de, Fritz.Froh@Home.de`

The message text is entered line-by-line. Each line begins with the keyword "`Message`". The maximum number of lines and characters depends on the particular message type. A space can be inserted into the message by simply not specifying any text to follow the keyword. But the 'equals'-sign may not be omitted by any means.

Quotation marks are regarded as components of the message text and are sent unchanged to the recipient. As a rule, then, the text should be entered without any quotation marks.

Certain variables can be entered in the message text, which are replaced by updated values at runtime (see chapter Variables 796).

Example:

```
Message=This is the message's first line.
Message=This is the message's second line.
Message=
Message=This is the message's fourth line.
```

(Blank line!)

10.2.4.2 E-mail

E-mails are the only form of messages for which no modem may be necessary. If a mail server is available in the LAN, then only a few data are sufficient for sending an e-mail. On the other hand, if the e-mail is to be sent via modem, this requires entering a whole series of parameters in addition to the specifications already described in chapter Templates 805.

Parameter	Keyword	Possible values
Modem initialization	`InitScript*`	(see below)
Dial-in number of the Internetprovider	`ProviderPhoneNumber*`	Telephone number
Login name for Internet access	`ProviderLoginName*`	Login name for the account with the Internet provider
Password for Internet access	`ProviderPassword*`	Password for the account with the Internet provider
Mail server	`SMTPServer`	Name or IP address of mail server
Login protocol for the mail server	`SMTPLoginMethod*`	login, cram-md5
Login name for the mail server	`SMTPLoginName*`	Login name for the account with the Mail server
Password for the mail server	`SMTPPassword*`	Password for the account with the mail server
Port available [1]	`SMTPServerPort*`	Unencrypted: 25 STARTTLS: 587
Connection encryption [1]	`SMTPServerSecurity*`	STARTTLS
Sender address	`SenderAddress`	An e-mail address which is accepted by the mail server as a return address
E-mail's subject line	`Subject*`	Any text with variables

* optional; [1] only for devices with serial number 16xxxx and higher

InitScript

An "`InitScript`" is only needed in special cases. As a rule the script can be omitted entirely.

Entering a script can be necessary if the modem requires special preparation prior to the dialing procedure. Thus, it may be necessary to operate an ISDN modem with the protocol X.75 by default, and only to temporarily change to Sync-PPP for dialing in to the Internet provider.

Example: `InitScript="" ATE0 OK ATS96=16 OK`

To understand the script's syntax, it's easiest to ignore the quotation marks at first. Then, the script can be regarded as a sequence of commands sent by the device to the modem, each separated by a space from the response which the device expects from the modem. In the example, the modem should answer each command with "`OK`".

Since scripts are also used where, for example, the modem's status message "`RING`" is awaited so that the call can be accepted with "`ATA`", they always start with a string which the modem must send before the subsequent command is transferred to it.

In cases where commands transfer is to begin right away, two empty quotation marks are used to indicate that no status message from the modem is to be awaited.

ProviderPhoneNumber

If an e-mail is to be sent by modem, the device must log in with the Internet provider for that purpose. The provider's phone number is specified in conjunction with the keyword "`ProviderPhoneNumber`".

Example: `ProviderPhoneNumber=00193670`

ProviderLoginName and ProviderPassword

Internet access also requires a login name and password. The respective values are passed in conjunction with "`ProviderLoginName`" and "`ProviderPassword`".

Example:

```
ProviderLoginName=msn@easysurfer-power.de
ProviderPassword=msn
```

If an e-mail is sent via the LAN interface, the login and password for the Internet provider are skipped.

SMTPServer

The keyword "`SMTPServer`", by contrast, must always be present and determines which mailserver is used to send the e-mail. The mail server can either be specified by its IP address or by its host name.

Example:

```
SMTPServer=10.0.0.13
SMTPServer=mx.freenet.de
```

SMTPLoginMethod

To prevent abuse (spam), almost every mail server requires logging in with a user name and password. The servers carry out this identity check according to a variety of protocols, of which one that is supported by the selected server must be specified in conjunction with "*SMTPLoginMethod*".

Example:

```
SMTPLoginMethod=cram-md5
```

SMTPLoginName and SMTPPassword

When the mail server prompts for login, the login name and password are entered in conjunction with "`SMTPLoginName`" or "`SMTPPassword`", respectively. Otherwise, this step is skipped.

Example:

```
SMTPLoginName=MyLoginName
SMTPPassword=MyPassword
```

SMTPServerPort

Specifies the port. If the keyword is not specified, the following ports are used automatically:

Unencoded: 25

STARTTLS: 587

SMTPServerSecurity

Selection of the connection encryption. Only for devices of serial number 19xxxx and higher. The following encryption settings are possible:

STARTTLS: communication begins unencrypted, but immediately switches to encrypted mode. Prompting for the password and transfer of the E-Mail are performed with encryption.

SenderAddress

When sending an e-mail, a return address must always be given, otherwise delivery is denied. Some servers also check whether the e-mail address is even valid. In extreme case, the server only permits sending from return addresses in the same domain as the server itself, or which belong to the account into which you've already logged in. These steps are taken to prevent the misuse of mail servers.

The return address is denoted by the keyword "`SenderAddress`". In order to provide the e-mail client with a more recognizable designation of the sender, the actual return address can be preceded by an extra text. In that case, however, the true return address must be bracketed by the characters "<>".

Example:

```
SenderAddress=MyDevice@Firm.de
SenderAddress=TestRig No.2 <MyDevice@Firm.de>
```

In the second example, the e-mail client states the sender as "TestRig Nr.2". Note that the text of this extra, identification aid must conform to the required character set of the server. For instance, German umlauts may not be accepted. You may need to find out what letters the servers involved accept!

Certain PC programs solve this issue by replacing the umlauts with a sequence of control characters, which are correctly reverse-translated by the e-mail client's system to display to original letters. However, imc Messaging does not support this capability!

Subject

Stating content for a subject line in conjunction with the keyword "`Subject`" is optional. The subject has the same syntax as "`Message`" and may also contain variables. The same constraints in terms of umlauts and special characters apply as for "`SenderAddress`".

(see chapter Variables 796)

Example:

```
Subject=Limit violation in test Rig No.2
```

Destination

With e-mails, the recipient's address is given with "`Destination`". In contrast to all other messages, e-mails can be sent to multiple recipients. For this purpose, the various target addresses are separated by commas.

As with the return address, the recipient's address can be preceded by a descriptive text which appears in the e-mail client as the addressee. The recipient's actual address must then be bracketed between the characters "<>".

For umlauts and special characters, the same constraints apply as for "`Subject`" and "`SenderAddress`".

Example:

```
Destination=Hans.Mustermann@Firma.de
Destination=Administrator <Hans.Mustermann@Firma.de>
```

Example

In the simplest case, the e-mail is passed to a server in the local network which requires neither any login nor a valid return address. In that case, the configuration would be, for example, as follows:

```
# Send an e-mail if Ether_Bit09 changes to 1.
[Ether_Bit09]
Event=TransitionToOne
MessageType=EMAIL

# IP address of mail server
SMTPServer=10.0.0.13

# Sender: If the mail server doesn't check the address, a
# pretend address can be entered here. (no umlauts)
SenderAddress=MyDevice@NoWhere.de

# E-mail address of recipient (no umlauts)
Destination=Hans.Mustermann@Firma.de

# Subject line (no umlauts)
Subject=Notification from Test Rig No.2!

# An example of a message text.
Message=This is an e-mail generated by an imc measurement device.
```

Example A configuration having all specifications

```
# Send an e-mail if Ether_Bit32 changes to 0.
[Ether_Bit32]
Event=TransitionToZero
MessageType=EMAIL

# Set Synch-PPP as the protocol for the ISDN modem.
# Usually not necessary. The commands are modem-specific.
InitScript="" ATBS96=16 OK

# Data for Internet access
# 0 for dial tone
ProviderPhoneNumber=00193670
ProviderLoginName=msn@easysurfer-power.de
ProviderPassword=msn
# Data for mail server access
SMTPServer=mx.freenet.de
SMTPLoginMethod=login
SMTPLoginName=Hans.Mustermann@freenet.de
SMTPPassword=qt3TzW

# Return address: Checked for validity by Internet servers!
SenderAddress=My Device <Hans.Mustermann@freenet.de>

# E-mail addresses of recipient (no umlauts)
Destination=Hans.Mustermann@Firma.de, Fritz <Fritz.Froh@Firma.de>
# Status line (no umlauts)
Subject=Notification from Test Rig No.2!

# An example of a message text.
Message=This is an e-mail generated by an imc measurement device.
```

Example A configuration encrypted with STARTTLS

```
# Send an e-mail if Virtual_Bit01 changes to 1.
# e-mail without dialing in
[Virt_Bit01]
Event=TransitionToOne
MessageType=EMAIL

SMTPServer=smtp.gmx.net
#SMTPServerPort=587
SMTPLoginMethod=login
SMTPServerSecurity=STARTTLS
SMTPLoginName=MyName@gmx.de
SMTPPassword=MyPassword
SenderAddress=My device <MyName@gmx.de>
Destination=Info@MyCompany.de
Subject=Message from test bench {DeviceName}!
Message={DeviceName}   {DateTime, %Y-%m-%d %H:%M:%S}
Message=Hallo!
Message=Keep smiling!
```

10.2.4.3 Default configuration

It is possible to save a default imc Messaging configuration in the device. It will be used whenever no measurement has been prepared or whenever the current measurement does not contain any imc Messaging configuration.

A complete configuration including UDP Status Monitoring and various e-mail-messages can be specified.

For creating, editing and deleting the default configuration, the XmlRpc-method "`SetimcMessagingDefaultConfig`" has been implemented. This command is automatically called when files are exchanged. By this means it is possible to set the default configuration solely by using FTP. For details, see the section "*Configure via FTP* 809".

10.2.4.4 Variables

The message texts and the subject lines of an e-mail can contain both permanent text portions and also variables which are replaced by updated values when the message is generated. A line may contain any desired number of embedded variables. The only limitations are the configuration file's maximum line length and the particular constraints affecting the message type concerned.

The following **variables** are available for **imc Messaging** 786 and **UDP Status Monitoring** 801:

Variable	Description
`AllMessageCounter`	Total count of messages generate. Format numerical 798.
`AllMessageLosses`	Total count of messages which were discarded because undeliverable or due to lack of resources. Format numerical 798.
`DateTime`	Date and time in any format 800.
`DeviceName`	User-defined name for the device.
`DirCounter`	Counts all closed folders which contain measured data and which were created since the last preparation procedure. Folders which contained no measured data upon being closed and which were consequently deleted are not counted. Format numerical 798.
`DisplayVar_01-32`	the display variable's current contents Format numerical 798.
`Ether_Bit01-32`	the ethernet bit's current state Format numerical 798.
`EventOverruns`	total number of events which couldn't be processed because the came too quickly in succession. Format numerical 798.

Variable	Description
`MeasurementStatus`	Indicates one of the following device states: 1: No measurement was started. It is not possible to distinguish whether a measurement has been prepared. 2: A timed start was set up, but the start time has not yet been reached. 3: A measurement has started and data capture is running. It is not possible to see whether a trigger has already been released. Format numerical 798.
`SerialNumber`	Returns the device's serial number as a text. For this variable, no formatting can be specified.
`SoftwareDateTime`	Returns the day on which the software was created. For formatting the output, the same formatting instructions are available as for the variable DateTime. If output of the time is demanded in the formatting instructions, then 0 is always returned, since the time is not available.
`ThisMessageCounter`	Placeholder in message text for the number of times this message was generated. Format numerical 798.
`ThisMessageLosses`	Placeholder in message text for the number of times a generated message of this kind was lost. Format numerical 798.
`TimeOfFirstStart`	Contains the starting time of the first measurement following the last preparation procedure in any format 800.
`Virt_Bit01-32`	the virtual bit's current state. Format numerical 798.

The values of the bits and Display variables, as well as of the date and time, are saved immediately when it is determined that a message must be generated. Even if the message can only be generated later, its text will reflect the values of the variables at the instant in time when the message was triggered.

By contrast, imc Messaging's counter variables are replaced with updated values only when the message text is actually generated.

The count begins when the message is prepared. Even if the measurement is stopped and restarted, this doesn't cause the counter to be reset! Only upon a measurements preparation are the counters all reset to zero.

All numerical variables can be expressed both in integer and floating-point format. If a Display variable is formatted as an integer, it will be rounded up for a decimal value of 0.5 and above, otherwise it's rounded down.

Since the bits can only take the values 0 or 1, only integer format makes sense for them. The counter variables' input range is 32-bit. Thus it's possible to tally up about $4*10^9$ events before a counter overflows and returns to zero. Nevertheless, these variables should preferably be formatted as integers.

But due to their somewhat greater complexity, the variables "`ThisMessageCounter`" and "`ThisMessageLosses`" are described in more detail here.

There are equally many instances of these variables as there are message configurations. A message configuration is always able to access only its own counters. Thus, it isn't possible to send a message stating the state of another message configuration's lost message counter.

If these counters are used in a message, it's possible to tell immediately how often the event concerned has already occurred, and how many messages generated in response were lost.

In contrast, the variables "`AllMessageCounter`" and "`AllMessageLosses`" pertain to every message and can be used for global monitoring of imc Messaging.

The a new variable **`DirCounter`**, which counts all closed folders which contain measured data and which were created since the last preparation procedure. Folders which contain no measured data upon being closed and which were consequently deleted are not counted!

The purpose of this new variable **`TimeOfFirstStart`** is to determine whether the device has undergone a new preparation procedure since the last session (Autostart, Suspend/Resume). This indicates whether the **`DirCounter`** is still increasing monotonously, or whether it has been reset since the device's last message. Otherwise, the receiver would not be able to distinguish whether, for instance, a counter reading of 3 on the **`DirCounter`** still refers to the last reading which the receiver was aware of, or whether the device has been restarted in the meantime and that there are 3 new closed folders ready to be retrieved. This case would otherwise only be recognized when the counter reads 4, since that is the next time when a connection to the device would be made – and the memory's exact state would be checked. The resulting delay could become intolerably long, depending on the circumstances.

Note that the variable contains the value 01.01.1900 00:00 immediately upon preparation! Only when the moment for starting the measurement is known, the variable's contents change accordingly.

For formatting the output, the same format instructions are available as for the variable `DateTime`.

10.2.4.5 Syntax

A variable is distinguishable in the text by appearing inside of curly brackets. The brackets contain the variables' pre-defined names, followed by a formatting instruction. For clearer visibility, commas and spaces can be inserted after the variable names.

The device's name ("`DeviceName`") is an exception. This doesn't need any formatting instruction because it is always outputted as a string. With the other variables, one distinguishes among simple numerical values and statements of the date and time.

Formatting of numerical variables

The formatting instructions always begin with a percent-symbol, followed, in the simplest case, by a single letter. This letter determines the numerical format as per the table below:

Format type	Definition
%d or %i	signed integer in decimal format
%u	unsigned integer in decimal format
%o	integer in octal format
%x	integer in hexadecimal format (0-9, a-f) without prefix (0x)
%X	integer in hexadecimal format (0-9, A-F) with prefix (0X)
%f	real number in fixed-point format
%e	real number in exponential format (e.g. 1.0e+4)
%E	real number in exponential format (e.g. 1.0E+4)
%g	like %f or %e, whichever is shorter
%G	like %f or %E, whichever is shorter

The complete syntax of the format specification takes the following form:

%	modifier	minimum width	.	Precision	Format type
necessary	optional	optional	optional	optional	necessary

Modifier:

Modifiers can be combined wherever it makes sense. They affect the output as follows:

Modifier	Effect	
-	left justified (default is right justified)	
0	output is zero-padded instead of filled out with spaces	
+	a plus-sign precedes positive numbers	
space	a space precedes positive numbers	
#	depends on the format type:	
	%#o (octal)	output prefixed with a zero
	%#x (hexadecimal)	output prefixed with "0x"
	%#X (hexadecimal)	output prefixed with "0X"
	%#f,%#e,%#E,%g,%#G	ensures output of a decimal point

Minimum width:

By default, as many characters are outputted as are necessary to display the value. For a formatted output it may be desirable to set a minimum width. The output is then filled up with spaces to achieve this width. The modifier "0" causes the output to be padded with zeroes instead of spaces.

Example:	`%d`	Value = 734	Output = "734"
	`%10d`	Value = 734	Output = " 734"
	`%010d`	Value = 734	Output = "0000000734"

Precision:

If a real number is to be rounded to a certain number of decimal places, a period followed by the number of places (digits) must be entered.

Example:	`%f`	Value = 734.458	Output = "734.458"
	`%.1f`	Value = 734.458	Output = "734.5"
	`%8.2f`	Value = 734.458	Output = "734.46"

In the third example, the output is intended to have a width of at least 8 characters, including the decimal point.

Example:

```
Message=The temperature is {DisplayVar_01, %5.1f}°C.
Message=Message Number {AllMessageCounter, %d}
```

Sample result: The temperature is 23.4°C.

Message Number 4

Formatted output of date and time

In the message texts, the date and time can be inserted in practically any format imaginable. Specifying the formatting of the date and time is accomplished in a totally different manner from that of numerical values.

It is possible to specify multiple formats for a single variable of the type "`DateTime`", so that the variable is replaced by that many different statements of the time. Each formatting specification consists of a percent-symbol followed by a letter which stands for the desired time representation. In the message text design, the formatting specification can be embedded among other text elements as desired (see examples below).

The following time representations are available:

Formatting specification	Definition
%d	day (01 – 31)
%e	day (1 – 31) with preceding space if single-digit
%m	month (01 – 12)
%y	two-digit year (00 – 99)
%Y	four-digit year
%G	four-digit year in conformance to the calendar week as per %V
%g	like %G but two-digit
%C	two-digit century
%F	as per %Y-%m-%d (ISO 8601 date format)
%D	as per %m/%d/%y (American date format)
%j	ordinal day of the year (001 – 366)
%u	weekday (1 – 7, Monday is 1)
%w	weekday (0 – 6, Sunday is 0)
%W	calendar week (00 – 53) beginning on first Monday
%U	calendar week (00 – 53) beginning on first Sunday
%V	ISO 8601:1988 calendar week (01 – 53) beginning with the first week having 4 days. The weeks begin on Monday
%H	hour (00 – 23)
%k	hour (0 to 23) with preceding space if single-digit.
%I	hour (01 – 12)
%l	hour (1 – 12) with preceding space if single-digit.
%M	minute (00 – 59)
%S	seconds (00 – 61) 60 & 61 only appear in special cases
%P	am/pm 12:00; noon is "pm", midnight "am"
%p	AM/PM 12:00; noon is "PM", midnight "AM"
%R	corresponds to %H:%M
%T	corresponds to %H:%M:%S
%r	corresponds to %I:%M:%S %p (American time representation)

The variable "%t" adds a tab stop to the output, "%n" causes a line break.

Example

```
Message=Date: {DateTime, %Y-%m-%d} Time: {DateTime, %H:%M:%S}
Message={DateTime, Year: %Y Month: %m Day:%d}
```

Example result:

Date: 2004-07-28 Time: 16:15:32

Year: 2004 Month: 07 Day:28

10.2.5 UDP status monitoring

UDP Status Monitoring has been developed to create data traffic within a certain time interval. That is necessary to hold a modem line, that would be interrupted by the provider, if there is no data transfer.

Therefore an entry in the messaging file "Devxxx.msg" has to be entered. This file must be imported as a supplemental file 212 into the experiment.

Example

```
[UDP Status Monitoring]
DestinationIP=x.x.x.x
DestinationPort=XXXX
Interval=x in seconds
NoLineFeed
```

```
[UDP Status Monitoring]
DestinationIP=10.0.2.7
DestinationPort=1205
Interval=10
NoLineFeed
```

As the target of an UDP Status Monitoring data package, it is now possible to specify either an IP address (`DestinationIP`) or a DNS-name (`Destination`) with the keyword Destination.

This is very useful if the target computer has no fixed IP address, but it does have a DynDNS name. The old keyword DestinationIP continues to be supported and works in the same way as always.

Note that the device requires a DNS-Server in order to convert the DNS name into an IP address! This must be announced to the device either via "*Interface-Configuration*", DHCP or, when dialing into the internet, by the provider server.

The name is converted every time a package is to be sent, because it is impossible to anticipate whether the target's IP address has changed. Therefore, it is sensible to provide a permanent IP address directly!

UDP Status Monitoring now supports all imc Messaging variables 796 with the exception of **`ThisMessageCounter`** and **`ThisMessageLosses`**. Due to the fixed time grid and the unsecured transmission without failure notification, these variables are not very useful. All other imc Messaging variables, particularly the two new variables mentioned above, are accepted by UDP Status Monitoring in the exact same way as by all other message types. However, the message text from UDP Status Monitoring is limited to 300 characters, as before. See also example here 802.

See also: "*Configure via FTP* 809"

10.2.5.1 Syntax / example / limitations

`[UDP Status Monitoring]`	Keyword for section UDP Status Monitoring
`Destination` or `DestinationIP`	DynDNS-Name or IP address for the destination PC, e.g. `DestinationIP= 192.168.0.1`
`DestinationPort`	Destination port that receive the data, e.g.`DestinationPort=9`. Port 9 is a reserved port of the PC. There the data are discarded and should work without problems.
`Interval`	Time interval in seconds, e.g. Interval=240 (every 4 minutes). Possible intervals are 1s to 4000000s (recommend: at least 10s)
`NoLineFeed`	If this command is entered, no automatic line breaks are inserted after the messages ("Message=...").

Afterwards, data can be sent. The message is divided by line feed, unless prevented by "`NoLineFeed`". The device name can be transferred by {Devicename}.

Example

```
[UDP Status Monitoring]
Destination=MyDNS_Name
DestinationPort=5000
Interval=10
Message={Devicename}
Message=This could be any text
Message=Last start {TimeOfFirstStart, %Y-%m-%d %H:%M:%S}
Message=Directories number = {DirCounter, %u}
Message=Temperature = {DisplayVar_01, %5.2f}
Message=State of measurement = {MeasurementStatus, %d}
```

Note: If a fixed IP is used instead of a dynamic DNS name, the line `Destination=MyDNS_Name` must be replaced with `DestinationIP=192.168.0.1`.

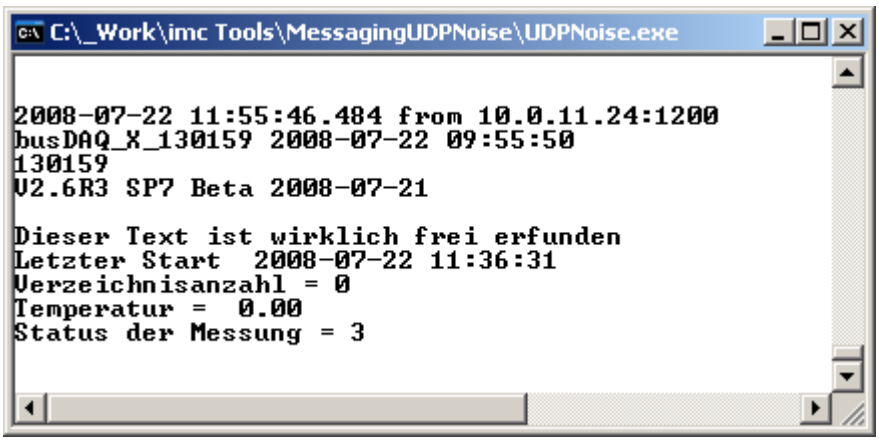
Find the variables and their format in chapter Variables 796.

Limitations

- The maximum pack of data is 300Bytes at max. Greater packages will be cut.
- Only one UDP Status Monitoring configuration can be defined per device.
- UDP Status Monitoring and imc Messaging has to be defined in one msg-file. It is possible to create UDP Status Monitoring packages in fixed intervals and conditionally send e-mails.
- The source port of the UDP Status Monitoring is always 1200.

10.2.5.2 Test of UDP status monitoring

For testing the UDP Status Monitoring, the program "*UDP Status Monitoring.exe*" is provided (included in the installation medium).



Messages arriving via UDP

10.2.6 WakeOnLAN (Magic Packets)

The entry *WakeOnLAN* enables automated starting of a computer when an imc measurement device is switched on. For this purpose, the section [WakeOnLAN] can be added in the Devxxx.msg file 790. The condition for this is that the target-PCs are configured for WakeOnLAN in the BIOS and under Windows!

The following parameters are necessary:

DestinationIP: The **Subnet-Broadcast-address**, e.g. 10.0.*255.255* or the **limited Broadcast address** *255.255.255.255.* When there is only one **target PC in Suspend mode**, it is also possible to use its IP address.

DestinationMAC: The MAC address(es) of the computer(s) to be woken up, the **maximum** count is **20** MAC addresses.

RepeatCount: The number of repetitions which the imc device performs after being activated. The repetitions are performed at a intervals of 3 seconds and are limited to a maximum of 200. In each repetition, three packets are sent at intervals of 300ms.

Example

```
[WakeOnLAN]
DestinationIP=10.0.255.255
DestinationMAC=00:0a:33:fd:41:01
DestinationMAC=00:0a:33:fd:41:02
DestinationMAC=00:0a:33:fd:41:03
DestinationMAC=00:0a:33:fd:41:04
RepeatCount=100
```

After the initialization (the self-starting), a series of 3 Wake On LAN packets are sent 100 times at intervals of 3 seconds. The 3 packets repetitions are sent at intervals of ca. 300ms. Up to 20 Mac-addresses can be specified.

In the example, four MagicPakets are sent with the respective MAC-addresses. After ca. 300ms, they are repeated. This process is repeated 100 times at intervals of 3s.

10.2.7 Possible error sources

To avoid problems arising in actual practice, you absolutely should test your message configurations!

Once you have checked your accessing data for an Internet provider or mail server, all you should still need to do is to copy and paste them into the configuration file. This way, you can avoid typing errors. The same applies to the phone numbers and e-mail addresses of the recipients.

Below are presented some familiar problems, their possible causes and the associated remedies:

Problem: The device doesn't send a message for every change of the bit. Only isolated messages are generated or none at all.

Cause: Either the bit changes too quickly, so that the change isn't recognized, or the sending queue is full.

Remedy: Ensure that the bits' states remain stabile long enough to be recognized reliably. Don't let too many messages be generated in rapid succession, and make sure that the messages can be sent without any problems.

Problem: The variables aren't replaced by updated values in the message. The line does appear as it was entered in the configuration file.

Cause: There is probably a syntax error.

Remedy: Check the spelling of the variables and check the formatting instructions for errors.

Problem: Having configured messages for ethernet bits, these messages are now constantly triggered although the bits can't have changed.

Cause: The ethernet bits are affected by every device in the network. It may happen that a device in use by someone else has changed some ethernet bit(s).

Remedy: Construct a separate network, or use the virtual bits.

Problem: When working with multiple devices, the first messages triggered by the ethernet bits aren't sent.

Cause: If you set a bit by means of imc Online FAMOS already during preparation of a measurement, it may not be recognized by all of the devices, since some of the devices may not have been prepared.

Remedy: Make sure that you don't make any changes to ethernet bits before concluding preparations for all of the devices.

10.2.8 Technical specifications imc Messaging

The following data reflect the state in effect in August, 2004. imc reserves the right to change the data in response to further developments.

Parameter	min. / max.
Event duration	min. 50ms; > 200ms preferred setting
Delay between occurrence of event and saving of updated value of variable	min. 20ms, max. approx. 200ms
Line length of the configuration file	max. 512 characters
Size of a configuration file	max. 300000 characters
Number of message configurations	max. 128 per device
Size of event queue	128 entries
Size of sending queue	128 messages
Number of sending attempts before message loss	30
Interval between successive sending attempts	1,4,9,16,25,36,49,60,60, ... 60min
Duration of attempt to dial in to Internet	maximum permitted: 60s
E-mail sending duration	maximum permitted: 80s
Number of characters in an e-mail	limited only by size of configuration file
Input ranges	
Bit variables	0 or 1
Counter variables	32 bits unsigned (0 to 4294967295)
Display variables	32 bits floating point (-3.4e+38 to +3.4e+38)

10.2.9 Templates

In order to keep the imc Messaging configuration as simple as possible, the following templates are offered, which contain the relevant access codes for a variety of service providers. You can adopt these data as appropriate in your configuration files. The examples refer to German providers. For access codes outside of Germany, inquire with the pertinent service providers.

Internet provider

These are some examples for internet providers. Check the WEB for further information of the supplier. **No responsibility is taken for the correctness of this information!**

Call by Call Internet access from land lines (ISDN and analog)

MSN (Easysurfer power)

```
ProviderPhoneNumber=0193670
ProviderLoginName=msn@easysurfer-power.de
ProviderPassword=msn
```

Call by Call Internet access via GSM

Inquire with your GSM provider as to the least expensive Internet access.

D1 T-Com Inland (Registration required!)

```
ProviderPhoneNumber=4122
ProviderLoginName=(assigned upon registration)
ProviderPassword=(assigned upon registration)
```

D1 and D2 Domestic (Germany): Freenet Call by Call

```
Per-minute price: 20 Cent + any fees for special numbers (debitel)
ProviderPhoneNumber=22243
ProviderLoginName=freenet
ProviderPassword=mobil
```

D1 Abroad (outside Germany): Freenet Call by Call (not available within Germany)

```
Costs: depends on provider
ProviderPhoneNumber=+491712522224 (substitute the appropriate country prefix for +49!)
ProviderLoginName=freenet
ProviderPassword=mobil
```

Free mail servers

You must apply for an e-mail address with a provider, in order to obtain a login name. This can be done via the provider's website.

GMAIL

```
#For GMAIL, "Less secure app access" must be enabled in the security settings.
MessageType=EMAIL
SMTPServer=smtp.gmail.com
SMTPServerPort=587
SMTPLoginMethod=login
SMTPServerSecurity=STARTTLS
SMTPLoginName="your login name with this provider"
SMTPPassword="your password with this provider"
SenderAddress="your e-mail address with this provider"
Destination=name@provider.de
Subject=My test message
Message=Hello!
#all sent e-mails are saved automatically!
```

GMX

```
MessageType=EMAIL
SMTPServer=mail.gmx.net
SMTPServerPort=587
SMTPLoginMethod=login
SMTPServerSecurity=StartTLS
SMTPLoginName="your login name with this provider"
SMTPPassword="your password with this provider"
SenderAddress="your e-mail address with this provider"
```

WEB.DE

```
MessageType=EMAIL
SMTPServer=smtp.web.de
SMTPServerPort=587
SMTPLoginMethod=login
SMTPServerSecurity=StartTLS
SMTPLoginName=user #without @web.de
SMTPPassword="your password with this provider"
SenderAddress="your e-mail address with this provider"
```

Smart-Mail (unencrypted)

```
MessageType=EMAIL
SMTPServer=smtp.smart-mail.de
SMTPLoginMethod=login
SMTPLoginName="your login name with this provider"
SMTPPassword="your password with this provider"
SenderAddress="your e-mail address with this provider"
```

Freenet

```
MessageType=EMAIL
SMTPServer=mx.freenet.de
SMTPServerPort=587
SMTPLoginMethod=login
SMTPServerSecurity=StartTLS
SMTPLoginName="your login name with this provider"
SMTPPassword="your password with this provider"
SenderAddress="your e-mail address with this provider"
```

10.2.10 Step by step

10.2.10.1 Setup connection

Reference

See Manual: "*Setting Up - Connect the device*" > "*Special options for connecting to the device*"

- "*Device connection via WLAN* 59"
- "*Device connection via LAN* 55"

10.2.10.2 Sending an e-mail

An e-mail will be sent, if virtual bit 01 will be set from 0 to 1. Replace the entries with blue background.

Save the following text as a text file that is called DEV001.msg. This file must be imported as a supplemental file 212 into the experiment.

For this example, the device must be connected to the internet via LAN.

Example

```
# Send an e-mail, if Virtual_Bit01 is set to 1
[Virt_Bit01]
Event=TransitionToOne
MessageType=EMAIL

# IP address of mailserver
SMTPServer=mx.freenet.de
SMTPLoginMethod=cram-md5
SMTPServerSecurity=StartTLS
SMTPLoginName=TestMail@freenet.de
SMTPPassword=MyPassword

# Sender: If the mailserver does not check the destination, it can be used as imaginary
address.
SenderAddress=My device <TestMail@freenet.de>

# E-mail addresse of recipient
Destination=john.smith@imc-tm.de

# reference line
Subject=This mail is send by an imc device {DeviceName}

# Example for message text
Message=The temperature is {DisplayVar_01, %3.1f}°C
```

10.3 Configure via FTP

It is possible to transfer any desired files to the device via FTP. This method can be used to write configuration files to the device. The device is configurable via FTP.

Warning

- When overwriting a file via FTP, any interruption of the transmission can cause the file to become invalid/damaged. If the file affected is the autostart configuration, the device will from that time on always attempt to carry out an autostart with the invalid file, each time the device is activated. It is even possible that the device will crash every time this happens. As a result, the devices may no longer be remotely accessible and may require repair.
- For this reason, **we strongly recommend that the autostart configuration(s) never be overwritten via an insecure (FTP) connection!**
- **imc STUDIO can not connect to the measurement!**

XmlRpcCmd

For configuration of devices via FTP, a **command interface** has been implemented. The command is called by **transmitting a text file to the root of the internal disk** which contains the command call incl. the associated parameters (xmlrpc method call, file: "xmlrpccmd.call").

A new configuration is transmitted as a separate file.

After the command is run, the result is saved in the form of a text file (xmlrpc response, file: "xmlrpccmd.response") which can be retrieved. To ensure the integrity of these files, their validity is confirmed by the creation of new files of length 0: "xmlrpccmd.call.valid" or respectively "xmlrpccmd.response.valid".

Possible configurations

1.Reconfiguring and starting the measurement

The ***autostart configuration*** will be started. Without autostart configuration, the device will be started but without recording.

2. Reconfiguring and starting the measurement with the specified configuration

The ***configuration must be present on the device***.The configuration may have been overwritten via FTP or newly created. A diskstart configuration (ume.zip) or autostart configuration (ums.zip) may be specified.

3. Reconfiguring and starting the measurement with the transmitted configuration

The new configuration is transmitted to the device via FTP. However, the configuration is only overwritten when the command is run in the device itself. This means that even a configuration in the internal Flash can be altered! An interrupted connection has no effect and the security is significantly increased.

This is the method recommended for altering an autostart configuration!

Constraints:

- The experiment to be altered must already exist.
- An autostart configuration (ums.zip) can only be overwritten with an autostart configuration.
- A diskstart configuration (ume.zip) can not be overwritten with an diskstart configuration.

10.3.1 Procedure

All file actions on the device are located directly in the ***root folder of the internal disk***.

1. User:

- If necessary,
  - delete from the device the files "*xmlrpccmd.call", "xmlrpccmd.call.valid", "xmlrpccmd.response"* and *"xmlrpccmd.response.valid".*
  - copy the new *configuration file* to the internal data storage medium.
- Copy the file
  - "*xmlrpccmd.call*". This file must contain a valid command description (see examples supplied).
  - "*xmlrpccmd.call.valid*". This file should be empty. It only serves to confirm that "*xmlrpccmd.call*" has been copied successfully.

2. Device:

- The files "*xmlrpccmd.call"* and *"xmlrpccmd.call.valid*" are read and subsequently deleted .
- If necessary, the new configuration file is copied to the target folder and deleted.
- The devices configures and starts the experiment.
- The device creates the file "*xmlrpccmd.response*", in which the results of the command is returned.
- To indicate that "*xmlrpccmd.response*" is valid, the file "*xmlrpccmd.response.valid*" is created.

3. User:

- Wait for "*xmlrpccmd.response.valid*". Provide a timeout! At least 1 min, but preferably 3 min., depending on the connection speed and quality.
- Copy "*xmlrpccmd.response*" for evaluation on the PC.
- Delete the files "*xmlrpccmd.response*" and "*xmlrpccmd.response.valid*" from the device.
- In case of a timeout, delete any files no longer needed.

10.3.2 Example of a configuration via FTP

The following example shows how an "*imcMessaging default configuration*" is activated. At the same time, the device is instructed to perform cyclical transmission of information via UDP Status Monitoring.

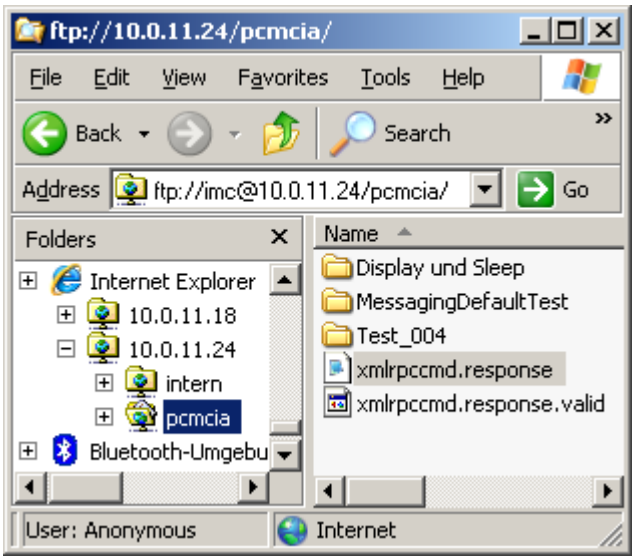
1. First create a small "autostart" experiment.

In the example, an e-mail is sent when virtual bit 1 changes from 0 to 1. Therefore, Virtual Bit 1 must be set while the experiment is running. The performance of this event can be automated with imc Online FAMOS, for example, using a timer or a ramp function. Save this experiment to the device as a self-activation configuration.

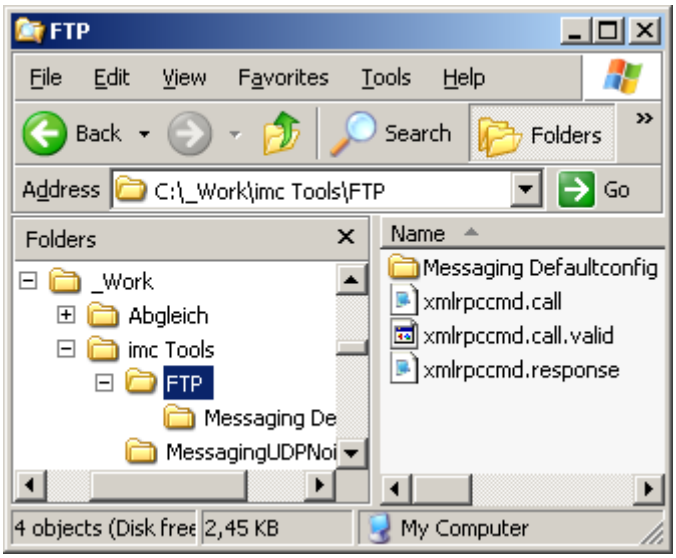
2. Open two instances of your Explorers.

One window shows your device. To do this, enter "ftp://" and the device's IP address, e.g. "ftp://imc@10.0.11.24/pcmcia/".

The second window shows your local folder in which the files to be transmitted are located.



The measurement device's data carriers



Source directory of the FTP configuration files

3. Create a text file "xmlrpccmd.call" and copy the following sample text to this file:

The segments of code appearing below in boldface must be adapted. The comments, appearing in green, are to be deleted subsequently.

File name: "*xmlrpccmd.call*":

```
<?xml version="1.0"?>
<methodCall>
   <methodName>SetimcMessagingDefaultConfig</methodName>
   <params>
     <param>
       <value>
         <struct>
           <member>
             <name>RequestID</name>
             <value>2008-06-02 13:29:38</value>
           </member>
           <member>
             <name>Configuration</name>
<value>[UDP Status Monitoring]
# The IP address of the destination PC
DestinationIP=15.0.1.17 ; here, enter your PC's target address.
# The UDP port to send the data to
# Port 9 is the "discard" port and should do no harm to the PC
DestinationPort=9 ; here. enter the port via which your computer receives UDP.
# The interval in seconds (e.g. send every 4min)
Interval=10 ; interval in seconds. Here, enter how often to send a UDP message.

# The data to send:
Message={DeviceName} {DateTime, %Y-%m-%d %H:%M:%S}
Message={SerialNumber}
Message={SoftwareVersion} {SoftwareDateTime, %Y-%m-%d}
Message={MeasurementStatus, %u}

[Virt_Bit01]
Event=TransitionToOne
MessageType=EMAIL
; here, enter your mail server's address
SMTPServer=15.0.0.19
; origin address of the e-mail sent. It is not applied and is merely for your own information.
SenderAddress=CS-7008SN122399@imc-tm.de
; e-mail address to which the device sends the message generated.
Destination=Max.Mustermann@imc-tm.de
Subject=Message from device {DeviceName}!
Message={DeviceName} {DateTime, %Y-%m-%d %H:%M:%S}
Message={SerialNumber}
Message={SoftwareVersion} {SoftwareDateTime, %Y-%m-%d}
Message={MeasurementStatus, %u}
</value>
           </member>
         </struct>
       </value>
     </param>
   </params>
</methodCall>
```

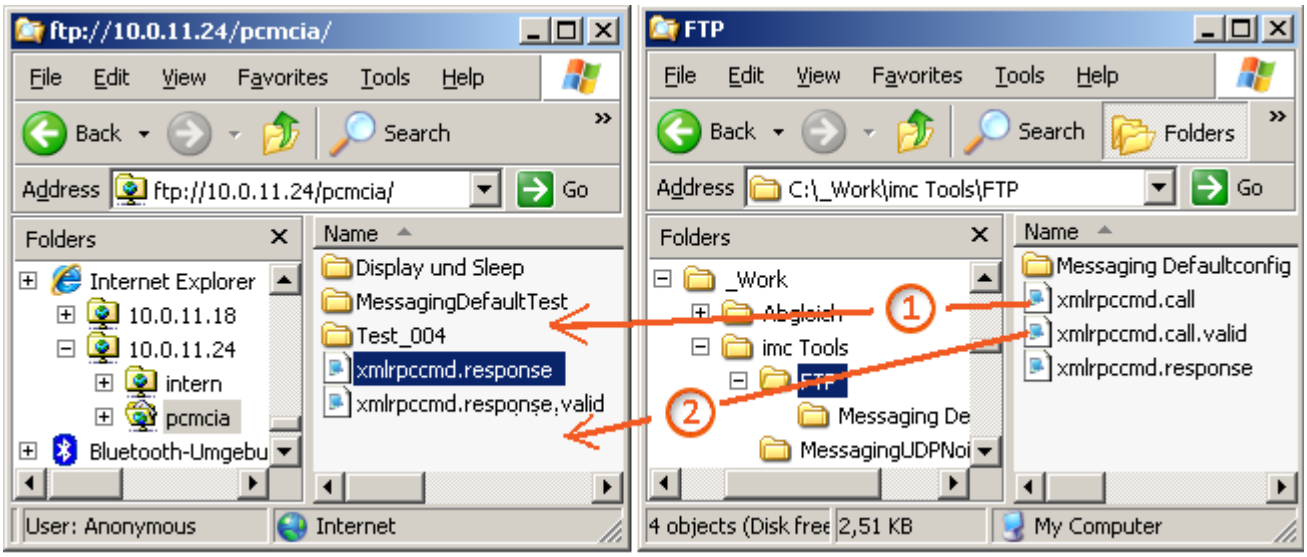
Response file name: "*xmlrpccmd.response*":

```
<?xml version="1.0"?>
<methodResponse>
   <params>
     <param>
       <value>
         <struct>
           <member>
             <name>RequestID</name>
             <value>2008-06-02 13:29:38</value>
           </member>
           <member>
             <name>Result</name>
             <value>
               <i4>0</i4>
             </value>
           </member>
         </struct>
       </value>
     </param>
   </params>
</methodResponse>
```

4. Create a text file with the name xmlrpccmd.call.valid.

This file has no content!

5. Next, copy the file xmlrpccmd.call and then the file xmlrpccmd.call.valid to the device folder.



Transfer of the configuration files

6. Wait a few seconds and then update the Explorer [F5].

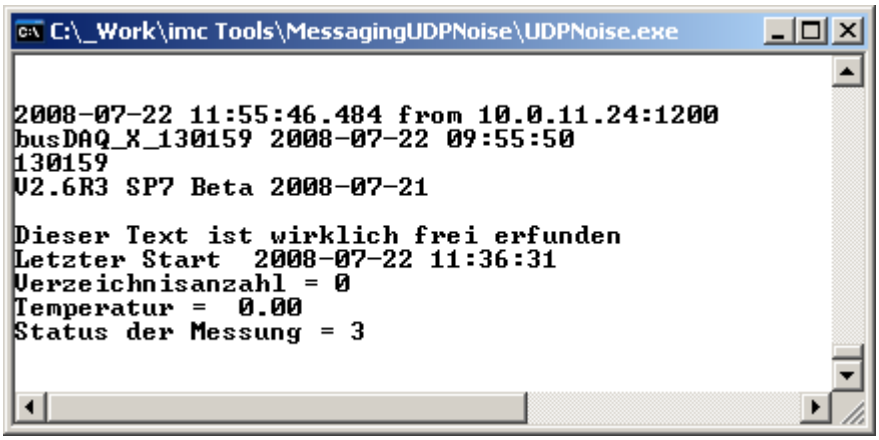
Now the copied files disappear and **xmlrpccmd.response** and **xmlrpccmd.response.valid** appear. The response file can be read with a text editor, and its return value is 0 if everything worked. Otherwise, an error code is returned. In this case, check your xmlrpccmd.call file again. If you don't find any errors, please contact our tech support.

```
...
- <member>
<name>Result</name>
- <value>
  <i4>0</i4>                    <---- here you find the return value
</value>...
```

7. The changes only take effect upon the next preparation procedure.

Cause a new preparation process by restarting, either by re-activation or by wakening from the SleepMode (only for busDAQ).

Check whether the e-mails arrive. The UDP messages can be checked using the program *UDP Status Monitoring.exe*.



Messages arriving via UDP

10.3.3 Syntax of command call and response

The following commands are implemented:

XmlRpcCmdReconfigure Reconfiguring and starting the measurement.

The following parameters can be specified in the command call (file: "*xmlrpccmd.call*"):

Parameter	Description
SourceDrive	• "*Removable*" -> CF-Card, ... • "*Nonremovable*" -> IDE or CF-Adapter
SourceFile	"*devXXX.ums.zip*" or "*devXXX.ume.zip*", XXX refers to the device number.
DestinationDrive	Which storage is addressed: • "*Removable*" -> CF-Card, ... • "*Nonremovable*" -> IDE or CF-Adapter • "*Internal*" -> Flashdisk of the interface
Experiment	Experiment name
RequestID (optional):	An arbitrary string, copied to "*xmlrpccmd.response*" by the device. If appropriate, unique strings are used, it is possible to recognize unambiguously, whether "*xmlrpccmd.response*" belongs to the current command.

The response (file: "*xmlrpccmd.response*") contains the following values:

Parameter	Description
Result	The imc STUDIO error code, or respectively 0, if no error occurred.
RequestID	The string from "*xmlrpccmd.call*", or empty, if no RequestID was specified.

SetimcMessagingDefaultConfig

Default configuration for imc Messaging 796

The following points must be observed:

1. The command "*SetimcMessagingDefaultConfig*" does not immediately reconfigure the device. The changes only take effect the next time a measurement is prepared!
2. The return value only indicates whether the command was able to be executed properly. There is no syntax check of the configuration supplied!

Parameter	Description
RequestID (optional)	An arbitrary string adopted by the device in "*xmlrpccmd.response*". If appropriate, non-repeating strings are used, it is possible to detect with certainty whether "*xmlrpccmd.response*" belongs to the current command.
Configuration	The new default configuration for the imcMessaging as an XML-encoded string. It is important for the configuration to be XML-encoded, since this means that some characters are replaced by Escape sequences which would otherwise lead to syntax errors. Within the configuration, the line formatting must be retained and indents must be avoided, since they would either lead to syntax errors or to corresponding extra indents in the messages. If an empty string is submitted as a configuration, the default configuration is deleted!

The response (file: "*xmlrpccmd.response*") contains the following values:

Parameter	Description
iRet	The imc STUDIO error code or 0, if no error has occurred.
Result	Identical to iRet, still retained for compatibility purposes
RequestID	The string from "*xmlrpccmd.call*", or empty, if no RequestID has been specified.

GetSoftwareVersion

Parameter	Description
RequestID (optional)	An arbitrary string adopted by the device in "*xmlrpccmd.response*". If appropriate, non-repeating strings are used, it is possible to detect with certainty whether "*xmlrpccmd.response*" belongs to the current command.

The response (file: "*xmlrpccmd.response*") contains the following values:

Parameter	Description
iRet	The imc STUDIO error code or 0, if no error has occurred.
Result	Identical to iRet, still retained for compatibility purposes
RequestID	The string from "*xmlrpccmd.call*", or empty, if no RequestID has been specified.
SoftwareVersion	The software version of the device as a string on the format "*Version 2.6R3 SP7 Beta (31.5.2008)*".

Note

- The **capitalization** of the parameter names is relevant!
- To see the exact syntax, observe the following examples.

10.3.4 Templates

Use the following templates and complete the necessary information. The explanation of the commands can be accessed via a link.

10.3.4.1 Reconfiguring and starting the measurement

File name: "*xmlrpccmd.call*":

```
<?xml version="1.0"?>
<methodCall>
   <methodName>XmlRpcCmdReconfigure 814</methodName>
</methodCall>
```

Response file name: "*xmlrpccmd.response*":

```
<?xml version="1.0"?>
<methodResponse>
   <params>
     <param>
       <value>
         <struct>
           <member>
             <name>RequestID</name>
             <value/>
           </member>
           <member>
             <name>Result</name>
             <value>
               <i4>0</i4>
             </value>
           </member>
         </struct>
       </value>
     </param>
   </params>
</methodResponse>
```

10.3.4.2 Reconfiguring and starting the measurement with the specified configuration

File name: "*xmlrpccmd.call*":

```
<?xml version="1.0"?>
<methodCall>
   <methodName>XmlRpcCmdReconfigure 814</methodName>
   <params>
     <param>
       <value>
         <struct>
           <member>
             <name>RequestID 815</name>
             <value>2007-10-18 12:34:38</value>
           </member>
           <member>
             <name>DestinationDrive 814</name>
             <value>Removable</value>
           </member>
           <member>
             <name>Experiment 814</name>
             <value>Test_001</value>
           </member>
         </struct>
       </value>
     </param>
   </params>
</methodCall>
```

Response file name: "*xmlrpccmd.response*":

```
<?xml version="1.0"?>
<methodResponse>
   <params>
     <param>
       <value>
         <struct>
           <member>
             <name>RequestID</name>
             <value>2007-10-18 12:34:38</value>
           </member>
           <member>
             <name>Result</name>
             <value>
               <i4>0</i4>
             </value>
           </member>
         </struct>
       </value>
     </param>
   </params>
</methodResponse>
```

10.3.4.3 Reconfiguring and starting the measurement with the transmitted configuration.

File name: "*xmlrpccmd.call*":

```
<?xml version="1.0"?>
<methodCall>
   <methodName>XmlRpcCmdReconfigure 814</methodName>
   <params>
     <param>
       <value>
         <struct>
           <member>
             <name>RequestID 815</name>
             <value>2007-10-18 15:49:06</value>
           </member>
           <member>
             <name>SourceDrive 814</name>
             <value>Removable</value>
           </member>
           <member>
             <name>SourceFile 814</name>
             <value>dev002.ume.zip</value>
           </member>
           <member>
             <name>DestinationDrive 814</name>
             <value>Removable</value>
           </member>
           <member>
             <name>Experiment 814</name>
             <value>Test_001</value>
           </member>
         </struct>
       </value>
     </param>
   </params>
</methodCall>
```

Response file name: "*xmlrpccmd.response*":

```
<?xml version="1.0"?>
<methodResponse>
   <params>
     <param>
       <value>
         <struct>
           <member>
             <name>RequestID</name>
             <value>2007-10-18 15:49:06</value>
           </member>
           <member>
             <name>Result</name>
             <value>
               <i4>0</i4>
             </value>
           </member>
         </struct>
       </value>
     </param>
   </params>
</methodResponse>
```

10.3.4.4 Get version with GetSoftwareVersion

File name: "*xmlrpccmd.call*":

```
<?xml version="1.0"?>
<methodCall>
   <methodName>GetSoftwareVersion 815</methodName>
   <params>
     <param>
       <value>
         <struct>
           <member>
             <name>RequestID 815</name>
             <value>2008-06-02 14:23:56</value>
           </member>
         </struct>
       </value>
     </param>
   </params>
</methodCall>
```

Response file name: "*xmlrpccmd.response*":

```
<?xml version="1.0"?>
<methodResponse>
   <params>
     <param>
       <value>
         <struct>
           <member>
             <name>RequestID</name>
             <value>2008-06-02 14:23:56</value>
           </member>
           <member>
             <name>Result</name>
             <value>
               <i4>0</i4>
             </value>
           </member>
           <member>
             <name>SoftwareVersion 815</name>
             <value>V2.6R3 SP7 Beta (31.5.2008)</value>
           </member>
         </struct>
       </value>
     </param>
   </params>
</methodResponse>
```

10.3.4.5 imc Messaging default configuration

10.3.4.5.1 Delete default configuration

File name: "*xmlrpccmd.call*":

```
<?xml version="1.0"?>
<methodCall>
  <methodName>SetimcMessagingDefaultConfig 815</methodName>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>RequestID 815</name>
            <value>2008-06-02 13:25:12</value>
          </member>
          <member>
            <name>Configuration 815</name>
            <value></value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodCall>
```

Response file name: "*xmlrpccmd.response*":

```
<?xml version="1.0"?>
<methodResponse>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>RequestID</name>
            <value>2008-06-02 13:25:12</value>
          </member>
          <member>
            <name>Result</name>
            <value>
              <i4>0</i4>
            </value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodResponse>
```

10.3.4.5.2 Set default configuration

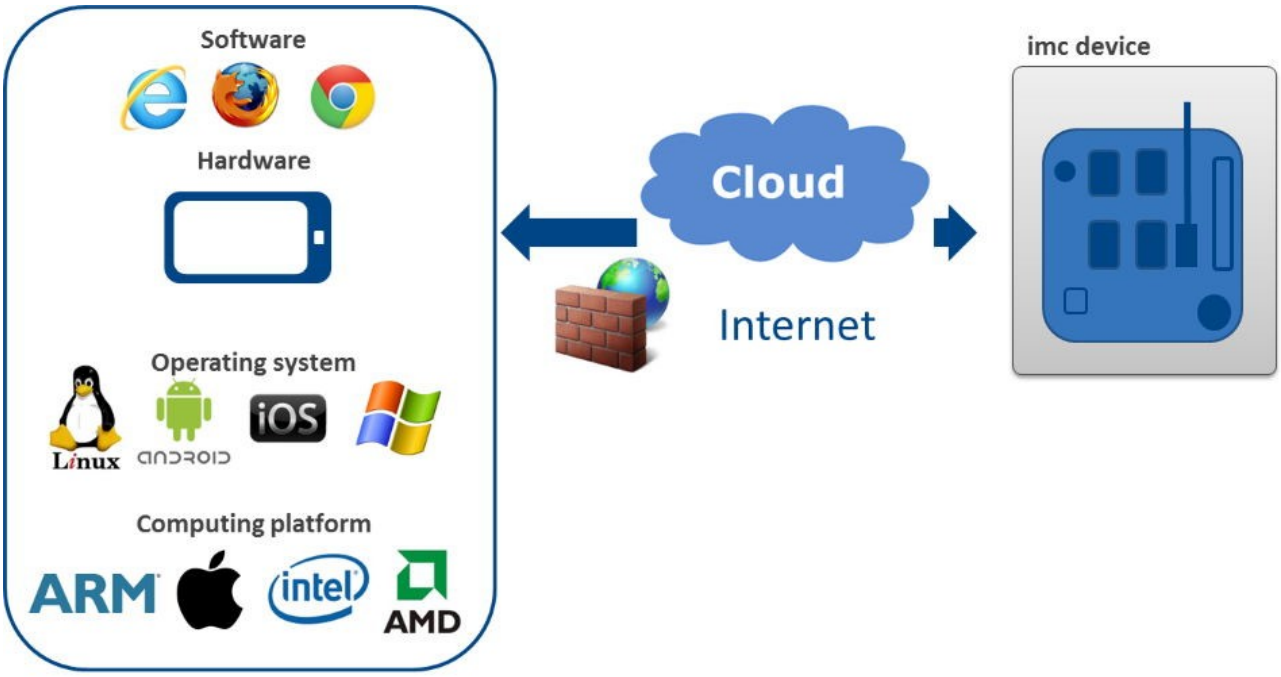
File name: "*xmlrpccmd.call*":

```
<?xml version="1.0"?>
<methodCall>
  <methodName>SetimcMessagingDefaultConfig 815</methodName>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>RequestID 815</name>
            <value>2008-06-02 13:29:38</value>
          </member>
          <member>
            <name>Configuration</name>
<value>[UDP Status Monitoring]
# The IP address of the destination PC
DestinationIP=10.0.2.7
# The UDP port to send the data to
# Port 9 is the &quot;discard&quot; port and should do no harm to the PC
DestinationPort=9
# The interval in seconds (e.g. send every 4min)
Interval=240
# The data to send:
Message={DeviceName} {DateTime, %Y-%m-%d %H:%M:%S}

Message={SerialNumber}
Message={SoftwareVersion} {SoftwareDateTime, %Y-%m-%d}
Message={MeasurementStatus, %u}

[Virt_Bit01]
Event=TransitionToOne
MessageType=EMAIL
SMTPServer=smpt.company.com
SenderAddress=Sender name &lt;sender@company.com&gt;
Destination=destination@company.com
Subject=Message from device {DeviceName}!
Message={DeviceName} {DateTime, %Y-%m-%d %H:%M:%S}
Message={SerialNumber}
Message={SoftwareVersion} {SoftwareDateTime, %Y-%m-%d}
Message={MeasurementStatus, %u}
</value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodCall>
```

Response file name: "*xmlrpccmd.response*":

```
<?xml version="1.0"?>
<methodResponse>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>RequestID</name>
            <value>2008-06-02 13:29:38</value>
          </member>
          <member>
            <name>Result</name>
            <value>
              <i4>0</i4>
            </value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodResponse>
```

10.4 imc REMOTE WebServer



imc REMOTE provides platform-independent remote access to imc measurement systems. This extension enables the imc system on which it is activated and running to be accessed by a terminal device/computer via a network connection, without the terminal device needing either to have imc STUDIO installed or to meet the usual requirements regarding its operating system. For communication with the imc system via its web-server, all that is required is a standard Internet browser which may run on any desired terminal device platform (e.g. MS Windows, Linux, iOS, Android etc.), so that hand-held mobile Internet-capable devices such as Tablet PCs, smart phones etc. can be used.

imc REMOTE WebServer is designed for remotely accessing measurement systems via Internet and in particular across the network's Firewall boundaries, because it communicates by means of secure https protocols.

Important properties and applications:

- Communication by means of html web pages
- Display of system variables (Display variables, pv-variables, virtual bits)
- Intervention in the device status/measurement process and interaction with the measurement parameters by setting (writing) Display variables, digital outputs, virtual bits and network bits (Ethernet bits)
- Altering of the system configuration by selection among Auto- and Diskstart experiments pre-loaded to the measurement device
- Downloading of measured data (files)

10.4.1 System requirements

License Management and Activation

- **imc REMOTE** includes two components **imc REMOTE WebServer** and **imc REMOTE LinkSecure.** imc REMOTE is a system extension permanently installed in a measurement device, which must be enabled with a serial number activation code. Users with multiple systems must obtain a separate activation code for each system.
- **imc REMOTE WebDesigner**, to create or change panels of websites that are provided via the **imc REMOTE Webserver** with a **imc measurement device**. imc REMOTE WebDesigner will be activated on your PC with an activation code.

System requirements

Internetbrowser	Device software
• Firefox Version 16 • Internet-Explorer Version 9 • Google Chrome 26.0 • Opera 12.02 • Safari 5.1.7	• Firmware (imc DEVICES) Version 2.9 R2 and higher • imc STUDIO Version 5.0 and higher • Option imc REMOTE with imc REMOTE WebServer (including LinkSecure)=3 225 (one license per device required)

Measurement devices for imc REMOTE WebServer	Measurement devices for imc REMOTE WebDesigner
The following devices are supported: • imc CRONOS*compact* • imc CRONOS*flex* • imc CRONOS-SL-N • imc CRONOS-XT • imc BUSDAQ*flex* • imc SPARTAN-N • imc C-SERIES-N, imc C-SERIES-FD • imc C1-N, imc C1-FD	Devices from group A7 on are supported: • imc CRONOS*compact* • imc CRONOS*flex* • imc CRONOS-XT

10.4.2 Activation

imc REMOTE WebServer is an optional product available for purchase. Upon purchase, you are supplied with an activation code which must be entered in the device properties.

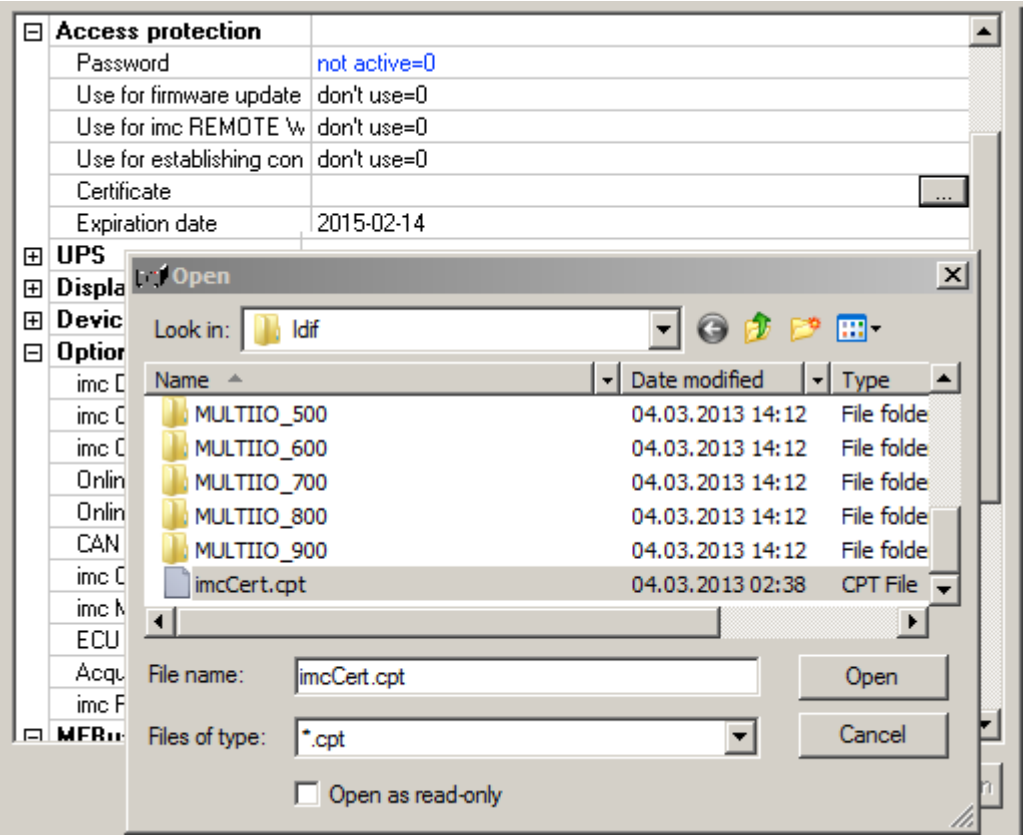
Activate certificate

The certificate is required for secure access via https.

- Open the dialog "*Device Properties*"

Ribbon	View
Setup-Configuration > Device Properties	Complete

- Load the certificate "*imcCert.cpt*". The software points directly to the correct folder.

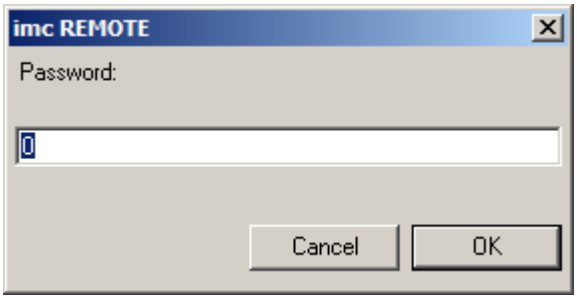


Activate certificate

Enter activation code

- Activate under "*imc REMOTE*": "*imc REMOTE WebServer (inkl. LinkSecure)=3*"

- Enter the activation code which you obtained by purchasing the option.



If access is provided via Internet, we recommend that in the branch "*Access protection*", you set the setting *Password* to "*active = 1*."

Close the dialog "*Device Properties*" by clicking on "*OK*".

The imc REMOTE WebServer is activated. Remember the password for the purpose of subsequent calls of WebServer.

Note — One connection is permanently occupied

If imc REMOTE WebServer is enabled in the device properties, a connection is permanently occupied. The maximum number of connections to the device depends on the device - see "*Device overview* 187".

10.4.3 Calling from the browser

Access to the WebServer from an Internet browser is provided via the device's IP address.

Enter https://IP address in the address bar, e.g. https://10.0.20.132.

Note — Find out the IP address

The IP address for the device can be viewed and set using the dialog "*Device interface configuration* 53".

Ribbon	View
Setup-Configuration > Device interfaces ( )	Complete

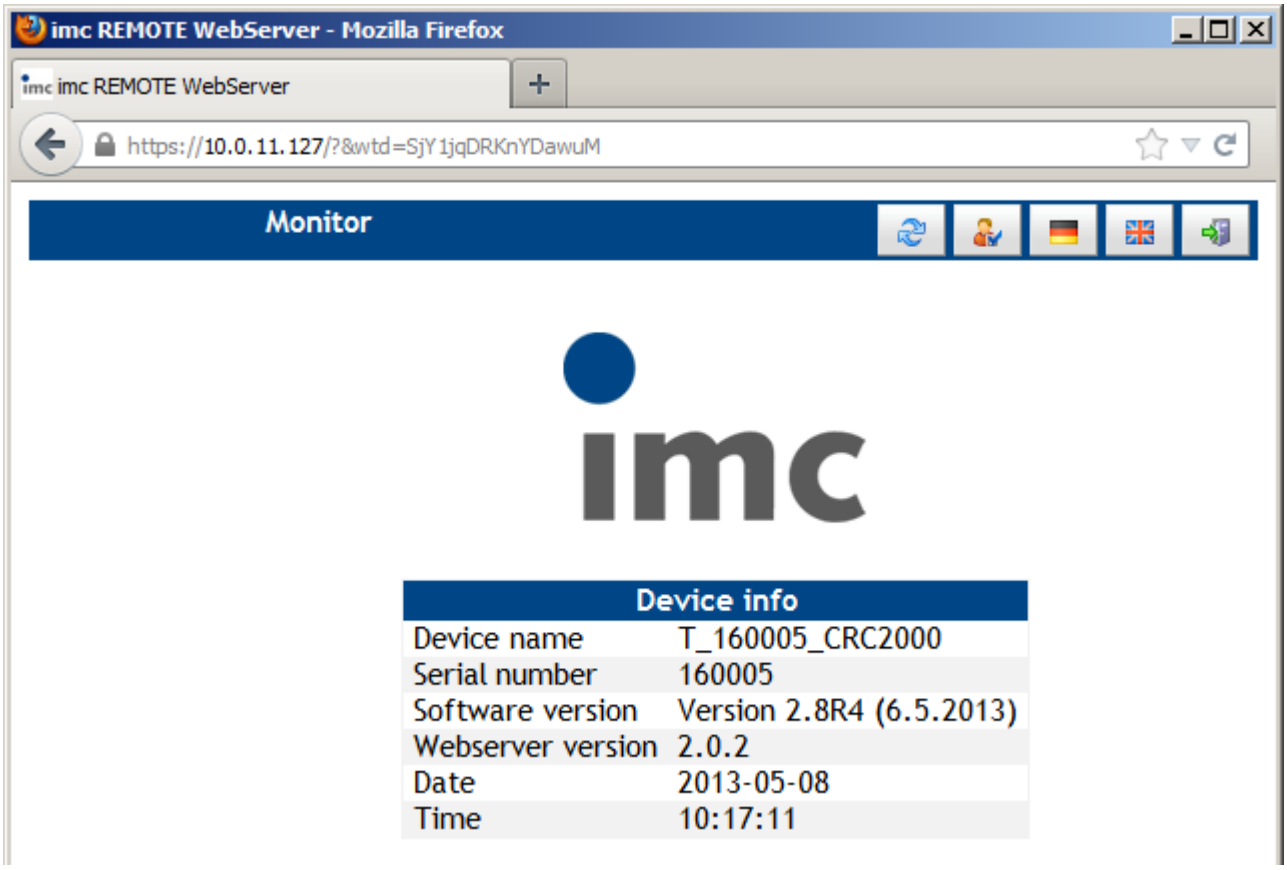
The input mask for the password you previously entered for the user-defined device appears.

Access to the device and Log-in

Note

The ***password*** for the Web server is assigned via Device properties:"*Access protection*" > "*Password* 222". With "*Access protection*" > "*Use for establishing connection* 222" , the query is activated or deactivated.

After successfully entering the password, the device information appears in the user role 828 "Monitor".



Main page of WebServer

10.4.3.1 Language selection

The language displayed can be changed at any time:

10.4.3.2 Closing a session

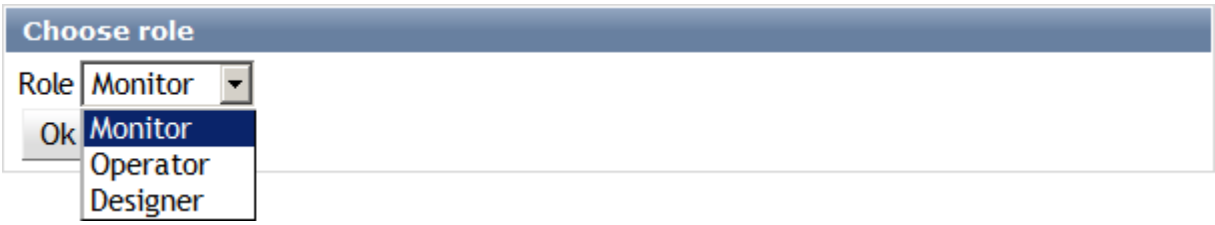
Always close a session by clicking on the Close symbol at the upper right corner.

Note

It is better not to simply close the Browser to end a session. If you do, the device receives no notification. The server program continues to run and is blocked from allowing re-login with Designer privileges until the device closes the session by Timeout.

10.4.4 User role

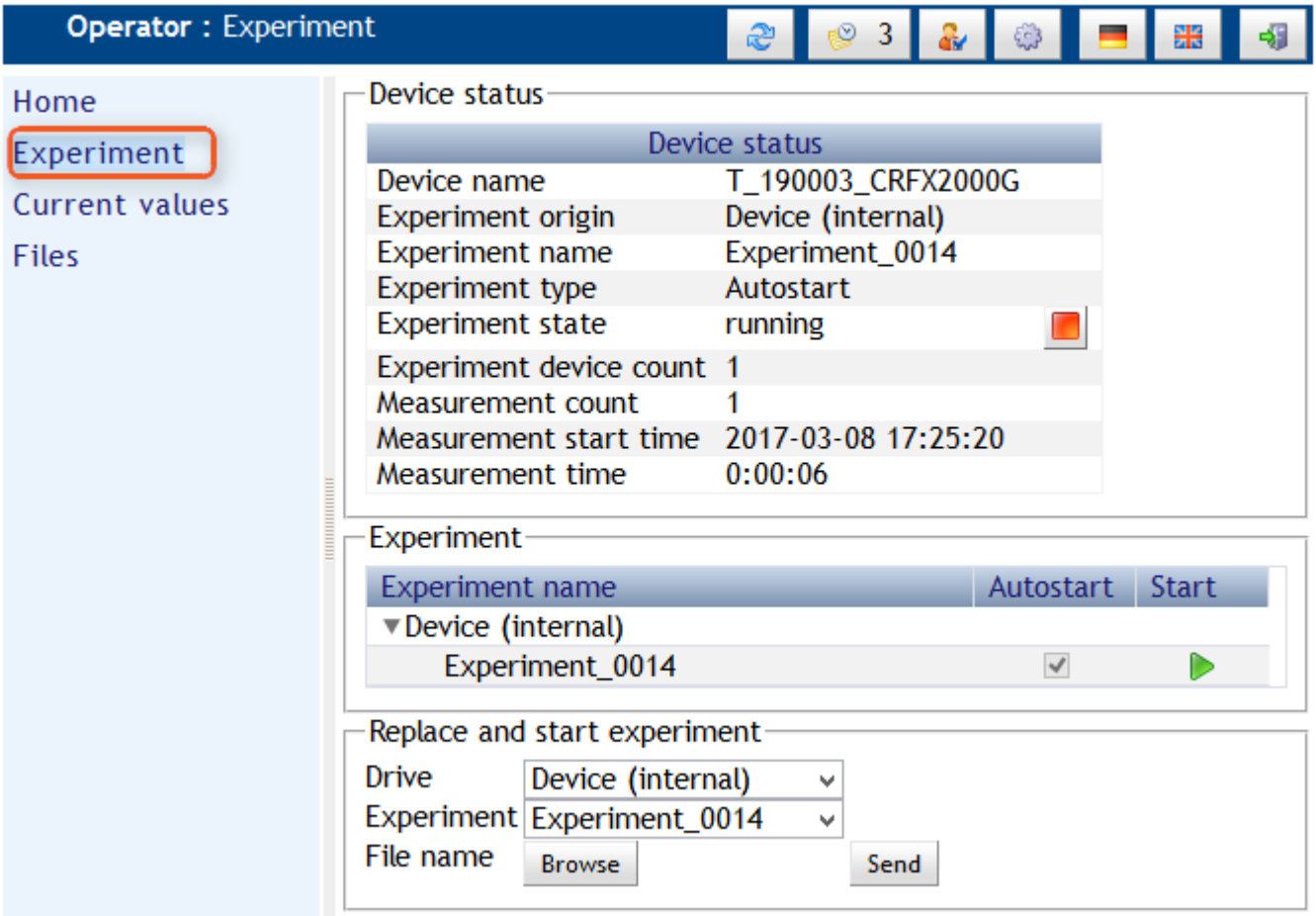
Version 2.0 and higher: There are three different user roles. By clicking on this icon, it is possible to change the selection of the user role.



User Role	Description
Monitor *(no password)*	• Observation of the system and measurement status • Querying of the device firmware-version (imc DEVICES) and the imc REMOTE WebServer version • Querying of system time and date • Downloading of measured data (files stored on onboard storage media) and the system experiment (configuration) • The Monitor role is read-only, i.e. interaction with the measurement (e.g. virtual bits or display variables) or changing of the running experiment (configuration) is prohibited. • It is not possible to change the selection for the current experiment. No intervention in the measurement state (start / stop) allowed.
Operator *(password optional)*	The Operator mode comes with the following functionality beyond that of Monitor: • Changing of the current configuration of the system/measurement, by selection and start of Diskstart-experiments • Changing of Auto-Start and Diskstart-configurations (experiments) by replacing experiments (upload/overwrite) • Starting/stopping the measurement (if the device is not connected and controlled by the user operation software imc STUDIO). • Ability to interact with a running measurement. For example, you can remotely intervene in a test by triggering a measurement, or to modify a calculation in imc Online FAMOS through a display variable, virtual bits, Ethernet bits, digital outputs and analog outputs.
Designer *(password protected*)*	• Configuration of which areas are accessible to the Operator role. • Creating and editing Panel pages • Display widgets: curve diagram, bar meter, pointer instrument, LED • I/O widgets: buttons, toggles, numerical input/output • Elements for graphical design: lines, circle, ellipse, etc. • License for imc REMOTE WebDesigner required * The password is distributed by purchasing the "imc REMOTE WebDesigner" option.

10.4.5 Experiment

Experiments already present on the system can be replaced via the WebServer. This is only possible in Operator mode.



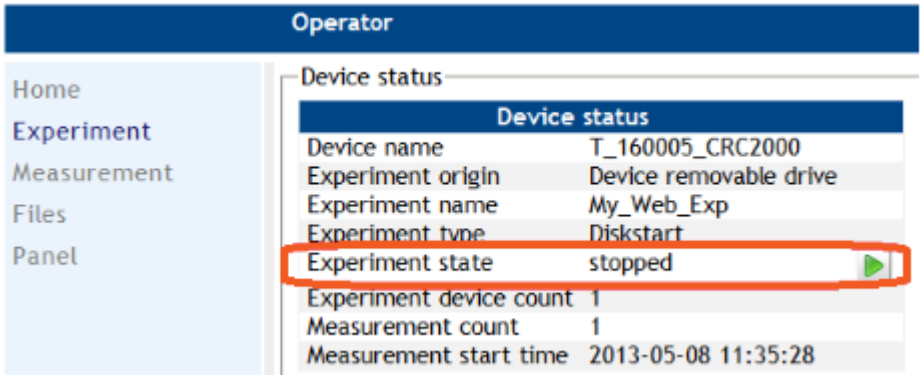
Step by step:

1. Create an Autostart- or Diskstart experiment 206 using the user operation software imc STUDIO. The experiment ("**.ume.zip*" or "**.ums.zip*"file) must be saved to the PC.
2. In the WebServer, select either the device's removable storage medium or its internal drive as the storage location.
3. Select the previously saved Auto-/Diskstart experiment by clicking on *Find*.
4. When you click on *Send*, the configuration is saved to the system and the measurement starts.
5. If you wish to keep multiple experiments available on the measurement system, repeat these steps for the other experiments.

10.4.5.1 Starting / stopping measurement

The measurement starts immediately once the experiment has been sent to the measurement system.

Otherwise, the measurement can be stopped at any time as well as restarted by clicking on the button next to *Experiment state*.

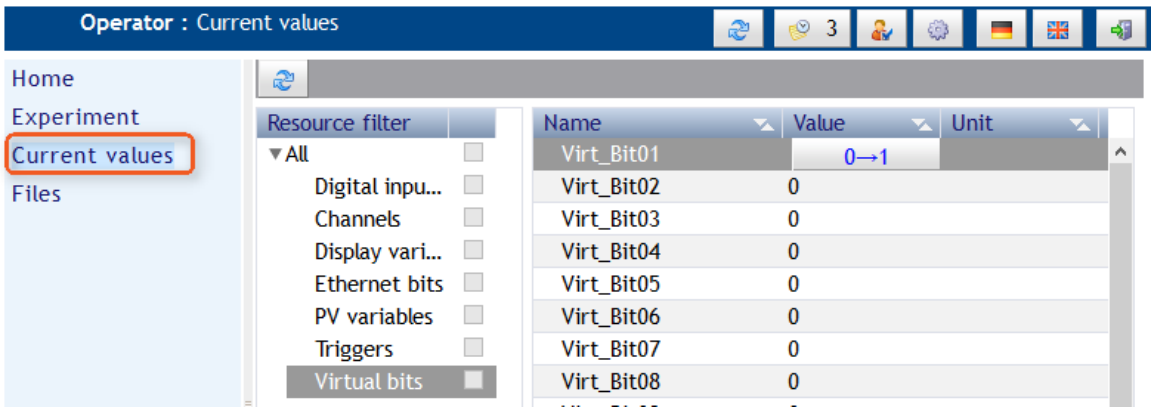


Note

If the measurement device is additionally connected with imc STUDIO, the function Start/Stop with imc Remote WebServer is deactivated. In this case, the Start/Stop button is hidden.

10.4.6 Current values

In both Operator and Designer modes, it is possible to influence the running measurement.



In Operator mode, you can change certain values and bits in the device. For instance, you can change a virtual bit's value, which can cause a trigger to be released in accordance with the respective configuration.

The same applies to digital output bits and network bits.

You can also set the value of the Display variable.

It is not possible to change pv-variables from imc STUDIO Automation or imc Online FAMOS here. Changing of the pv-variables can be performed in the Panel using either of the widgets *Bar meter* 839 or *Numerical input/output* 840.

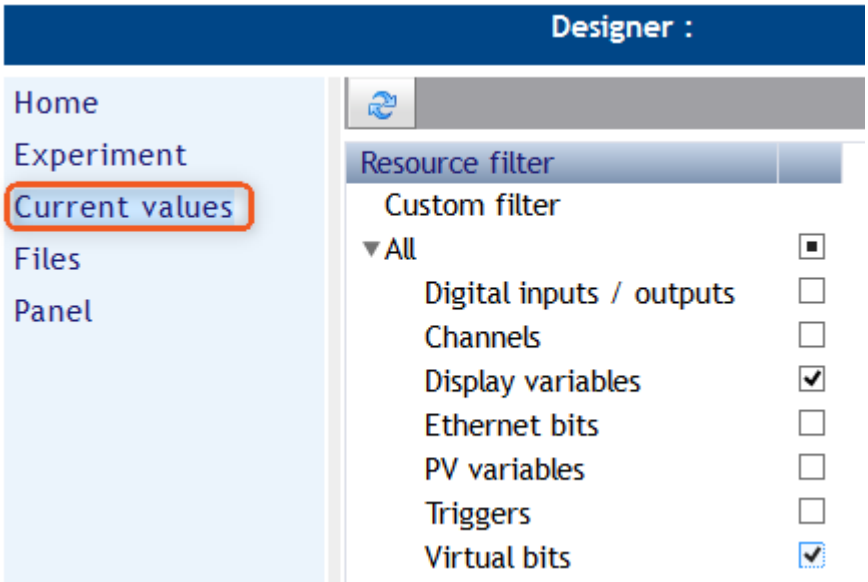
It is not possible to write values triggers.

Note

The values of the channels are **not** updated in this dialog. The channels' current values are displayed by the pv-variables.

10.4.6.1 Adapting the design

In the Designer mode 828, you set which resources are visible to the Operator.



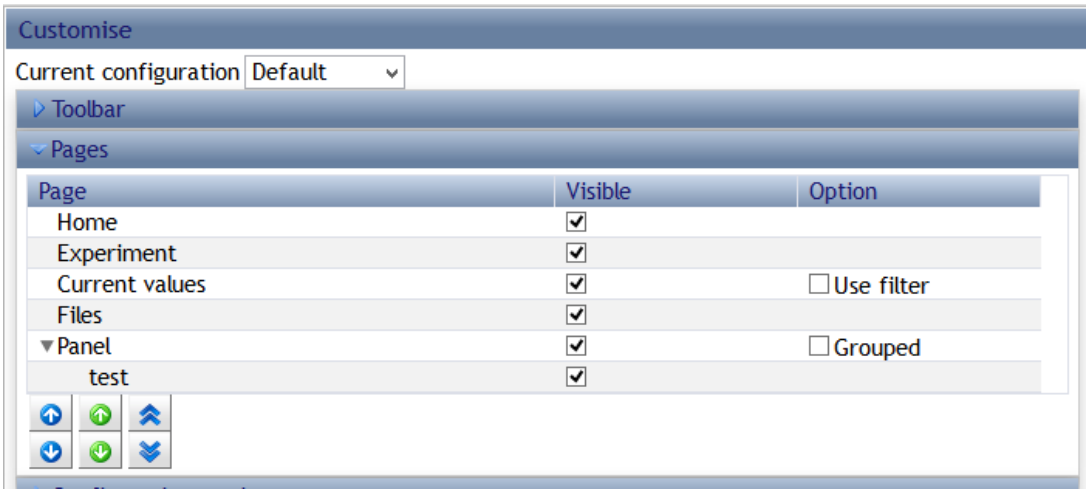
Only resources activated here are visible in Operator mode and can be changed (on the right side).

You can block either complete resources or individual elements such as bits or Display variables.

The dialog for the settings is only available in Designer mode 828:

Pages

In the column *Visible*, you can restrict the selection of pages (*Home, Experiment, Current values, Files, Panel)*. However, the pages are only effectively hidden once you activate the option *Use filter*.



10.4.7 Settings im-/export, save

All settings can be set as default, imported and exported if Designer mode 828 is enabled:

New, Save, Set as default, OK

Settings for the current session are adopted when you click on the button *OK (7)*.

To adopt the settings in a permanent way, select *Set as default (5).*

Using the button *New (2)*, you can set up a new configuration. This will be initialized with the settings which were last adopted by *Set as default*.

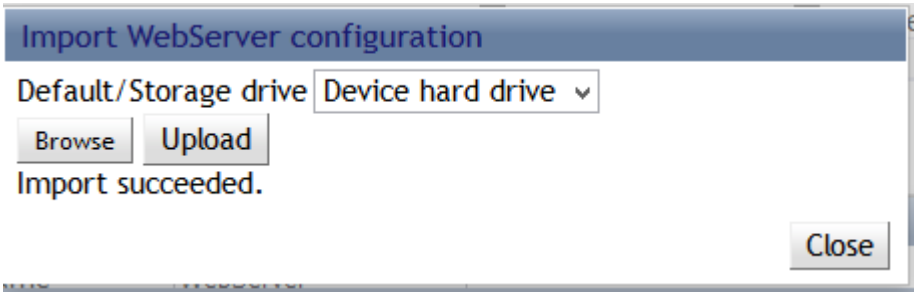
By means of *Save* and *Save as (4)*, the current configuration can be saved in the device's memory.

Using *Delete*, the current configuration can be deleted from the device memory.

Import, Export

You can *export (2)* the settings: A dialog having a link appears. Once you click on this, the selection dialog for saving the file *WebServer.imcwsc* appears.

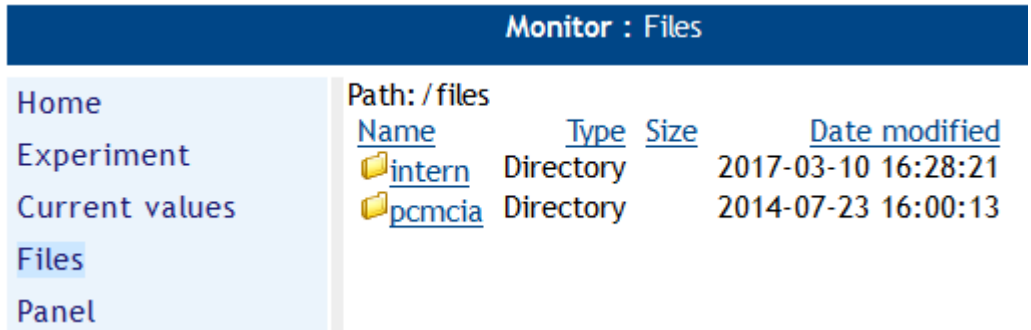
Import (1) also opens a dialog for selecting *WebServer.imcwsc.* Import is performed when you click on the button *"Upload"*.



When you copy the file ***WebServer.imcwsc*** to the experiment's folder on the PC, the settings are applied upon the next upload.

10.4.8 Files

The item Files provides access to the internal and to the removable data storage.



The single files can be downloaded.

10.4.9 Panel

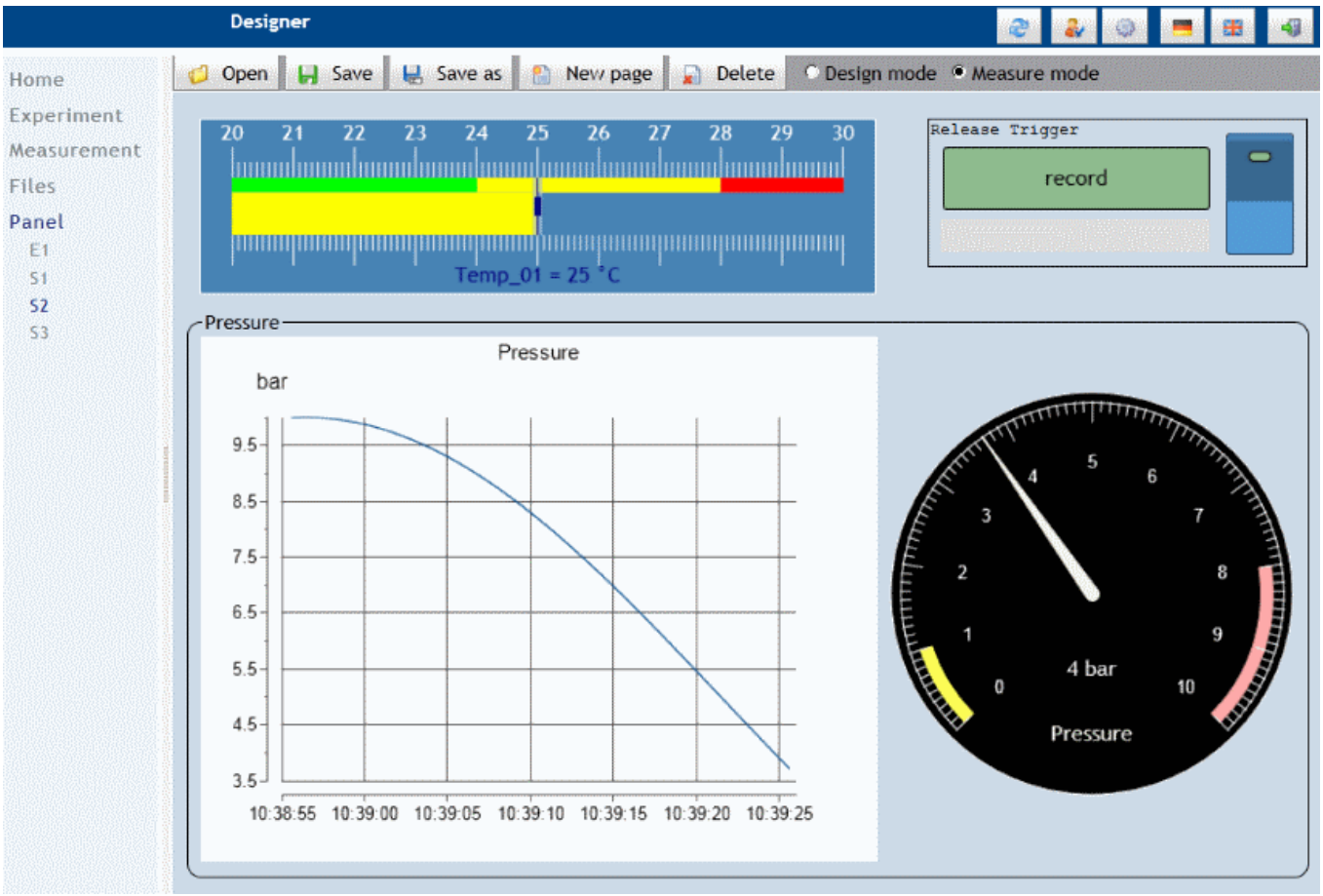
The web server from version 2.3 allows the use of panels. In the panel pane, it is possible to create multiple pages, by means of which the measurement's status can conveniently be displayed.

Note

We recommend stopping measurement while designing the web-pages. This can prevent possible performance bottlenecks, which, especially on smaller devices (CRC/CRFX-400), can cause Out-Of-Memory exceptions or overflow of measurement data to occur.

Note

For the purpose of creating Panel-pages, the option **imc REMOTE WebDesigner** is required and for the *User Role* 828, the selection *Designer* must be chosen.



Panel pages are saved in the device (Save As...) and can also be deleted from there. Saved Panel pages can be selected by mouse click, if they are made visible by adapting the design. Furthermore, saved Panel pages can be loaded using the button "Open".

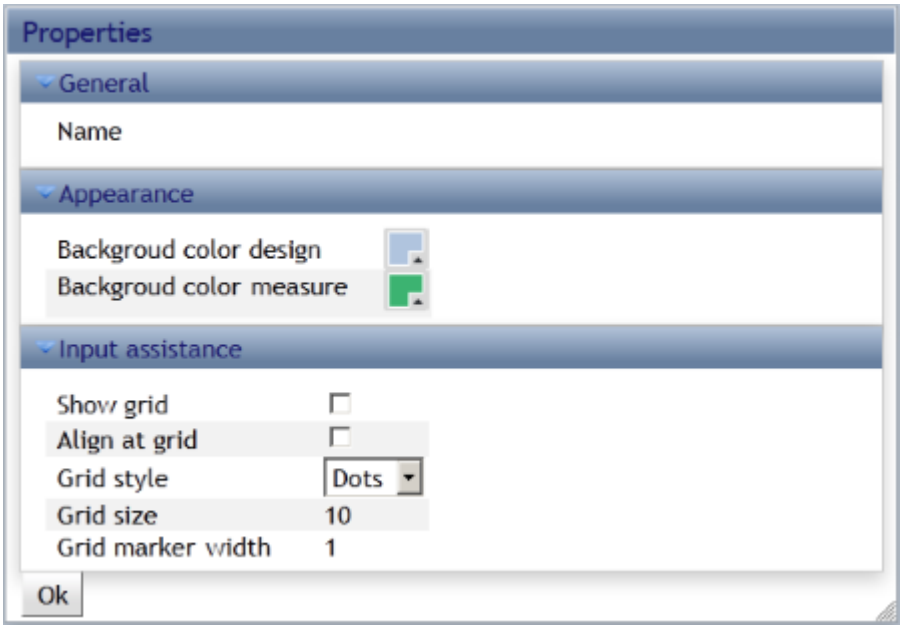
Note

- Panel pages are not saved along with the experiment. Once a page has been deleted, it can not be retrieved. Links with variables' names only work after switching to a different experiment if the same ones exist in that experiment.
- An Export/Import function for Panel pages is in preparation.

The Panel Page pane is only available in **Designer Mode**. In order that the Panels also be displayed in Monitor and Operator modes, they need to be set as visible in the configuration 831.

Once again the Panel itself distinguishes between Design and Measurement mode. In Design mode, the widgets are created. In Measurement mode, the data is exchanged with the device.

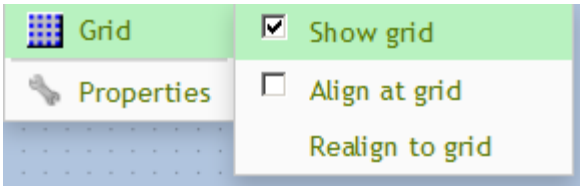
By right-clicking on the Panel page, the Properties become available for editing:



Panel properties

Along with the background color, the grid is also defined here.

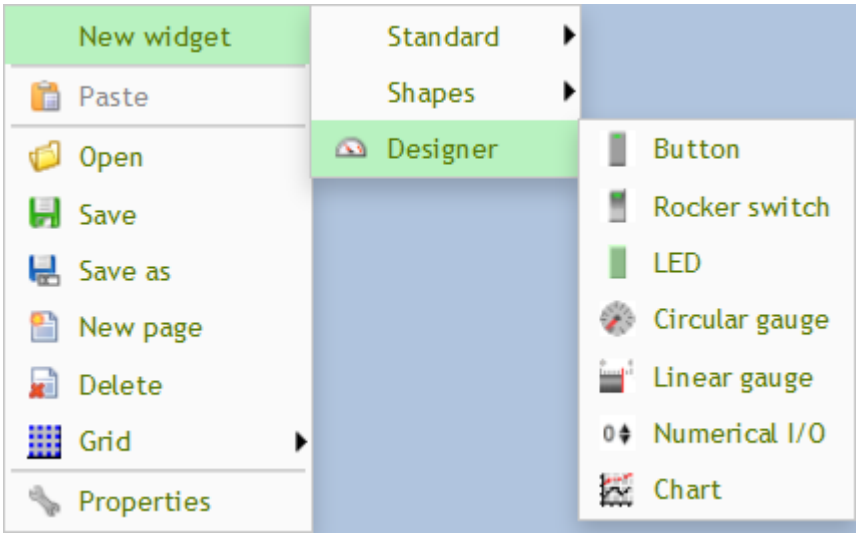
However, the grid can also be used by means of the menu item Grid:

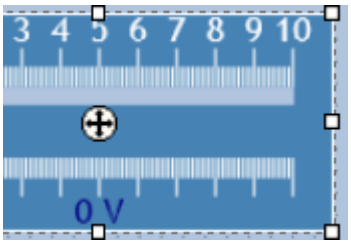


Grid context menu

10.4.9.1 Widgets

Widgets are created by right-clicking the mouse over the Panel.



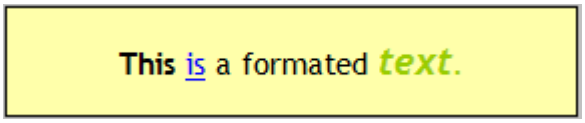


To call the widget's properties, right-click the mouse on the crosshairs in the center.

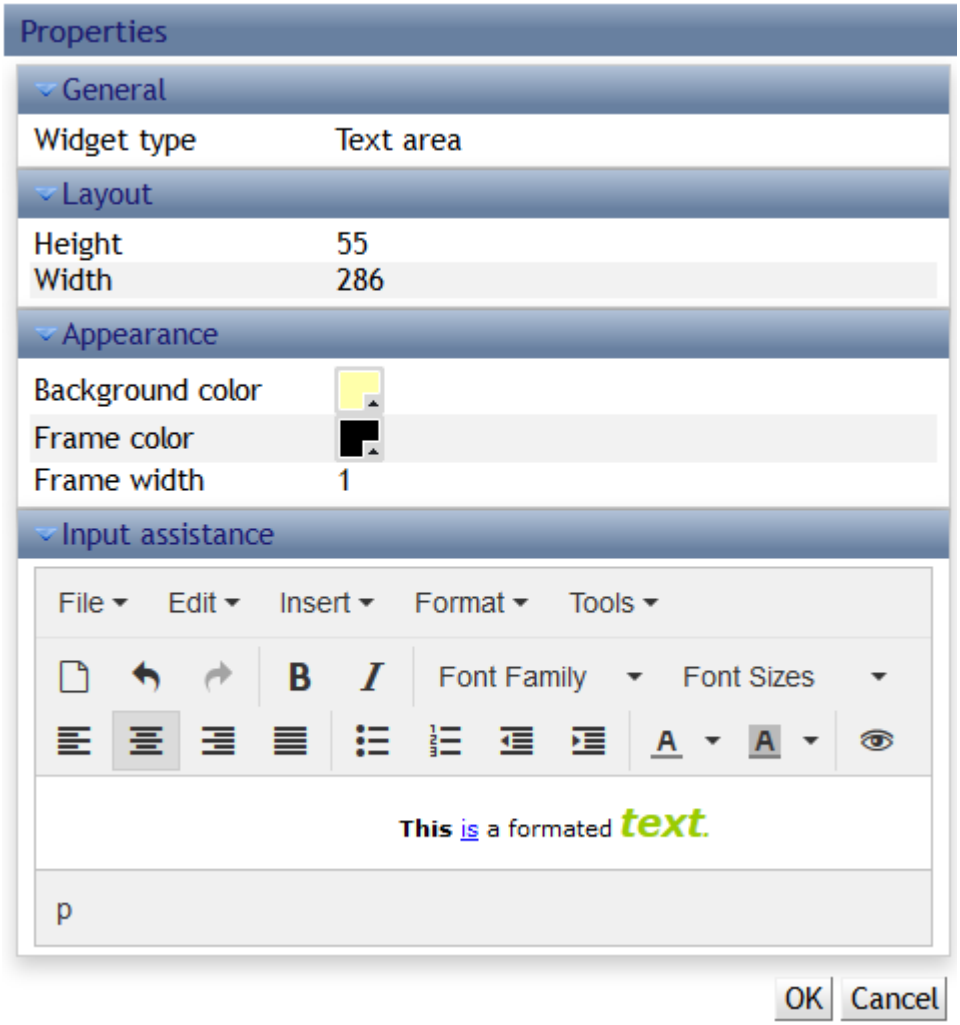
Note

- Note that the Web-Browser only receives single values from the device. Other channel parameters such as the sampling rate, unit or input range are not available. For this reason, these parameters must be entered manually in the widgets' properties.
- Although the Panel closely resembles a program's GUI, you are working with an Internet browser. For this reason, there are certain constraints on operation which can vary according to the browser selected. Thus, many key functions, such as Delete, are not supported but must be called via a context menu.

Standard - Text area

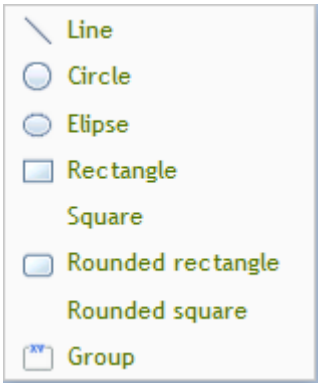


Widget for displaying text. It is possible to set font and colors, background and frame, as well as the frame width. The frame can be shaped by means of the mouse or entered directly in terms of pixels.

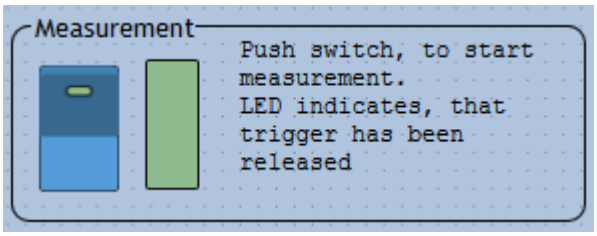


Shapes

As graphics elements, a variety of shapes are available.



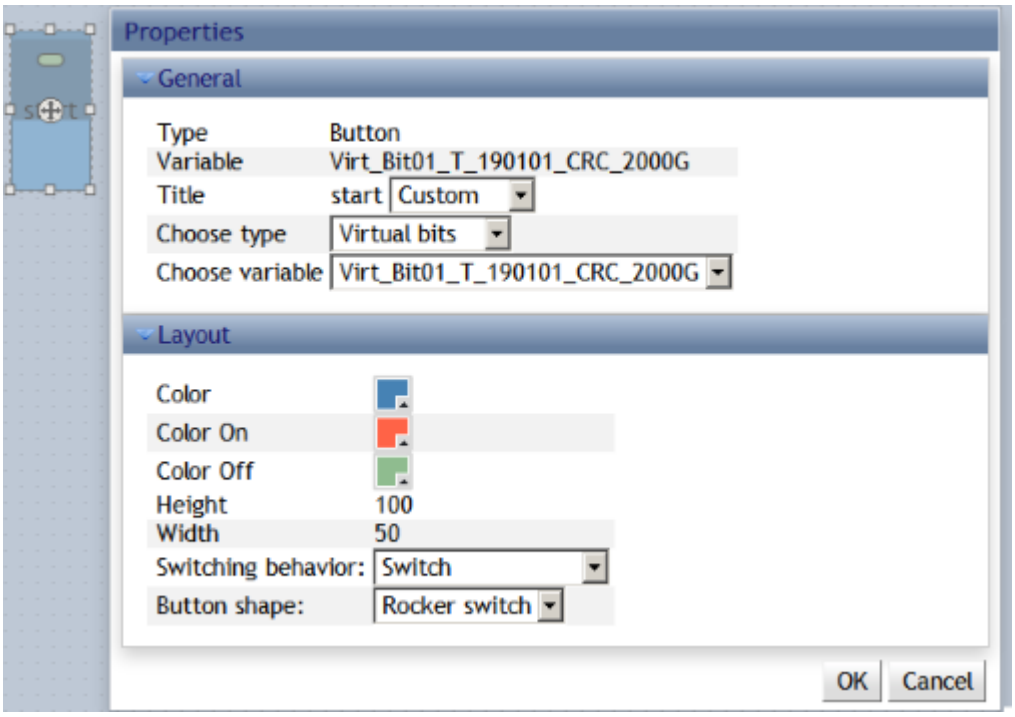
The group box can additionally be supplied with text.



Designer - Press key/toggle switch

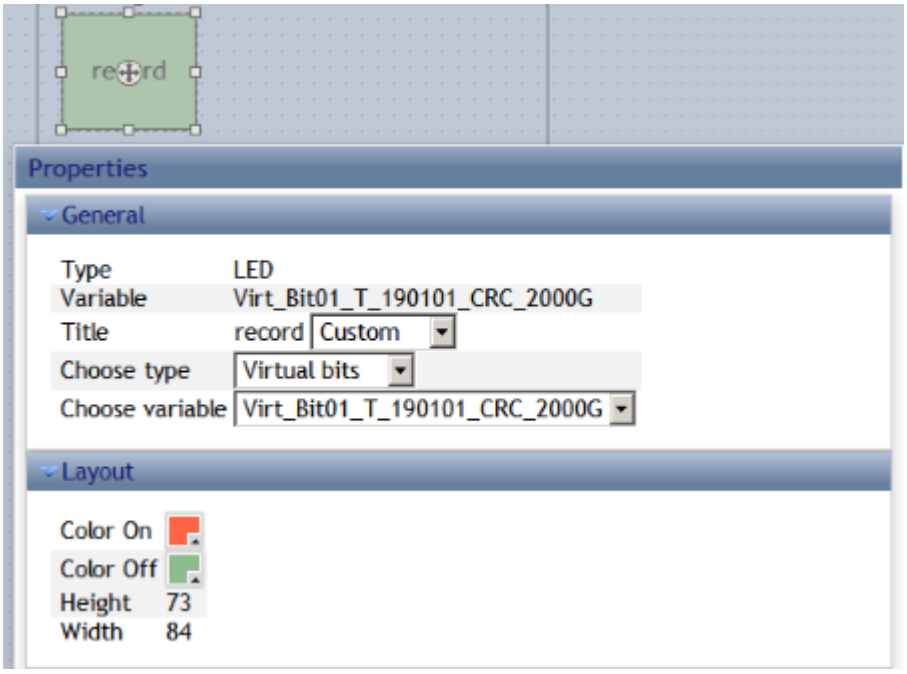
Switches/Pushbuttons can be linked with any digital variable (Virtual_Bits, Ethernet bits, digital inputs)

Their behavior and appearance can be set in the properties: Switch/Pushbutton and Rocker switch/Button.



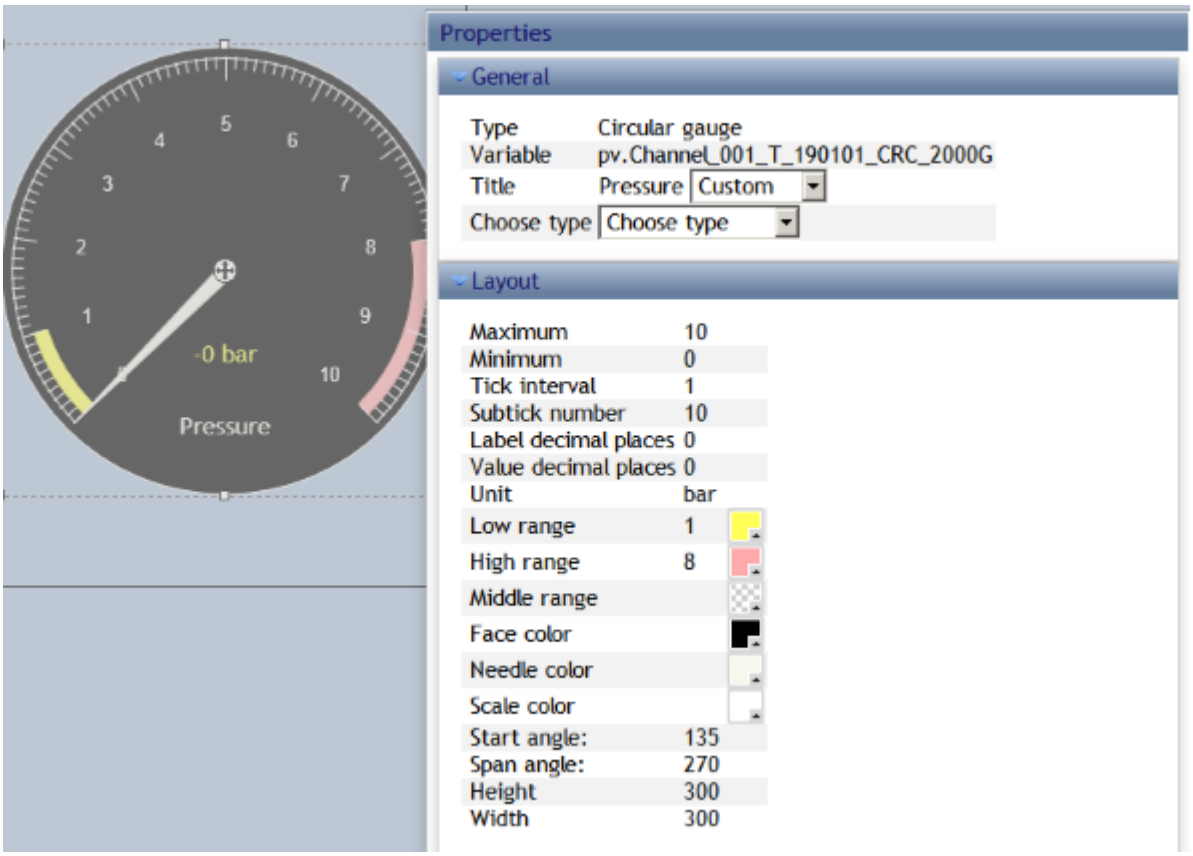
Designer - LED

LEDs can be linked with any digital variables (Virtual_Bits, Ethernet bits, digital inputs)



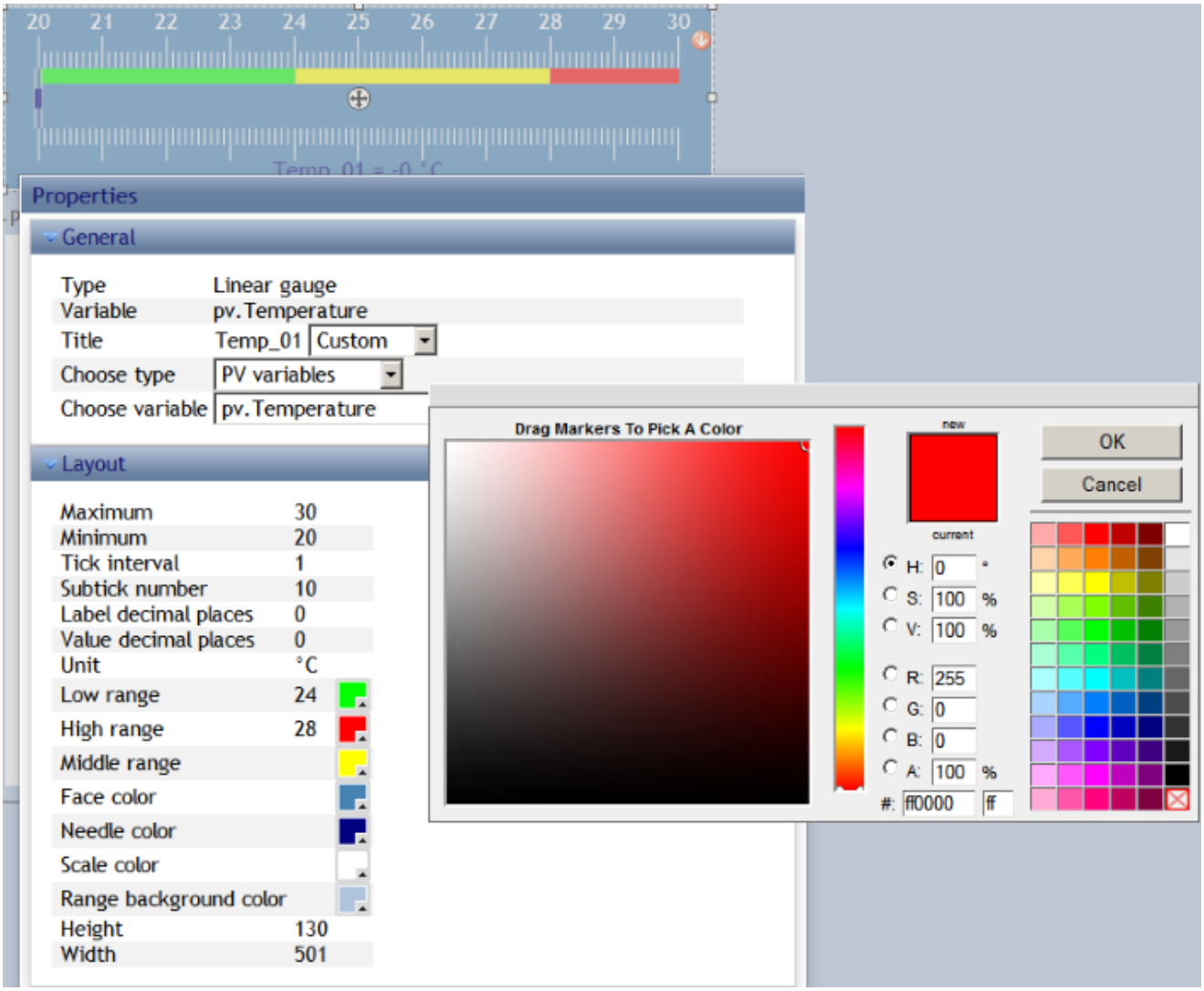
Designer - Pointer instrument

The available settings for the pointer instrument correspond to those of the bar meter.



Designer - Bar meter

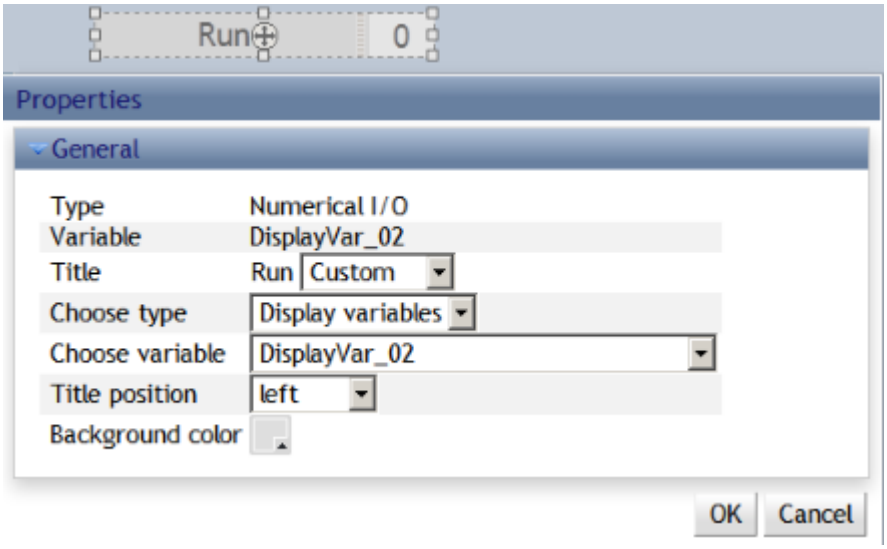
Display of the instantaneous value by a bar meter



Enter the minimum and maximum value of the linked channel. This is now the range in which the variable will be displayed. The colors change according to the current value; the value boundaries and colors can be freely selected.

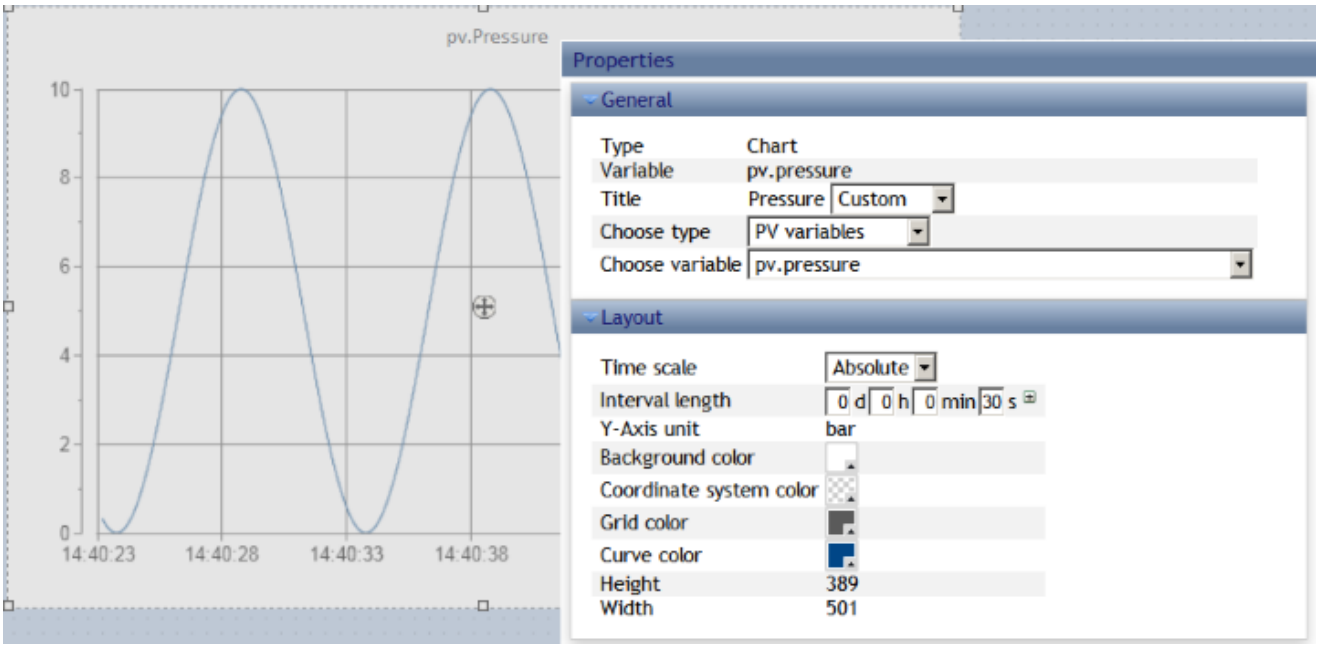
Designer - Numerical input/output

For the purpose of numerical input and output, Display variables and pv-variables can be used.



Designer - Curve diagram

The curve window can display exactly one parameter. This parameter may come from a Display variable, a pv-variable, or even from digital variables(Virtual_Bits, Ethernet bits, digital inputs)



The scaling is set automatically along with the maximum and minimum values of the data measured.

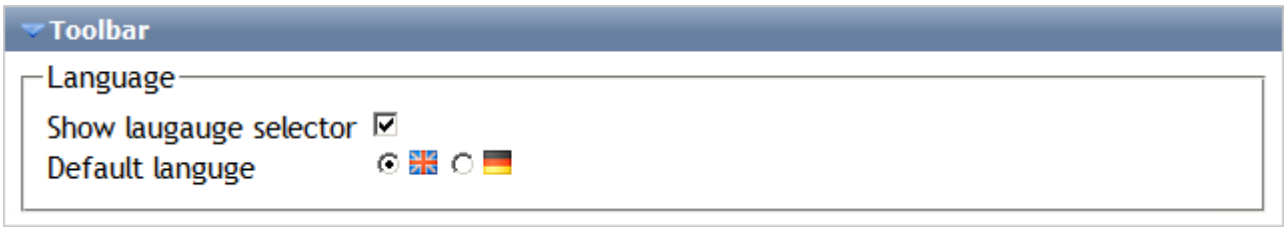
The x-axis can be set to relative or absolute.

10.4.10 Miscellaneous

To apply miscellaneous settings, call the settings dialog via the associated button, which is only visible in ***Designer mode***:

Toolbar

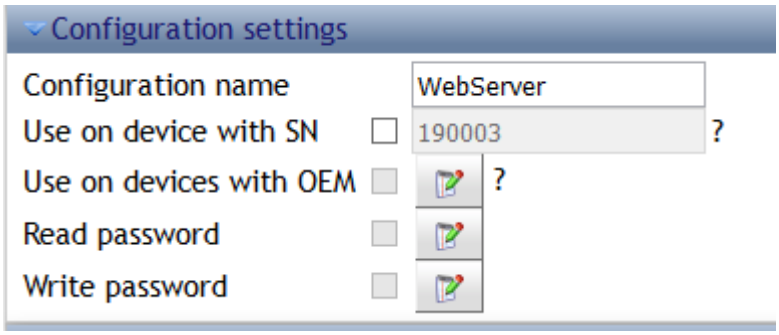
Set the display language in which the WebServer starts up.



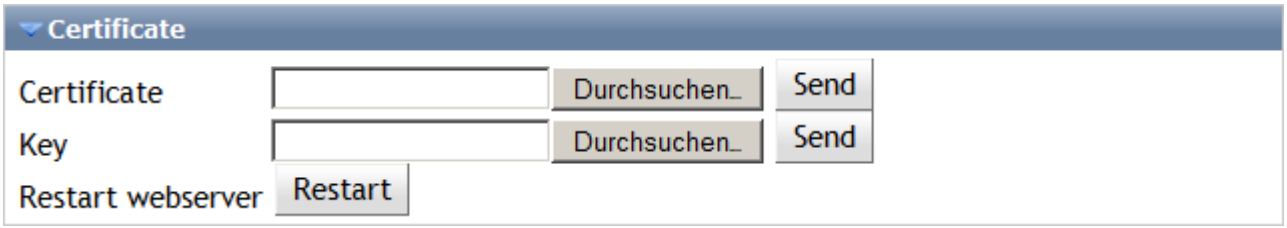
Configuration

Protect the settings by assigning passwords.

The **OEM identifier** is required for custom devices and applications in order to enable fixed Panel configurations. If this setting is active, the configuration can only be used in devices having the specified OEM identifier.



Certificate



The certificate required for running WebServer can be renewed using this function. This entails exchanging the certificates saved in the device and uploading ("sending") a valid certificate and key.

In the uploading process, the Webserver verifies whether the certificate and the key match each other. The validity can only be checked the next time the Webserver is started. if the certificate is not valid, it is rejected and the old one remains in force.

This can only be performed in either ***Operator*** or ***Designer mode***.

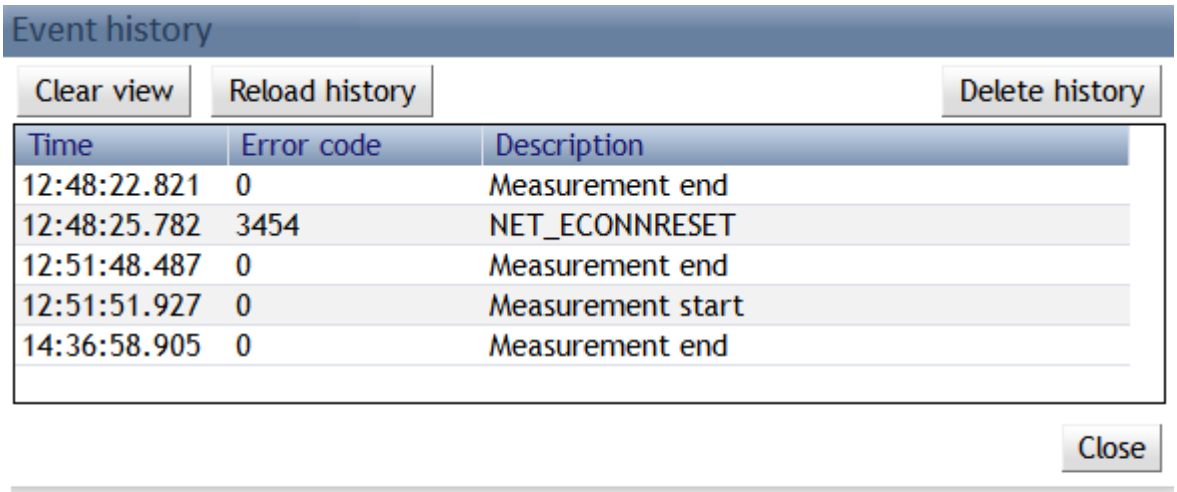
For the encrypted connection to the device, use an X.509 certificate in the form of a certificate and key file in PEM format.

If you have any questions, please contact our tech support 8.

10.4.11 Event history

The event history is opened via the toolbar. Here, the entry count is already displayed.

The event history is a log of the course of the procedures and error messages:



11 Data analysis and signal processing

This chapter describes the various data analysis and signal processing possibilities available.

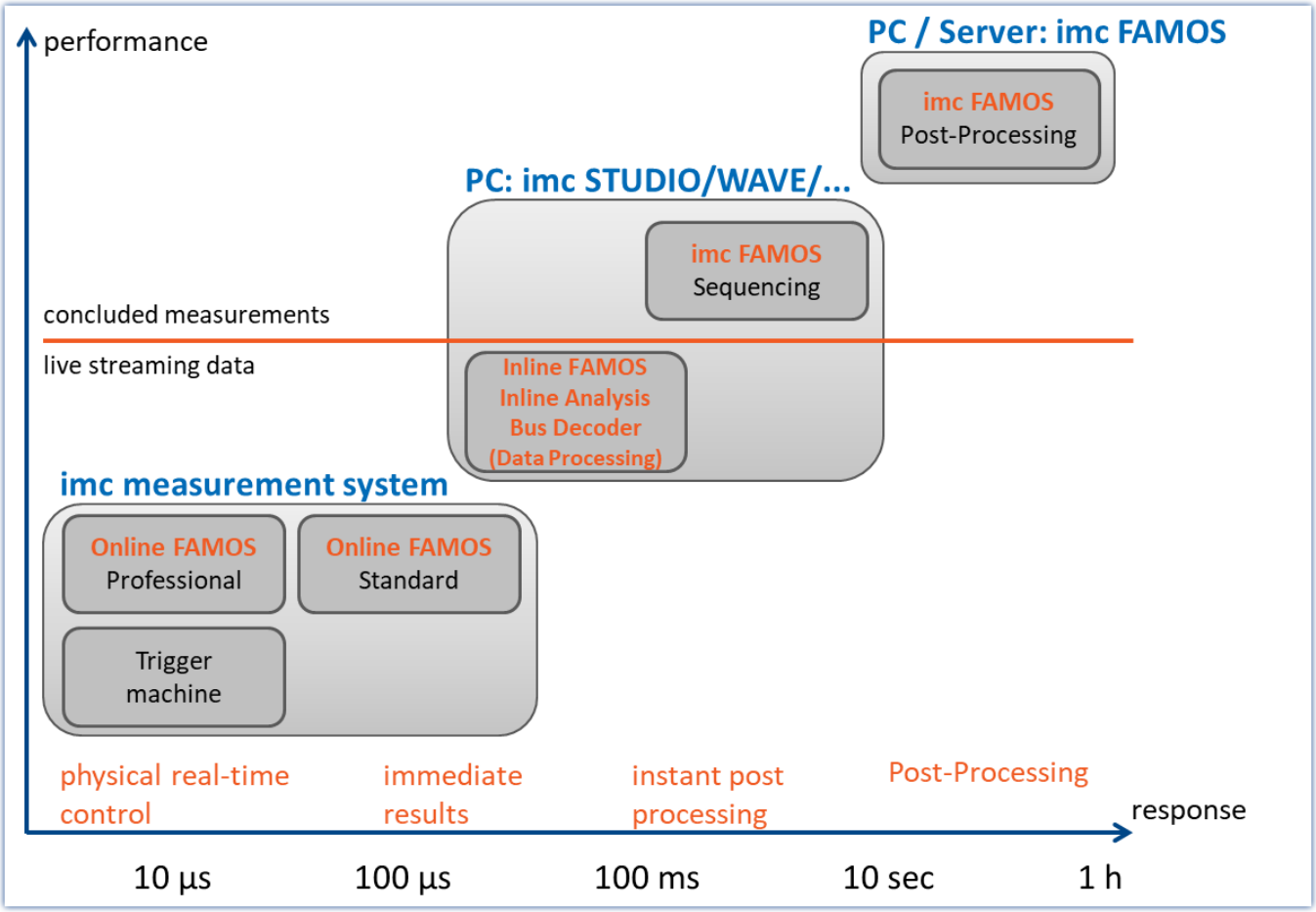
11.1 Overview of the tools

Tool	Description
imc Online FAMOS 851	imc Online FAMOS is a powerful extension for imc DAQ systems. It offers a variety of real-time functions for pre-processing and signal analysis. The mathematical analysis functions are executed on the signal analysis platform integrated in the measurement device. This means that analysis results are available immediately and also independently of the PC.
imc Inline FAMOS 851	imc Inline FAMOS enables calculations to be performed on data streams from the measurement currently runnning. The calculations are performed on the PC, taking advantage of the PC's processing power. By contrast, with imc Online FAMOS, the calculations are performed by the device. A number of pre-defined functions are available for calculation purposes. Most of the imc Online FAMOS functions are available in the same way and with the same syntax as in imc Inline FAMOS. There are a few additional functions exclusively in imc Inline FAMOS.
Inline Analysis 1064	Inline Analysis enables calculations to be performed on data streams from the measurement currently runnning. The calculations are performed on the PC, taking advantage of the PC's processing power. It is possible to compute a derivative or calculated channel from a input channel. For this purpose, a variety of functions packages are available.
imc FAMOS 1088	From within imc STUDIO it is possible to transfer variable currently being measured to imc FAMOS and to process them with a imc FAMOS sequence there. imc FAMOS processes entire data sets. When you load a measurement to imc FAMOS, you load the complete measurement. You have equal access to the first and last values of a measurement. And you can also write imc FAMOS-sequences consisting of various loops and conditions for processing data. In OFA/IFA, data are processed already while being captured. This means that the raw data are **no longer** in memory. The current raw data are used to generate calculation results. For all applications having online functionality requirements, it is not adequate to wait until the end of the measurement.
Bus Decoder 1101	This component allows either all or individual measurement channels belonging to a log-channel to be decoded/extracted. A log-channel can be a logged Fieldbus communication ("Logfile").This provides more flexibility and dynamic capability for deciding on targeted extraction of individual channels from the compressed logfile. The decoding is performed on the basis of the data streams of the measurement currently running on the PC. This utilizes the PC's computation resources.
Powertrain Monitoring 1106	The component imc STUDIO Powertrain Monitoring has been developed in cooperation with the company GfM (Gesellschaft für Maschinendiagnose mbH) for power train diagnostics.

When should which tool be used?

Application fields:

- Real-time analysis and brief reaction time: imc Online FAMOS
- Complex scope of functions (calculation-intensive): imc Inline FAMOS
- Calculation-intensive subsequent analysis (post-processing): imc FAMOS



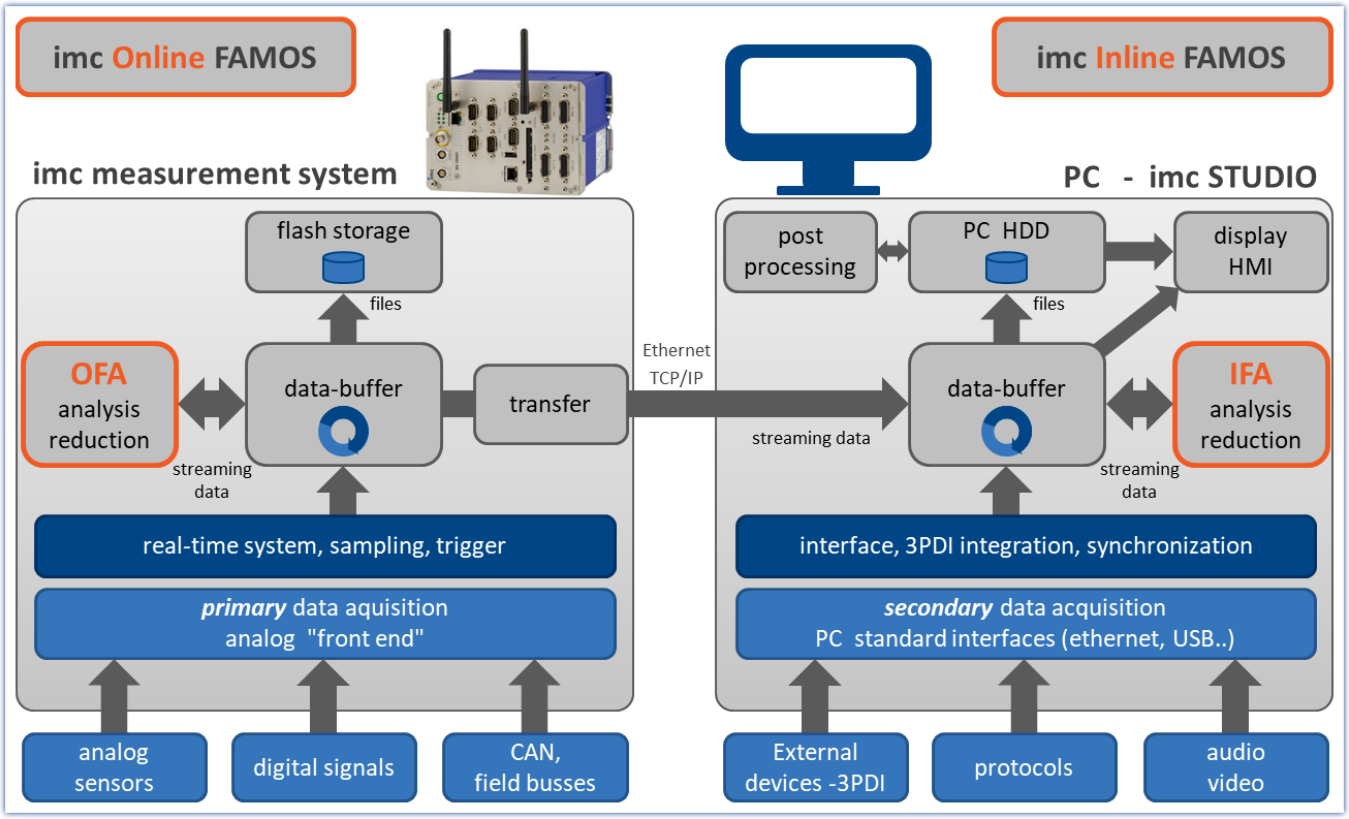
11.1.1 Comparing imc Online FAMOS with imc Inline FAMOS

imc Online FAMOS	imc Inline FAMOS
Device-based, classical real-time analysis	PC-based analysis of live streaming data diametrically different from imc FAMOS (post-processing of completed data sets)
Processing occurs where the data are captured, inside of the measurement device • no calculations across multiple devices possible • stand-alone capability	Processing occurs on the PC, and **not** where the data are captured. • Calculations across multiple devices possible (applies to all channels which are captured as of measurement start (BaseTrigger)), optionally also of 3rd-party devices (via 3PDI) • not stand-alone capable • correspondingly reduced real-time reaction • conversely: use of the powerful and scalable PC-platform
Resolution of the calculations and results: 4 byte	Resolution of the calculations and results: 8 byte
Commonalities: • Live-analysis: immediate visual feedback • Processing of live data streams: running, not concluded, measurements (not post-processing) • Unified syntax, same scope of functions • Application of calculation operations to combinations of multiple channels assigned to the same trigger	

Professional-Version

In imc Inline FAMOS, there are no version grades as with imc Online FAMOS (imc Online FAMOS / imc Online FAMOS Professional).

Data flow schematic



Data flow schematic

Comparison measurement

An example is presented in the section "*Performance comparisons* 848".

Scope of Functions

Most imc Online FAMOS functions are also available in imc Inline FAMOS in the same way and with the same syntax. Additionally, there are some functions available only in imc Inline FAMOS.

Among others, the following function groups are not provided:

- all closed-loop control functions and many system functions
- all CAN-functions and all ECU-functions
- ReadyForPowerOff, SyncOverload
- OnSyncTask (Synchronous Tasks), OnPowerOff
- IntegralP/IntegralP2

In the Functions Reference, it is noted if a particular function is available only in imc Online FAMOS or only in imc Inline FAMOS.

What kind of tasks can be accomplished?

Any task which could be performed by an electrical circuit can be handled. A circuit doesn't include loops or conditions, which are inappropriate to the streaming data we are dealing with. The data streams can only be processed and subjected to operations jointly with other data streams. The data streams can be transferred to the PC, to the storage medium, to DACs and to digital output bits.

If, instead of digital processing by OFA/IFA, we put a complex analog circuit (e.g. filter, ...) in front of the device' inputs and digitalize the outputs of such a system, the channels recorded directly correspond to the virtual channels. They also possess all properties of a normal channel. Furthermore, if we consider logic operations and filtering (mean value, etc.) to be the main OFA/IFA functions, then this is comparable to an analog circuit. Thus, OFA/IFA can be viewed as a substitute for a freely configurable analog pre-processing circuit.

What can OFA/IFA not do?

- It is impossible to formulate any processes which do **not** utilize **continuously streaming data**.
- imc Online FAMOS does not replace a synthesizer! For output rates of up to 10 kHz, the Synchronous Task 857 can be used. For this purpose, imc Online FAMOS Professional is necessary.
- OFA/IFA works asynchronously. It processes data according to the formula that you have entered as quickly as possible. If the formula that you specify is somewhat long or involved, however, it may lead to delays. It is impossible to state a fixed time interval within which the loop can be processed. If your application requires definite response times, then these can be guaranteed with an Synchronous Task 857 of imc Online FAMOS Professional.
- Fieldbus channels recorded in **time stamping mode, can't** be proceeded!
- **Unsigned 32-bit integer values** (32-bit UINT) are not calculated correctly.
  - Workaround: The channel must be set up in the CAN Assistant as an unsigned 31-bit integer.
    32-bit UNIT channels are mostly created when a DBC file is imported. If such a channel is processed any further in OFA/IFA, the bit count must be set to 31 bits following import of the DBC file.

11.1.1.1 Performance comparisons

A test comparing the performance of imc Online FAMOS with that of imc Online FAMOS is conducted using familiar hardware.

Test devices and configuration

Hardware used

Computer:

- WINDOWS 10
- Intel(R) Xeon(R) CPU E3-1270 V2 @ 3.50GHz 3.50GHz
- RAM: 8 GB
- Normal magnetic hard disk drive (not SSD)

Device:

- imc CRONOS compact 400 with two amplifiers of model UNI2-8

Test – configuration:

- Channel_001: 100kHz
- imc Online FAMOS: 2x FFT and 5x filtering, second order (bandpass, band-stop, high-pass, low-pass)
- RAM-buffer duration of all channels: 1s

```
; initialization prior to first measurement
OnInitAll
    sample = 0
End

OnTriggerMeasure(BaseTrigger)
    FFT_001 = fft(Channel_001, 1, 1024)
    FFT_002 = fft(Channel_001, 1, 1024)

    Res_BP = FiltBP(Channel_001, 0, 0, 2, 100.0, 1000.0)
    Res_BS = FiltBS(Channel_001, 0, 0, 2, 100.0, 1000.0)
    Res_HP = FiltHP(Channel_001, 0, 0, 2, 100.0)
    Res_LP = FiltLP(Channel_001, 0, 0, 2, 100.0)

    Res_LP_ = FiltLP(Channel_001, 0, 0, 2, 100.0)

    sample = GetSampleCount(Channel_001)
    GetSample = Channel_001 * 0 + sample
End
```

Test procedure and results

imc Online FAMOS:

Handling a 100kHz channel, the imc Online FAMOS test program is at the edge of its capacities. The virtual channel "GetSample" continuously shows the limit value. In the curve window, the distance between the analog channel "Channel_001" and the calculated channel continually increases. As soon as the RAM-buffer of 1 second is no longer sufficient, data overflow occurs. This results in a **loss of data**.

Endurance test (= 24h) capability is just possible in this example with 2xFFT + 4 filters.

imc Inline FAMOS:

Endurance test capability still prevailed with 25xFFT + 50 filters, which is more than 12 times the performance of imc Online FAMOS. Here, the qualification for endurance test capability is considered to be simultaneous display of the analog and the virtual channel in the curve window.

As well, significantly more substantial calculations are possible in imc Inline FAMOS. However, depending on the performance of the computer's respective hardware configuration, the curve window will **eventually no longer display the data in real time**.

Since the data storage management is handled by Windows, this moment in time is not predictable. With a 24-fold load (50xFFT + 100 filters) by comparison with the OFA experiment, this hardware experienced a **data overflow in the curve window** after 1.5 hours. However, there was no resulting loss of data since the data can be retrieved after the end of the measurement. When true data loss would occur in the computer is not possible to determine.

This result applies exclusively for the hardware and software settings specified!

Conclusion

The dynamic data storage management provided by WINDOWS enables multiple times the computation power of the fixed RAM-buffer structure in the device. This is scalable in accordance with the processor speed and PC's RAM size. imc Inline FAMOS is thus future-proof, able to keep pace with technological progress without requiring replacement of the actual measurement hardware.

11.1.2 imc Online FAMOS - System requirements and limitations

imc Online FAMOS and imc Online FAMOS Professional

With the exception of imc BUSLOG, all devices which support imc STUDIO are also able to run imc Online FAMOS (or imc Online FAMOS Professional).

Maximum amount of variables in imc Online FAMOS:

Internally, imc Online FAMOS administers up to 999 variables (single values + virtual channels). However, be aware that only a maximum of 512 channels can be administered per device.

Note

If more than 500 variables are renamed when performing editing in the imc Online FAMOS Assistant before the Assistant is closed, an administration error occurs. This effect does not occur if the Assistant is closed and then later reopened before more than 500 variables were altered.

When imc Online FAMOS is Available?

imc Online FAMOS and imc Online FAMOS Professional can be accessed as an option. It must be enabled for each device individually. If the device is ordered with imc Online FAMOS or imc Online FAMOS Professional, it will be delivered already enabled.

If you wish to upgrade a device with imc Online FAMOS or imc Online FAMOS Professional subsequent to delivery, contact our tech support 8 for more details.

11.1.3 imc Inline FAMOS - System requirements and limitations

The limitations of imc Online FAMOS do not apply to imc Inline FAMOS!

The amount and scope of the tasks which can be performed depends on the computational demands of the functions used and on the power of the PC used. Since all calculations are performed by the PC, the PC's performance determines the possible scope.

Minimum PC requirements[1]	Recommended PC configuration [2]
The minimum requirements for imc STUDIO apply here.	The recommended configuration of imc STUDIO applies here. Additional recommendation: Processor with at least 2+n processor core, where n is the number of Tasks.

1 A system with the minimum requirements is not sufficient for intensive calculations.

2 To ensure smooth performance of calculations, storage and visualization of the measured data, one processor core per Data Processing-Task should be available. Ideally 2+n; where n is the number of Tasks.

Which channels can be combined in calculation operations?

1. **Channels belonging to one device**
   Here, the same prerequisites apply as for imc Online FAMOS. Channels can be combined together when they are subject to the same trigger. E.g. Start/Stop-trigger (BaseTrigger) or a trigger x.
2. **Channels belonging to different devices**
   Channels which are assigned to the Start/Stop trigger (BaseTrigger) can be combined in calculations across multiple-devices.
   Calculations on combinations of triggered channels from different kinds of devices will not work because:
   - Trigger x may be defined differently on differing devices
   - Device triggers do not propagate and synchronize globally, i.e.: a device "A" channel can not be triggered by a device "B" trigger

Process vector and other device variables

All device variables, including process vector and Display variables, network and virtual bits can be processed and set like in imc Online FAMOS.

Process vector variables can not be set up in imc Inline FAMOS. Process vector variables are device variables explicitly created for a specific device, e.g. via imc Online FAMOS, via the plug-in Setup or via Automation.

Process vector variables can be is saved and restored exclusively on the device.

Refresh/writing rate of the device variables

The reading and writing access to the devices is limited.

11.2 imc Online FAMOS and imc Inline FAMOS

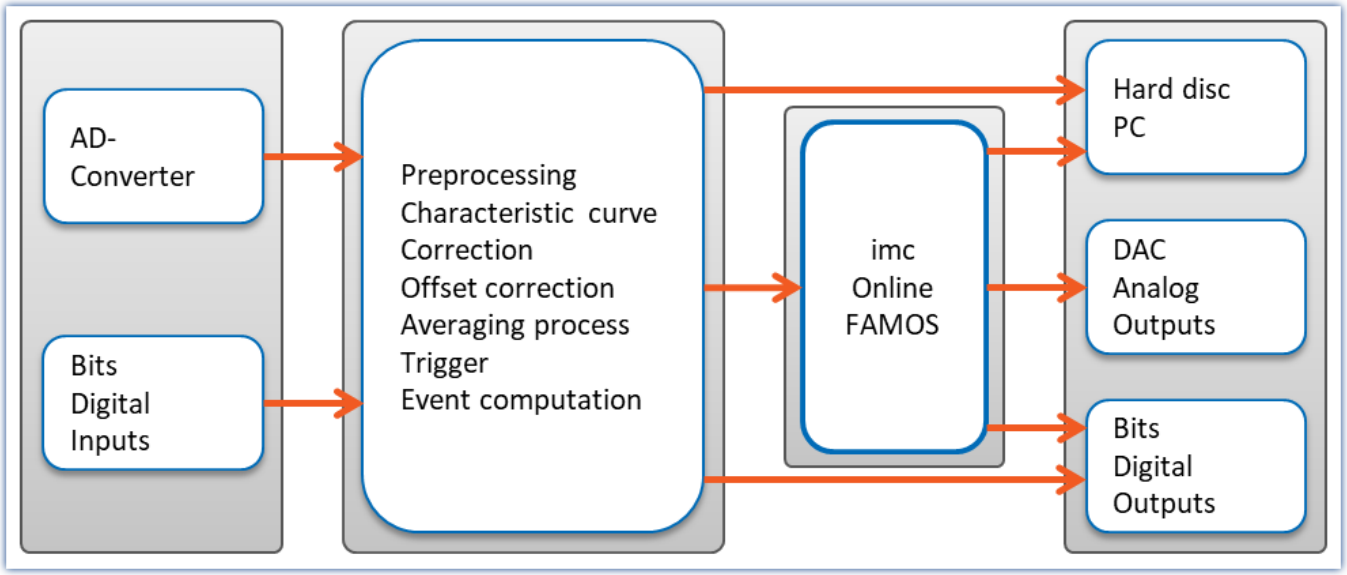
imc Online FAMOS

imc Online FAMOS is a powerful extension for imc DAQ systems. It offers a variety of real-time functions for pre-processing and signal analysis. The mathematical analysis functions are executed on the signal analysis platform integrated in the measurement device. This means that analysis results are available immediately and also independently of the PC. Such pre-processing can also yield significant data reduction and thus reduce the amount of data to be exchanged between the DAQ system and the PC.

The results are available in imc STUDIO as virtual channels. Virtual channels contain data calculated on the basis of existing input channels. This may be, for example, an averaged input channel, or the difference between the signal values form two input channels.

The virtual channels can be redirected to DAC (digital-analog-converter) or DIO (digital input/output, bits) outputs on the device. For example, you compute the difference between two analog input signals (digitally), and convert the result back to an analog signal at the DAC output. The DIO outputs can be used, for example, to set a digital bit whenever a certain signal level has been reached.

The following diagram shows how data stream through the device and which of them can be accessed by imc Online FAMOS:



Analog/ digital conversion of analog input data and/ or acquisition of digital input data is carried out on the unit's main board. All of these input data are usually subjected to a certain amount of preprocessing, for example, characteristic and offset correction, event and trigger computations etc. The trigger system can also be used to directly set output bits.

Instead of direct transfer to the PC, the data streams with measured channels signals are rerouted through imc Online FAMOS. These are imc Online FAMOS's inputs. Results calculated by imc Online FAMOS are mostly virtual channels which are passed on to the PC like other input channels. Additionally, note that even digital output bits and the DACs can be outputted as virtual channels.

imc Inline FAMOS

Processing and analysis of measured data during a running measurement

imc Inline FAMOS is a functions package for Data Processing.

imc Inline FAMOS enables calculations to be performed on data streams from the measurement currently runnning. The calculations are performed on the PC, taking advantage of the PC's processing power. By contrast, with imc Online FAMOS, the calculations are performed by the device.

Scope of functions 846:

A number of pre-defined functions are available for calculation purposes. Most of the imc Online FAMOS functions are available in the same way and with the same syntax as in imc Inline FAMOS. There are a few additional functions exclusively in imc Inline FAMOS.

Cross-device calculations

In contrast to imc Online FAMOS, imc Inline FAMOS provides the ability to apply calculation operations to channels belonging to different devices, if the channels are assigned to "*BaseTrigger*" (measurement Start/Stop).

Displaying results:

The results generated are treated as device variables/channels. They are configured on the Setup pages (e.g. Storage) and can be displayed on Panel pages.

Tasks:

Multiple independent, complete calculation sequences (**Tasks**) can be processed in parallel. The maximum possible scope/amount of these tasks depends on the computational resources required by the functions used, in conjunction with the computational resources available to the PC used.

Note Notes on the product names

This documentation applies to imc Online FAMOS and imc Inline FAMOS. For the purpose of easier readability, the product names are not spelled out in their entirety when referring to both products.

imc Online FAMOS: OFA

imc Inline FAMOS: IFA

11.2.1 Overview

OFA/IFA works with streams of binary data.

Take a simple example where the difference between two channels is computed:

```
Difference = Input1 - Input2
```

If data from both `Input1` and from `Input2` are present, then the `Difference` is computed and assigned to the virtual channel `Difference`.

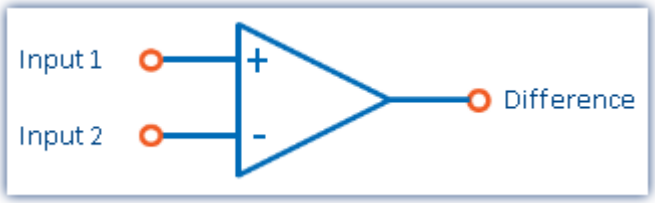
When measurement is started, the device performs the following procedure:

Repeat while measurement continues
Wait for **`Input1`** data
Wait for **`Input2`** data
Retrieve last value of **`Input1`** from memory
Retrieve last value of **`Input2`** from memory
Compute difference
Assign computed difference to virtual channel: **`Difference`**

The virtual channel only contains values if the channels on which its calculations are based contains measured values. This means that only after the input channel is triggered is it possible for the virtual channel to be calculated. When the input channel's signal is concluded, the virtual channel is also concluded.

In the diagram of the internal process above, you see a "Repeat while..." loop. In the equation defined, `Difference = Input1 - Input2`, however, there is no loop. In spite of this, a loop is running internally in order for the term `Difference` to be calculated while the input channels are returning input channels.

Calculation of the difference can be compared to the following simplified electrical circuit diagram:



Schematic circuit diagram of a calculated difference

Output of an analog output

We now consider assigning the difference to an analog output channel (DAC channel):
`DAC1 = Input1 - Input2`

The following steps are executed internally:

Repeat while measurement continues
Wait for **`Input1`** data
Wait for **`Input2`** data
Retrieve last value of **`Input1`** from memory
Retrieve last value of **`Input2`** from memory
Compute difference
Assign computed difference to **`DAC1`**

The procedures for calculating a virtual channel and for generating a DAC output are the same. Thus, the DAC can be referred to as a virtual channel identified by its permanent name (a system name). The DAC is not saved but outputted directly to the respective output terminal. The output to digital output bits proceeds in the same way.

11.2.1.1 What happens when imc Online FAMOS overflows?

General notes on data overflow in software

OFA/IFA operates without synchronization to the data capture. This means that OFA/IFA works through it program in an uninterrupted loop.

OFA/IFA working without overflow

If the data rate is low enough, and there are no new samples available for calculations, then there is nothing for OFA/IFA to do.

If the online calculation is so complex that new samples accumulate by the time the calculation is done, then OFA/IFA enters the new data into the next calculation iteration. If the average number of new samples remains equal, there is no overload.

OFA/IFA overflows

So what happens if the data rate is so high that the number of new samples increases after every iteration? In such a case, OFA/IFA falls behind in performing the calculations. The amount of data increases and with it the time required for calculation.

You will notice this overflow if you display the input data and the virtual channels in a curve window. The virtual channels' values are still pending calculation and are not yet visible, while the input data is already on display. In consequence, the gap between the input data and the virtual data grows at the end of the channel graph.

The following applies to imc Online FAMOS: For brief measurements, of course, this is tolerable if the RAM-buffer duration 427 is adequate. But if the accumulated data exceeds the RAM buffer depth, the software will announce a data overflow!

Special case applicable to imc Online FAMOS: Overflow in the synchronous task

When using the Synchronous Task 857, it is necessary to additionally ensure that the commands are performed within the allocated time interval. If this does not succeed, the device announces the overflow with `LED_06` and activation of the beeper. When the function `SyncOverload` is called in the synchronous task, neither `LED_06` nor the beeper are activated. Instead, the overflow can be indicated using a variable.

Example

```
OnSyncTask( 0.01 )
    if Greater( pv.Channel_002, 5)
       Virt_Bit01 = 1
        ; More functions...
    else
        Virt_Bit02 = 1
        ; More functions...
    end
    ; Upon overflow, the virtual channel changes from 0 to 10
    pv.SyncOverloaded = SyncOverload( 10 )
End
```

11.2.1.2 Source text with Control commands

Conventional **source text without control commands** mainly provided the ability to define virtual channels and subject them to calculations. An OFA/IFA program is executed line-by-line; all instructions entered are carried out in the order they appear in the program.

If the **Control Commands** are activated, the ability to use branching instructions is provided. Program branching can be achieved using conditions and case differentiation, for instance: "if condition is fulfilled, carry out one operation, otherwise carry out other operation". Thus, different procedures can be defined in OFA/IFA to depend on values of variables.

Also, operations can be directly assigned to various states within the course of a measurement (e.g. at the start, end, or during the measurement). An example shall illustrate the difference between "with" and "without" control commands:

Example

```
OnTriggerStart( BaseTrigger )     ; to be carried out at start of measurement
    Reset = 1                    ; Reset is a digital output
End

OnTriggerMeasure( BaseTrigger )   ; to be carried out during measurement
    If Temperature > 25          ; branch, querying temp. channel signal
        Reset = 1
    Else
        Reset = 0
    End
End
```

At the start of the measurement, the digital output bit *Reset* is activated. During the measurement, Reset will be set depending on the Temperature.

More information on syntax is presented in the section: "*Variables and Syntax with Control Commands* 872".

11.2.1.2.1 Transforming to imc Online FAMOS with/without Control commands

By default, OFA/IFA starts without control commands.

Activate the Control Commands via the context menu 865 in the Editor or the menu item "*Extra*" > "*With control commands*" (via the menu applies only to imc Online FAMOS):

Menu item	Description
With control commands	Deactivates/activates the Control Commands.

Upon making this switch, control commands are automatically added to the source text in the Editor. To prevent making this switch inadvertently, it must first be confirmed in order to take effect.

The control commands added automatically are:

Control command	Description
`OnInitAll`	initialization prior to first measurement, or for imc Inline FAMOS: after starting the Task
`OnAlways`	for code segment performed constantly
`OnTriggerStart`	for code segment performed once at the start of a measurement
`OnTriggerEnd`	for code segment performed once at the end of a measurement
`OnTriggerMeasure`	for code segment performed for duration of measurement

If operations were already in the editor when the switch is made, these operations are automatically assigned to the control commands `OnAlways` and `OnTriggerMeasure`. Usually, no revisions are necessary.

Deactivating Control Commands

Deactivation of the control commands is also effected by means of the menu item stated above.

Only such operations are directly transferred which can also be used "*without control commands*", while all other operations still appear, but commented out. After such a transformation, alteration of the source text is usually also necessary, and will sometimes cause significant reduction of functionality.

As a rule, it is inadvisable to port a program created with control commands to "*without control commands*".

11.2.1.2.2 Additional groups in the function list

When you activate the Control Commands, then among others, the following three additional groups appear in the function list: "*Controls*", "*Control functions*" and "*Comparison operators*".

The "*Controls*" group lists all control commands supported. This comprises control commands for conditions and case differentiation (e.g. `IF`), on the one hand, and on the other hand control commands concerning various measurement process states, e.g. `OnTriggerStart` (for operations performed at the start of a measurement).

The "*Control functions*" group contains special functions; such as timer functions.

The "*Comparison operators*" group contains comparison operators such as `>`, `<`, `<>` and `=`. As well, the logic operators `AND`, `OR` and `NOT` are provided here.

Reference	Function Reference

The functions are described in the chapter "*imc Online/Inline FAMOS function reference* 896".

11.2.1.3 imc Online FAMOS Professional

imc Online FAMOS Professional is the package for effective use of the process vector and for operation of the measurement device on the test station. By this means, it is possible to perform monitoring, and open- and closed loop control functions. The package also provides a significant performance enhancement for all purely measurement engineering applications (data-logger operation) in calculating virtual channels.

With imc Online FAMOS Professional you have

- Performance enhancement 857
- Synchronous Tasks 857 including accessories
- Direct access to the process vector 1141
- PID-Controller 886

11.2.1.3.1 Performance enhancement

imc Online FAMOS Professional serves to accelerate online computations. Thus it increases the system's maximum possible throughput without causing data overflow in long-term measurements.

Note Internal information

The accuracy is not reduced by that feature. The increased speed is achieved by as many functions as possible using the signal analysis platform's internal memory. Accessing commands in such memory space can happen much more quickly.

However, this memory is much smaller, so that it's usually not possible to keep all program components within it. By means of the imc Online FAMOS Professional, the compiler shifts the functions into the internal memory until it's full. Therefore those functions whose gain on speed is the highest will be preferred.

In general all functions are considered. But memory intensive functions will be preferred, because they speed up most.

Examples

- FFT: 1,5 –2,5x
- Basic calculations: 2x
- Digital filters: 3-4x

11.2.1.3.2 Synchronous tasks

Along with the control command `OnTimer`, imc Online FAMOS Professional **with control commands** includes a control command for deterministic reactions. At a specified clock rate, imc Online FAMOS' otherwise asynchronous evaluation process is interrupted by a higher priority task. During this interrupt, the command lines entered for the synchronous task are executed.

```
1  ⊞ OnInitAll...
3
4  ⊞ OnAlways...
6
7  ⊟ OnSyncTask(0.1)
8        Virt_Bit01 = 1 - Virt_Bit01
9        DOUT001_Bit01 = Virt_Bit01
10   End
11
12 ⊞ OnTriggerStart(Trigger_48)...
14
```

- Enhancement of OnTimer
- True interrupt handler
- Precise timing
- Access to input channels via the process vector
- Loop control

The user can configure up to **5** synchronous tasks. The cycle times extend from **100µs up to 1s** in steps of 1-2-5. With Synchronous Tasks, there is an effective way to access the current measurement values with the help of the process vector.

By this means, monitoring, and open- and closed-loop control functions can be realized. In particular, the switch/case control commands can be used to achieve state control.

In Synchronous Tasks, not all of imc Online FAMOS' functions are allowed. This pertains to functions, which use a collection of samples like `FFT`, `Mean`, `RMS` etc. But comparison functions, basic math functions and control functions are possible.

In imc Online FAMOS, *closed-loop control* is only possible in a Synchronous Task: two-point controls and PID-controls are included with the Synchronous Task functions. The PID-controls are provided with a variety of additional functions, e.g. controller block, output limiting, and change of input parameters during operation.

It is possible to use the vector functions `VMax`, `VMean`, `VMin`, `VSum`, `VRMS`, `VIsAnyGreater`, `VValueAtXValue`, `VXValueOfMax`, `VXvalueOfMin`, `VXValueWithYValue` in synchronous tasks. In the synchronous task, the functions named can be used on vectors generated using the function `VectorFromFile`. Also supported are functions `Monoflop`, `MonoflopRT`, `JKFlipFlop` and `RSFlipFlop`.

Restrictions

LEDs can't be controlled inside a synchronous task. Instead a virtual bit has to be controlled inside the `SyncTask`. Than, that virtual bit set the LED in `OnAlways`.

11.2.2 Operation

11.2.2.1 Opening the editor

imc Online FAMOS

Select "imc Online FAMOS" from the Ribbon "*Home*" (or "*Setup-Configuration*"). The imc Online FAMOS Editor is displayed in a dialog.

Note

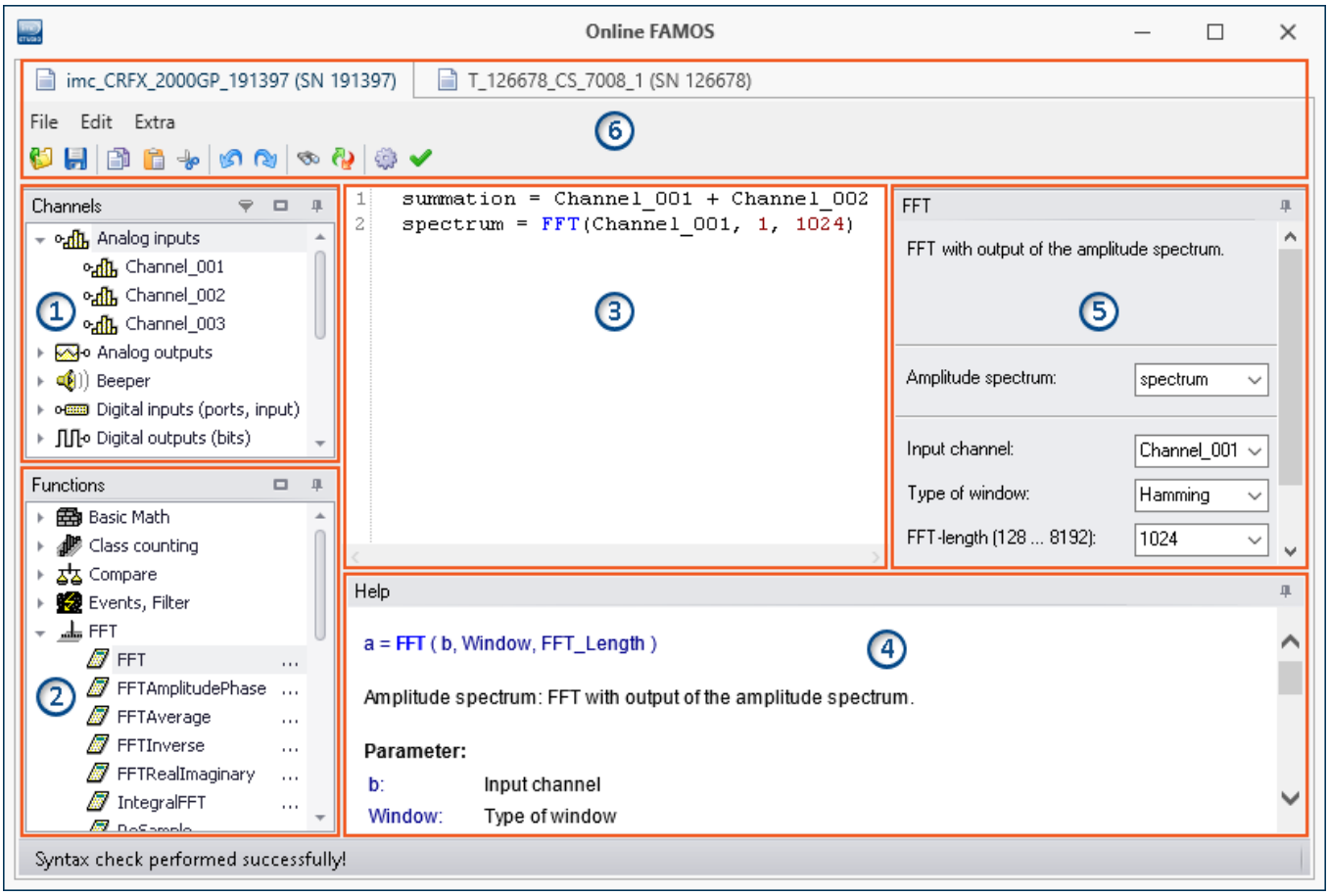
If the menu item imc Online FAMOS is not accessible, check in your calibration certificate whether your device comes with this optional feature. For the software to detect whether imc Online FAMOS is available, the device must have been connected at least once.

imc Inline FAMOS

Dialog: Select "imc Inline FAMOS" from the Ribbon "*Home*" (or "*Setup-Configuration*") to start imc Inline FAMOS. The imc Inline FAMOS Editor is displayed in a dialog.

Embedded: Go to the Data Processing via the Navigation pane. There is a page tab (Task:) for the respective function package imc Inline FAMOS . If it is not yet selected, click the left mouse button on the Task name. The imc Inline FAMOS Editor is displayed in the main window.

11.2.2.2 User interface



The window is subdivided into six regions:

1. Variables list 860
2. Function list 860
3. Editor 860 (text box) for the calculation operations
4. Help 861
5. Formula assistant 861
6. Menu and actions 863 (only in imc Online FAMOS)

Region 1: Variables List

Here is the list of all available variables supported (imc Online FAMOS: device variables | imc Inline FAMOS: among others device variables and user-defined variables).

You can group the variables according to the variable type. With or without grouping, a symbol indicating the respective variable type appears before each name. Variables which were created in the current Task have a green symbol.

Icon	Description	Icon	Description
	active analog input channels and incremental counter channels		Digital outputs (bits) (of all DIO-ports)
	Digital inputs (ports)		Analog outputs (DAC)
	Digital outputs (ports)		housing LED's
	calculated virtual channels		Virtual bits
	calculated local channels		Ethernet bits
	calculated local single value variables		Display variables
	Process vector variables		Beeper (miniature speaker)
	Digital inputs (bits) (of all DIO-ports set for "*Bit-input*")		Trigger which starts an analog channel. Only those triggers are listed to which at least one channel is assigned.

Region 2: Functions List

Here is a list of all available mathematics functions and characters. With the help of the functions, you can perform calculations on the channels and variables. As a result, you obtain virtual and local channels or local single value variables.

The mathematical functions are always updated instantaneously. To learn about their workings, see the Help-box or the Functions Reference.

Region 3: Editor for the calculation operations

In the text box, enter the calculation operations. You may insert carriage returns and spaces as desired. However, any instruction must be coded in its entirety on one line!

Adding a variable or function to the editor

There are multiple ways to add an element to the editor:

- by **double-clicking** on the element
- by applying the **Drag&Drop** technique to the editor
- by calling the **Formula Assistant**: using the Formula Assistant, it is easy to set function parameters and enter the functions into the editor.
- by **input** with **auto-completion** support
  The "first suggestion" from the list is applied by means of the Tab-key. When the ENTER-key is pressed for the "first suggestion", a line break is inserted. When navigating through the list of suggestions, either TAB or ENTER can be used to apply the suggestion.

Functions are entered into the editor along with parenthesis.

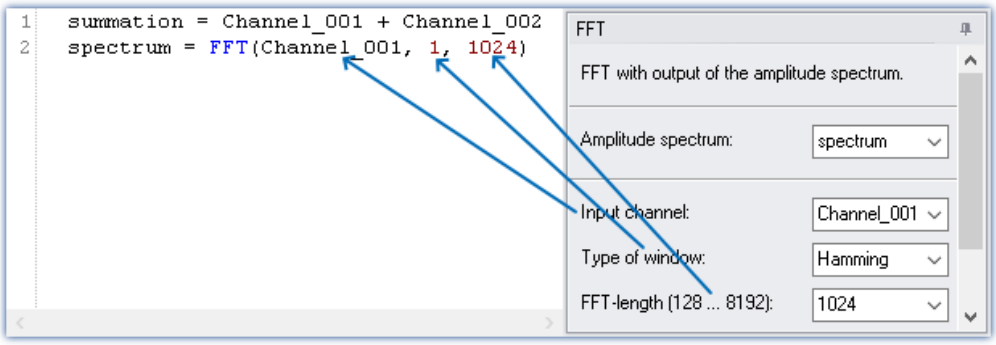
Region 4: Help

Here you find an explicit description of the functions and of the variables' properties.

To open the help text for a variable or a function, left click the mouse to select the desired item. The editor remains unchanged.

Region 5: Formula-Assistant

The Formula-Assistant provides support in parameterizing the functions. The Assistant shows the function selected in the editor.



Formel-Assistent

What parameters are needed depends on the math function selected. For instance, for the Rainflow-function, there are 10 parameters to set. Procedures, in contrast to functions, don't have result values.

For the result, a variable name can be entered manually or an output channel can be selected from the pop-down list.

Changes to the parameters in the Formula-Assistant are immediately adopted in the editor and vice versa.

11.2.2.3 Brief introduction

The Formula Assistant helps you in selecting and parameterizing a mathematical function.

Clicking on a function in the Function List calls a help window displaying the associated help text. You can magnify the help window if desired.

Enter the function along with its parameters in the Editor. You may wish to use the Formula Assistant for this purpose. To assign an operation to a virtual channel, use an equals sign. The name for the virtual channel can be freely specified.

Example

```
summation = Channel_001 + Channel_002
difference = Channel_003 - 5
DAC_01 = difference
```

In the example, we have assigned a sum and a difference to two virtual channels as well as outputting channels on a DAC channel.

Note Note on analog outputs (DAC)

The analog outputs of the measurement device have an output voltage range of -10 V...10 V. You can therefore only assign values lying within this range! A scaling in the plug-in Setup for the DAC channels is taken into account.

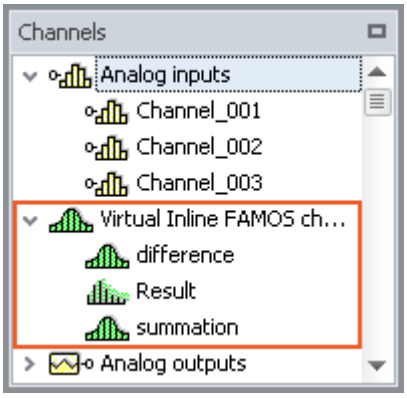
Adding a variable or function

- You can use Drag&Drop to take names from the Variables list into the Editor. Or, alternatively, double-click on the name to insert it in the editor.
- Analogously, you can move functions from the Function List to the editor using Drag&Drop. Just double-clicking on a function's entry in the Function List move the function into the editor.

Definitions for each individual function are made on its own **separate line**.

Check / Syntax check ✔ |

Click on the button to perform a "*Syntax check*". If an error occurs, a notification will appear in the status bar! The faulty portion will appear highlighted in the Editor.



If the syntax check concludes without an error, all newly defined virtual channels and variables will be adopted in the Channels list.

Optional: Saving the source text

Extra saving of the source text is not necessary, since the entries are saved with the experiment. However, it is possible to create an extra copy of the source text. This file can be stored externally, but is not used in the experiment. The source text file can be re-imported at a later time or used in a different experiment.

To save the source text, open the Editor's context menu and select "*Save source text*" ( ).

imc Online FAMOS - Close

Close imc Online FAMOS now. In the table of channels in the Setup, the newly defined channels appear as additional **virtual channels**.

Applying the imc Inline FAMOS-configuration

If the Syntax-Check concludes without finding an error, the Task will still not run. As soon as the configuration has been **applied**, the Task starts and calculations are performed. To do this, click on the button *Apply* ( ).

In the table of channels in the Setup, the newly defined channels appear as additional **virtual channels**.

11.2.2.4 Menu

The menu is not included in imc Inline FAMOS. All functions can be accessed via the "context menu 865". The description of the Data Processing menu ribbon is presented here: "*Data Processing*" > "*Ribbon* 1100"

File - Menu

Menu item	Hot key	Description
Load source text	(Shift + F2)	Loads a text file.
Save source text	(Shift + F3)	The contents of the editor are saved in a text file. This does not replace the Save to Configuration command!
New	-	This clears the contents of the editor and deletes the virtual channels from the Variables List.

Edit - Menu

Menu item	Hot key	Description
Undo	(Ctrl + Z)	Reverses the last change in the Editor. This function can be used multiple times in succession.
Redo	(Ctrl + Y)	Restores the previously reversed change. This function can be used multiple times in succession.
Undo all		Reverses any changes which have been made to the source text since the Editor was called.
Find	(Strg + F)	Finds texts in the imc Online FAMOS editor.
Find and Replace	(Strg + H)	Finds and replaces texts in the imc Online FAMOS editor.
Copy	(Ctrl + C)	Copies the highlighted region in the Editor to the Clipboard.
Paste	(Ctrl + V)	Inserts the contents of the Clipboard at the highlighted location.
Cut	(Ctrl + X)	Extracts the highlighted region of the Editor and moves it to the Clipboard.
Delete	(Ctrl + Del)	This deletes the selected area from the editor.
Properties 881	(F5)	Opens the Properties window. There, the properties of virtual channels can be edited subsequently.
Syntax check	(F6)	The source text is checked for errors. If an error occurs, a notification appears in the status bar! The faulty spot is highlighted in the Editor. If the Syntax-Check concludes without any error located, the status bar reads: "Syntax-Check performed successfully!". All newly defined virtual channels and variables are adopted into the Variables list.

Extras - Menu

Menu item		Description
	Options	It is possible to deactivate automatic LED-flashing. (See LED-flashing during measurement 893)
	Supplemental files	See "*Supplemental files* 891" - *Menu item is only available for devices of firmware group B* 187.
	Directories	In imc STUDIO, the default folder can not be changed. See "*Supplemental files* 891".
	imc Online FAMOS with control commands	Activate control commands 855.
	imc Online FAMOS without control commands	Deactivate control commands 855.

11.2.2.5 Context menu

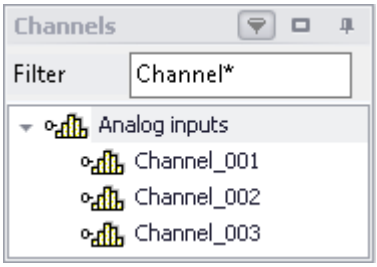
Editor

Right-clicking in the editor calls the following context menu:

Menu item	Hot key	Description
Load source text	(Shift + F2)	Loads a previously saved source text file and thus overwrites the current configuration.
Save source text	(Shift + F3)	Saves the source text and/or the properties at a specific location.
Copy	(Ctrl + C)	Copies the highlighted region in the Editor to the Clipboard.
Paste	(Ctrl + V)	Inserts the contents of the Clipboard at the highlighted location.
Cut	(Ctrl + X)	Extracts the highlighted region of the Editor and moves it to the Clipboard.
Undo	(Ctrl + Z)	Reverses the last change in the Editor. This function can be used multiple times in succession.
Redo	(Ctrl + Y)	Restores the previously reversed change. This function can be used multiple times in succession.
Syntax check	(F6)	The source text is checked for errors. If an error occurs, a notification appears in the status bar! The faulty spot is highlighted in the Editor. If the Syntax-Check concludes without any error located, the status bar reads: "Syntax-Check performed successfully!". All newly defined virtual channels and variables are adopted into the Variables list.
With control commands	-	Deactivates/activates the Control Commands 855.
Properties	(F5)	Opens the Properties window. There, the properties of virtual channels can be edited subsequently (see Virtual channels 867).
Show line numbers	-	Turns the display of line numbers in the Editor on/off.
Show folding	-	Turns on/off the code-folding in the Editor for the control commands, used to group logically associated source text sections. OnTriggerStart(Trigger_48) End
Collapse foldings	(Ctrl + J)	All folds are folded.
Unfold foldings	(Ctrl+Shift + J)	All folds are folded out.
Autocomplete	(F7)	The automatic text predicting function provides you with suggestions of possible terms while you are typing. You can use the Tab-key to adopt the current selection in the Editor. • The "first suggestion" from the list is applied by means of the Tab-key. • When the ENTER-key is pressed for the "first suggestion", a line break is inserted. • When navigating through the list of suggestions, either TAB or ENTER can be used to apply the suggestion.

Variables List

Right-clicking in the Variables List calls the following context menu:

Menu item	Description
Tree view	The variables are displayed in groups by channel type.
List view	All variables are displayed in a list. The channel types are not indicated explicitly. The type is denoted by the symbol in front of the name.
Expand tree	Expands the groups in the Tree view.
Collapse tree	Collapses the groups in the Tree view.
Sort by types	Sorts by channel type in the List view.
Sort by names	Sorts by name in the List view.
Filter	Shows a filter bar.  *Filtering the channel list with Wilcards*

Function List

Right-clicking in the Function List calls the following context menu:

Menu item	Description
Tree view	All functions are displayed in alphabetical order by function group.
List view	All functions are displayed as an alphabetically ordered list.
Expand tree	Expands the function groups in the Tree view.
Collapse tree	Collapses the function groups in the Tree view.

Help

Right-clicking in the Help box the following context menu:

Menu item	Description
Copy all	Copies the complete text from the current help text to the Clipboard.
Copy example	Copies any existing examples from the current help text to the Clipboard.

11.2.3 Variables and syntax

11.2.3.1 Virtual channels and local variables (channels and single value)

Defining virtual channels

Virtual channels are defined in the editor.

A channel assignment consists of

- the virtual channel's freely selected name,
- the equals sign, and
- an arithmetic expression, which may, for example, include an already existing channel.

Example

```
Sum = Channel_001 + Channel_002
CorrectedOffset = Channel_001 + 5
Stretch = 2 * Channel_003
```

You can also use parentheses and minus signs:

```
Complicated = -( 3 + 4 * ( Channel_001 + 1 ) )
```

Function arguments must be written in parentheses:

```
SquareRoot = sqrt ( Channel_001 )
```

Formulas which are not permitted:

Incorrect	Explanation
`a = DAC1`	This is not allowed because an analog output (DAC) can **only be written** to but not read from.

Local channels and single value variables

If you need intermediate results to compute a virtual channel, it is possible to keep them local by prefixing an **underline** to their names. This applies both to single values and to channels.

Access to local variables is twice as fast as normal virtual channels or single values. This increases the computation power. The disadvantage is that these variables are then no loinger visible to the PC.

Local variables appear in the Variables List with a trash bin symbol when they were created but no longer used. For example, if they became obsolete in the process of developing the source text but were forgotten rather than deleted.

Creating local channels

A virtual channel remains local if the variable's name begins with an underline "_".

Example

```
_LocalChannel = Channel_001 * 3
Virtual1 = _LocalChannel + 1
Virtual2 = _LocalChannel + 2
Virtual3 = _LocalChannel + 3
```

Creating local single value variables

Local single value variables are assigned once and can later be used in subsequent lines. A single value is an isolated number without any further properties. Single values only exist locally and do not require an underline: "`a = 1`" or "`_a = 1`".

Example

```
constant = 3 + 4 * sqrt ( 5 )
Virtual1 = Channel_001 * constant
Virtual2 = Channel_002 * constant
Virtual3 = Channel_003 * constant
```

11.2.3.2 Polling digital Inputs

Digital inputs can be polled by imc Online FAMOS. However, they are not polled at a fixed sampling rate, but rather as fast as the signal analysis platform can process the internal loop. The digital inputs are thus not a data stream, but are in the nature of single values.

A data stream can only be created in conjunction with a channel.

Example

```
BitStream = Channel_001 * 0 + DIO01_Bit11
```

Virtual bits can also be polled:

```
Constant1 = Virt_Bit_04 ; SW only locally possible
```

The return value of a digital bit is 0 for "LOW" and 1 for "HIGH".

11.2.3.3 Setting system outputs

The system outputs must be defined in a single statement. The left side of the assignment contains the system output name. The following system outputs are possible (as long as they are displayed in the Variables List):

- Analog outputs (DAC)
- Digital outputs
- Beeper
- LED
- Virtual bit and Ethernet bits
- Display variables
- Triggers

Analog outputs (DAC)

The analog outputs of the measurement device have an output voltage range of -10V...10V. You can therefore only assign values lying within this range! A scaling in the plug-in Setup for the DAC channels is taken into account.

Definition:

`DAC1 = 5` `DAC2 = Channel_001 / 2`	You can assign constants or data streams to a DAC. With data streams, the most recent valid value appears at the DAC.

Not allowed:

`Virtuell = DAC3`	appearance on the right side of an equation
`DAC1 = 1` `DAC1 = 2`	multiple assignment
`DAC4 = 125`	exceeding the value range

Digital outputs

For digital outputs, the assignment may take either of the values zero or one. 0 for LOW and 1 for HIGH. Internally, any assignment not equal to zero is set to 1. Only exactly zero equals 0.

Definition:

`DIO02_Bit01 = 1`
`DIO02_Bit02 = 0`
`DIO02_Bit03 = STRI(Channel_001, -5, 5)`
	If you use mathematics functions, ensure that the resulting return value is either exactly 0 or 1.
	The Schmitt trigger function is particularly suited for generating digital signals from analog signals.

Not allowed:

`Virtuell1 = DIO02_Bit04`	appearance on the right side of an equation
`DIO02_Bit01 = 1` `DIO02_Bit01 = 0`	multiple assignment

Beeper

The beeper is to be considered a digital output which is directly connected with the beeper. Only a tone of fixed pitch can be emitted.

Definition:

`BEEP1 = 1`
`BEEP1 = 0`
`BEEP1 = STRI( Channel_001, -5, 5 )`
	Assigning 1 (TRUE) to the beeper turns it on and assigning 0 (FALSE) to it turns it off.

LED

Some imc measurement devices have LEDs on the housing, which can be operated as a form of digital output. An LED can either shine or be off. (See also "*LED-flashing during measurement* 893")

Definition:

`LED1 = 1`
`LED1 = 0`
`LED1 = STRI( Channel_001, -5, 5 )`
	Assigning 1 (TRUE) to the LED turns it on and assigning 0 (FALSE) to it turns it off.

Virtual bits and Ethernet bits

Virtual bits and Ethernet bits are set in the same way as digital outputs:

Definition:

`Virt_Bit_01 = 1`
`Virt_Bit_01 = 0`
`Virt_Bit_01 = STRI(Channel_001, -5, 5)`
	Assigning 1 (TRUE) or 0 (FALSE).

Display variables

Display variables are set in the same way as digital outputs. However, they can take on a wider (4Byte) range of values.

Definition:

`DisplayVar_01 = 1`
`DisplayVar_01 = 123.456`
`DisplayVar_01 = STRI(Channel_001, -5, 5)`
	Assigning a number.

Triggers

A trigger can be released not only in response to events as defined in the plug-in Setup, but also by OFA/IFA.

As soon as an assignment of 1 is made to an armed trigger, this trigger is released.

This does not change the arming of the trigger. A trigger can initiate recording of data. OFA/IFA releases the trigger just as if a different event had released the trigger.

Triggers with the combination "*Passive*" or "*Immediate*" cannot be released using this method.

Definition:

`Trigger_01 = 1`
`Trigger_01 = 0`
`Trigger_01 = STRI( Channel_001, -5, 5)`
	By assignment of a 1 (TRUE), the trigger is released.

11.2.3.4 Syntax: channel name

Usually the channel name can be used as variable name directly.

Example

```
; Channel name "My_channel"
Res= FFT( My_channel, 2, 1024)
```

If a channel name begins with a number or contains special characters such as \?+! or a space, the channel name must be set into **curly braces**.

Example

```
; Channel name "123 My channel 100% display"
Res= FFT( {123 My channel 100% display}, 2, 1024)
```

11.2.3.5 Syntax: comment

Single line comment

A complete line or a part of a line can be turned into a comment with a ***semicolon***.

Example

```
; Computation begins here
Sum = Channel_001 + Channel_002 ; Computes the sum
; Computation ends here
```

Multiline comments / Block comments

Several lines can be commented using `(* .....*)`.

Example

```
(* the following lines are commented
Sum1 = Channel_001 + Channel_002 ; computes the sum
Sum2 = Channel_003 + Channel_004 ; computes the sum
Sum3 = Channel_005 + Channel_006 ; computes the sum
this is the end of the commented block *)
```

11.2.3.6 Syntax: multiple channels in a formula

Time Base for Channels in a Formula

If multiple channels are jointly subjected to a function, they must all have the same time base. Since the calculation functions process all a channel's data points one-by-one, the following parameters must be the same for all channels in a formula:

- Sampling Time
- Trigger Affiliation
- Pretrigger
- Measurement Duration

Processing a function on two channels having different sampling rates would require the function to use interpolation. This would vastly increase the demands on the system.

The data must be simultaneous. This means that the channels must be started by the same trigger and with the same pre-trigger duration.

Note

If it is impossible to avoid having differing sampling intervals for the input channels, there are various functions to use to compensate. Two good possibilities are `ReSample` and `Mean`.

11.2.4 Variables and syntax with Control commands

The following descriptions apply to **OFA/IFA with Control Commands**.

In order to be able to use the advanced functions and options offered by the control commands, activate them. More information is presented in the section: "*Source Text with Control Commands* 855".

11.2.4.1 Creating a variable

With activated Control Commands, it is additionally possible to generate single values, process vector variables, as well as local and global data arrays. The values each occupy 4 Byte or 8 Byte and can be created in either Integer or Float format.

They are created in the `OnInitAll` block (see: Types of Variables 879).

Note

Notes on precision

- **int**: a pure numerical value (disregarding factor and offset)
  with 32-bit precision
- **float**: a scaled numerical value (factor and offset are applied)
  imc Online FAMOS: with 24-bit precision
  imc Inline FAMOS: with 52-bit precision

Double-initialization

Avoid double-initialization in the OnInitAll-block. However if this is necessary, the first statement defines the type; e.g.

```
OnInitAll
   int pv.x = 0
   ...
   pv.x = 5
End
```

In any subsequent lines, any differing type will not be applied.

Example 1 Process vector

```
OnInitAll
    ; int creates a variable in Integer format:
    int pv.NewEntryA = 0

    ; without int, the variable is in Float format:
    pv.NewEntryB = 0
    float pv.NewEntryC = 0
End
```

Example 2

```
; Executed once at the start
OnInitAll
   v[2] ; local array
   vs          = VectorStatic( BaseTrigger, 4)
   int VarInt  = 1
   VarFloat    = 0.0
   int pv.Var1 = 0
   pv.Var2     = 0
   VChanReal   = SingleValueChannel( BaseTrigger, 1000 )
End

; run upon start of the measurement
OnTriggerStart(BaseTrigger)
   v[1]  = pv.Channel_001 ; Channel values upon start
   v[2]  = pv.Channel_002

   vs[1] = 0
   vs[2] = 0
   vs[3] = 0
   vs[4] = 0

   VarInt   = 0
   VarFloat = 0.0
   pv.Var1  = 0
   pv.Var2  = 0
End

; Executed during a running measurement
OnTriggerMeasure(BaseTrigger)
   ;...
   if Virt_Bit01
      Virt_Bit01  = 0
      VChanReal   = pv.Channel_001
   end
End

; Executed at the end of every measurement
OnTriggerEnd(BaseTrigger)
   DisplayVar_01 = pv.Channel_001 - v[1] ; difference end - start
   DisplayVar_02 = pv.Channel_002 - v[2] ; difference end - start
End
```

11.2.4.2 Comparison operators

Comparison operators are used for queries within condition statements. Here, we will describe only the `>` operator (Greater), and the other comparison operators `>=`, `=`, `<=`, `<` and `<>` are used in an analogous manner. Comparison operators return results of the BOOL data type, as required for condition statements. Single values and channels can also be used as operands.

Example: "Greater than"-operator

Compares the two operands to check if the first is greater than the second.

```
IsGreater = a > b
```

IsGreater: Result	**A**: 1st operand
	B: 2nd operand

IsGreater = 1, if A is greater than B.

IsGreater = 0, if A is less than or equal to B.

Example

```
OnInitAll
    Value = 0
    VrtBit_01 = 0
End
OnTriggerMeasure( BaseTrigger )
    Value = CurrentValue( Channel_001, 0, 0.0 )
    If Value > 5
        VrtBit_01 = 1
    End
End
```

Combinations of different conditions can be implemented using the operators `AND` (logical And) or `OR` (logical Or); for instance,

```
If VirtBit_01 > 0 AND VirtBit_02 = 0
```

or

```
Virt_Bit01 = DisplayVar_01 < -10 OR DisplayVar_01 > 10
```

11.2.4.3 Defining virtual channels with conditions

- **Deactivated Control Commands:** Virtual channels are always filled with values (according to the clock rate of the parameter channels, etc.).
- **Activated Control Commands:** Virtual channels are only filled with values if they are defined by the currently operable branch of the condition. If the condition in the example below is met (the virtual bit `VrtBit_01` is set), then the virtual channel `VrtChannel_001` is filled with values, otherwise it remains empty:

```
If VrtBit_01 > 0
    VrtChannel_001 = Channel_001 + 10
End
```

If the condition's default branch also defined the virtual channel, then the virtual channel is always filled with values. But then, the virtual channel's particular values depend on the operative branch of the condition, for example,

```
If VrtBit_01 > 0
    VrtChannel_001 = Channel_001 + 10
Else
    VrtChannel_001 = Channel_001 + 20
End
```

If a virtual channel is used in a formula (or if a virtual channel is queried), the virtual channel must be defined in every program branch which leads to the formula or query in question. The following example is only permitted if the default branch of the condition is called. Otherwise, the virtual channel `VrtChannel_001` wouldn't be defined for `VrtBit_01 = 0`.

```
If VrtBit_01 > 0
    VrtChannel_001 = Channel_001 + 10
Else
    VrtChannel_001 = Channel_001 + 20
End
VrtChannel_002 = 2*VirtChannel_001 + Channel_001
```

11.2.4.4 Conditions, case-differentiation and CAN-transmission with channels

If conditions, case-differentiation or CAN-transmission with channels is called in `OnTriggerMeasure`, then the respective current (or most recent) value in the channel is used for the query or transmission. For this purpose, OFA/IFA automatically inserts a `CurrentValue`-function before the querying expression, and then the resulting current value of the channel is retrieved. The following examples illustrate channels subjected to conditions. The behavior of channels in case-differentiation and CAN-transmissions is analogous.

If a physical input channel is used in at least one query (e.g. `If Channel_001 > 0`), the current value of the input channel is noted at the start of `OnTriggerMeasure`. Whenever the physical channel is queried, the single value noted stands in for it:

Coded in OFA/IFA	OFA/IFA automatically transforms this to

`LED_01 = 1`
`If Channel_001 > 0`
`  .....`
`If Channel _001 < 0`
`  .....`
	`LED_01 = 1`
	`_cv = CurrentValue(Channel_001, 0, 0.0)`
	`If _cv > 0`
	`  .....`
	`If _cv < 0`
	`  .....`

If an expression is queried, e.g. `Channel_001 - 1 > 0`, the current value of the expression `Channel_001 - 1` is automatically generated by OFA/IFA and placed before the query, and the single value generated is then queried:

Coded in OFA/IFA	OFA/IFA automatically transforms this to

`LED_01 = 1`
`If Channel_001 + 1 > 0`
`  .....`
`If Channel_001 - 1 < 0`
`  .....`
	`LED_01 = 1`
	`_cv = CurrentValue(Channel_001 + 1, 0, 0.0)`
	`If _cv > 0`
	`  .....`
	`_cv = CurrentValue(Channel_001 - 1, 0, 0.0)`
	`If _cv < 0`
	`  .....`

If virtual channels are queried, e.g. `If VrtChan_001 > 0`, then the current value in the channel is determined and noted after each definition (with activated control commands, virtual channels can be defined at various points in the source text). If the virtual channel is queried, the query retrieves the noted current value of the virtual channel rather than the channel itself:

Coded in OFA/IFA	OFA/IFA automatically transforms this to

`If VrtBit_01 > 0`
`  VrtChannel_001=`
`Ch01+3`
`Else`
`  VrtChannel_001=`
`Ch01+5`
`End`
`LED_01 = 1`
`If VrtChannel_001 > 0`
`  .....`
`If VrtChannel_001 < 0`
`  .....`
	`If VrtBit_01 > 0`
	`  VrtChannel_001 = Channel_001 + 3`
	`  _cv = CurrentValue(VrtChannel_001, 0, 0.0)`
	`Else`
	`  VrtChannel_001 = Channel_001 + 5`
	`  _cv = CurrentValue(VrtChannel_001, 0, 0.0)`
	`End`
	`LED_01 = 1`
	`If _cv > 0`
	`  .....`
	`If _cv < 0`
	`  .....`

Note — **Note on CurrentValue-funktion**

The `CurrentValue`-functions automatically generated by OFA/IFA reflect the option "last value". However, a high sampling rate can lead to situations where many sample values from a channel are still being processed. If "last value" is the valid option for the `CurrentValue` function, this can lead to values being ignored.

To avoid this, there are other options which can be set for the `CurrentValue` function (e.g. use of maximum value, if multiple values are available for the channel). But then, the `CurrentValue`-function, along with the corresponding setting for the option, must be written into the source text; it isn't added automatically.

Note Boolean variables from files

Boolean (True/False) variables can be queried in `If` conditions without comparison:

```
Switch_A = Channel_01 > 5
If Switch_A
   ...
```

OFA/IFA allows this, because the compiler makes the result `Switch_A` a boolean variable by comparing `Channel_01 > 5`.

When **importing variables from files**, however, the compiler cannot always clearly recognize them as boolean variables.
Earlier versions of OFA/IFA accepted this, but this could lead to incorrect queries. The following construct leads to an error message in current OFA/IFA versions:

```
Bools= VectorFromFile("Bools.dat")
Switch_B= Bools[2]
...
If Switch_B ; -> Error message
   ...
```

Therefore If queries in current OFA/IFA programs must always be written with comparison, e.g. :

```
Bools= VectorFromFile("Bools.dat")
Switch_B= Bools[2]
If Switch_B = 1 ; -> OK
   ...
```

This change may cause old experiments that worked with earlier imc STUDIO versions to generate error messages in current versions. In this case the OFA/IFA code must be changed according to the last example.

11.2.4.5 Example - Simplified motor simulation

The use of various control commands is illustrated (in imc Online FAMOS) below in reference to a simplified motor simulation. The voltage and current signals from a motor during run-up, normal operation and wind-down are simulated (for 5 s each). The current state of the simulation can be frozen using a Pause-bit.

```
;-----------------------------------------------------------------------
; In the OnInitAll block, initializations are performed prior to the
; first measurement. All single value variables used in the formulas
; must be initialized here.
;-----------------------------------------------------------------------
OnInitAll
    Status   = 0        ; Initialization of single value variables
    Counter  = 0
    Data     = 0
    PauseOld = 0
    PauseNew = 0
End

;-----------------------------------------------------------------------
; The block OnAlways is run constantly. The Pause-function is
; implemented here. If you click on the Pause bit, the timer is stopped.
; If the Pause-bit is currently deactivated, clicking on it starts the
; periodic timer. After 10ms, the timer ticks at 10ms intervals.
;-----------------------------------------------------------------------
OnAlways
    ; PauseNew is the 1st virtual bit
    if PauseNew <> 0 AND PauseOld = 0           ; Pause-bit is clicked on
        RecordText ( "Pause!!!" )               ; output text
        StopTimer ( 1 )                         ; stop timer
    end
    if PauseNew = 0 AND PauseOld <> 0           ; Pause-bit is clicked off
        RecordText ( "Pause finished!!!" )      ; output text
        StartTimerPeriodic ( 1, 0.01 , 0.01 )   ; start periodic timer
    end
    PauseOld = PauseNew
End
```

```
;-----------------------------------------------------------------------
; The OnTriggerStart block is executed at the start of
; the measurement. When the measurement starts, a periodic timer is
; set up. After 10ms, the timer ticks at 10ms intervals.
;-----------------------------------------------------------------------
OnTriggerStart ( BaseTrigger )
    LED_01 = 1                              ; switch on LED_01
    StartTimerPeriodic ( 1, 0.01 , 0.01 )   ; start timer
End

;-----------------------------------------------------------------------
; The OnTriggerEnd block is executed at the end of a measurement.
; Here, the timer is stopped and LED_01 switched off.
;-----------------------------------------------------------------------
OnTriggerEnd ( BaseTrigger )
    LED_01 = 0                      ; switch off LED_01
    StopTimer ( 1 )                 ; stop timer with the ID 1
End

;-----------------------------------------------------------------------
; The OnTriggerMeasure block runs constantly during measurement.
; The virtual channel Voltage is created under the control of a timer,
; and the virtual channel Current is computed from the channel Voltage.
;-----------------------------------------------------------------------
OnTriggerMeasure (BaseTrigger )
    Voltage = channel_001 * 0 + Data
    _x = Voltage + sin ( sawtooth ( Voltage, 0, 1, 10000 ) )
    rpm = upper ( 0, filtlp ( _x, 0, 0, 3, 1 ) * 60 + _x * 20 )
    _y = diff ( _x )
    Current = 10*filtlp ( _y, 0, 0, 2, 2 )
    if Voltage > 5      ; queries virtual channel Voltage. If the
                        ; current value is greater than 5,
        LED_05 = 1      ; switch on LED_05
        LED_06 = 0      ; switch off LED_06
    else                ; otherwise (current value in channel Voltage <= 5)
        LED_05 = 0
        LED_06 = 1
    end
End

;-----------------------------------------------------------------------
; The OnTimer block is run upon every click of Timer 1. The periodic
; timer, Timer 1, was set up either in OnTriggerStart or OnAlways
; using the function StartTimerPeriodic. In the OnTimer block,
; the numerical values for the virtual channel Voltage are computed.
;-----------------------------------------------------------------------
OnTimer ( 1 )                   ; OnTimer-Block for Timer 1
    switch Status               ; case differentiation based on the
                                ; single value Status
        case 0                      ; case: Status = 0
            Data = Data + 0.02      ; compute the data
            Counter = Counter + 1   ; increment counter
            if Counter > 500        ; query counter
                Counter = 0         ; re- initialize counter
                Status = 1          ; set Status
            end                     ; finish condition
        end                         ; finish case-differentiation
        case 1                      ; case: Status = 1
            Counter = Counter + 1
            if Counter > 500
                Counter = 0
                Status = 2
            end
        end
        case 2                      ; case: Status = 2
            Data = Data - 0.02
            Counter = Counter + 1
            if Counter > 500
                Counter = 0
                Status = 0
            end
        end
        default                     ; default treatment for Status,
                                    ; if Status is not equal to 0, 1 or 2
        end
    end
End
```

11.2.5 Types of variables

The following is a list of the usual variable types which you can define in OFA/IFA. Special types of variable such as CAN-structures, controller-structures, ... are not mentioned here. These are presented in the respective chapters.

Virtual channels ( )

Description	Example	Syntax	Definition
Virtual channel, (Datenfeld equidistant)	`Virt = Channel_001`	various functions	Without control command Otherwise: OnTriggerMeasure
Data array global	`vs = VectorStatic (BaseTrigger, 10)`	Array = VectorStatic (Trigger#, Size)	OnInitAll
Data array dynamic	`vsChan = SingleValueChannel (BaseTrigger, 1000)`	Array = SingleValueChannel (Trigger#, DataRate) various additional functions	OnInitAll
Vector series dynamic	`vChan = VectorChannel (BaseTrigger, 100, 5)`	Array = VectorChannel (Trigger#, DataRate, Size)	OnInitAll
Vector series equidistant	`Virt_fft = FFT (Channel_001, 0, 1024)`	various functions	Without control command Otherwise: OnTriggerMeasure

Local channels ( )

Description	Example	Syntax	Definition
Local channel	`_Local_Virt = Channel_001`	various functions	Without control command Otherwise: OnTriggerMeasure
Local vector, Data array local	`v[10]` `int v[10]`	Array[Size] various additional functions	OnInitAll

Local single value variables ( )

Description	Example	Syntax	Definition
Single value as Float	`Lokal_Var1 = 4` `Lokal_Var2 = 0xA5 ; hex`	Variable = 0	Without control command Otherwise: OnInitAll
Single value as Integer	`int Var1= 1` `int Var4= 0xA5 ; hex`	int Variable = 0	OnInitAll

Single value as Float	`Var2= 0.0`
	`float Var3= 0`
	`Var4= 0xA5 ; hex`
		VariableA = 0
		float VariableB = 0
			OnInitAll

pv-variables (single value;) (only in imc Online FAMOS Professional)

Description	Example	Syntax	Definition
Single value as Integer	`int pv.Var1= 1` `int pv.Var4= 0xFFFF ; hex`	int pv.Variable = 0	OnInitAll

Single value as Float	`pv.Var2= 0.0`
	`float pv.Var3= 0`
	`pv.Var4= 0xFFFF ; hex`
		pv.VariableA = 0
		float pv.VariableB = 0
			OnInitAll

Local texts ( )

Description	Example	Syntax	Definition
Text	`Text = "Hello"`	various functions	

11.2.5.1 Local vectors

Local Vectors in OnInitAll-Block

Local vectors can be created using the function `VectorFromFile` and now also in the `OnInitAll` block:

```
OnInitAll
    Vector = VectorFromFile( "Vector_01.DAT" )
    ; or
    int sv1[VectorLength]
    ; or
    sv2[VectorLength]
End
```

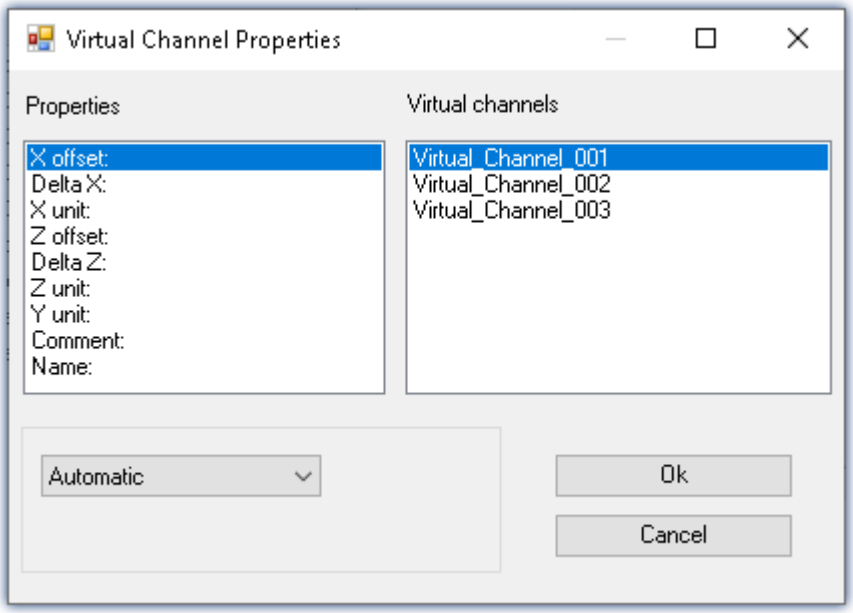
- The vector elements are initialized at 0.
- The vector length of `sv` must be selected appropriately. Make sure to use only possible vector indexes, especially for variable vector indexes.
  `sv[1]…sv[vector length]`
- Local vectors can be used in synchronous tasks, too.
- Elements of local vectors can be used on the left or right side of an expression:
  `Virt_Bit01 = sv[1]`
  or
  `if sv[2] > 0`
  or
  `sv[1] = sv[1] + 1`
- The elements of local vectors can be addressed with variable indexes, e.g.:
  `sv[Index + 2] = sv[Index + 1]`
- Functions which return streaming vectors can be assigned to static vectors:
  `sv2 = FFT(...)`
- The function `GetSampleCount` of group Control functions returns for local vectors the length of the vector. This is useful, if the a local vector has been created with the function `VectorFromFile`. Thus the possible vector indexes for this vector can be checked.

11.2.6 Properties of virtual channels

Already defined virtual channels can subsequently be assigned certain properties. A virtual channel's name, comment and units can be set. If any virtual channels are available as vectors, it is additionally possible to set special vector properties.

To do this, open the dialog: "*Virtual Channel Properties*" via the context menu 865 in the Editor, or via the menu item "*Edit*" > "*Properties*" (via the menu item applies only to imc Online FAMOS):

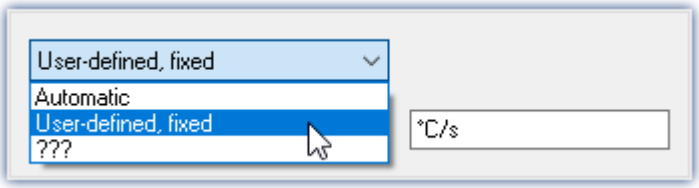
Menu item	Description
Properties	Opens the Properties window. There, the properties of virtual channels can be edited subsequently (see "*Virtual channels* 867").



The virtual channels appear on the right side of the window. Multiple selection is possible for assigning the same properties to several channels.

The various properties are displayed on the left side of the list. You can define one property at a time for the selected virtual channel(s) on the right.

The options for the selected property are made in the drop-down list boxes at the bottom of the window. The default setting for all properties is *automatic*. To define a property select *User-defined* in the list box. Then, a text box will appear in which you can enter the value desired.



The properties of virtual channels can only be set in this dialog. If you wish to change properties which were already set or to make settings for new virtual channels, call the dialog again.

Property	Description
X-offset	The offset in the x-direction can be set to a fixed value for purposes of display in the curve window. Otherwise, the value is automatically determined. This property can only be set for **vectors**.

Property	Description
Delta-X	The distance between two sample values in the x-direction can be set to a fixed value for purposes of display in the curve window. Otherwise, the value is automatically determined. This property can only be set for **vectors**.
Defining X-axis units	For the display of a virtual channel in the curve window, the units of the X-axis can either be determined automatically or given a fixed setting by the user. It is best to specify only SI-units. Do not specify any orders of magnitude such as milli- and kilo- (exception: kg). The same guidelines for units are valid as for defining the units of input channels. For channels whose X-axis is the time-axis, it is recommended to leave the X-axis unit on automatic. In that case, the unit displayed will be "s" for seconds.
Z-offset	The z-offset is the z-value given for the "front" curve in a 3D curve window. The z-offset can either be determined automatically or given a fixed setting by the user. This property can only be set for **vectors**!
Delta-Z	Delta-Z is the distance between two adjacent curves in the curve window. This property can only be set for **vectors**!
Comment	Comments can be added to virtual channels in the same way as with normal input channels. You can use any characters in this field - you can also leave it blank.
Name	Instead of the virtual channel's name which you used when defining it, you can specify a different name here. This saves you the work of renaming all names in the entire source text, for example for the purpose of the display in the curve window. It also makes sense to insert a shorter name in the formulas while stating the whole name here. The names of all channels must be unique within the system. The same limitations apply for constructing a name as for the input channels.

11.2.7 Calculation examples

11.2.7.1 Determining the RMS of a mixed signal's alternating component

The root-mean-square (RMS) of the signals alternating components is to be determined; the signal contains a direct as well as an alternating component.

For example, it is possible to determine the RMS of a noise signal in an amplifier which has an offset voltage. Another use would be for determining the noise component in a supply current.

Problem background:

The signal *u(t)* consists of direct and alternating components and can be represented as $u(t) = u_(t) + u_0$ where u_0 is the signal's direct component.

If one tries to find the alternating component's RMS by using the equation

$$U_ = \sqrt{\frac{1}{T}\int_{t-T}^{T}[u(t) - Meanvalue(u(t))]^2 dt}$$

directly, T being the time over which the data are collected, an error message will be generated when the source text is compiled. The reason is that the RMS, a single value, is to be subtracted from $u(t)$, which is a vectored data stream.; in other words, the data types don't match.

Solution:

If the initial equation $u(t) = u_(t) + u_0$ is incorporated into the definition of the RMS, and it is noted that the mean of the alternating quantity is eliminated, it follows that the sum of the RMS components' squares $U^2 = U_\sim^2 + U_0^2$ equals the square of their sum. This makes it possible to determine $U_\sim$ as the difference of two data sets of matching type.

Example

Letting Ch_01 designate the complete signal, the RMS of the alternating component can be computed as

```
Rms_noise= Sqrt(mean(Ch_01*Ch_01, 1000,1000)- mean(Ch_01, 1000, 1000)^2)
```

11.2.7.2 Determining one or more frequencies' contributions to a signal

Purpose:

The 150 Hz contribution to a non-sinusoid current of mains frequency is to be displayed as a curve plotted over time. A digital output could additionally be set in response to this contribution's exceeding a threshold value.

Problem background:

If the FFT function is applied directly to a measurement, the resulting frequency spacing is given by

$$\Delta f = \frac{f_T}{N}$$

where f_T is the sampling rate and N is the number of data points used, which must be expressed as a power of two (e.g., 1024 = 210). If, for instance, the sampling rate 1 kHz and N=1024 are specified, the frequency spacing is 0.9765625 Hz. This means that it is impossible to determine the 150 Hz line exactly.

Solution:

One could extract exactly 1000 values from the continuous stream of measurement data. This excerpt can afterwards be extended to 1024 points by resampling. The resulting, resampled data set then has a sampling frequency of 1.024 kHz, and the 150th spectral line is located exactly at 150 Hz.

Example

Declare `Current` as the measurement channel to be analyzed, which is sampled at a rate of 1 kHz and whose 150 Hz spectral line is to be charted over time. One appropriate source text would be the following:

```
_I_Channel=VectorizeAndSkip(Current ,1000 , 0 )
_I_Res=ReSample(_I_Channel ,1024 )
Spec = FFT(_I_Res, 0, 1024)
I_150 = VValueAtXValue(Spec, 150)
```

With the function `VectorizeAndSkip`, 1000 samples are extracted from the continuous data stream designated by the variable "Current". A data "skip" is not performed in this instance, as indicated by the zero as the last parameter in the function call.

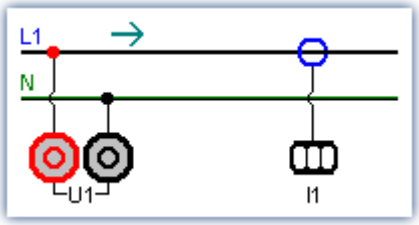
The assigned name, `_I_Channel`, begins with an underline, identifying it as an internal variable. This isn't strictly necessary, but it helps to conserve memory.

Next, the ReSample function is employed to produce 1024 samples. Now the FFT calculation can be performed. The function `VValueAtXValue` makes it possible to extract the Y-value at a specified X-value. In this case, the spectral line #150 is at 150 Hz, since the frequency spacing is 1 Hz, and it is designated by the variable `I_150`. All available functions such as triggering in response to threshold crossing, can be applied to this variable.

11.2.7.3 Power measurements

Here follows the description for the Single-phase, Two-phase- and Three-phase-power measurement.

11.2.7.3.1 Single-phase measurement power1



Single-phase power measurement

Effective value (root-mean-square)

$$y = \sqrt{\frac{1}{T}\int_T x^2 dt}$$

The effective value is the square root of the mean-square of the input signal.

Instantaneous power

$$p = u \cdot i$$

The operation Instantaneous Power supplies the product of two respective sampling values.

Active power

$$P = \frac{1}{T}\int_T (u \cdot i) dt$$

The active power is the average of the instantaneous powers during an averaging interval. This describes the actual power recorded by the user.

Apparent power

$$P_S = U \cdot I$$

The apparent power is the product of the root-mean-squares for current and voltage. The root-mean-squares are calculated according to the algorithm described under the operation effective value (root-mean-square).

Reactive Power

$$Q = \sqrt{S^2 - P^2}$$

The reactive power is the geometric difference between apparent and active power. Apparent and active power are calculated according to the above algorithm. The reactive power is the portion of the apparent power not recorded by the user.

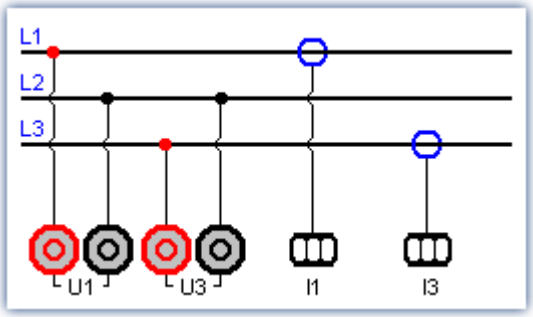
Power Factor

$$\cos(\varphi) = \frac{P}{S}$$

The power factor is the ratio between active power and apparent power. Active power and apparent power are calculated according to the above algorithm. The power factor corresponds to the cosine of the phase shift between current and voltage.

11.2.7.3.2 Two-phase measurement Power2 (ARON)

The power can be determined from only two phase voltages and two conductor currents, assuming that the zero-wire of the three-phase circuit has no current. For that balanced load the third quantity is thus clearly defined. This type of power measurement will be referred to below as an ARON measuring circuit. The ARON measuring circuit is often the only way to measure the power on a user whose star point is not accessible.



ARON measuring circuit

In the triangular circuit, the connection of the measurement unit results analogous to the illustrated star circuit.

Instantaneous power

The operation instantaneous power (ARON) supplies the sum of the products of two respective sampling values.

$$p = u_1 \cdot i_1 + u_3 \cdot i_3$$

Active power

The active power is the average of the instantaneous powers during an averaging interval.

$$P = \frac{1}{T}\int(u_1 \cdot i_1 + u_3 \cdot i_3)dt$$

This describes the actual power recorded by the user.

Apparent power

The apparent power results from the sum of the products of the effective values (root-mean-squares) for the currents and voltages during the individual phases.

$$S = \frac{\sqrt{3}}{2}\int(U_{rms\,1} \cdot I_{rms\,1} + U_{rms\,3} \cdot I_{rms\,3})dt$$

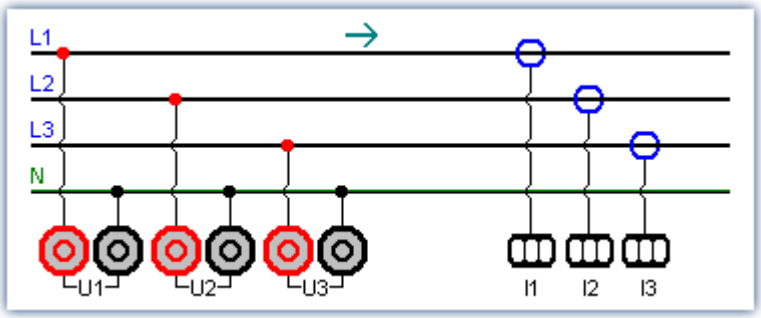
It is required that the symmetry of the supply circuit (three equal voltages with phase shifts of 120°) be stable. The root-mean-squares are calculated according to the algorithm under the operation effective value (root-mean-square).

Reactive power and **Power Factor**

See Single-phase power measurement 884

11.2.7.3.3 Three-phase measurement with N: Power3

Measurements on the three-phase circuit **with** N result according to the following diagram. For measurements on the three-phase circuit **without** N use ARON measuring circuit 884.



with N (Y-connection)

Instantaneous power

The operation instantaneous power supplies the sum of the products of two respective sampling values.

$$p = u_1 \cdot i_1 + u_2 \cdot i_2 + u_3 \cdot i_3$$

Active power

The active power is the average of the instantaneous powers during an averaging time.

$$P = \frac{1}{T}\int(u_1 \cdot i_1 + u_2 \cdot i_2 + u_3 \cdot i_3)dt$$

This describes the actual power recorded by the user.

Apparent power

The apparent power results from the sum of the products of the effective values (root-mean-squares) for the currents and voltages during the individual phases.

$$S = U_{rms\,1} \cdot I_{rms\,1} + U_{rms\,2} \cdot I_{rms\,2} + U_{rms\,3} \cdot I_{rms\,3}$$

The root-mean-squares are calculated according to the algorithm under the operation effective value (root-mean-square).

Reactive power and **Power Factor**

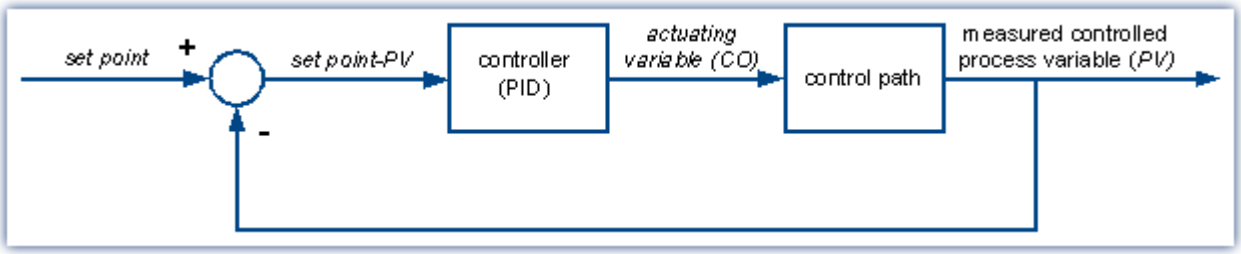
See Single-phase power measurement 884

11.2.7.4 PID-controller

The controller functions are only available if imc Online FAMOS Professional is enabled for the device.

In a controller, a process value (PV) measured by sensors is continually compared with a specified setpoint value. If a system deviation (setpoint – PV) occurs, the controller generates a controller output (CO) which affects the system's energy flow in such a way to cause the process value to approach the setpoint. Then the system remains unaffected by the controller until the setpoint is changed or a disturbance occurs. Whether and how the controlled variable/ process value reaches the setpoint depends on the controller. Depending on the quantity to be controlled and the desired control response, a variety of different types of controller can be employed.

The following is a simplified sketch of a control loop:



Simplified sketch of a control loop

A pure *P-controller* is not able to completely correct disturbances, and can cause oscillation if the P-component (KP) is too large. It may react immediately to disturbances or changed setpoints, but still operates inexactly due to the remaining system deviation.

An *I-controller* (integrator), on the other hand, can completely correct a system deviation, but it does so much more slowly and also has a tendency to lead to oscillations.

A *D-controller* working alone is useless due to its differentiating response, since it only puts out a controller output when the system deviation changes, and thus doesn't correct system deviation as such. It is used in conjunction with other controllers in order to accelerate the correction of system deviations.

A *PI-controller* unites the advantages of a P- with an I-controller, since the P-component quickly reacts to system deviations and the I-components ensures that no system deviation remains. A PD-controller is very quick in providing adjustment thanks to its P- and D-components, but it leaves a persistent system deviation.

A PID-controller is constructed by combining the three basic elements P-controller, I-controller and D-controller. It possesses the positive properties of the individual controllers, i.e. it adjusts quickly, leaves no deviation and reaches the setpoint without overshooting it. However, it has the drawback that the D-component amplifies high-frequency disturbance signals. To prevent this, there is the possibility of specifying an upper cutoff frequency for the D-component. This type of controller is known as continuously acting, since it reacts to any change. Because each component or proportional coefficient like P-component (KP), I-component (KI) and D-component (KD) can be defined separately, then all types of controllers from the simple P-controller to the PID-controller can be achieved. The controller output CO then results from the addition of the three outputs of the P-, I- and D-controllers:

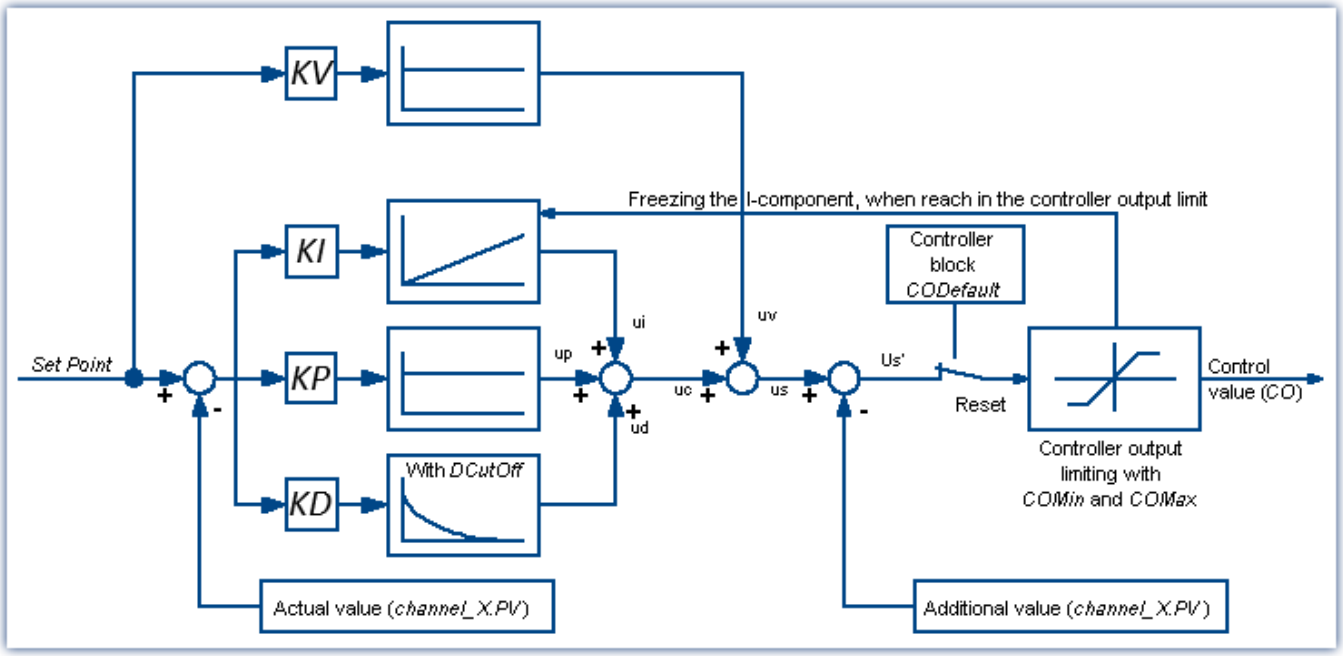
$$CO = KP \cdot (SetPoint - PV) + KI \cdot \int (SetPoint - PV) + KD \cdot (SetPoint - PV)'$$

11.2.7.4.1 The PID-controller in imc Online FAMOS Professional

The constants KP, KI and KD are specified for a continuous (non-discrete) controller and can be changed online at any time during the control process. The PID-controller is computed in the *synchronous task* with the selected cycle time, and the process value PV (any input channel) is provided to it in a process vector.

The entire controller structure in imc Online FAMOS consists, besides the PID-controller, of:

- an input control element (P-element with proportional coefficient KV),
- an additional summation point, via which a further measurement channel can be inputted in the form of an extra value,
- a controller block with adjustable default value and controller output limiting. The controller output limiting also comes with tracking of the I-component when the control variable limit has been reached. Toward this end, the I-component is frozen until the control variable limit is canceled.
- Additionally, it's possible to set an upper cutoff frequency (DCutOff) for the D-component, in order to avoid amplifying noise signals.



This as well as all of the PID-controller's other configuration possibilities are described below and can also be found in the imc Online FAMOS help dialog.

Parameter	Description
KP	**P-component** Use in the controller equation: CO = KP * (SetPoint - PV) + ... with PV = process value, CO = control output, SetPoint = setpoint KP's unit is: [unit for CO] / [unit for PV]
KI	**I-component** Use in the controller equation: CO = … + KI * Integral (SetPoint - PV) + ... with PV= process value, CO = control output, SetPoint = setpoint KI's unit is: [unit for CO] / (s * [unit for PV]) When the function .Calc is called, one integration step is carried out. The synchronous task's cycle time is used as the basis.
KD	**D-component** Use in the controller equation: CO = … + KD * differentiator (SetPoint - PV) + ... with PV= process value, CO = control output, SetPoint = setpoint KD's unit is: s * [unit for CO] / [unit for PV] When DCutOff is not equal to 0, be sure that the differentiator is band-limited.
KV	**Factor for the input control** Use in the controller equation: CO = … + KV * SetPoint + ... with CO = control output, SetPoint = setpoint KV's unit is: [unit for CO] / [unit for PV]
SetPoint	**setpoint** The setpoint is used when the .Calc function is called.
Reset	**Controller block** if = 0, the controller is working normally. if = 1, the controller is blocked. When the controller is blocked, its output is set to the default output value .CODefault (generally = 0). Also, when the controller is blocked, the integrator is reset. The integral is thus kept at zero. If .Reset is changed, it only takes effect the next time the function .Calc is called.
CODefault	**Default for the controller output** When a controller block is in effect, the controller's output is set to this default value. This value is generally = 0. This value should lie within the range [COMin...COMax].
COMin	**Lower limit for the setpoint** The controller output is limited to the range [COMin...COMax]. When controller output limiting is in effect, the PID-controller's integrator is frozen at its current value.
COMax	**Upper limit for the setpoint** The controller output is limited to the range [COMin...COMax]. When controller output limiting is in effect, the PID-controller's integrator is frozen at its current value.

Parameter	Description
Xinput	**Extra input**, fed into the controller after the P-, I-, D-components. Use in the controller equation: CO = ... - Xinput + ... With CO = controller output. Xinput's unit is: [unit for CO]
DcutOff	**Upper cutoff frequency** (in Hz) for the D-component To prevent amplifying the noise, the D-component is frequency-limited. Thus it is then only effective up to a certain frequency .DCutOff. While at lower frequency components the gain is proportional to the frequency, at higher frequency components the gain is constant. This constitutes a 1st order high-pass filter whose cutoff frequency is located just at .DCutOff. If this value is set to 0, the frequency limiting is deactivated. In that case, a digital differentiator is computed. Use in the controller equation: for DCutOff <> 0: CO = … + KD * high-pass (SetPoint - PV) + ... for DCutOff = 0: CO = … + KD * differentiator (SetPoint - PV) + ...
CO	**controller output** The controller output can be queried as .CO. It is also already returned by the function .Calc.
PV	**process value** The process value is the instantaneous value measured for the controlled variable and is queried as .PV. It is already set when the function .Calc is called and therefore generally needs not be set explicitly.

The controller is computed in imc Online FAMOS *with Control Commands* in the command OnSyncTask with the function .Calc. It must be initialized previously, since it initially contains no values, but already needs values upon the first cycle pass in order to compute the PID-controller.

Initialization is performed in the *OnInitAll*-block with the values specified for KP, KI and KD. All other elements are initialized as 0.0, except DCutOff, which is set to approx. 0.1 / [controller cycle interval].

```
Controller = CtPID( P_component, I_ component, D_ component )
Controller output = CtPID.Calc( Process Value)
```

Example

```
OnInitAll
    EngineController = CtPID( 20.0, 0.5, 0 ) ; P, I, D
    EngineController.SetPoint = 6000          ; set rpm to 6000
    EngineController.COMin = 0.0              ; min and min range must be defined
    EngineController.COMax = 10000.0
    ; DiplayVar_11 - 13 pass KP, KI and KD values if Virt_Bit03 is set by the user.
End
OnTriggerMeasure( BaseTrigger ) ; when the measurement is running
    If Virt_Bit02 = 1 ; take new set point from display variable
        EngineController.SetPoint = DisplayVar_01
        Virt_Bit02 = 0
    End
    If Virt_Bit03 = 1 ; Read K-components from display variable
        EngineController.KP = DisplayVar_11
        EngineController.KI = DisplayVar_12
        EngineController.KD = DisplayVar_13
        Virt_Bit03 = 0
    End
End
OnTriggerStop()
     EngineController.SetPoint = 0
End
OnSyncTask( 0.1 )
    ; controller output scaled for DAC
    DAC_VoltageEngine = EngineController.Calc( pv.Speed )

    If Virt_Bit01 = 1 ; reset controller
        EngineController.Reset = 1
        Virt_Bit01 = 0
    End
End
```

The `.Calc` function calculates exactly one controller step. For the currently transferred process value, the new controller output is determined as the return value. The return value is then typically outputted at a DAC. The process value is subsequently available `.PV`, the controller output in `.CO`.

If controller parameters are changed, then these changes only take effect upon the next call of the function `.Calc`, because then and only then is a new controller output computed.

The function `.Calc()` is only allowed in synchronous tasks. The synchronous task's cycle interval is used to discretize the controller.

11.2.7.4.2 Two-position controller

The two-position controller (discrete controller) is, like the PID-controller, computed in the synchronous task with the function `.Calc`. However, it must previously be initialized in the `OnInitAll`-block by creating the necessary controller structure along with it. In the process, the controller's setpoint `.SetPoint` is set `= 0`, and the controller output `.CO` is set `= 0`.

The controller's method of functioning: It only outputs specific controller outputs when pre-defined conditions are met. If the process value is greater than the upper reversal point, then the controller *output = 1* is generated and returned by the function *.Calc()*. If the process value is lower than the lower reversal point, then the resulting controller *output* is = 0. If the process value is between the two reversal points, the controller output remains unchanged. By means of the parameter *Output_Inversion*, the controller output can normally be inverted, so that when the process value is too high, the resulting controller *output* is = 0; and when the process value is too low, the resulting *CO* is = 1.

```
Controller = CtTwoPos(Hysteresis, Output_Inversion )
```

Example

```
OnInitAll
    Thermostat = CtTwoPos( 2, 0 )   ; Hysteresis, invert output
    Thermostat.SetPoint = 20        ; optional: set setpoint
    DigitalOut_01 = 0               ; optional: output in rest state
End
OnSyncTask( 0.1 )
    DigitalOut_01 = Thermostat.Calc( pv.Temperatur_01 )  ; set controller output
    ...
    If VirtBit_01 <> 0
        Thermostat.SetPoint = 22.0 ; change setpoint
        VirtBit_01 = 0
    End
End
```

The hysteresis is the distance between the upper and lower reversal points. It should be > 0. The two reversal points lie symmetrically around the setpoint. If, for instance, the setpoint is = 20 and the hysteresis = 2, then the resulting reversal points are 21 and 19.

11.2.8 Information and tips

11.2.8.1 Transferring a configuration

The possible configurations depend on how the device is equipped:

When imc Online FAMOS configurations are copied to other devices, their behavior may well be altered. This also applies when copying the imc Online FAMOS configuration to imc Inline FAMOS. For instance, an assignment `LED_01 = Greater(Channel_001, 5)` will cause LED_01 to be switched on, if the device has LEDs, otherwise a virtual channel by that name will be created.

The same goes for beepers, DAC-outputs and DIO-bits.

11.2.8.2 Virtual channels and the trigger machine

Virtual channels don't appear along with the channel selections in the Trigger dialog. Thus, a virtual channel cannot be triggered directly. To trigger the value of such a computed channel, the pertinent event must be detected in OFA/IFA, e.g. using the function `Greater`. The event sets a virtual bit, which in turn can be used by the trigger machine:

```
Virtual_Bit01 = Greater(Temp_difference, 5)
```

11.2.8.3 Supplemental files

Supplemental file such as characteristic curves for functions, must be imported to the experiment.

Reference

On this section, see the description for: "*Setup*" > "*Ribbon*" > "*Supplemental files* 212".

11.2.8.4 Comments for class-counting functions

To apply class-counting functions, a number of parameters need to be specified. However, the data presented in a curve window are not sufficient to determine all of these parameters. For this reason, the parameters for class-counting functions, in particular, are automatically inserted in the comment for the class-counting result. To save space, code-words corresponding to the various parameter settings options are used. The comment is produced when a syntax-check is successfully performed. The class-counting function names all begin with "Cl", e.g. ClRainflow.

With class-counting functions which return a matrix and a residue as their results, the automatic comment is produced only for the virtual channel containing the matrix.

If you write your own comment for a virtual channel which results from a class-counting function (Properties dialog in OFA/IFA), the comment will be displayed in the curve window. The comment which would otherwise be made automatically is then no longer available.

List of possible code-words and their meanings for class-counting function comments:

Code-word	Definition
OFA:	Comment intro for any class-counting function
RowMin	Lower limit of bin range (for rows)
RowMax	Upper limit of bin range (for rows)
ColMin	Lower limit of bin range (for columns)
ColMax	Upper limit of bin range (for columns)
Hyst	Hysteresis width
RowCl	Axis arrangement: X-axis is target class
RowAmplitude	Axis arrangement: X-axis is amplitude
RowClStart	Axis arrangement: X-axis is start class
RowMeanValue	Axis arrangement: X-axis is mean value
UnitCl	Result unit: classes
UnitInput	Result unit: as input channel
EndClOpen	Open-ended outer bins
EndClClosed	Closed outer bins
OptAlgBasic	Calculating method: base algorithm
OptAlgClor	Calculating method: with Clormann-Seeger correction
RowUnitStr	Unit (row-wise); in the comment: <Unit>
ColUnitStr	Unit (column-wise); in the comment: <Unit>
Level	Reference level
RowRevs	Axis arrangement: rpm in X-axis
RowNotRevs	Axis arrangement: rpm in Z-axis
Func	Function name of class-counting function

After the code words RowMin, RowMax, ColMin and ColMax, the corresponding value is stated, e.g. RowMax 1.00000E+2.

After the code words RowUnitStr and ColUnitStr, the unit is stated. The unit text is framed by "greater than" and "less than" symbols, e.g. RowUnitStr <Unit>.

11.2.8.5 Nesting depth, maximum stack

For conditions, etc., the maximum nesting depth is 90.

```
If (...)
  If (...)
    If (...) ...
```

With formulas, it is also possible to keep up to 90 expression elements.

For example: Result = a+b comprises 4 elements: a, b, + and "Result"

Note that there is **no** improvement of performance if the entire expression is written in one line. Instead it only reduces clarity!

11.2.8.6 LED-flashing during measurement

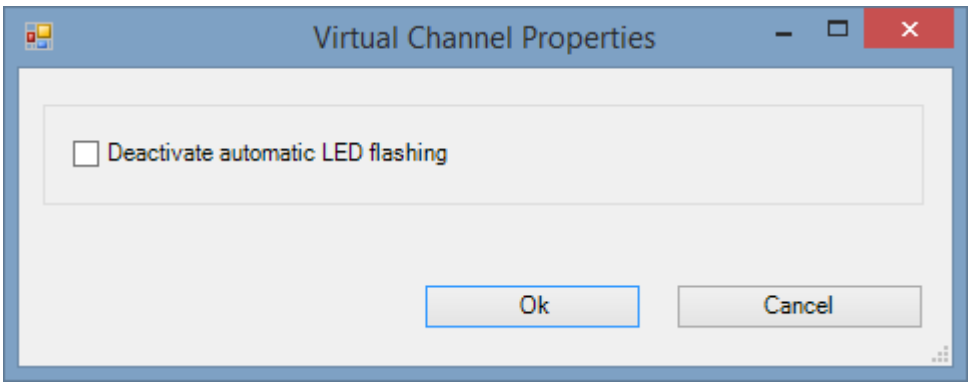
During a running measurement, LED 6 flashes at a 1-second rhythm. This provides a simple visual indication of whether the measurement is running.

LED 6 does not flash,

- if it is used in the imc Online FAMOS source code,
- if flashing is deactivated in the Options,
- if imc Online FAMOS is disabled.

Deactivating flashing in the Options

- Open the imc Online FAMOS options: Menu "*Extra*" > "*Options*"
- Activate the option: "*Deactivate automatic LED flashing*"



11.2.8.7 Using an external program editor

It's possible to enter the imc Online FAMOS source text using an external program editor (e.g. Notepad++, ...).

- Open the imc Online FAMOS options: Menu "*Extra*" > "*Options*"
- Activate the option: "*Edit imc Online FAMOS source text in the external program editor?*"
- Select the editor

Open the external editor by menu "*Edit*" > "*Start External Editor*" or push function key "*F4*".

Exchange of the imc Online FAMOS source text is performed by means of a temp file. This file is deleted as soon as the external editor is closed.

Note

In order that the content in the imc Online FAMOS editor and in the external editor match, the external editor may not keep the exchange file open after import, but should be able to automatically recognized any changes in the file.

- Example: WINWORD doesn't keep files open upon loading them. Automatic update of the temporary file in WINWORD is not possible, because imc Online FAMOS cannot access the open file.
- Example: NOTEPAD opens file and displays their contents. The file can later be altered by imc Online FAMOS, but the editor doesn't recognize the file alteration.

11.2.8.8 Tips and tricks

imc Online FAMOS - Computation uncertainty in summations

Goal: An incremental encoder is to be used to find a cumulative displacement. An incremental encoder delivers 400 pulses per meter = 2.5 mm/pulse.

If imc Online FAMOS computes the total displacement as:

```
Total_Path = sum (incremental_Channel, 1)
```

then the computed path will deviate from the actual displacement after a while.

Why does this happen?

imc Online FAMOS processes channel data in so-called Float format:

Float = 4 byte real number = 7 significant digits

4-Byte Float

-3.4028235E+38 to -1.1754944E-38, 0.0E+0, +1.1754944E-38 to +3.4028235E+38

In this system, rational numbers such as 2.5 can only be represented with limited precision. The number 10/4 is not expressed as 2.5 but rather as 2.4999999... This amounts to a discrepancy of 0.000223516% and is actually negligible. However, if this error is compounded throughout a calculation, larger errors can result. In this case, a new value is added to the sum every 2ms (assuming every 2ms 1 pulse = 2.5mm), in other words 500 times per second = 30.000 times per minute = 1.800.000 times per hour etc.

Thus a small cause can produce a large effect.

→ Approaches to a solution: imc Online FAMOS

1. Rounding

```
_inc=Round(Inc_1*400) ;400 pulses/m
Path_online2=sum(_inc,1)/400
```

2. Fewer samples = set a longer sampling interval.

Reduce the error in consequence of aggregate summation by reducing the terms summated:

- On the page: **Analog/Digital channels** in the dialog: **Sampling & Preprocessing**, set sampling for every 1 sec, for instance, instead of 2ms.
- Or in **imc Online FAMOS**

```
;500 = 1/ sampling interval = 1 / 2 ms    400 = pulse count / m
;take path increments every 1sec with resolution 1.
Inc_1s=mean(Inc_1,500,500)*(400*500)
Inc_1s_round=Round(Inc_1s)  ;round to whole number
;summation and division by pulse count.
Path_1s=sum(Inc_1s_round,1)/400
```

imc Online FAMOS - Insufficient Global RAM

When imc Online FAMOS is exited, a check is made of whether there is enough space in the device's global RAM for the virtual channels which were set up. If the input channels already take up too much of the device's buffer memory, the virtual channels cannot be created.

Tip: Temporarily slow down input channels

- Exit imc Online FAMOS, in spite of syntax-errors. Set a large sampling time for the input channels.
- Call imc Online FAMOS. Change anything, which will cause imc Online FAMOS to be re-compiled (e.g. enter and delete a space). Then exit imc Online FAMOS again.
- Now set on the page: **Analog/Digital channels** in the dialog: **Data transfer** the desired RAM buffer duration for each channel.
- Then go to the page: **Analog/Digital channels** in the dialog: **Sampling & Preprocessing** and set the desired sampling time for the input channels.

imc Online FAMOS - Digital Filters

Due to numerical problems, the ratio of the sampling frequency to the cutoff frequency with filters may not exceed a certain amount.

The following rule applies:

fsample/fg < (Order)th Root(1000000); where fsample= sampling frequency and fg = cutoff frequency

Examples:

Order	Ratio	fsample/fg
1st	1st Root(1000 000)	< 1000000
2nd	2nd Root(1000 000)	< 1000
3rd	3rd Root(1000 000)	< 100
4th	4th Root(1000 000)	< 32
5th	5th Root(1000 000)	< 16
6th	6th Root(1000 000)	< 10
7th	7th Root(1000 000)	< 7
8th	8th Root(1000 000)	< 6

The occurrence of numerical problems also depends on the filter characteristics set and on the particular implementation of the filter.

FAQ

Question: A data stream in imc Online FAMOS is processed in conjunction with a Display variable. This isn't performed accurately down to a single sample.

Answer: When trying to multiply a status cell with a waveform, the processing depends on how the internal FIFOs are filled. Thus, slight offsets can occur.

Question: There are problems in imc Online FAMOS when using experiments at long path names.

Answer: When trying to multiply a status cell with a waveform, the processing depends on how the internal FIFOs are filled. Thus, slight offsets can occur.

Question: The fieldbus channels are not available in imc Online FAMOS, but they are listed in the main dialog.

Answer: Fieldbus channels recorded in time stamping mode, can't be proceeded in imc Online FAMOS.

Question: When outputting to a DAC, not every value is outputted?

Answer: Two possible causes:

1. imc Online FAMOS needs longer than the sampling interval for the throughput of the channels to be processed. This means that the data come in chunks, of which only the respective last value can be transferred on to the DAC.
2. The sampling rate multiplied by the input channel's RAM buffer duration is > 2^16. However, the addressing is accomplished using a 16-bit pointer. This means that the samples are not addressed individually but rather in blocks. In imc Online FAMOS, the data are then also transferred in blocks. However, only the respective last value in a block is transferred to the DAC.

Question: How many channels @100kHz sampling rate can be proceeded online by the function SoundPressureLevel simultaneously?

Answer: imc Online FAMOS can proceed up to 3 channels (independently from the main board).

11.2.9 imc Online/Inline FAMOS function reference

This document contains the reference for the imc Online FAMOS and imc Inline FAMOS functions.

The functions are sorted in alphabetical order.

Functions for imc Online FAMOS or imc Inline FAMOS with control commands are listed in one group.

11.2.9.1 Operators and symbols

(+) Addition

Addition: Basic arithmetic

Sum = Summand1 + Summand2

If one of the two summands is a real number, the result is also a real number. In the addition of two integers, the result is an integer.

No handling mechanism is provided against potential data overflow; the user is required to take precautions for this possibility.

(-) Subtraction

Subtraction: Basic Arithmetic

Difference = Subtrahend - Minuend

If the subtrahend or the minuend is a real number, the result is a real number. In the subtraction of one integer from another integer, the result is an integer.

No handling mechanism is provided against potential data overflow; the user is required to take precautions for this possibility.

(*) Multiplication

Multiplication: Basic Arithmetic

Product = Factor1 * Factor2

If either of the two factors is a real number, the result is a real number. In the multiplication of two integers, the result is an integer.

No handling mechanism is provided against potential data overflow; the user is required to take precautions for this possibility.

(/) Division

Division: Basic Arithmetic

Quotient = Numerator / Denominator

With division, the result is always a real number.

For division between integers, please us the operator `iDiv` 951.

(^) Power

Power: Base raised to exponent;

Result = Base ^ Exponent

0 ^ 0 is defined to be 1.

(%) Modulus

Modulus: Remainder from division of two variables

Result = Numerator % Denominator

Example

```
Result = 10 % 3 ; Result = 1
```

If either the numerator or the denominator is a real number, the result is a real number.

When using the Modulo operator on two integers, the result is an integer. See also: iDiv 951.

(=) Equal sign (Assignment)

Equal sign: The "=" (equal sign) is used for making assignments to local and virtual channels.

Character: "=" (equal sign)

Example

```
_LocChan = Data + 1
VirtChan = _LocChan * 2
```

(=) Equal operator (Comparison)

"Equals"-operator. Compares the two operands for equality.

Prerequisite:
Only usable with control commands!

IsEqual = A = B

IsEqual: Result	**A**: 1st operand **B**: 2nd operand

IsEqual = 1, if A = B.

IsEqual = 0, if A is not equal to B.

Example

```
OnTriggerMeasure( BaseTrigger )
    Value = CurrentValue( Channel_001, 0, 0.0 )
    If Value = 1
        VirtChannel = Channel_001 + 10
    Else
        VirtChannel = Channel_001 + 5
    End
End
```

(<>) Inequality operator

Inequality-operator. Compares the two operands for inequality.

Prerequisite:
Only usable with control commands!

IsUnequal = A <> B

IsUnequal: Result	**A**: 1st operand **B**: 2nd operand

IsUnequal = 1, if A is not equal to B.
IsUnequal = 0, if A = B.

(<) Less than operator

"Less than"-operator. Compares the two operators to check whether the first is less than the second.

Prerequisite:
Only usable with control commands!

IsLess = A < B

IsLess: Result	**A**: 1st operand **B**: 2nd operand

IsLess = 1, if A is less than B.
IsLess = 0, if A is greater than or equal to B.

Example

```
OnTriggerMeasure( BaseTrigger )
    Value = CurrentValue( Channel_001, 0, 0.0 )
    If Value < 1
        VirtChannel = Channel_001 + 5
    Else
        VirtChannel = Channel_001 + 10
    End
End
```

(<=) Less than or equal operator

"Less than or equal to"-operator. Compares the two operands to check if the first is either less than or equal to the second.

Prerequisite:
Only usable with control commands!

IsLessEqual = A <= B

IsLessEqual: Result	**A**: 1st operand **B**: 2nd operand

IsGreater = 1, if A is less than or equal to B.
IsGreater = 0, if A is greater than B.

(>) Greater than operator

"Greater than"-operator. Compares the two operands to check if the first is greater than the second.

Prerequisite:
Only usable with control commands!

IsGreater = A > B

IsGreater: Result	**A**: 1st operand **B**: 2nd operand

IsGreater = 1, if A is greater than B.
IsGreater = 0, if A is less than or equal to B.

(>=) Greater than or equal operator

"Greater than or equal to"-operator. Compares the two operands to check if the first is either greater than or equal to the second.

Prerequisite:
Only usable with control commands!

IsGreaterEqual = A >= B

IsGreaterEqual: Result	**A**: 1st operand **B**: 2nd operand

IsGreaterEqual = 1, if A is greater than or equal to B.
IsGreaterEqual = 0, if A is less than B.

( ) Opening closing bracket

Opening parenthesis: The parentheses "(" and ")" are used in nested expressions and with functions.

Character: "(" (opening parenthesis)

Example

```
VirtChannel1 = 2 * ( Data + 1 )
VirtChannel2 = Max( Data, 5, 10 )
```

(;) Semicolon

Semicolon: A semicolon ";" is used to precede remark comments. Any information to the right of a semicolon is ignored by the program.

Character: ";" (Semicolon)

Example

```
VirtChan = Data + 1 ; first virtual channel
```

11.2.9.2 Functions (alphabetic)

11.2.9.2.1 A

ABCRating

ABC-Rating: Conducts either an A-, B- or C-frequency evaluation of a signal according to DIN IEC 651. In addition, subsequent time-rating (moving RMS-value with exponential averaging) and resampling can be performed.

a = ABCRating(b, FrequencyRating, TimeConstant, ReductionFactor)

a: Results	**TimeConstant**: Time constant for averaging
b: Input channel (Signal to be evaluated)	**In imc Online FAMOS:**
FrequencyRating: The signal's frequency rating	≥ 0.0: Time constant is defined by the user in s
1: A-rating	e.g. 0.125 for FAST-weighting, 1.0 for SLOW-weighting
2: B-rating	**In imc Inline FAMOS:**
3: C-rating	-1: Fast (0.125s)
	-2: Slow (1s)
	-3: Pulse
	-4: Peak
	-5: RMS in interval
	-6: RMS from start
	≥ 0.0: Time constant is defined by the user in s
	ReductionFactor: Factor for resampling, ≥ 1
	for a ReductionFactor = 1, no resampling

For a time constant = 0, no subsequent time weighting is performed. The reduction factor must in this case be exactly 1.

The A-rating corresponds to IEC 61672-1, 1st edition, 2002-05, Class1 and DIN IEC 651, 1981, class 0.

Example

```
SignalA = ABCRating ( Signal, 1, 0.125, 1000 )
```

- The signal is subjected to A-rating.
- The frequency-weighted signal is then time-weighted with a time constant of 0.125 s (FAST) and resampled with a factor of 1000.
- The signal originally has a sampling frequency of 20 kHz
- The A-rated result has a sampling frequency of 20 Hz.

Abs

Absolute value of input channel

a = Abs(b)

a: Results channel	**b**: Input channel

General information about the Accu* functions

The functions determine result over the entire measurement duration.

The result is a virtual channel or a single value.If the result is a virtual channel, this virtual channel is filled with exactly one result value at the end of the measurement.

It is possible to perform calculations on the results of the Accu-functions. Toward this end, be aware that these results of the Accu-functions are only available once the measurement is finished.For this reason, the results of the Accu-functions can not be taken into account in virtual channels which are calculated during the measurement.However, it is also possible to generate new virtual channels from the results of the Accu-functions, which are only available at the end of the measurement.

Example — Example of including results of the Accu-functions in a calculation

Determining the input signal's minimum and maximum values and sample count over the entire measurement duration. The results are subjected to calculations afterwards.

```
AccuMin1 = AccuMin( signal )
AccuMax1 = AccuMax( signal )
AccuLength1 = AccuLength( signal )
Res1 = (AccuMax1 - AccuMin1) * 10
Res2 = AccuLength1 / 2
```

Example — Example of an incorrect application

Determining an input channel's count of samples throughout the entire measurement duration.
The result of the AccuLength-function is determined at the end of the measurement. This result is generated too late to be used in generating calculated virtual channels during the measurement.
The virtual channel Res returns 0 as its result.

```
AccuLength1 = AccuLength( signal )
Res = signal*0 + AccuLength1
```

AccuLength

Count of samples over the entire measurement

Result = AccuLength(Signal)

Result: Results channel	**Signal**: Input signal

Remarks:

The function determines the count of samples in the input signal over the entire measurement.

The result is a virtual channel or a single value. If the result is a virtual channel, this virtual channel is filled with exactly one result value at the end of the measurement.

Example

Determining the input signal's minimum and maximum values and sample count over the entire measurement duration.
"*signal*" is the input channel.

```
AccuMin1 = AccuMin( signal )
AccuMax1 = AccuMax( signal )
AccuLength1 = AccuLength( signal )
```

AccuMax

Maximum value of entire measurement

Result = AccuMax(Signal)

Result: Results channel	**Signal**: Input signal

Remarks:

The function determines the maximum value of the input signal over the entire measurement duration.

The result is a virtual channel or a single value. If the result is a virtual channel, this virtual channel is filled with exactly one result value at the end of the measurement.

Example

Determines the minimum, maximum and mean value of the input signal over the entire measurement duration.

"*signal*" is the input channel.

```
AccuMin1 = AccuMin( signal )
AccuMax1 = AccuMax( signal )
AccuMean1 = AccuMean( signal )
```

AccuMean

Mean value over entire measurement

Result = AccuMean(Signal)

Result: Results channel	**Signal**: Input signal

Remarks:

The function determines the mean value of the input signal over the entire measurement duration.

The result is a virtual channel or a single value. If the result is a virtual channel, this virtual channel is filled with exactly one result value at the end of the measurement.

Example

Determines the minimum, maximum and mean value of the input signal over the entire measurement duration.

"*signal*" is the input channel.

```
AccuMin1 = AccuMin( signal )
AccuMax1 = AccuMax( signal )
AccuMean1 = AccuMean( signal )
```

AccuMin

Minimum value of entire measurement

Result = AccuMin(Signal)

Result: Results channel	**Signal**: Input signal

Remarks:

The function determines the minimum value of the input signal over the entire measurement.

The result is a virtual channel or a single value. If the result is a virtual channel, this virtual channel is filled with exactly one result value at the end of the measurement.

Example

Determines the minimum, maximum and mean value of the input signal over the entire measurement duration.

"*signal*" is the input channel.

```
AccuMin1 = AccuMin( signal )
AccuMax1 = AccuMax( signal )
AccuMean1 = AccuMean( signal )
```

AccuRMS

RMS-value over the entire measurement

Result = AccuRMS(Signal)

Result: Results channel	**Signal**: Input signal

Remarks:

The function determines the RMS-value of the input signal over the entire measurement duration.

The result is a virtual channel or a single value. If the result is a virtual channel, this virtual channel is filled with exactly one result value at the end of the measurement.

Example

Determines the minimum, maximum and RMS-value of the input signal over the entire measurement duration.

"*signal*" is the input channel.

```
AccuMin1 = AccuMin( signal )
AccuMax1 = AccuMax( signal )
AccuRMS1 = AccuRMS( signal )
```

AccuStDev

Standard deviation over the entire measurement

Result = AccuStDev(Signal)

Result: Results channel	**Signal**: Input signal

Remarks:

The function determines the standard deviation of the input signal over the entire measurement duration.

The result is a virtual channel or a single value. If the result is a virtual channel, this virtual channel is filled with exactly one result value at the end of the measurement.

Example

Determines the minimum, maximum and standard deviation of the input signal over the entire measurement duration.

"*signal*" is the input channel.

```
AccuMin1 = AccuMin( signal )
AccuMax1 = AccuMax( signal )
AccuStDev1 = AccuStDev( signal )
```

Acos

Inverse cosine of input channel

a = Acos(b)

a: Result; in radians	**b**: Input channel

AND

Logical AND operator applied to A and B.

Prerequisite:
Only usable with control commands!

LogicalAnd = A AND B

LogicalAnd: Result	**A**: 1st operand **B**: 2nd operand

LogicalAND = 1, if A is nonzero and B is nonzero.
LogicalAND = 0, if A is nonzero and B = 0.
LogicalAND = 0, if A = 0 and B is nonzero.
LogicalAND = 0, if A = 0 and B = 0.

The And-operator may only be used on operands of type BOOL. A result of type BOOL is returned by the operators <, <=, >, >=, =, <>, AND, OR and NOT.

Example

```
OnTriggerMeasure( BaseTrigger )
    Value1 = CurrentValue( Channel_001, 0, 0.0 )
    Value2 = CurrentValue( Channel_002, 0, 0.0 )
    If Value1 > 0 AND Value2 > 0
        VirtChannel_001 = Channel_001 + 10
    Else
        VirtChannel_001 = Channel_001 + 5
    End
End
```

Asin

Inverse sine of input channel

a = Asin(b)

a: Result; in radians	**b**: Input channel

Atan2

Inverse tangent of b/c

a = Atan2(b, c)

a: Result	**b**: Numerator **c**: Denominator

11.2.9.2.2 B

BitAnd

Bit wise AND connection of b and c

a = BitAnd(b, c)

Example

```
lownibble = BitAnd( DIO_Port01, 0x0f )
```

Extracts the DIO-Port's bottom four bits.

BitNot

Negation, bit-wise

a = BitNot(b [,Dataformat])

a: Result **b**: Argument	**Dataformat**: Assumed (integer) data format (only in imc Inline FAMOS) Allowed data types -32: 32 bit signed -16: 16 bit signed -8: 8 bit signed 1: 1 Bit (digital)

Bitwise inversion of the parameter's value.

A logical inversion is performed for each bit. The result bit is 0 if the input bit is 1, else 1. Negative integers are represented as a two's complement.

For imc Inline FAMOS applies: Each value is converted to the specified integer data format. When converting to the data type "*Digital*", all values which do not equal 0 are regarded as 1. The result is identical to that of the logical NOT-operator.

Example imc Online FAMOS

Bit-wise Negation of DIO_Port_01.

```
Res = BitNot( DIO_Port_01 )
```

Example imc Inline FAMOS

Inversion of the digital bit 'Virt_Bit01'.

```
Res1 = BitNot( Virt_Bit01, 1 )
```

All bits of the 2-Byte wide physical channel 'Signal_01' are inverted.

```
Res2 = BitNot( Signal_01, -16 )
```

All bits of the 4-Byte wide CAN channel 'CAN_01' are inverted.

```
Res3 = BitNot( CAN_01, -32 )
```

BitOr

Bit wise OR connection of b and c

a = BitOr(b, c)

BitXor

Exclusive OR, bit-wise: Bit-wise exclusive OR operation on b and c

a = BitXor(b, c)

11.2.9.2.3 C

Charact

Characteristic: The input channel is corrected using a characteristic curve. For each original value, a result is calculated from the characteristic in the file.

a = Charact(b, "Filename")

"Filename": characteristic in file

Values which lie between two characteristic points are linearly interpolated.

It is possible to calculate characteristic curves from **equidistant** data sets and from **XY**-data sets in imc FAMOS format. The X-coordinates of an XY-characteristic curve must be monotonically increasing. If they are not strictly monotonically increasing, then for one x-value there are multiple possible y-values; this means that the characteristic curve has a vertical portion at this location. If the input channel has such a value, then one of the associated y-values is arbitrarily selected as the result.

Driving cycle: This function can be used to output a waveform to a analog output or fieldbus.

Reference

See the example in the chapter: "*CAN-Bus Interface* 514" > ... > "*Outputting a waveform to a DAC module* 580".

Load the file

In imc STUDIO, the file of the characteristic curve is permanently integrated in the experiment.

- Import this file using the ribbon "*Setup-Configuration*" > "*Supplemental files*".

Note

imc Online FAMOS: The characteristic curve is limited to 50000 values.
imc Inline FAMOS: The characteristic curve is not limited.

ClHistogram

Classification histogram: amplitude time-of-stay procedure

a = ClHistogram(b, MinValue, MaxValue, Total, ResUnit, EndClass)

a: Results	**ResUnit**: Result units
b: Input channel	0: classes
MinValue: Lower limit of range	1: like input channel
MaxValue: Upper limit of range	**EndClass**: End classes are
Total: Number of classes; ≥ 1	0: closed
	1: open

The classification results are not complete until the end of measurement and therefore cannot be used in further computations.

This data classification procedure is based on DIN 45667 (German standard).

- For each class, the sum of the times during which the signal lies within the respective class boundaries is averaged separately.

In effect, this function counts the number of samples which occur in each class during a measurement.

- The time-of-stay in a class is computed by dividing the no. of samples.

Example

```
Histo1 = ClHistogram( Chan01, -10.0, 10.0, 32, 0, 0 )
```

- The input channel is divided into 32 classes ranging from -10 to 10, with closed end classes.
- The results are displayed as classes 0 to 31.

ClLevelCrossing

Level crossing histogram: Class counting with the level crossing method.

a = ClLevelCrossing(b, MinValue, MaxValue, Total, Reference, Hysteresis, ResUnit, EndClasses)

a: Results

b: Input channel

MinValue: Lower limit of range

MaxValue: Upper limit of range

Total: Number of classes; ≥ 1

Reference: Reference level

Hysteresis: hysteresis for peak identification; ≥ 0.0
specified in percent of class width

ResUnit: Result units
0: classes
1: like input channel

EndClass: End classes are
0: closed
1: open

The number of class level crossings are being counted, thus the result is divided into the number of classes + 1.

- If closed end classes are specified, the end classes of the result will necessarily be empty.
- With open end classes, level crossings through the upper and lower classes are counted.

The class counting results are not complete until the end of measurement and therefore cannot be used in further computations.

Example

```
Histo1 = ClLevelCrossing( Chan01, -10.0, 10.0, 64, 0.0, 10.0, 0, 0 )
```

- The input channel is divided into 64 classes ranging from -10 to 10. Level crossings relative to 0.0 are counted with 10 percent hysteresis relative to the class width.
- The end classes are closed. The results are displayed as classes 0 to 64

ClMarkov

Data classification using Markov procedure

a = ClMarkov(b, MinValue, MaxValue, Total, Hysteresis, Axes, ResType, ResUnits, EndClass, Option)

a: Results

b: Input channel

MinValue: Lower limit of range

MaxValue: Upper limit of range

Total: Number of classes; ≥ 1

Hysteresis: Hysteresis for peak identification; ≥ 0.0

Axes: Axes configuration, x-axis is

0: target class or amplitude

1: starting class or mean value

ResType: Result matrix is of type

0: target/starting class

1: amplitude/mean

ResUnits: Result units

0: classes

1: like input channel

EndClass: End classes are

0: closed

1: open

Option: Algorithm versions

0: main algorithm

The classification results are not complete until the end of measurement and therefore cannot be used in further computations.

Class-counting according to the Markov procedure counts all consecutive class transitions between local signal extrema, e.g. from Maximum 1 to Minimum 1, from Minimum 1 to Maximum 2, ...

The hysteresis suppresses noise and minor (i.e. unimportant) cycles. A new peak or valley is recognized only when the distance to the previous one is larger than the specified hysteresis width.

The result matrix can use the starting/target class or amplitude/mean as dimensions.

Example

```
Matrix1 = ClMarkov(Chan01, -10.0, 10.0, 64, 0.15, 0, 0, 0, 1, 0)
```

- The input channel is divided into 64 classes ranging from -10 to 10 with open end classes.
- The results are stored in a starting / target class matrix with classes from 0 to 63.
- The hysteresis width is approx. one-half the class width.

CloseSaveInterval

With this function, a new storage folder is set up on the internal storage medium. The folder is closed along with all data captured up to this time.

All subsequent measurement data are then saved in the new folder.

Prerequisite:
Only available in: imc Online FAMOS!
Only usable with control commands!

CloseSaveInterval()

If this function occurs within a pre-set saving interval, then the saving interval in progress is terminated, a new folder is created and filled with measured data until the actual end of the interval is reached. Subsequently, interval storage resumes according to the usual procedure.

The function can be used when data storage in the device is activated. At each call, an additional folder is created. For this, a command is triggered whose execution can be delayed depending on the configuration.

Constant calling of the function may cause the creation of an uncontrolled amount of folders.

Example

```
OnTriggerMeasure( BaseTrigger )
    If Virt_Bit01 > 0
        CloseSaveIntervall()
        Virt_Bit01 = 0
    End
End
```

ClRainFlow

Data classification using rainflow procedure

a = ClRainFlow(b, MinValue, MaxValue, Total, Hysteresis, Axes, ResType, ResUnits, EndClass, Option)

a: Results

b: Input channel

MinValue: Lower limit of range

MaxValue: Upper limit of range

Total: Number of classes; ≥ 1

Hysteresis: Hysteresis for peak identification; ≥ 0.0

Axes: Axes configuration, x-axis is

0: target class or amplitude

1: starting class or mean value

ResType: Result matrix is of type

0: target/starting class

1: amplitude/mean

ResUnits: Result units

0: classes

1: like input channel

EndClass: End classes are

0: closed

1: open

Option: Algorithm versions

0: main algorithm

1: with Clormann-Seeger correction

The classification results are not complete until the end of measurement and therefore cannot be used in further computations.

The rainflow procedure counts the cycles in a measured signal. Each cycle consists of a peak-valley pair (e.g. maximum and minimum).

The hysteresis suppresses noise and minor (i.e. unimportant) cycles. A new peak or valley is recognized only when the distance to the previous one is larger than the specified hysteresis width.

The result matrix can use the starting/target class or amplitude/mean as dimensions.

Example

```
Matrix1 = ClRainFlow( Chan01, -10.0, 10.0, 64, 0.15, 0, 0, 0, 1, 0 )
```

- The input channel is divided into 64 classes ranging from -10 to 10 with open end classes.
- The results are stored in a starting / target class matrix with classes from 0 to 63.
- The hysteresis width is approx.one-half the class width.

ClRainFlowRes

Rainflow with residues: data classification using rainflow procedure, output of remainder

ClRainFlowRes(a, Remainder, b, MinValue, MaxValue, Total, Hysteresis, Axes, ResType, ResUnits, EndClass, Option)

a: result matrix

Remainder: remainder

b: input channel

MinValue: lower limit of range

MaxValue: upper limit of range

Total: number of classes; ≥ 1

Hysteresis: hysteresis for peak identification; ≥ 0.0

Axes: axes configuration, x-axis is

0: target class or amplitude

1: starting class or mean value

ResType: Result matrix is of type

0: target/starting class

1: amplitude/mean

ResUnits: result units

0: classes

1: like input channel

EndClass: end classes are

0: closed

1: open

Option: algorithm versions

0: main algorithm

1: with Clormann-Seeger correction

The classification results are not complete until the end of measurement and therefore cannot be used in further computations. The remainder is not output until the end of the measurement.

The rainflow procedure counts the cycles in a measured signal. Each cycle consists of a peak-valley pair (e.g. maximum and minimum).

The hysteresis suppresses noise and minor (i.e. unimportant) cycles. A new peak or valley is recognized only when the distance to the previous one is larger than the specified hysteresis width.

The result matrix can use the starting / target class or amplitude / mean as dimensions.

Example

```
ClRainflowRes( Matrix1, Res1, Chan01, -10.0, 10.0, 64, 0.15, 0, 0, 0, 1, 0)
```

- The input channel is divided into 64 classes ranging from -10 to 10 with open end classes.
- The results are stored in a starting target class matrix with classes from 0 to 63. The remaining partial cycles are also displayed.
- The hysteresis width is approx. one-half the class width.

ClRainFlowTM

Rainflow with TrueMax filter: Rainflow class counting with TrueMax filter and hysteresis in percent of the class width.

a = ClRainFlowTM(b, MinValue, MaxValue, Total, Hysteresis, Axes, ResType, ResUnits, EndClass, Option)

a: Results

b: Input channel

MinValue: Lower limit of range

MaxValue: Upper limit of range

Total: Number of classes; ≥ 1

Hysteresis: Hysteresis for peak identification; ≥ 0.0

Specify as a percentage of the class width.

Axes: Axes configuration, x-axis is

0: target class or amplitude

1: starting class or mean value

ResType: Result matrix is of type

0: target/starting class

1: amplitude/mean

ResUnits: Result units

0: classes

1: like input channel

EndClass: End classes are

0: closed

1: open

Option: Algorithm versions

0: main algorithm

1: with Clormann-Seeger correction

The class counting results are not complete until the end of measurement and therefore cannot be used in further computations.

The rainflow procedure counts the cycles in a measured signal. Each cycle consists of a peak-valley pair (e.g. maximum and minimum).

The hysteresis suppresses noise and minor (i.e. unimportant) cycles. A new peak or valley is recognized only when the distance to the previous one is larger than the specified hysteresis width.

The dimension of the result matrix can be the starting/target class or the amplitude/mean.

Because the extreme points of a signal are not usually accurately sampled, the TrueMax filter attempts to interpolate them more accurately.

Example

```
Matrix1 = ClRainFlowTM( Chan01, -10.0, 10.0, 64, 50.0, 0, 0, 0, 1, 0)
```

- The input channel is divided into 64 classes ranging from -10 to 10 with open end classes.
- The results are stored in a starting/target class matrix with classes from 0 to 63.
- The hysteresis width is one-half the class width.

ClRainFlowTMRes

RainflowTM with Residue: Data classification using rainflow procedure, output of remainder. With TrueMax filter and hysteresis in percent of the class width.

ClRainFlowTMRes(a, Remainder, b, MinValue, MaxValue, Total, Hysteresis, Axes, ResType, ResUnits, EndClass, Algorithm)

a: Result matrix

Remainder: Remainder

b: Input channel

MinValue: Lower limit of range

MaxValue: Upper limit of range

Total: Number of classes; ≥ 1

Hysteresis: Hysteresis for peak identification; ≥ 0.0

in % of class width

Axes: Axes configuration, x-axis is

0: target class or amplitude

1: starting class or mean value

ResType: Result matrix is of type

0: target/starting class

1: amplitude/mean

ResUnits: Result units

0: classes

1: like input channel

EndClass: End classes are

0: closed

1: open

Algorithm: use

0: main algorithm

1: main algorithm with Clormann-Seeger correction

The class counting results are not complete until the end of measurement and therefore cannot be used in further computations. The remainder is not outputted until the end of the measurement.

The rainflow procedure counts the cycles in a measured signal. Each cycle consists of a peak-valley pair (e.g. maximum and minimum).

The hysteresis suppresses noise and minor (i.e. unimportant) cycles. A new peak or valley is recognized only when the distance to the previous one is larger than the specified hysteresis width.

The dimensions of the result matrix can be the starting/target class or amplitude/mean.

Because the extreme points of a signal are not usually accurately sampled, the TrueMax filter attempts to interpolate them more accurately.

Example

```
ClRainflowTMRes(Matrix1, Res1, Chan01, -10.0, 10.0, 64, 50.0, 0, 0, 0, 1, 0)
```

- The input channel is divided into 64 classes ranging from -10 to 10 with open end classes.
- The results are stored in a starting/target class matrix with classes from 0 to 63. The remaining partial cycles are also displayed.
- The hysteresis width is one-half the class width.

ClRangePairCount

Classification histogram: Using the range pair counting procedure.

a = ClRangePairCount(b, MinValue, MaxValue, Total, Hyst, ResUnit)

a: Results

b: Input channel

MinValue: Lower limit of range

MaxValue: Upper limit of range

Total: Number of classes; ≥ 2

Hyst: Hysteresis; in physical units, ≥ 0

ResUnit: Unit of the Result

0: classes

1: like input channel

The class width is calculated according to the following equation:

Classwidth = (MaxValue - MinValue) / (Total + 1)

In other words, with a range from -10 V to 10 V, it is necessary to create 19 classes in order to achieve a class width of 1 V. The lowest, 'zero-th class' would represent a range from 0 V to 1 V, and since nothing is ever counted in that range, it is omitted and the remaining total is 19 instead of 20 classes.

Only the magnitude is counted. So all classes would range from 1 V to 20 V.

The class counting results are not complete until the end of measurement and therefore cannot be used in further computations. This data classification procedure is based on DIN 45667 (German standard): the sum of the range-pairs is counted and allocated in classes.

Example

```
Histo1 = ClRangePairCount( Chan01, -10.0, 10.0, 19, 0.1, 0 )
```

- The input channel is divided into 19 classes ranging from 1 to 20.
- The results are displayed as Classes 1 to 19.

ClRevolutionsHistogram

Revolutions histogram: Counts the revolutions into the classes of the input channel.

a = ClRevolutionsHistogram(b, c, MinValue, MaxValue, Total, ResUnit, EndClass)

a: Result

b: Input channel

c: speed in rpm

MinValue: Lower limit of range

MaxValue: Upper limit of range

Total: Number of classes; ≥ 1

ResUnit: Result units

0: classes

1: like input channel

EndClass: End classes are

0: closed

1: open

The class counting results are not complete until the end of measurement and therefore cannot be used in further computations.

ClRevolutionsMatrix

Two-channel revolutions matrix: counts the revolutions in a matrix.

a = ClRevolutionsMatrix(b, c, MinValue1, MaxValue1, Total1, MinValue2, MaxValue2, Total2, Axes, ResUnits, EndClasses)

a: Results

b: first input channel

c: Speed in rpm (second input channel)

MinValue1: Lower limit of range

MaxValue1: Upper limit of range

Total1: Number of classes; ≥ 1

MinValue2: Lower limit of range

MaxValue2: Upper limit of range

Total2: Number of classes; ≥ 1

Axes: Axis order, speed is

0: Z axis

1: X axis

ResUnits: Result units

0: classes

1: like input channel

EndClasses: End classes are

0: closed

1: open

The class counting results are not complete until the end of measurement and therefore cannot be used in further computations.

ClTrueMax

Extreme value approximation: Approximates extreme values using interpolation.

a = ClTrueMax(b)

ClTwoChannelHistogram

Two channel histogram: Amplitude time-at-level procedure with two input channels and result matrix.

a = ClTwoChannelHistogram(b, c, MinValue1, MaxValue1, Total1, MinValue2, MaxValue2, Total2, ResUnit, EndClass)

a: Results

b: first input channel

c: second input channel

MinValue1: Lower limit of range 1

MaxValue1: Upper limit of range 1

Total1: Number of classes; ≥ 1

MinValue2: Lower limit of range 2

MaxValue2: Upper limit of range 2

Total2: Number of classes; ≥ 1

ResUnit: Result units

0: classes

1: like input channel

EndClass: End classes are

0: closed

1: open

The class counting results are not complete until the end of measurement and therefore cannot be used in further computations.

CodeRange

Returns specifiable values in response to a signal's value falling inside certain ranges.

a = CodeRange(b, Delay, "Filename")

a: Result	**Delay**: delay time
b: Input channel	**"Filename"**: coefficient vector file

The coefficients must be supplied in a group of three in the following manner (Minimum, Maximum, Code value). If the input channel lies within one of the specified ranges, the result is the code value, otherwise the result is zero.

If a switching time is specified, the output of the result values' data stream is delayed by this switching time. The result is only then a code value, if the input value stays within the corresponding range for at least the duration of the switching time, otherwise it is zero.

If the resulting data stream and other channels are to be used together as parameters of functions, the other channels must also be supplied with a delay (Function DelayLine).

Example

```
Transmission = GearRatio (Nin, Nout, 1, 1, 10, 0 )
Gears = CodeRange( Transmission, 0.1, "CoGear.dat")
DelayedLoad = DelayLine( Stress, 0.1 )
Gear1Load = SamplesGate( DelayedLoad, Gears, 1 )
```

In the channel Gear1Load, all loading measured while the vehicle was in first gear is recorded.

Cos

Cosine of input channel

a = Cos(b)

a: Result	**b**: Input channel; in radians

CreateVChannel

By means of this function, measurement channels which are represented by only a single value can be provided with a time track. In this way, it is possible to trace how a value changes over time.

To this end, a virtual channel (*real numbers*) is generated with the specified value. This channel has the properties of the input channel.

a = CreateVChannel(b, Value)

a: Virtual channel	**Value**: single value, which is supplied to the virtual channel
b: input channel	

The value supplied must be a single value. It can be a number or also, for instance, a virtual bit or a process vector element.

Example

```
Virt1 = CreateVChannel( Channel_001, 2.5 )
Virt2 = CreateVChannel( Channel_001, Virt_Bit01 )
Virt3 = CreateVChannel( Channel_001, pv.X123 )
```

CreateVChannelInt

By means of this function, measurement channels which are represented by only a single value can be provided with a time track. In this way, it is possible to trace how a value changes over time.

To this end, a virtual channel (*integers*) is generated with the specified value. This channel has the properties of the input channel.

a = CreateVChannelInt(b, Value)

a: Virtual channel	**Value**: single value, which is supplied to the virtual channel
b: input channel	

The value supplied must be a single value. It can be a number or also, for instance, a virtual bit or a process vector element.

Example

```
VirtInt1 = CreateVChannelInt( Channel_001, 2 )
VirtInt2 = CreateVChannelInt( Channel_001, Virt_Bit01 )
VirtInt3 = CreateVChannelInt( Channel_001, pv.X123 )
```

CrossCorrelation

Cross-correlation between a test channel and a reference channel.

Prerequisite:
Only available in: imc Inline FAMOS!

Result = CrossCorrelation(Reference, Test, BlockLength)

Result: Results of the cross correlation	**Test**: test channel
Reference: Reference channel	**BlockLength**: Maximum delay (128 ... 131072)

The cross-correlation function (CCF) indicates how similar two data sets are for different shifts in the x direction. The cross-correlation function returns values between -1 and +1.

A value of 1 at the location t signifies that shift the test channel in the x-direction by t from the reference channel achieves the maximum correlation between the two channels. The two signals then have the same curve plot. A value of -1 signifies that the two signals are equal opposites (where one signal is positive, the other is negative by the same magnitude). A value of 0 signifies that the two signals are entirely uncorrelated at this shift. All values between -1 and +1 are possible.

Using the cross-correlation function, it is possible to determine whether one signal is present (hidden) within another, and additionally, by what delay one signal appears in another channel. For this purpose, the maximum of the cross-correlation function is of particular interest. Its location indicates the size of the delay by which the signal appears within the other channel. Its magnitude indicates how similar the signal having the delay is to the other.

In order to be able to interpret the delay between signals correctly, it is important to specify the parameters of the function `CrossCorrelation` in the correct order. The first parameter represents the reference channel containing the original signal (having no delay. The second parameter represents the test channel which contains a signal having a delay (and sometimes also interference) compared to the original. The x-coordinate of the cross-correlation function's maximum value directly indicates the shift between the test channel from the reference channel.

During calculation of the cross-correlation, the input channel is factorized into vectors. The length of these vectors represents the maximum delay between the test channel and the reference channel which can be determined by using the function `CrossCorrelation`.

Another effect of factorizing the input channel is that the data generated are considered periodic; i.e. the vectors are imagined to extend in both directions (left and right) in a periodic pattern. Thus, if one vector (window) contains one single pulse, the signal is interpreted as representing a chain of multiple pulses. One particular consequence is that when the cross-correlation function returns a large shift of 0.9 periods, this has the same meaning as a small shift in the negative direction, namely by -0.1 periods.

The x-scaling of the function `CrossCorrelation` is that of the two channels specified. Both channels should have the same x-scaling, otherwise the results may not have any sensible interpretation.

The function `CrossCorrelation` is normalized to the product of the RMS (root-mean square) of the two channels specified. Thus, the channel generated has no y-unit.

Example

```
Res = CrossCorrelation( RefChannel, TestChannel, 1024 )
```

The function `CrossCorrelation` is applied to a reference channel and a test channel (which may feature interference and a delay).

For example, if the channels have a sampling rate of 100 Hz, then due to the window width of 1024 samples, maximum delays of 10 s can be detected.

CurrentValue

Current value: A single value is isolated from out of the given channel.

Prerequisite:
Only usable with control commands!

SingleValue = CurrentValue(Channel, Option, Init)

SingleValue: Result	**Option**: Which value?	**Init**: Initialization value
Channel: Channel from which the value is to be taken	0: Last value 1: Minimum 2: Maximum 3: Mean value	

The return value is the initialization value, until an actual value from the channel is obtained. The parameter "Option" is important if, due to a high sampling rate for the channel, there are multiple values which were not processed yet. In that case, the parameter "Option" decides how the return value is derived from these values. The selection "last value" refers to the current value, the data point most recently measured on the channel.

Example

```
OnTriggerMeasure( BaseTrigger )
    Value = CurrentValue( Channel_001, 2, 0.0 )
    If Value > 100
        VrtChannel_001 = Channel_001*0 + 100
    Else
        VrtChannel_001 = Channel_001*0
    End
End
```

11.2.9.2.4 D

dB

Converts input channel to decibel

a = dB(b)

The calculation is performed according to the formula: a = 20 * log(abs(b))

General information about the DelayBuffer functions

Prerequisite:
Only usable with control commands!
To use the `DelayBuffer` function, a `DelayBuffer` structure in the `OnInitAll` segment must be defined by means of the function `DelayBuffer`.

DelayBuffer

Initialization of the delay buffer: Sets up a delay line while specifying the maximum delay.

a = DelayBuffer(MaxSize)

a: The result is a DelayBuffer structure	**MaxSize**: Maximum buffer size, maximum size of the delay line

A signal can be delayed by N clock pulses. A buffer for measurement values is created having a defined length. In a processing step the current measured value is appended to the end of the buffer and the oldest measured value is retrieved from the start of the buffer. The buffer's length and the pulse rate of the processing steps determines the effective delay (see example).

The buffer's size is set to the specified maximum size. The buffer is completely pre-initialized with the default value 0.0. The actual delay can be set at any time using the function `DelayBuffer.SetSize`. The buffer also be completely refilled at any time with the specified value using the function `DelayBuffer.Fill`.

The most recently accumulated measured value are always held in the buffer, whose size is MaxSize. It is necessary to take account of the delay line's warmup time. This is because it must first be filled with a sufficient number of measured values before properly delayed values can be read out.

The function `DelayBuffer` must be called in the `OnInitAll` segment. In the process, a DelayBuffer structure is created and initialized.

Example 1

```
OnInitAll
    Dlb1 = DelayBuffer( 100 )
    Value = 0
    DelayValue = 0
End

OnSyncTask( 0.01 )
    ...
    ; delay of the output values by 1s (100*0.01s):
    DelayValue = Dlb1.Next( Value )
End

; Alternatively to the synchronized task, it is also possible to call
; the function .Next() within the timer:
OnTriggerStart( BaseTrigger )
    StartTimerPeriodic( 1, 0.1, 0.0 )
End

OnTimer( 1 )
    ...
    ; delay of the output values by 10s (100*0.1s):
    DelayValue = Dlb1.Next( Value )
End
```

Example 2 Delay measuring channel

A channel sampled at 1 kHz is delayed by 200 ms.

```
OnInitAll
    Dlb1 = DelayBuffer( 200 )
    DelayValue = 0
End

OnSyncTask( 0.001 )
    ; delay of the output values by 200 ms (200*0.001 s):
    DelayValue = Dlb1.Next( pv.Channel_001 )
End

OnTriggerMeasure( BaseTrigger )
    ; Create a virtual channel
    Channel_Delay = Channel_001 * 0 + DelayValue
End
```

DelayBuffer.Fill

Fill buffer: The entire buffer is filled with the value specified.

DelayBuffer.Fill(Value)

Value: The value with which the entire buffer is filled

All values ever previously stored in the buffer are overwritten.

Example

```
OnInitAll
    Dlb1 = DelayBuffer( 100 )
    Value = 0
    DelayValue = 0
End

OnSyncTask( 0.01 )
    If Virt_Bit01 > 0
        Dlb1.Fill( -1000.0 )
        Virt_Bit01 = 0
    End
    ...
    ; delay if the output value by 1s (100*0.01s):
    DelayValue = Dlb1.Next( Value )
End

; Alternatively to the synchronized task, it is also possible to call
; the function .Next() within the timer:
OnTriggerStart( BaseTrigger )
    StartTimerPeriodic( 1, 0.1, 0.0 )
End

OnTimer( 1 )
    ...
    ; delay of the output values by 10s (100*0.1s):
    DelayValue = Dlb1.Next( Value )
End
```

DelayBuffer.Next

Perform processing step: The new measured value (b) is appended to the end of the buffer and the oldest measurement value at the beginning of the buffer is returned.

a = DelayBuffer.Next(b)

a: The result is the delayed value, the oldest measurement value at the start of the buffer	**b**: The new measured value to be appended at the end of the buffer

The warmup time must be taken into account. If a delay of N measured values is set, then the very first N calls of this function only return the pre-initialized value.

Example

```
OnInitAll
    Dlb1 = DelayBuffer( 100 )
    Value = 0
    DelayValue = 0
End

OnSyncTask( 0.01 )
    ...
    ; delays the output value by 1s (100*0.01s)
    DelayValue = Dlb1.Next( Value )
End

; alternatively to the synchronous task, it is also possible to call
; the function .Next() within the timer:
OnTriggerStart( BaseTrigger )
    StartTimerPeriodic( 1, 0.1, 0.0 )
End

OnTimer( 1 )
    ...
    ; delay of the output value by 10s (100*0.1s)
    DelayValue = Dlb1.Next( Value )
End
```

DelayBuffer.SetSize

Set buffer size: Changing the buffer's size

DelayBuffer.SetSize(Size)

Size: New number of values in the buffer, new size of delay line

The size specified determines the delay. The signal is delayed by this many steps. The size can be changed at any time. A value of 0 means no delay. The maximum size specified in `DelayBuffer` may not be exceeded, however.

If the size is reduced, then stored values are skipped the next time `DelayBuffer.Next` is called. If the size is increased, then an older value which is still saved is fetched from the buffer the next time `DelayBuffer.Next` is called.

Example

```
OnInitAll
    Dlb1 = DelayBuffer( 100 )
    Value = 0
    DelayValue = 0
End

OnSyncTask( 0.01 )
    If Virt_Bit01 > 0
        Dlb1.SetSize( 50 )
        Virt_Bit01 = 0
    End
    ...
    ; delay of the output values by 1s (100*0.01s):
    DelayValue = Dlb1.Next( Value )
End

; Alternatively to the synchronized task, it is also possible to call
; the function .Next() within the timer:
OnTriggerStart( BaseTrigger )
    StartTimerPeriodic( 1, 0.1, 0.0 )
End

OnTimer( 1 )
    ...
    ; delay of the output values by 10s (100*0.1s):
    DelayValue = Dlb1.Next( Value )
End
```

DelayLine

Delay stream: The data stream is delayed by the specified time.

a = DelayLine(b, Delay)

a: result	**Delay**: delay time
b: input channel	

The result value data stream is delayed, a value from the input channel only appears at the output after a delay interval.

If other functions used produce delayed results, and the channel in question is to be combined with others in computations, then it is necessary to amend the channel with the same delay time that the others have.

Example

```
Gears = CodeRange (Transmission, 0.1, "CoGear.dat" )
DelayedLoad = DelayLine( Load, 0.1 )
Gear1Load = SamplesGate( DelayedLoad, Gears, 1 )
```

- In the channel Gear1Load, all loading which occurs while the vehicle is in first gear is recorded.
- Since the function `CodeRange` delays its result by 0.1 seconds, the Loading-channel must also be delayed by 0.1s.

DFilt

Digital filter: sends input channel through a digital filter

a = DFilt(b, "Filename")

a: Result	**"Filename"**: Coefficient vector file
b: Input channel	

The file of coefficients must be saved in imc FAMOS format.

The coefficients come in two different formats:

1. Product sum format

$y[t] = b_0 / a_0 * x[t] + ... + b_n / a_0 * x[t-n] - a_1 / a_0 * y[t-1] - ... - a_n / a_0 * y[t-n]$

The a_n terms must lead, followed by the b_n terms.

The coefficient vector must include the same number of denominator and numerator terms. Set any non-appearing terms to zero.

The denominator coefficients are to be specified first and the first value of the denominator coefficients a_0 must be equal to 1. All higher order denominator coefficients a[n] are assumed equal to zero.

2. Biquad format

The filter consists of 2nd order elements connected in "series". Each of these filters is defined as a biquad term of the form

$y[t] = b_0 * x[t] + b_1 * x[t-1] + b_2 * x[t-2] + a_1 * y[t-1] + a_2 * y[t-2]$

The coefficient vector must be specified in the following form:

$0\ a_2\ a_1\ b_2\ b_1\ b_0\quad a_2\ a_1\ b_2\ b_1\ b_0\quad ...\quad a_2\ a_1\ b_2\ b_1\ b_0$

The 0 in the first position designates the biquad format.

Example

Suppose the filter coefficients for a Butterworth 2nd-order low pass filter with a cutoff frequency of 100 Hz are given. The filter is to process a channel sampled at 5 kHz. The file of coefficients contains the following numbers:

```
0
-0.950212 a2
1.94894 a1
0.000317864 b2
0.000635728 b1
0.000317864 b0
```

```
y[t] = b0 * x[t] + b1 * x[t-1] + b2 * x[t-2] + a1 * y[t-1] + a2* y[t-2]
```

```
y[t] = 0.000317864 * x[t] + 0.000635728 * x[t-1] + 0.000317864 * x[t-2] + 1.94894 * y[t-
1] + -0.950121* y[t-2]
```

The file of coefficients must be saved in imc FAMOS format. If you have an ASCII table, load that file into imc FAMOS first and save it as *.dat file.

Note

- `DFilt` is initialized in imc Online FAMOS and imc Inline FAMOS as if the first measured value has already existed for a long time. With "quiet" signals, then, such as temperatures, transients are barely noticed. The drawback becomes clear, however, if a disturbance pulse is the first value and causes the filter to oscillate strongly.
- The `DFilt` function in the analysis software imc FAMOS initializes the filter with zeroes. For this reason, the two varieties respond differently during filter transient times.

Diff

Differentiation of argument

a = Diff(b)

a: Results	**b**: Input channel

The result values are the difference of the last and next to last input value divided by the sampling rate.

For the first input value a zero is returned.

DiskFreeSpace

Returns the free disk space on internal storage medium.

Prerequisite:
Only available in: imc Online FAMOS!

a = DiskFreeSpace(b)

a: Free disk space in kBytes (1 kByte = 1024 Bytes).	**b**: Reserved parameter (set to 0)

Zero will be returned in case of failure or absent storage medium.

DiskFreeTime

Free time: Estimate of time remaining for recording measurement data until internal storage medium is full.

Prerequisite:
Only available in: imc Online FAMOS!

a = DiskFreeTime(b)

a: Remaining free time in s	**b**: Reserved parameter (set to 0)

In general, the remaining free time cannot be determined exactly. The accuracy of the approximation made by this function depends on several factors. If you doubt the accuracy of this function, try using the function `GetDiskFreeSpace` which returns a more reliable estimate.

The approximation assumes that all triggers have been released and that data from these events is being recorded continuously to the hard drive.

Note the following:

- This function makes use of the function `DiskFreeSpace`. Consequently, its maximum resolution is 32 kByte.
- If new triggers are released, the hard drive space is used up more quickly.
- This function cannot make accurate predictions when data is not streaming at a constant sampling rate. In these cases the following mean data rates can be used as approximations:
  - Transitional Recording: 50% reduction
  - DIO Port reduced: 20 bytes/s
  - CAN data with time stamps: 50 bytes/s
  - Alarm channels: 1 byte/s
  - Histograms, matrices: not possible (space required one time only at trigger event)
  - Residues: not possible (space required one time only at the end of the measurement)

If clusters on the hard drive are defective, this also reduces the remaining time from the function estimate.

If you expect large deviations, the return value of the function has to be corrected!

For example, you could multiply the function result with an appropriate factor to account for uncertainties.

Example

Note that the result is a single value and cannot be treated as a channel.

```
Display_01 = GetDiskFreeTime (0) * 0.9 / 60 ;remaining time in min, 10% uncertainty
Duration_Channel = Channel_001 * 0 + GetDiskFreeTime ( 0 )
; returns time duration continuously
```

DiskRunDir

Current folder number on the storage medium

Prerequisite:
Only available in: imc Online FAMOS!
Function only for imc DEVICES; in imc STUDIO it is not used.

a = DiskRunDir(b)

a: Current folder number	**b**: Reserved parameter (set to 0)

This function gets the number of the current folder on the device's drive when "Continuous numbering" is active.

When "Continuous numbering" is activated for data storage, then upon starting measurement, or with interval data saving, the folder number is incremented by 1.

"0" will be returned in case of failure or absent storage medium.

DisplaySetButton

Function for setting a display button

Prerequisite:
Only available in: imc Online FAMOS!

DisplaySetButton(Page, Button)

Page: Page number The enumeration of the page number starts at 1. This corresponds to the enumeration in the function `DisplaySetPage`.	**Button**: Button index. The enumeration of the button number starts at 0. This corresponds to the preset enumeration that one obtains by using the option 'Quick Edit' in the display editor.

Note

If a display button is linked to a function like balance of channels or jump calibration, such a procedure can be executed with `DisplaySetButton` by imc Online FAMOS also.

DisplaySetPage

Function to show a new Displaypage

Prerequisite:
Only available in: imc Online FAMOS!

DisplaySetPage(Page)

Page: Pagenumber

11.2.9.2.5 E

General information about the ECU functions

Prerequisite:
Only available in: imc Online FAMOS!
Only usable with control commands!
ECU-functions are only available if the measurement device has a CAN-MultiIO.

When the function is called several times in succession, the commands are carried out separately one at a time. For this purpose, there is a small buffer which is sufficient for a few commands.

The following ECU functions are evaluated with `OnECUCmdReturn_ECU_001`. There you will find a comprehensive example.

ECUReadTroubleCodes_ECU_*

The function runs the ECU-command 'ReadDiagnosticTroubleCodesByStatus'.

ECUReadTroubleCodes_ECU_*

When the function `ECUReadTroubleCodes_ECU_001` is called, only one command is activated, and then the online configuration is processed further.

The activated command is only completed when `OnECUCmdReturn_ECU_001` returns the value ECU command 18h.

ECUStartSession_ECU_*

The function runs the ECU-command 'StartDiagnosticSession'.

ECUStartSession_ECU_*

When the function `ECUStartSession_ECU_001` is called, only one command is activated, and then the online configuration is processed further.

The activated command is only completed when `OnECUCmdReturn_ECU_001` returns the value ECU command 10h.

ECUStopSession_ECU_*

The function runs the ECU-command 'StopDiagnosticSession'.

ECUStopSession_ECU_*

When the function `ECUStopSession_ECU_001` is called, only one command is activated, and then the online configuration is processed further.

The activated command is only completed when `OnECUCmdReturn_ECU_001` returns the value ECU command 20h.

ECUClearDiagInformation_ECU_*

The function runs the ECU-command 'ClearDiagnosticInformation'.

ECUClearDiagInformation_ECU_001()

When the function `ECUClearDiagInformation_ECU_001` is called, only one command is activated, and then the online configuration is processed further.

The activated command is only completed when `OnECUCmdReturn_S1` returns the value ECU command 4h.

When the function is called several times in succession, the commands are carried out separately one at a time. For this purpose, there is a small buffer which is sufficient for a few commands.

Example

```
OnTriggerMeasure( BaseTrigger )
    If Virt_Bit01 > 0
        ECUClearDiagInformation_ECU_001( )
        Virt_Bit01 = 0
    End
End

OnECUCmdReturn_S1( Return, ECUCmd, CmdID )
    ECUCmd1     = BitAnd( ECUCmd, 0x0000FFFF )
    ECUIndex1   = BitAnd( ECUCmd, 0x00FF0000 )
    NodeIndex1 = BitAnd( ECUCmd, 0x0F000000 )
    If Return <> 0
        If ECUCmd1 = 0x4
            RecordText( "Error occurred in command 0x4" )
        Else
            RecordText( "General error occurred" )
        End
    End
End
```

OnECUCmdReturn_ECU_*

Return-treatment: Evaluation of the command-return values

OnECUCmdReturn_ECU_001(Return, ECUCmd, CmdID)

Return: return value; possible options: = 0: performed successfully = 1: timeout error occurred > 1: protocol-specific error occurred, referred to in the manual under the pertinent protocol	**ECUCmd**: ID of the ECU command, see below on list of ECU commands with associated IDs **CmdID**: ID or address of the ECU command (e.g. for routines), else 0

The function parameters Return, ECUCmd and CmdID are treated as local single value variables in the `OnECUCmdReturn_ECU_001` block, and may only be used within this command. These three variables are filled when the command is carried out and can be inquired in the `OnECUCmdReturn_ECU_001` block. They are single values.

With compound commands, the parameter ECUCmd returns the ID of the last command carried out. For instance, for the compound command ECU-Start with security access, the parameter ECUCmd returns the security access ID and not the ID of the ECU-Start command.

Example

```
OnTriggerMeasure( BaseTrigger )
    If VirtBit_01 > 0
        ECUReadTroubleCodes_ECU_001( )
        VirtBit_01 = 0
    End
    If VirtBit_02 > 0
        ECUStartSession_ECU_001( )
        VirtBit_02 = 0
    End
    If VirtBit_03 > 0
        ECUStopSession_ECU_001( )
        VirtBit_03 = 0
    End
End
OnECUCmdReturn_ECU_001( Return, ECUCmd, CmdID )
    If Return <> 0
        If ECUCmd = 0x18
            RecordText( "Error occurred in command...!" )
        End
    End
End
```

KWP 2000 commands set in the CAN-Assistant, IDs hexadecimal:

StartDiagnosticSession: 10 86	ReadMemoryByAddress: 23
SecurityAccess: 27	WriteDataByLocalID: 3b
LogInKey: 00 00	WriteDataByCommonID: 2e
StopDiagnosticSession: 20	WriteMemoryByAddress: 3d
TesterPresent: 3e	StartRoutineByLocalID: 31
ReadDiagnosticTroubleCodesByStatus: 18 00 ff 00	StartRoutineByAddress: 38
ReadDataByLocalID: 21	RequestRoutineResultsByLocalID: 33
ReadDataByCommonID: 22	RequestRoutineResultsByAddress: 3a

Equal

Compares for equality

a = Equal(b, c)

a: Result	**b**: 1. argument
	c: 2. argument

a = 1 if the arguments are equal, otherwise 0.

ExpoRMS

Moving RMS: Computes moving root-mean-square of an exponentially-weighted data window and applies reduction.

a = ExpoRMS(b, tau, RF)

a: Result	**tau**: Time constant for exponential weighting
b: Input channel	**In imc Online FAMOS:** ≥ 0.0: Time constant is defined by the user in s **In imc Inline FAMOS:** -1: Fast (0.125s) -2: Slow (1s) -3: Pulse -4: Peak -5: RMS in interval -6: RMS from start ≥ 0.0: Time constant is defined by the user in s
	RF: Reduction factor in samples; ≥ 1

The ExpoRMS function computes the root-mean-square of an exponentially weighted window.

A new result value is generated for every RF values.

- If RF > 1, then data reduction is applied.

Example

```
eff = ExpoRMS( Channel_001, 0.125, 1000 )
```

Computes RMS of exponentially weighted window and reduces by 1000.

11.2.9.2.6 F

FFT

Magnitude spectrum: FFT with output of the magnitude spectrum

a = FFT(b, Window, FFT_length)

a: Result; Magnitude spectrum	**Window**: Type of window	**FFT_length**: FFT-length
b: Input channel	0: Rectangle 1: Hamming 2: Hanning 3: Blackman 4: Blackman-Harris 5: Flat-Top	**In imc Online FAMOS:** 128 ... 8192 **In imc Inline FAMOS:** 128 ... 131072

The FFT length may only take one of the following values:

In imc Online FAMOS: 128, 256, 512, 1024, 2048, 4096, 8192.

In imc Inline FAMOS: 128, 256, 512, 1024, 2048, 4096, 8192, 16384, 32768, 65536, 131072.

The length of the result vector is determined as 1+(FFT_Length/2).

Example

```
Res = FFT( Signal_01, 3, 1024 )
```

Calvculation of an FFT with Blackman window and an FFT-length of 1024. The result is a streaming vector of length 513.

FFTAmplitudePhase

Magnitude-phase spectrum: FFT with output of magnitude and phase

FFTAmplitudePhase(Amplitude, Phase, b, Window, FFT_length)

Amplitude: Result 1; Amplitude (magnitude) spectrum **Phase**: Result 2; Phase spectrum **b**: Input channel	**Window**: Type of window 0: Rectangle 1: Hamming 2: Hanning 3: Blackman 4: Blackman-Harris 5: Flat-Top	**FFT_length**: FFT-length **In imc Online FAMOS:** 128 ... 8192 **In imc Inline FAMOS:** 128 ... 131072

The results of the procedure are the two first parameters: Magnitude and Phase. The FFT length may only take one of the following values:

In imc Online FAMOS: 128, 256, 512, 1024, 2048, 4096, 8192.

In imc Inline FAMOS: 128, 256, 512, 1024, 2048, 4096, 8192, 16384, 32768, 65536, 131072.

The length of the result vectors is determined as 1+(FFT_Length/2).

Note Peak values or RMS-values

The function FFTAmplitudePhase calculates the peak values when applied in imc Online FAMOS, but in imc Inline FAMOS it calculates the RMS-values.

For this reason, different calculation results are usually obtained.

Example

```
FFTAmplitudePhase( Amplitude, Phase, Signal_01, 1, 2048 )
```

Calculation of an FFT with Hamming window and an FFT length of 2048. The results are streaming vectors of length 1025.

FFTAverage

Overlapped, resampled and/or averaged amplitude spectrum: FFT with overlapped or skipped windows, arbitrary window size and averaging across a certain number of spectra or from the beginning.

Spectra = FFTAverage(Data, Width, Window, Overlap, Average, Count)

Spectra: Result; Averaged FFTs

Data: Input channel, time-domain data

Width: window width

In imc Online FAMOS:

100 ... 8192

In imc Inline FAMOS:

100 ... 131072

Window: Type of window

0: rectangle
1: Hamming
2: Hanning
3: Blackman
4: Blackman-Harris
5: Flat-Top

Overlap: stated in percent

-400: every 5th FFT
-100: every 2nd FFT
0: every FFT
33: 33.33% overlap
50: 50% overlap
66: 66.66% overlap
75: 75% overlap
90: 90% overlap

Average: averaging type

0: none
1: arithmetic mean
2: maximum
3: minimum
4: average from beginning

Count: number of data averaged

If the count is given as 10, then at first, 10 time data spectra are taken and averaged. The average spectrum is returned. All averages are carried out with the spectrum's magnitude.

The window width does not need to be a power of 2, in which case the data automatically gets interpolated. It is thus possible, for example, for a sampling rate of 5 kHz and 1000 points window width, to obtain spectral lines at exact multiples of 5 Hz.

- If the overlap is > 0, then the time windows overlap by the amount specified. However, in that case the computation time increases greatly!
- If the overlap is negative, not all time data are considered when forming the spectra.

The result values are to be interpreted as RMS values.

It is possible to calculate a Peak-Hold spectrum from the beginning of the measurement in combination with the function VmaxV 1039; see Example 2.

Example

```
Res = FFTAverage( Signal_Vib, 1000, 3, 0, 1, 100 )
```

Calculation of averaged FFTs with a Blackman window and window widths of 1000. Every FFT is factored in and the arithmetic mean of 100 spectra is taken.

Example 2 Peak-Hold spectrum

```
Res = FFTAverage( Signal_Vib, 1024, 0, 0, 1, 10 )
ResPeakHold = VMaxV( Res, 2, 1 )
```

Calculation of averaged FFTs with a rectangle window and window widths of 1024. Every FFT is factored in and the arithmetic mean of 10 spectra is taken. Calculation of the Peak-Hold spectrum from the beginning of the measurement.

FFTInverse

Inverse FFT: Inverse FFT from complex data

a = FFTInverse(RealPart, ImaginaryPart)

a: Result, Output time signal	**RealPart**: Real part of complex spectra **ImaginaryPart**: Imaginary part of complex spectra

The internal FFT function works with up to 4096 (actually complex) points.

A special technique deconstructs 8192 real (time-domain) points so that the spectrum can be computed by means of the 4 k internal FFT. The 4 k internal FFT contains a 4 k Real part and a 4 k Imaginary part.

However, the technique can not be used on complex data in the frequency domain. The data are already complex, for which reason it is not possible to transfer 8 k complex data.

Example

```
FFTRealImaginary( Real1, Imag1, Signal_01, 3, 1024 )
Res = FFTInverse( Real1, Imag1 )
```

Calculation of a complexe FFT with Real- and Imaginary parts. Subsequently, determination of the time signal from the Real- and Imaginary parts.

FFTRealImaginary

Complex FFT: FFT with real- and imaginary part output

FFTRealImaginary(RealPart, ImaginaryPart, b, Window, FFT_length)

RealPart: Result 1; Real part of the spectrum (returned vector) **ImaginaryPart**: Result 2; Imaginary part of the spectrum (returned vector) **b**: Input channel	**Window**: Type of window 0: Rectangle 1: Hamming 2: Hanning 3: Blackman 4: Blackman-Harris 5: Flat-Top	**FFT_length**: FFT-length **In imc Online FAMOS:** 128 ... 8192 **In imc Inline FAMOS:** 128 ... 131072

The FFT length may only take one of the following values:

In imc Online FAMOS: 128, 256, 512, 1024, 2048, 4096, 8192.

In imc Inline FAMOS: 128, 256, 512, 1024, 2048, 4096, 8192, 16384, 32768, 65536, 131072.

The length of the result vectors is determined as 1+(FFT_Length/2).

Example

```
FFTRealImag( Real1, Imag1, Signal_01, 5, 8192 )
```

Calculation of a complexen FFT with Flat-Top window and an FFT length of 8192. The results are streaming vectors of vector length 4097.

General information about the filters

The filter functions `FiltBP`, `FiltBS`, `FiltHP` and `FiltLP` are like their analog examples. It is recommended to set the parameters in a way, the analog circuit would be chosen. Parameters like order and cutoff frequency could create filters, that couldn't be build as a real analog circuit. However, usually extreme settings don't end in practicable results and stress the online processor unnecessary.

Due to numerical problems, the ratio of the sampling frequency to the cutoff frequency with filters may not exceed a certain amount.

The following rule applies:

fsample/fg < (Order)th Root(1000000); where fsample = sampling frequency and fg = cutoff frequency

See also "*Tips and Tricks*" > "*imc Online FAMOS - Digital Filters* 895".

FiltBP

Processes channel data with a band pass filter

Result = FiltBP(Data, Characteristic, Para, Order, CutoffFreqLow, CutoffFreqUp)

Result: Filtering result	**Para**: Reserved
Data: Channel to be filtered	**Order**: Filter order
Characteristic: Filter characteristic	Bessel: 1..40
0: Butterworth	otherwise: 1..100
1: Bessel	**CutoffFreqLow**: Lower cutoff frequency (Hz)
2: Chebychev	**CutoffFreqUp**: Upper cutoff frequency (Hz)
3: Critical damping	

FiltBS

Processes channel data with a band stop filter

Result = FiltBS(Data, Characteristic, Para, Order, CutoffFreqLow, CutoffFreqUp)

Result: Filtering result	**Para**: Reserved
Data: Channel to be filtered	**Order**: Filter order
Characteristic: Filter characteristic	Bessel: 1..40
0: Butterworth	otherwise: 1..100
1: Bessel	**CutoffFreqLow**: Lower cutoff frequency (Hz)
2: Chebychev	**CutoffFreqUp**: Upper cutoff frequency (Hz)
3: Critical damping	

© 2023 imc Test & Measurement GmbH

FiltHP

Performs high pass filtering of channel data

Result = FiltHP(Data, Characteristic, Para, Order, CutoffFreq)

Result: Filtering result

Data: Channel to be filtered

Characteristic: Filter characteristic

0: Butterworth
1: Bessel
2: Chebychev
3: Critical damping

Para: Reserved

Order: Filter order

Bessel: 1..20
Chebychev: 1..50
otherwise: 1..100

CutoffFreq: Cutoff frequency (Hz)

FiltLP

Performs low-pass filtering of channel data

Result = FiltLP(Data, Characteristic, Para, Order, CutoffFreq)

Result: Filtering result

Data: Channel to be filtered

Characteristic: Filter characteristic

0: Butterworth
1: Bessel
2: Chebychev
3: Critical damping

Para: Reserved

Order: Filter order

Bessel: 1..20
Chebychev: 1..50
otherwise: 1..100

CutoffFreq: Cutoff frequency (Hz)

Floor

Truncates the argument to the next smallest or equal integer.

a = Floor(b)

This function returns the **next lower integer** for a real number input, i.e. it rounds down to the next lower integer.

Example

The function returns

Input channel value	Result value
1,2	1
1,9	1
-1,2	-2
-1,9	-2

11.2.9.2.7 G

GearRatio

Transmission ratio: A gear system's transmission ratio is calculated on the basis of the engine rpm and the live axle rpm.

a = GearRatio(Ne, Na, MinNe, MinNa, Ft, Code)

a: result	**MinNe**: minimum input rpm
Ne: input rpm, engine	**MinNa**: minimum output rpm
Na: output rpm, live axle	**Ft**: low-pass cutoff frequency
	Code: code value, output less then minimum

The two rpm values are filtered with a low-pass and the transmission ratio Ne to Na is computed. Additionally, the rpm values are checked against their respective minima.

If the output rpm value falls short of MinNa or both of the minima are fallen short of, the output is a specified result value.

If only the input rpm value falls short of the minimum, the output is zero.

Example

```
Transmission = GearRatio( Nin, Nout, 1, 1, 10, 0 )
Gears = CodeRange( Transmission, 0.1, "CoGear.dat" )
DelayedLoad = DelayLine( Last, 0.1 )
Gear1Load = SamplesGate( DelayedLoad, Gears, 1 )
```

In the channel Gear1Load, all loading measured while the vehicle was in first gear is recorded.

GetDateTime

The function determines the current date and time.

Prerequisite:
Only usable with control commands!

GetDateTime(SecondsDecimal, Seconds, Minute, Hour, Day, Month, Year)

SecondsDecimal: current fraction of seconds in decimal notation, 0.0 ... 0.99	**Day**: current day, 1 ... 31
Seconds: current seconds, 0 ... 59	**Month**: current month, 1 ... 12
Minute: current minutes, 0 ... 59	**Year**: current year, e.g. 2009
Hour: current hours, 0 ... 23	

The decimal fraction of seconds is precise to within one 64th of a second.

The function parameters may be local single value variables, display variables or pv-variables.

Channels are not permitted as parameters. If any parameter is not required, a zero can be entered in its place.

Example

```
OnInitAll
    FindDate = 0
    FindTime = 0
    pv.Minute = 0
    pv.Hour = 0
    Time1 = ""
End

OnTriggerMeasure( BaseTrigger )
    If FindDate > 0
        GetDateTime( 0, 0, 0, 0, DisplayVar_Day, DisplayVar_Month, DisplayVar_Year )
        FindDate = 0
    End

    If FindTime > 0
        GetDateTime( 0, 0, pv.Minute, pv.Hour, 0, 0, 0 )
        FindTime = 0
        ; output as text "Time: ss:mm"
        Time1 = "Time: "
        If pv.Hour < 10
            Time1 = TextAdd( Time1, "0" )
        End
        Time1 = TextAdd( Time1, TextFormatI( pv.Hour ) )
        Time1 = TextAdd( Time1, ":" )
        If pv.Minute < 10
            Time1 = TextAdd( Time1, "0" )
        End
        Time1 = TextAdd( Time1, TextFormatI( pv.Minute ) )
        RecordText( Time1 )
    End
End
```

Example

```
OnInitAll
    SecondsDp = 0
    Seconds = 0
    Minute = 0
    Hour = 0
End

OnSyncTask( 0.1 )
    If Virt_Bit01 > 0
        GetDateTime( SecondsDp, Seconds, Minute, Hour, 0, 0, 0 )
        Virt_Bit01 = 0
    End
End
```

GetDuration

A channel's measurement duration: Finds the duration of measurement for a physical channel.

MeasurementDuration = GetDuration(InputChannel)

MeasurementDuration: Measurement duration for the physical channel in s.	**InputChannel**: Physical channel

The measurement value can be determined for all physical channels.

In particular for analog inputs, incremental counter inputs, Fieldbus inputs, digital Fieldbus ports and digital ports.

If no measurement duration is defined, the value 0 is returned.

Example

```
; measurement duration of physical channel Channel_001 in s
_MeasDuration_01 = GetDuration( Channel_001 )
```

GetHistoValue

Value from a histogram: Extracts the value from the specified index of a histogram.

HistogramValue = GetHistoValue(Histogram, Index)

HistogramValue: Histogram value at specified index **Histogram**: Histogram from which the value at the specified index location is to be extracted	**Index**: Index of histogram position; 0, 1, ..., number of classes -1

Example

```
OnTriggerMeasure( BaseTrigger )
    Histo = ClHistogram( Channel_001, -10.0, 10.0, 32, 1, 1 )
    ; 1st histogram value, histogram index: 0
    Histo_0 = Channel_001*0 + GetHistoValue( Histo, 0 )
    ; Last histogram value, histogram index: 31
    Histo_1 = Channel_001*0 + GetHistoValue( Histo, 31 )
End
```

GetHistoValue2

Value from a histogram: Extracts the value from the specified column index (X_Index) and row index (Y_Index) of a histogram.

HistogramValue = GetHistoValue2(Histogram, X_Index, Y_Index)

HistogramValue: Histogram value at specified indices **Histogram**: Histogram from which the value at the specified index locations is to be extracted.	**X_Index**: Column index in the histogram; 0, 1, ..., number of x-classes -1 **Y_Index**: Row index in the histogram; 0, 1, ..., number of y-classes -1

The histogram must be given in the form of a matrix of dimensions (number of x-classes) * (number of y-classes).

The histogram's X-index denotes the index in the X-direction in the curve window (histogram column index), the histogram's Y-index denotes the index in the Y-direction in the curve window (histogram row index).

Example

```
OnTriggerMeasure( BaseTrigger )
    Histo = ClRainFlow( Channel_001, -10.0, 10.0, 32, 0, 0, 0, 1, 1, 0 )
    ; 1st histogram value, histogram index: 0, column index: 0, row index: 0
    Histo_0 = Channel_001*0 + GetHistoValue2( Histo, 0, 0 )
    ; 69th histogram value, histo index: 68, column index: 4, row index: 2
    ; Histogram index = row index * 32 + column index = 2 * 32 + 4 = 68
    Histo_68 = Channel_001*0 + GetHistoValue2( Histo, 4, 2 )
End
```

GetLastError

Determines the error number of the last error. Determines the error number of the last error depends on what filter is configured.

GetLastError(ErrorNumber, ErrorOrigin, Filter)

ErrorNumber: Result. Error number of last error. The result values are <= 0, e.g. -5613.

ErrorOrigin: Result. Denotes where error occurred. The result values are >= 0 and <= 11.

0: No error
1: Error in Fieldbus system, Slot 1
2: Error in Fieldbus system, Slot 2
3: Error in Fieldbus system, Slot 3
4: Error in Fieldbus system, Slot 4
5: Error in Fieldbus system, Slot 5
6: Error in Fieldbus system, Slot 6
7: Error in Fieldbus system, Slot 7
8: Error in Fieldbus system, Slot 8
9: Error in the online-system
10: Error on the internal drive
11: Error in data capture

Filter: Which errors are accounted for?

-1: All errors
0: All errors in the Fieldbus system
1: Errors in the Fieldbus system, Slot 1
2: Errors in the Fieldbus system, Slot 2
3: Errors in the Fieldbus system, Slot 3
4: Errors in the Fieldbus system, Slot 4
5: Errors in the Fieldbus system, Slot 5
6: Errors in the Fieldbus system, Slot 6
7: Errors in the Fieldbus system, Slot 7
8: Errors in the Fieldbus system, Slot 8
9: Errors in the online-system
10: Errors on the internal drive
11: Error in data capture

Each time the function is called, the last occurring error number and the error origin are found, in accordance with the specified filter. If no error occurred, the function returns a value of 0 as the result. The result values can be outputted in a single value variable (see example) or in a device variable (e.g. DisplayVar_01). Instead of a return variable for the error origin, the value 0 is also allowed. Then the error origin is not determined.

Any occurring error number is overwritten the next time the function is called. For purposes of analysis, having an additional variable can be useful (see example).

The function can only be applied in imc Online FAMOS with control commands and dependent upon the device. As a fundamental rule, the function `GetLastError`() may only be called once per device with a particular parameter. If the function `GetLastError`(..., -1) is called, there may not be any further calls of the function `GetLastError`(). If the function `GetLastError`(..., 0) is called, then only the functions `GetLastError`(..., 9) and `GetLastError`(..., 10) may additionally be called.

To avoid burdening the device's computational power, it is recommended to call the function `GetLastError`()with a controlling timer (see example).

Excerpt of possible error numbers

CAN-Bus	Description
-5100	Overflow in an int. data repository of the CAN-Bus system, e.g. data red. of channels needed.
-5101	Aggregate sampl. frequ. of the CAN channels is too high, e.g. longer sampl. intervals needed.
-5102	At least one CAN-message has been lost.
-5103	Access to the CAN-Bus has been interrupted due to too many bus errors, e.g. wrong Baud rate.
-5106	Failure to send message during the starting phase of the synchronization.
-5107	Measurement values for a synchr. channel were discarded, e.g. the sampling rate is too low.
-5108	For a synchr. channel, supplem. readings were provided (bus error, or sampl. rate too high).
-5109	A CANSAS module has failed to execute a command, e.g. due to data transmission error.
-5110	No acknowledgement received from a CANSAS module!
-5111	Other bus nodes have been detected. Unable to convert the baud rate!
-5112	The modules are not suitable to the modules connected to the bus. Unable to convert Baud rates!
-5113	After conversion of the Baud rate, a module was no longer found!
-5114	A CANSAS unit's firmware is not up-to-date.
-5115	A received norm display number does not correspond to the configuration. The value is ignored.
-5116	A CAN-channel's bits lie outside of the message received. No value is generated.
-5117	Error in loading a SeedKey-library.
-5118	Unable to find a SeedKey-function.

© 2023 imc Test & Measurement GmbH

EtherCAT	Description
-5600	Internal error of imc CRONOSflex DAQ-system.
-5601	Overflow in an internal hardware data volume belonging to the imc CRONOSflex system.
-5602	The aggregate sampling frequency of the channels is too high.
-5603	The number of events (trigger inputs) to calculate is too high.
-5604	The max. channel count has been exceeded. Reduce number of act. channels/Mon. channels.
-5605	Error of imc CRONOSflex system in setting the synchronization.
-5606	Error in synchronizing the CRFX-modules connected to the imc CRONOSflex system bus.
-5607	Synchronization error, VCXO outside of capture range.
-5610	Communication error 1 in the imc CRONOSflex system. Please re-initialize the device.
-5611	Communication error 2 in the imc CRONOSflex system. Please re-initialize the device.
-5612	Communication error 3 in the imc CRONOSflex system. Please re-initialize the device.
-5613	Communication between imc CRONOSflex system and CRFX module failed.
-5614	Communication between imc CRONOSflex system and CRFX module back in order.
-5615	Overload of imc CRONOSflex system. Reduce data load and re-initialize device.
-5616	Error in event calculation in the imc CRONOSflex system.
-5617	A CRFX-module connected to the imc CRONOSflex system is in an invalid state.
-5618	Unable to process data transmitted via the imc CRONOSflex system bus.
-5619	Capture of meas. data by CRFX-modules o the CRONOSflex system bus no longer synchronized.
-5620	Interference on the imc CRONOSflex system bus. Please check the cabling.

Example

```
; If an error occurred, the error number and slot of the respective last fieldbus
; error are outputted as a virtual channel

OnInitAll
    DisplayVar_32 = 0
    int LastError = 0
    int Slot = 0
End

OnTriggerStart( BaseTrigger )
    StartTimerPeriodic( 1, 0.1, 0 )
End

OnTriggerMeasure( BaseTrigger )
    V_Error = CreateVChannelInt( Channel_001, DisplayVar_31 )
    V_Slot = CreateVChannelInt( Channel_001, DisplayVar_32 )
End

OnTimer( 1 )
    GetLastError( LastError, Slot, 0 )
    If LastError = -5613 or LastError = -5618
        DisplayVar_31 = LastError
        DisplayVar_32 = Slot
    End
End
```

GetSampleCount

Sample counting: Determines the number of values in a channel, which were not yet processed.

Samplenumber = GetSampleCount(Channel)

Samplenumber: Result	**Channel**: Input channel

If the parameter channel contains no values which apply, the function returns the value 0.

Example

```
OnInitAll
    sc = 0
End
OnTriggerMeasure( BaseTrigger )
    sc = GetSampleCount( Channel_001 )
    If sc > 1
        RecordText( "Proceeding more than 1 sample..." )
    End
End
```

GetSamplingTime

Channel sampling interval: Determines the sampling interval for a physical channel.

Samplinginterval = GetSamplingTime(Inputchannel)

Samplinginterval: Sampling interval of the physical channel	**Inputchannel**: Physical channel

The sampling interval can be found for any physical channel.

This means for any analog, incremental counter inputs, fieldbus inputs, as well as digital and digital fieldbus ports.

Example

```
_Samplinginterval_01 = GetSamplingTime( Channel_001 )
```

Greater

Checks whether the first argument is greater than the second argument

a = Greater(b, c)

a = 1 if b is greater than c, otherwise 0.

Example

```
LED_01 = Greater( Channel_001, 8 )
```

The LED lights up whenever the signal value exceeds 8.

GreaterEqual

Checks whether the first argument is greater than or equal to the second argument

a = GreaterEqual(b, c)

a = 1 if b is greater than or equal to c, otherwise 0.

11.2.9.2.8 H

HighLowRatio

The pulse duty ratio of the signal in a window, with reduction

a = HighLowRatio(b, RF)

a: Result	**RF**: Reduction factor in samples; ≥ 1
b: Input channel	

The function returns a value for each set of RF input samples as the result.

The ratio is the number of samples not equal to zero divided by the number of samples equal to zero, for all complete pulses.

A complete pulse is a sequence of three edges, an edge being a transition from zero to a non-zero value, or vice versa.

Analog signals should be preprocessed (STri 1017).

Hyst

Hysteresis: Applies a hysteresis filter to the input channel

a = Hyst(b, Width)

a: Result	**Width**: Hysteresis width
b: Input channel	

If the slope of the input channel changes sign, the output channel follows only if the difference between the last two samples is larger than the hysteresis width. Otherwise the output channel stays constant.

If the slope does not change sign, the result is equal to the input channel. This filters out all vibrations with amplitudes smaller than the hysteresis width.

11.2.9.2.9 I-J

iDiv

Integer division: Division of two integer values. The result is an integer value. Decimal places are omitted.

Result = Numerator iDiv Denominator

Both the numerator and denominator must be of data type Integer, numbers in Real-format are not permitted.

Single value variables of type Integer may only be defined with control commands.

Example

```
Quotient1 = 14 iDiv 4  ; Quotient1 = 3
Quotient2 = 14 / 4     ; Quotient2 = 3.5
```

Integral

a = Integral(b)

The result values are the sum of all current input samples multiplied by the sampling rate.

Integral2

Integral with reset to 0. The return values are the respective sums of all input values accumulated since the last reset to 0 or since the start of the measurement, multiplied by the sampling interval.

Result = Integral2(Channel, Reset)

Result: Integral	**Reset**: If this value is not 0, the integral is reset to 0
Channel: Input channel	

In contrast to the function `Integral`, the function `Integral2` offers the option to reset the integral to 0 during measurement. Otherwise, the behavior of the two functions is the same.

Example

```
Result = Integral2( Channel_001, VrtBit_01 )
```

IntegralFFT

Integral over amplitude spectrum: Integration in the amplitude spectrum's frequency range.

IntFFT = IntegralFFT(Vector)

IntFFT: Result of integration	**Vector**: Amplitude spectrum Magnitude spectrum

Integration is performed by multiplying the amplitude spectrum with an appropriate weighting function.

For a frequency of 0 Hz, the result of integration is 0 and thus meaningless.

Example

```
vFFT = FFT( InputChannel, 0, 1024 )
IntFFT = IntegralFFT( vFFT )
```

Calculation of an FFT with rectangular window and an FFT-length of 1024. The result is a streaming vector of length 513.

Subsequent integration in the frequency domain of the amplitude spectrum

IntegralP

Precise integral

Prerequisite:
Only available in: imc Online FAMOS!

a = IntegralP(b)

a: result	**b**: input channel

The respective result values are the sum of all previous input values multiplied by the sampling interval.

Internally, the precise integral uses a numerical notation with higher precision in order to suppress the compounding of errors in longer integrations.

However, `IntegralP` is significantly slower than the function `Integral`.

IntegralP2

Precise integral with reset to 0. The return values are the respective sums of all input values accumulated since the last reset to 0 or since the start of the measurement, multiplied by the sampling interval.

Prerequisite:
Only available in: imc Online FAMOS!

Result = IntegralP2(Channel, Reset)

Result: Precise Integral	**Reset**: If this value is not 0, the integral is reset to 0
Channel: Input channel	

In contrast to the function `IntegralP`, the function `IntegralP2` offers the option to reset the integral to 0 during measurement.

Otherwise, the behavior of the two functions is the same.

In contrast to the function `Integral2`, the function `IntegralP2` calculates the integral precise.

The function `IntegralP2` requires much more calculating time.

Example

```
Result = IntegralP2( Channel_001, VrtBit_01 )
```

General information about the interval funktions

Interval functions enable analysis of one or more channels in relation to the signal plot of one channel.

- For example, in taking measurements of a motor, the RPM values, vibration, etc. are only interesting if the RPM exceeds a certain level.
- Or, in conjunction with the particular angular ranges of a rotating system, the static values of associated channels are to be determined.

Using the trigger machine and imc Online FAMOS, it would be possible to do this calculation for one range. The interval functions can achieve the same thing with fewer command lines, no trigger and for a variety of ranges.

The example below illustrates this:

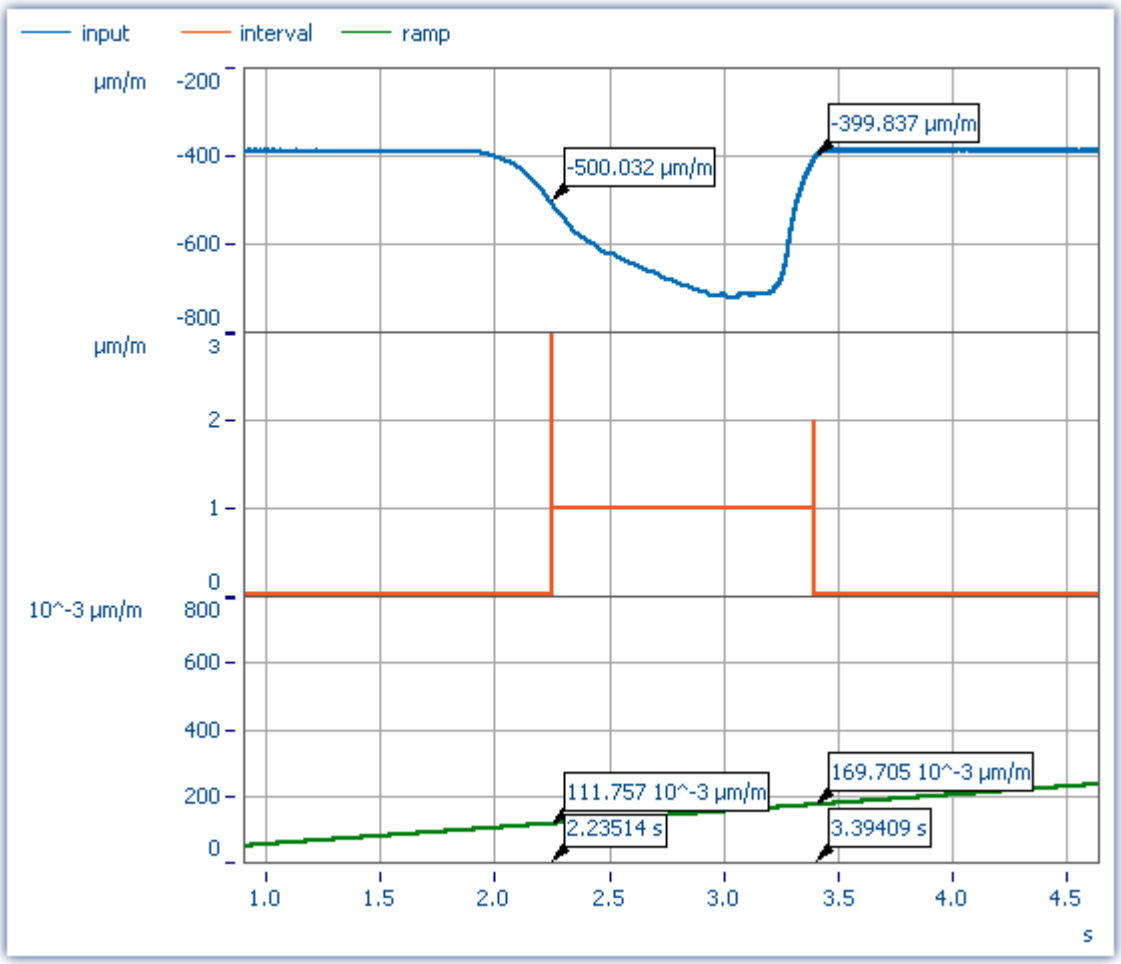
Example

An input channel is analyzed in the interval from one descending edge at -500 µm/m until an ascending edge at -400 µm/m. In this interval, the maximum and minimum of a different channel (ramp) are determined.

The program for this appears as follows:

```
ramp     = sawtooth( input, 0, 0.00001, 62800 )
interval = IntervalFromLevels( input, -500, -400, 1)
Maximum  = IntervalMax( ramp, interval )
Minimum  = IntervalMin( ramp, interval )
```

The interval channel is only an auxiliary channel which can only be used internally.



The functions IntervalMax, IntervalMin, etc. return exactly one result per interval and have no more time reference.

Maximum =	0.1697 µm/m
Minimum =	0.1118 µm/m

Note

Result channels created with *IntervalMax, -Min, Mean* and *RMS* no longer have a time reference and **cannot be sent via XCPoE!**

IntervalFrom1Level

Generates a channel specifying whether the signal's values are within a specifiedinterval.

Result = IntervalFrom1Level(Signal, Level, Levelcode)

Result: results channel	**Level**: interval limit
Signal: input channel	**Levelcode**: Reserved, always set to 0

Depending on the specified Levelcode, the interval limit must be crossed either in the rising or falling direction to count as entering the interval.

Subsequent interval functions use the results of `IntervalFrom1Level` to limit their calculations to the resulting intervals. The interval channel that is generated by `IntervalFrom1Level` contains, unlike `IntervalFromLevels`, additional data that interpolate the exact point in time when the interval limit is crossed between 2 samples of the given input signal. Subsequent interval functions may evaluate these data to increase their precision.

Example

```
; Determine the mean value of a squared sine signal
; Theoretical value = 0.5, independent of the interval limit
; set in IntervalFrom1Level
saw1 = SawTooth(Channel_001, 0, 0.01, 628)
signal1 = Sin(saw1)
signal2 = signal1 * signal1
interval = IntervalFrom1Level(saw1, 1.0, 0)
iMean1 = IntervalMean(signal2, interval)
```

IntervalFromLevels

Generates a channel stating whether the signal's values are located within the specified interval.

Result = IntervalFromLevels(Signal, Level1, Level2, Levelcode)

Result: results channel	**Level1**: first interval limit
Signal: input channel	**Level2**: second interval limit
	Levelcode: Levelcode

Depending on the specified Levelcode, the first interval limit must be crossed either in the rising or falling direction to count as entering the interval. Correspondingly, the second interval limit must be crossed in the rising or falling direction in order to count as leaving the interval.

Permissable values for Levelcode:

0	Level1 must be crossed in rising direction, so that the interval is considered entered, Level2 must be crossed in rising direction, so that the interval is considered left.
1	Level1 must be crossed in falling direction, so that the interval is considered entered, Level2 must be crossed in rising direction, so that the interval is considered left.
2	Level1 must be crossed in rising direction, so that the interval is considered entered, Level2 must be crossed in falling direction, so that the interval is considered left.
3	Level1 must be crossed in falling direction, so that the interval is considered entered, Level2 must be crossed in falling direction, so that the interval is considered left.

`IntervalFromLevels` designates the beginning and end of the resulting intervals by special values so that it is possible to append the intervals directly to each other.

Subsequent interval functions use the results of `IntervalFromLevels` to limit their calculations to the resulting intervals.

Example

```
; Determine min and max of a periodic signal between 20 and 40 degrees.
; the angle is obtained from an incremental encoder sensor in Angle-absolute-mode
intervall = IntervalFromLevels( angle, 20, 40, 0 )
iMin = IntervalMin(signal, interval)
iMax = IntervalMax(signal, interval)
```

IntervalFromPulse

Creates an interval channel from a incremental encoder signal.

Result = IntervalFromPulse(Impulse time, Period duration, Multiplier)

Result: Virtual channel with result	**Period duration**: Length of period
Impulse time: Incremental counter channel set to measurement mode impulse time.	**Multiplier**: Reserved, always = 0

The input signal comes from an incremental encoder sensor in pulse-time mode. `IntervalFromPulse` uses this to generate interval codes indicating the exact pulse time which can be used by subsequently called functions, for example, ones which use these time points for interpolation.

The interval is entered upon arrival of the first pulse and is only exited when the measurement ends or when more than one pulse arrives during a sampling interval. In the latter case, the system waits for the number of pulses associated with the period length before entering the interval again.

This serves to prevent any phase shift due to periodic data. For nonperiodic data, a period length of 0 must be specified.

Example

```
ivl = IntervalFromPulse( Inc_Counter_01, 60, 0 )
res = IntervalResample( Channel_001, ivl, 360.0, 60, "Deg", 1 )
```

IntervalMax

Determines the input signal's maximum within the specified interval.

Result = IntervalMax(Signal, Interval)

Result: result channel	**Intervall**: interval channel
Signal: input signal	

The interval channel is generated by the functions `IntervalFromLevels` or `IntervalFrom1Level`.

`IntervalMax` determines the input signal's maximum, while limiting this computation to the intervals specified by the interval channel. Since the time reference is lost, the result cannot be charged any further in many imc Online FAMOS functions and cannot be sent via XCPoE 952.

Example

```
; Determine min and max of a periodic signal between 20 and 40 degrees.
; the angle is obtained from an incremental encoder sensor in Angle-absolute-mode
interval = IntervalFromLevels( angle, 20, 40, 0 )
iMin = IntervalMin(signal, interval)
iMax = IntervalMax(signal, interval)
```

IntervalMean

Determines the mean of the input signal within the given interval.

Result = IntervalMean(Signal, Interval)

Result: results channel	**Interval**: interval channel
Signal: input signal	

The interval channel is generated by the functions `IntervalFromLevels` or `IntervalFrom1Level`.

`IntervalMean` determines the mean of the input signal, while limiting this computation to the intervals specified by the interval channel.

`IntervalMean` incorporates the additional data provided by `IntervalFrom1Level` to increase the precision of the computed results.

Since the time reference is lost, the result cannot be charged any further in many imc Online FAMOS functions and cannot be sent via XCPoE 952.

Example

```
; Determine the mean of a squared sine signal
; Theoretical value = 0.5, independent of the interval limit specified by
; IntervalFrom1Level
saw1 = SawTooth(Channel_001, 0, 0.01, 628)
signal1 = Sin(saw1)
signal2 = signal1 * signal1
interval = IntervalFrom1Level(saw1, 1.0, 0)
iMean1 = IntervalMean(signal2, interval)
```

IntervalMin

Determines the input signal's minimum value within the specified interval.

Result = IntervalMin(Signal, Interval)

Result: result channel	**Interval**: interval channel
Signal: input signal	

The interval channel is generated by the functions `IntervalFromLevels` or `IntervalFrom1Level`.

`IntervalMin` determines the input signal's minimum, while limiting this computation to the intervals specified by the interval channel. Since the time reference is lost, the result cannot be charged any further in many imc Online FAMOS functions and cannot be sent via XCPoE 952.

Example

```
; Determine min and max of a periodic signal between 20 and 40 degrees.
; the angle is obtained from an incremental encoder sensor in Angle-absolute-mode
interval = IntervalFromLevels( angle, 20, 40, 0 )
iMin = IntervalMin(signal, interval)
iMax = IntervalMax(signal, interval)
```

IntervalMult

Pulse count multiplier: From the input interval channel, generates a new interval channel with multiple pulse count.

Results = IntervalMult(Intervals, MinFrequency, Multiplier)

Result: Channel containing the multiplied interval codes	**MinFrequency**: Minimal frequency of the pulses contained, stated in Hz.
Intervals: Input channel containing the original interval codes. Sampling interval stated in s.	**Multiplier**: The pulse count is multiplied by this factor.

The function is mainly used for signals from sensors which generate one pules per revolution. Many algorithms (such as Resampling, ...) however require a higher pulse count per revolution. For this purpose, an interval signal is generated from the pulse signal, which this function in turn multiplies by the pulse count. The results can only be processed with other interval functions.

Multiplication only occurs if the frequency of the pulses the signal contains is low enough. The reason is that the results channel can only contain a single pulse per sampling interval. If the input signal has one pulse every 10 samples, for instance, then the maximum multiplication factor is 10, since in that case each result sample contains one pulse. To compensate small variations and to avoid numerical problems, this maximum resulting pulse frequency has to be reduced by the factor 0.999. So, for an input signal of 1000 samples per pulse the multiplicator may be at most 999, for 100 samples per pulse at most 99, and for the 10 samples per pulse in the example above at most 9.

If the input signal contains a too rapid sequence of pulses, the pulses are deleted. The function either multiplies, or discards. Thus, in the result, either there are no pulses or the multiplied amount.

Also, the multiplication only takes place if the signal has enough pulses per time unit. The parameter MinFrequency specifies the minimum pulse frequency which is enough for multiplication to take place.

If the pulses arrive at a slower rate, they will be deleted. Thus, the pulse frequency may only lie within the range [MinFrequency ... SamplingFrequency/Multiplier].

The function requires temporary memory of the size [SamplingFrequency / MinFrequency], for which reason the minimum frequency may not be chosen too low.

Example

```
; A drive shaft sensor connected to the incremental counter interface returns
; one pulse per revolution. For subsequent resampling by the angle, however,
; 120 pulses per revolution are needed.
; The shaft spins at 600..6000 RPM. 600 RPM = 10 Hz = MinFrequency.
; SamplingFrequency = 20 kHz > 12 kHz = 120*100 Hz, 100 Hz = 6000 RPM.
ivl1 = IntervalFromPulse(ImpulseTime, 0, 0) ; create intervals
ivl120 = IntervalMult(ivl1, 10, 120) ; multiply pulses
res = IntervalResample(Channel_001, ivl120, 360, 120, "Degree", 1) ; resampling
```

IntervalResample

Resampling of a periodic signal

Result = IntervalResample(Signal, Interval, MaxAngle, AmtPoints, Unit, IsSegmented)

Result: Results channel

Signal: Measurement signal

Interval: Interval signal

MaxAngle: Value of total angle or unit length

AmtPoints: Number of encoder pulses per total angle or unit length

Unit: Value of the result

IsSegmented: Is the result segmented?

0: no

1: yes

Two channels are captured simultaneously which contain time-domain data: one channel with the actual measured signal returning position-dependent measured data, and one channel with the associated interval data.

These interval data were generated from an incremental encoders's signal using the function IntervalFromPulse, which works in the impulse time mode. By means of IntervalResample, values for the measurement signal at the time points of interest are linearly interpolated from the pulse time data. In the process, no more values may be generated than specified y the sampling interval.

MaxAngle is the value to which the number of pulses corresponds, e.g. 60 encoder pulses per 360 degrees, per 2*pi or per revolution (for oscillation signals) or even 2 encoder pulses per meter.

Example

```
interval = IntervalFromPulse(Inc_Encoder_01, 60, 0) ; Interval codes
res = IntervalResample( Channel_001, interval, 360.0, 60, "Deg", 1) ; resampling
```

IntervalRMS

Determines the root mean square of the input signal within the specified interval.

Result = IntervalRMS(Signal, Interval)

Result: results channel

Signal: input signal

Interval: interval channel

The interval channel is generated by the functions `IntervalFromLevels` or `IntervalFrom1Level`.

`IntervalRMS` determines the root mean square of the input signal, while limiting this computation to the intervals specified by the interval channel.

`IntervalRMS` incorporates the additional data provided by `IntervalFrom1Level` to increase the precision of the computed results.

Since the time reference is lost, the result cannot be charged any further in many imc Online FAMOS functions and cannot be sent via XCPoE 952.

Example

```
; Determine the root mean square of a sine signal
; Theoretical value = 0.707107, independent of the interval limit specified
; by IntervalFrom1Level
saw1     = SawTooth( Channel_001, 0, 0.01, 628 )
signal1  = Sin( saw1 )
interval = IntervalFrom1Level( saw1, 1.0, 0 )
iRMS1    = IntervalRMS( signal1, interval )
```

IsSynchronized

This function returns whether the internal device clock is synchronized.

Prerequisite:
Only available in: imc Online FAMOS!

a = IsSyncronized(b)

a: Result 1: synchronized 0: not synchronized -1: unable to determine/no clock present	b: clock, only for establishing the data rate.

JKFlipFlop

JK flip-flop function

a = JKFlipFlop(J, K)

a: Result	J: J input K: K input

Returns a 1 for HIGH (H) state or 0 for LOW (L) state.

A change to HIGH state occurs if J is not equal to zero and K is equal to zero.

Likewise, a change to LOW state occurs if J is equal to zero and K is not equal to zero.

If both J and K are both equal to zero, the state does not change.

If neither J nor K are equal to zero, the state changes.

The function may also be called with single values (e.g. virtual bit or Display variable) as its parameter. In order to obtain useful results in such a case, calls of the function should occur at constant time intervals, e.g. in a Timer or a synchronous task.

Example

```
; The LED flashes while the signal is larger than 9.
res = Greater( Channel_001, 9 )
LED_01 = JKFlipFlop( res, res )
```

11.2.9.2.10 L

LEQ

LEQ-value: The function determines the LEQ-value in dependence on the time. The result states the LEQ for all previously processed values of the input signal.

LEQSignal = LEQ(Signal, FrequencyRating, ReductionFactor)

LEQSignal: Evaluated signal **Signal**: Signal to be evaluated	**FrequencyRating**: Frequency rating of the signal 0: no rating 1: A-rating 2: B-rating 3: C-rating	**ReductionFactor**: Factor for resampling, ≥ 1. For a reduction factor = 1, no resampling

At the end of the measurement, the last value of the LEQ-signal displays the LEQ for the entire measurement. The signal is first frequency-weighted, e.g. with an A-rating according to DIN IEC 651.

The function requires a sound signal expressed in the unit Pa (Pascal). The result is referenced to $20*10^{-6}$ Pa and expressed in dB. The result can always be interpreted as the mean sonic pressure level of all data already captured.

For display in a curve window, representation as "Last value as number" is recommended. This way, the number appearing will gradually level off at the ultimate value, which can then be read at the end of the measurement.

The reduction factor can be selected in such a way that multiple values per second are generated.

Example

```
LEQ_Chan1 = LEQ( Chan1, 1, 1000 )
```

The signal is subject to an A-rating and resampled by the factor 1000. The signal originally had a sampling frequency of 20kHz, the result only has 20 Hz. The result is the mean sound pressure level LEQ, expressed in dB.

Less

Checks whether the first argument is less than the second argument

a = Less(b, c)

a = 1 if b is less than c, otherwise 0.

LessEqual

Checks whether the first argument is less than or equal to the second argument

a = LessEqual(b, c)

a = 1 if b is less than or equal to c, otherwise 0.

Ln

Natural logarithm of input channel

a = Ln(b)

The logarithm with the base e (Euler number) is formed.

LogAnd

Logical AND operator applied to B and C.

A = LogAnd(B, C)

A = 1, if B is nonzero and C is nonzero.
A = 0, if B is nonzero and C = 0.
A = 0, if B = 0 and C is nonzero.
A = 0, if B = 0 and C = 0.

LogNot

Logical inverse of B.

A = LogNot(B)

A = 1, if B = 0.
A = 0, if B is nonzero.

LogOr

Logical OR operator applied to B and C.

A = LogOr(B, C)

A = 1, if B is nonzero and C is nonzero.
A = 1, if B is nonzero and C = 0.
A = 1, if B = 0 and C is nonzero.
A = 0, if B = 0 and C = 0.

LogXor

Logical XOR operator applied to B and C.

A = LogXor(B, C)

A = 1, if B = 0 and C is nonzero.
A = 1, if B is nonzero and C = 0.
A = 0, if B = 0 and C = 0.
A = 0, if B is nonzero and C is nonzero.

Lower

Returns the lesser value of the arguments

a = Lower(b, c)

11.2.9.2.11 M

Max

Maximum of a window, with reduction

a = Max(b, WS, RF)

a: result	**WS**: window size in samples; **In imc Online FAMOS:** 1 * RF, 2 * RF, ... ,10 * RF **In imc Inline FAMOS:** 1 * RF, 2 * RF, ... ,1000 * RF
b: input channel	**RF**: reduction factor in samples; ≥ 1

Returns the maximum value in a window with WS samples for every RF input samples.

Thus for RF > 1, data reduction is performed.

The window size must be an integer multiple (1 to 10/1000) of the reduction factor.

Example

```
high = Max( Channel_001, 20, 4 )
```

Returns for every fourth input sample the maximum of the last 20 input samples.

Note

If the windows size is not 1*RF, there are not enough input samples in the beginning. The result is then computed with the available input samples.

Mean

Mean value of a window, with reduction

a = Mean(b, WS, RF)

a: result	**WS**: Window size; in samples;
b: input channel	**In imc Online FAMOS:** 1 * RF, 2 * RF, ... ,10 * RF **In imc Inline FAMOS:** 1 * RF, 2 * RF, ... ,1000 * RF
	RF: Reduction factor in samples; ≥ 1

The mean function returns the mean value of a window with WS samples for every RF input samples.

Thus for RF > 1, data reduction is performed.

The window size must be an integer multiple (1 to 10/1000) of the reduction factor.

Example

```
average = Mean( Channel_001, 6, 2 )
```

Returns for every second input sample the mean value of the last 6 input samples.

Note

If the windows size is not 1*RF, there are not enough input samples in the beginning. The result is then computed with the available input samples.

Internal calculation precision: 40 bit.

Median3

Median of three values: The input channel is smoothed using the median of the last three samples

a = Median3(b)

Each group of three samples is sorted by amplitude in decreasing order. The result is the middle element of the sorted list.

Median5

Median of five values: The input channel is smoothed using the median of the last five samples

a = Median5(b)

Each group of five samples is sorted by amplitude in decreasing order. The result is the middle element of the sorted list.

Min

Minimum of a window, with reduction

a = Min(b, WS, RF)

a: result	**WS**: Window size in samples;
b: input channel	**In imc Online FAMOS:**
	1 * RF, 2 * RF, ... ,10 * RF
	In imc Inline FAMOS:
	1 * RF, 2 * RF, ... ,1000 * RF
	RF: Reduction factor in samples; ≥ 1

The minimum function gives the minimum of a window with WS samples for every RF input samples.

Thus for RF > 1, data reduction is performed.

The window size must be an integer multiple (1 to 10/1000) of the reduction factor.

Example

```
Low = Min( Channel_001, 10, 2 )
```

Returns for every second input sample the minimum of the last 10 input samples.

Note

If the windows size is not 1*RF, there are not enough input samples in the beginning. The result is then computed with the available input samples.

Monoflop

a = Monoflop(b, Duration)

a: result	**Duration**: Pulse duration in samples; ≥ 1
b: input channel	

If Monoflop detects a positive edge in the input signal (a transition of zero to non-zero) it produces a pulse of predetermined duration with amplitude 1.

Not retriggerable, i.e. signal edges are only evaluated after the end of the pulse.

The function may also be called with a single value (e.g. virtual bit or Display variable) as its first parameter. In order to obtain useful results in such a case, calls of the function should occur at constant time intervals, e.g. in a Timer or a synchronous task.

MonoflopRT

Retriggerable monoflop

a = MonoflopRT(b, Duration)

a: result	**Duration**: Pulse duration in samples; ≥ 1
b: input channel	

If MonoflopRT detects a positive edge in the input signal (a transition of zero to non-zero) it produces a pulse of predetermined duration with amplitude 1.

Additional positive edges within this duration prolong the output pulse.

The function may also be called with a single value (e.g. virtual bit or Display variable) as its first parameter. In order to obtain useful results in such a case, calls of the function should occur at constant time intervals, e.g. in a Timer or a synchronous task.

11.2.9.2.12 N

NorthCorrection

Correction of north transition in a window with WF samples.

a = NorthCorrection(b, WF)

a: result	**WF**: window size in samples; ≥ 1
b: argument	

The function `NorthCorrection` corrects the discontinuity at due North according to the addition method. It prevents skipping from 360° to 0° in the averaging interval.

For values fluctuating around 360°, an average of 180° is avoided.

The result of averaging could lie outside of the wind rose range 0°..360°, e.g 365°.

The function `WindRoseCorr` 1042 returns the result to the wind rose range (0°..360°), e.g. changes 365° to 5°.

This procedure is conditional on the wind direction not changing more than 100 degrees.

The only sensible way to use the functions `NorthCorrection` and `WindRoseCorr` 1042 is in a combination as illustrated in the example below. Instead of averaging, the standard deviation, i.e. the function `StDev` 1016, can be used.

Example

```
NC = NorthCorrection( Channel, 10 )
NC_Mean = Mean( NC, 10, 10 )
Result = WindRoseCorr( NC_Mean )
```

NOT

Logical inverse of A.

Prerequisite:
Only usable with control commands!

LogicalNot = NOT A

LogicalNot: Result	**A**: Operand

LogicalNOT = 1, if A = 0.
LogicalNOT = 0, if A is nonzero.

The NOT-Operator may only be applied to individual variables.

If applied to whole expressions, for example `If NOT( pv.x > 0) = 0`, the NOT-operator will not return correct results. In such cases, the function `LogNot` should be used instead; thus for example: `If LogNot(pv.x > 0) = 0`.

Example

```
OnTriggerMeasure( BaseTrigger )
    Value = CurrentValue( Channel_001, 0, 0.0 )
    If NOT Value > 0
        VirtChannel_001 = Channel_001 + 5
    Else
        VirtChannel_001 = Channel_001 + 10
    End
End
```

NumberOfPulses

Returns the number of pulses in reduction window

a = NumberOfPulses(b, RF)

a: Result	**RF**: Reduction factor in samples; ≥ 1
b: Input channel	

Returns the number of pulses in the window of RF samples and does a reduction with a factor of RF.

A pulse is a sequence of two edges, an edge being a transition from zero to non-zero or vice versa.

Analog signals should be preprocessed (`STri` 1017).

Example

```
_tmp = Stri( Channel_001, 0, 0 )
pulses = NumberOfPulses( _tmp, 100 )
```

Returns for every 100th input sample the number of pulses in the last 100 input samples.

11.2.9.2.13 O

OnECUCmdReturn_ECU_001

See ECU-functions 933.

OR

Logical OR operator applied to A and B.

Prerequisite:
Only usable with control commands!

LogicalOr = A OR B

LogicalOr: Result	**A**: 1st operand **B**: 2nd operand

LogicalOR = 1, if A is nonzero and B is nonzero.
LogicalOR = 1, if A is nonzero and B = 0.
LogicalOR = 1, if A = 0 and B is nonzero.
LogicalOR = 0, if A = 0 and B = 0.

The Or-operator may only be used on operands of type BOOL. A result of type BOOL is returned by the operators <, <=, >, >=, =, <>, AND, OR and NOT.

Example

```
OnTriggerMeasure( BaseTrigger )
    Value1 = CurrentValue( Channel_001, 0, 0.0 )
    Value2 = CurrentValue( Channel_002, 0, 0.0 )
    If Value1 > 0 OR Value2 > 0
        VirtChannel_001 = Channel_001 + 10
    Else
        VirtChannel_001 = Channel_001 + 5
    End
End
```

General information about the Otr functions

Prerequisite:
Only available in: imc Online FAMOS!
Device enabled for: "*Online Order tracking*"

OtrAngleAdd

Addition of an angle to an angle signal

Result = OtrAngleAdd(Angle, Add)

Result: results channel

Angle: angle signal

Add: angle to be added

The function adds a user-specified angle to the angle signal supplied, which is derived from an incremental encoder sensor operating in absolute angle mode. The resulting angle channel can be processed by `OtrResampleAngle`.

Changes to the angle to be added are always made only when the resulting angle signal's negative signal edge arrives. This results in segments having a negative angle, which are ignored by `OtrResampleAngle`.

Example

```
; angle offset for zero correction
angle = OtrAngleAdd(Inc_encoder_01, -0.1)
; resampling
res = OtrResampleAngle(Channel_001, angle, 360.0, 360, "Degrees", 1)
```

OtrEncoderPulsesToRpm

Determines the RPM-value: The function determines an RPM time history from a pulse signal.

a = OtrEncoderPulsesToRpm(b, Signal_Type, EncoderType, EncoderPulse, RPM_Min, Reduction)

a: Result - RPMHistory: Time-history of RPM-signal

b: Input channel - Pulse_Signal: Time-history of the pulse signal

Signal_Type: Type of pulse signal

- 0: Number of events
- 1: Time of pulse
- 2: Sampled rectangle
- 3: Sine signal

EncoderType: Type of encoder, how many missing teeth?

- 0: Default
- 1: 1 tooth missing
- 2: 2 teeth missing

EncoderPulse: Pulses per revolution, ≥ 1

RPM_Min: Minimum occurring RPM-value

Reduction: Factor by which the number of data points in the result is less than the input data

The time-history of the RPMs is determined. The function returns 0.0 until at least 2 pulses have been detected. Only then is a computation of RPM possible. The quality of the computation depends on the signal type. For instance, Time of pulse measurement allows very good computation. Since the function is carried out online in the device, it must return an estimated value in some instances (due to unknown location of future).

Notes on Signal_type

0	Number of events. The pulse signal reflects the number of pulses counted in a sampling interval. The signal is received if the device's incremental counter input is set to "Event counting". The signal comprises a sequence of integers. Each number represents the number of pulses counted within the current sampling interval. The RPM-value is assumed to be directly proportional to the number of pulses counted. The function smoothes the RPM signal in order to allow a better RPM estimate. If the signal consisted of the sequence of values { ...., 3, 4, 4, 4, 3, 4, 4, 4, 3... }, then for this portion, a RPM-value of approx. 3.75 can be estimated. The function treats each value of the signal as an number of pulses. If the signal consisted of the sequence of values { 0, 0, 1, 0, 1, 2, 1 }, this amounts to 5 pulses detected. This signal type comes with a certain amount of imprecision, since it doesn't reflect at what time within the sampling interval the pulses were counted.

1	Time of pulse. The signal is received if the device's incremental counter input is set to "Time of pulse". In this mode, an encoder pulse's exact position in time in the frame of the sampling interval is determined. This involves very precise capture of the pulse. Only this signal type allows such precise sampling in relation to the angle. In this mode it is also important that the measured RPM value be small enough so that there is never more than one encoder pulse per sampling interval. The sampling interval may have to be adjusted to a small enough value for this purpose. RPM_Max must then also be set to a correspondingly low value. Maximum possible RPM [revs/min] = 60 / (encoder pulses * sampling interval [s])
2	Sampled rectangle. One of the device's digital inputs is used to sample the time-history of the encoder's digital output. At every transition from zero to nonzero, the encoder is assumed to have turned by one increment. If the signal consisted of the sequence of values { 0, 0, 1, 1, 1, 0, 0, 1, 1, 0 }, this amounts to 2 pulses detected. If an analog voltage is sampled and there is no hardware comparator, the analog voltage data must be converted to a digital data sequence. If the signal is noisy, it may need to be smoothed beforehand and then have a Schmitt-Trigger performed. This signal type comes with a certain amount of imprecision, since it doesn't reflect at what time within the sampling interval the edge occurred.
3	Sine signal. One of the device's digital inputs is used to sample the time-history of the encoder's (sinusoid) analog output. Sinusoid or other signals with zero-crossing in a positive slope can be processed. If the signal is noisy, it may need to be smoothed beforehand and then have a Schmitt-Trigger performed. A zero-crossing in a positive slope is interpreted as a pulse from the encoder. If the signal consisted of the sequence of values { -3.0, -1.0, +1.5, +2.8, +1.3, +0.1, -0.6 }, a pulse is detected at the transition from -1.0 to +1.5.

Note — Notes on EncoderType

Special features of encoders with missing pulses:

The number of encoder pulses is always specified to include the missing tooth. E.g. for an encoder emitting a pulse every 10 degrees of rotation, which should then have 36 teeth, the number of encoder pulses to specify is then also 36. But the encoder only generates 35 pulses because one is missing. Also typical are encoders generating a pulse every 6 degrees. In this case, 60 teeth are to be specified although 2 are missing and only 58 actually present. The first tooth after the gap is interpreted as the zero-pulse. This marks the start of sampling. Encoders with missing pulses can only be used with the signal type "Time of pulse". Recognizing the missing teeth is only possible if the RPM-value is relatively constant in the region around the gap. Particularly with extremely low RPM-values, this cannot be guaranteed. Since in that case, the gap cannot be clearly recognized, the minimum RPM must be set to a non-zero value. At higher RPM-values, the gap is usually clearly recognizable due to mechanical inertia. The function tries to re-synchronize itself after any error in the pulse sequence (or a presumably incorrectly interpreted pulse sequence). However, incorrect sampled values may occasionally occur.

Note — Notes on the RPM-range

If a value for RPM_Min > 0 is set, then signal components up to that RPM-value are disregarded for the result, and whole periods of OrderRef can be skipped in the process. The minimum RPM must not be set too low since the tracking filter is dynamically limited. RPM_Max may only be set so high that the data rate after sampling isn't greater than that of the input data. When OrderRef or Oversampling have high values, this generally is accompanied by a high data rate. In this case, the maximum RPM-value must be set correspondingly low.

Note — Notes on Reduction

The number of data points can be reduced by this factor. Default is 1. Since a high sampling rate sometimes must be used to pinpoint the location of the pulses, a correspondingly large reduction factor can be used to adapt the data rate to the RPMs. This is because the RPM-value often doesn't change so quickly.

Example 1

The RPMs of a toothwheel are to be determined. To this end, an inductive encoder is used to generate a rectangular voltage signal. This voltage is recorded at a constant sampling rate of 1 kHz. The voltage is about 0 V .. 3 V in the tooth-gap, and about 18 V .. 22 V at the tip of the tooth. The toothwheel has 8 teeth.

```
_Pulses = stri( Voltage, 15, 5 )
Revs = OtrEncoderPulsesToRpm( _Pulses, 2, 0, 8, 0, 1 )
; _STyp = 2 ; sampled rectangle signal
; _ETyp = 0
; _EPulse = 8 ; number of teeth
; _MinRevs = 0
; _Red = 1
; Revs = OtrEncoderPulsesToRpm( _Pulses, _STyp, _ETyp, _Pulse, _MinRevs, _Red )
```

Since the voltage isn't in an appropriate format, it is altered to produce a proper 0-1sequence.

Example 1

The device records the signal "Tacho1" of an encoder with 128 pulses using the Time-of-pulse mode. The sampling time is 0.1ms.

```
Revs = OtrEncoderPulsesToRpm( Tacho1, 1, 0, 128, 0, 1 )
; _STyp = 1 ; Time-of-pulse
; _ETyp = 0
; _EPulse = 128 ; Pulses per revolution
; _MinRevs = 0
; _Red = 1
; Revs = OtrEncoderPulsesToRpm( Tacho1, _STyp, _ETyp, _EPulse, _MinRevs, _Red )
```

The RPM-value may reach up to 4680 revs/min [= 60 / (128 * 0.0001)].

OtrFrequLine

Determines the frequency line: The function finds the magnitude or phase of a sinusoidal oscillating signal having a fixed period duration.

a = OtrFrequLine(VibrationSignal, PeriodLength, PeriodAmount, Option)

a: Result - Magnitude or phase time history	**Option**: Determines what is calculated
VibrationSignal: Signal with sinusoidal oscillation	0: Magnitude (effective value)
PeriodLength: Number of samples per period; ≥ 2.0	1: Phase (in degrees)
PeriodAmount: The number of periods over which an average is taken	

The function finds the value for the magnitude or for the phase of the oscillation for each interval of length PeriodLength * PeriodAmount. In other words, the number of oscillation in such an interval is given by PeriodAmount. The duration of the oscillations must be fixed and constant. PeriodLength does not have to be a integer number of samples. But the Product PeriodLength*PeriodAmount must be of a integer number of samples in length, determined by PeriodLength*PeriodAmount divided by the sampling interval.

The function finds one line of the discrete Fourier-spectrum (DFT) using Hanning-windowing.

If the signal contains significant components at other frequencies, a large number should be set for the number of periods in order to reduce the distorting effects of these other frequencies. If only a small number of periods is used, it may be advisable to connect a bandpass filter in front.

PeriodLength ≥ 2.0, PeriodAmount ≥ 1. The product of these two parameters may not exceed 2e9. The phase is stated in the range -180 degrees .. +180 degrees. the value of the phase is 0 degrees for a cosine oscillation, -90 degrees for a sine oscillation.

Example

A vibration signal is sampled in reference to the rotation angle (vib_revs), so that all oscillation components up to the 8th order are present. The signal thus contains 16 points per revolution. The phase of the 1st order is to be determined. One value for the phase is desired for every 5 revolutions.

```
Phase = OtrFrequLine( vib_revs, 16, 5, 1 )
```

OtrFrequLine2

Determining frequency lines: A signal containing an vibration of fixed period duration is approximated by a sinusoidal waveform. The magnitude and phase of this approximation are determined.

OtrFrequLine2(Magnitude, Phase, OscillationSignal, PeriodLength, PeriodCount)

Magnitude: Result of calculation (RMS-value)

Phase: Result of calculation (in degrees).

The phase is determined as in the range -180 degrees ... +180 degrees.
The phase's value is 0 degrees for a cosine oscillation, -90 degrees for a sine oscillation.

VibrationSignal: Input channel with vibration

PeriodLength: Number of sampling values in a period; ≥ 2.0

PeriodCount: Averging over this many periods; ≥ 1.

The product of PeriodLeng and PeriodCount may not exceed 2e9.

Within every interval of length PeriodLength * PeriodCount, this function determines the magnitude and the phase of the oscillation. This interval contains exactly the number PeriodCount oscillations.

The duration of the oscillations must be fixed and constant. The period length does not have to be an integer number of samples. But the Product PeriodLength*PeriodAmount must be an integer number of samples in length, determined by PeriodLength*PeriodAmount divided by the sampling interval.

The function determines a line of the discrete Fourier transform (DFT) with rectangular windowing.

If the signal additionally contains significant other frequency components, a large number of periods should be selected in order to reduce their distorting effects. When working with a low number of periods, it may be necessary to connect a bandpass filter upstream.

Example

A vibration signal is sampled over the angle (vib_revs), so that all vibration components up to the 8th order are included.

The signal thus comproses 16 points per revolution. The magnitude and phase of the 1st order are to be determined.

Every 5 revolutions, a value for the magnitude and phase each is to be returned.

```
OtrFrequLine2( mag, phase, vib_revs, 16, 5 )
```

OtrFrequLine3

Determining frequency lines: A signal containing an vibration of fixed period duration is approximated by a sinusoidal waveform. The magnitude and phase of this approximation are determined. Also, a sinusoidal oscillation is formed based on these results.

OtrFrequLine3(Magnitude, Phase, Sine, VibrationSignal, PeriodLength, PeriodCount, SampleCount)

Magnitude: Result of calculation (RMS-value)

Phase: Result of calculation (in degrees).

The phase is determined as in the range -180 degrees ... +180 degrees.
The phase's value is 0 degrees for a cosine oscillation, -90 degrees for a sine oscillation.

Sinus: Results of the calculation (sinusoidal oscillation)

VibrationSignal: Input channel with sinusoidal oscillation

PeriodLength: Number of sampling values in a period; ≥ 2.0

PeriodCount: Averging over this many periods; ≥ 1.

SampleCount: Length of the sinusoidal oscillation found as the results (in Samples)

The product of PeriodLeng and PeriodCount may not exceed 2e9.

The function determines in each interval of length PeriodLength * PeriodCount the value of the magnitude and phase of the oscillation. In each such interval, there are PeriodCount oscillations.

The duration of the oscillations must be fixed and constant. The period length does not have to be an integer number of samples. But the Product PeriodLength*PeriodAmount must be an integer number of samples in length, determined by PeriodLength*PeriodAmount divided by the sampling interval.

The function determines a line of the discrete Fourier transform (DFT) with rectangular windowing.

If the signal additionally contains significant other frequency components, a large number of periods should be selected in order to reduce their distorting effects. When working with a low number of periods, it may be necessary to connect a bandpass filter upstream.

The resulting sinusoidal oscillation can only be used to display instantaneous values. E.g. in the curve window by means of "last N samples". An interpretation of this channel to indicate time is generally not possible.

Example

A vibration signal is sampled over the angle (vib_revs), so that all vibration components up to the 8th order are included. The signal thus comproses 16 points per revolution. The magnitude and phase of the 1st order are to be determined. Every 5 revolutions, a value for the magnitude and phase each is to be returned.

A resulting oscillation with 50 points resolution is to be generated.

```
OtrFrequLine3( mag, phase, sinus vib_revs, 16, 5, 50 )
```

OtrOrderSpectrum

Order spectrum over RPMs: The order spectrum is determined from the vibration and pulse signal time-histories referenced to the RPMs.

OrderSpectrum = OtrOrderSpectrum(Vibration, RPM, RPM_Min, RPM_Max, RPM_Bin_Width, Resolution, OrderMax, AveragingType)

OrderSpectrum: Order spectrum referenced to the RPMs

Vibration: Time-history of vibration signal

RPM: Time-history of the RPMs. In scaled to U/min

RPM_Min: Lower limit of desired RPM-range

RPM_Max: Upper limit of desired RPM-range

RPM_Bin_Width: Width (interval) of an RPM-class

Resolution: Resolution of the order spectrum [1, 1/2, 1/3, 1/4, ...]

OrderMax: The highest order (line) in the order spectrum

AveragingType: The processing of all spectra within one RPM-class

0: Arithmetic mean

1: Maximum

2: Minimum

RPM_Min, RPM_Max and RPM_Bin_Width are scaled in revs/minute. For sampling, the absolute value of the RPMs is used, but the original RPM value is used for assignment to an RPM-class.

Resolution: 0.1, if 0.1 orders is the distance between lines in the order spectrum. The resolution must be an integer fraction of 1.0, in other words, 1, 1/2, 1/3, 1/4, ... The inverse value of the resolution is a measure of how many revolutions are used to determine the order spectrum. For example, with a resolution of 0.1, each spectrum is determined from 10 revolutions.

The spectral lines are stated as rms-values. Since the FFT is internally calculated from a somewhat larger number of data (a power of 2), some spectral lines are truncated. The first line in the spectrum is always set as zero. Rectangular windowing is used for the FFT.

The averaging works on the magnitude spectrum. The average RPM-value in a spectrum determines the RPM-class.

A tracking Butterworth lowpass filter is used for anti-aliasing. The signal thus filtered is sampled at intervals of rotational angle. Then, an FFT is computed. As a result of this procedure, higher intermediate order lines may show up during sampling than are present in the ultimately returned spectrum. The 3dB-order of the anti-aliasing filter is located, i.e., at the end of the spectrum. The FFT can consist of anywhere between 16 and 8192 points. A rectangular window is used.

The function works properly for:

OrderMax < 16 / (SampleTime_Vibration * RPM_Max)

...

OrderMax < 32 / (SampleTime_Vibration * RPM_Max)

where SampleTime_Vibration is the sampling time of the signal 'Vibration'.

Use the value 16 for a worst case estimate. But, depending on the current number values, it can be as high as 32. The factor of 2 for feasible values is due to the internal rounding up to a power of 2 of values for the FFT used. Obtaining the highest possible order is sometimes possible by changing the resolution.

The RPM's should change only slowly. The RPM's should not fall far below 1% of capacity. The RPM's must be used to compute a good approximation of the rotational angle by means of integration (summing up). Therefore, the RPM values should be reasonably exact.

If no spectrum is computed in a certain RPM class, this spectrum is filled with zeroes. During measurement, the results are transferred like a histogram, i.e., regular but infrequent transmission of intermediary results to the PC. The result is a segmented waveform. Each segment is an order spectrum.

Example

From the time-history of a vibration vib and of the RPMs rpm, the order spectrum is to be determined in relation to the RPMs. The RPM-time trace is sampled at a rate of 0.2 ms.

```
_Min = 1000.0  ; minimum of RPM range
_Max = 6000.0  ; maximum of RPM range
_Delta = 100.0 ; bin spacing of RPM classes
_Res = 0.1     ; spectrum resolution, i.e. 1/10 order visible, calculated
               ; from 10 revolutions
_OMax = 6.0    ; up to this order, lines in the spectrum are to be displayed.
_MeanType = 0  ; 0 (arithmet. mean)
OSpectrum = OtrOrderSpectrum( vib, rpm, _Min, _Max, _Delta, _Res, _OMax, _MeanType)
; _OMax = 6.0 < 16 / ( 0.0002 * 6000 ) = 13.3
```

OtrOrderSpectrumP

Order spectrum over RPMs: The order spectrum is determined from the vibration and pulse signal time-histories referenced to the RPMs.

a = OtrOrderSpectrumP(Vibration, PulseSignal, SignalType, EncoderType, EncoderPulse, RPM_Min, RPM_Max, RPM_Bin_Width, Resolution, OrderMax, AveragingType)

a: Result - Order spectrum referenced to the RPMs

Vibration: Time-history of vibration signal

PulseSignal: Time-history of pulse signal

SignalType: Type of pulse signal

- 0: Number of events
- 1: Time of pulse
- 2: Sampled rectangle
- 3: Sine signal

EncoderType: Type of encoder, how many missing teeth?

- 0: Default
- 1: 1 tooth missing
- 2: 2 teeth missing

EncoderPulse: Pulses per revolution, ≥ 1

RPM_Min: Lower limit of desired RPM-range

RPM_Max: Upper limit of desired RPM-range

RPM_Bin_Width: Width (interval) of an RPM-class

Resolution: Resolution of the order spectrum [1, 1/2, 1/3, 1/4, ...]

OrderMax: The highest order (line) in the order spectrum

AveragingType: The processing of all spectra within one RPM-class

- 0: Arithmet. mean
- 1: Maximum
- 2: Minimum

The RPM-values are derived from the pulse signal by aggregating the angle contribution of each pulse. The vibration signal is sampled in reference to the rotational angle. Then the spectrum is determined. That is the complicated, but exact algorithm.

RPM_Min, RPM_Max and RPM_Bin_Width are scaled in revs/minute. For sampling, the absolute value of the RPMs is used, but the original RPM value is used for assignment to an RPM-class.

Resolution: 0.1, if 0.1 orders is the distance between lines in the order spectrum. The resolution must be an integer fraction of 1.0, in other words, 1, 1/2, 1/3, 1/4, ...

The inverse value of the resolution is a measure of how many revolutions are used to determine the order spectrum. For example, with a resolution of 0.1, each spectrum is determined from 10 revolutions.

The spectral lines are stated as rms-values. Since the FFT is internally calculated from a somewhat larger number of data (a power of 2), some spectral lines are truncated. The first line in the spectrum is always set as zero. Rectangular windowing is used for the FFT.

The averaging works on the magnitude spectrum. The average RPM-value in a spectrum determines the RPM-class.

A tracking Butterworth lowpass filter is used for anti-aliasing. The signal thus filtered is sampled at intervals of rotational angle. Then, an FFT is computed. As a result of this procedure, higher intermediate order lines may show up during sampling than are present in the ultimately returned spectrum. The 3dB-order of the anti-aliasing filter is located, i.e., at the end of the spectrum. The FFT can consist of anywhere between 16 and 8192 points. A rectangular window is used.

The function works properly for:

OrderMax < 16 / (SampleTime_Vibration * RPM_Max)

...

OrderMax < 32 / (SampleTime_Vibration * RPM_Max)

where SampleTime_Vibration is the sampling time of the signal 'Vibration'.

Use the value 16 for a worst case estimate. But, depending on the current number values, it can be as high as 32. The factor of 2 for feasible values is due to the internal rounding up to a power of 2 of values for the FFT used. Obtaining the highest possible order is sometimes possible by changing the resolution.

Notes on signal type

0	Number of events. The pulse signal reflects the number of pulses counted in a sampling interval. The signal is received if the device's incremental counter input is set to "Event counting". The signal comprises a sequence of integers. Each number represents the number of pulses counted within the current sampling interval. If the signal consisted of the sequence of values { 0, 0, 1, 0, 1, 2, 1 }, this amounts to 5 pulses detected. This signal type comes with a certain amount of imprecision, since it doesn't reflect at what time within the sampling interval the pulses were counted.
1	Time of pulse. The signal is received if the device's incremental counter input is set to "Time of pulse". In this mode, an encoder pulse's exact position in time in the frame of the sampling interval is determined. This involves very precise capture of the pulse. Only this signal type allows such precise sampling in relation to the angle. In this mode it is also important that the measured RPM value be small enough so that there is never more than one encoder pulse per sampling interval. The sampling interval may have to be adjusted to a small enough value for this purpose. RPM_Max must then also be set to a correspondingly low value. Maximum possible RPM [revs/min] = 60 / (encoder pulses * sampling time [s])
2	Sampled rectangle. One of the device's digital inputs is used to sample the time-history of the encoder's digital output. At every transition from zero to nonzero, the encoder is assumed to have turned by one increment. If the signal consisted of the sequence of values { 0, 0, 1, 1, 1, 0, 0, 1, 1, 0 }, this amounts to 2 pulses detected. If an analog voltage is sampled and there is no hardware comparator, the analog voltage data must be converted to a digital data sequence. If the signal is noisy, it may need to be smoothed beforehand and then have a Schmitt-Trigger performed. This signal type comes with a certain amount of imprecision, since it doesn't reflect at what time within the sampling interval the edge occurred.
3	Sine signal. One of the device's digital inputs is used to sample the time-history of the encoder's (sinusoid) analog output. Sinusoid or other signals with zero-crossing in a positive slope can be processed. If the signal is noisy, it may need to be smoothed beforehand and then have a Schmitt-Trigger performed. A zero-crossing in a positive slope is interpreted as a pulse from the encoder. If the signal consisted of the sequence of values { -3.0, -1.0, +1.5, +2.8, +1.3, +0.1, -0.6 }, a pulse is detected at the transition from -1.0 to +1.5.

Note Notes on EncoderType

Special features of encoders with missing pulses:

The number of encoder pulses is always specified to include the missing tooth. E.g. for an encoder emitting a pulse every 10 degrees of rotation, which should then have 36 teeth, the number of encoder pulses to specify is then also 36. But the encoder only generates 35 pulses because one is missing. Also typical are encoders generating a pulse every 6 degrees. In this case, 60 teeth are to be specified although 2 are missing and only 58 actually present. The first tooth after the gap is interpreted as the zero-pulse. This marks the start of sampling. Encoders with missing pulses can only be used with the signal type "Time of pulse". Recognizing the missing teeth is only possible if the RPM-value is relatively constant in the region around the gap. Particularly with extremely low RPM- values, this cannot be guaranteed. Since in that case, the gap cannot be clearly recognized, the minimum RPM must be set to a non-zero value. At higher RPM-values, the gap is usually clearly recognizable due to mechanical inertia. The function tries to re-synchronize itself after any error in the pulse sequence (or a presumably incorrectly interpreted pulse sequence). However, incorrect sampled values may occasionally occur.

The RPM-value should change only slowly. It should not sink substantially below 1% of the maximum value. It follows that the minimum RPM-value must be > 0.0.

If no spectrum is determined for a particular RPM-class, this spectrum is filled with zeroes. During measurement, the result is transferred like a histogram, i.e., regular but few transfers of intermediate results to the PC. The result is a segmented data set. Each segment is an order spectrum.

Example

The order spectrum referenced to the RPMs is to be determined from a vibration signal vib and the pulse signal Inc01. The time signals are sampled at rate of 0.2 ms. Inc01 is recorded in the Time-of-pulse mode of the incremental counter input. The encoder has 8 markings per revolution.

```
OSpectrum = OtrOrderSpectrumP( vib, Inc01, 1, 0, 8, 1000, 6000, 100, 0.1, 6.0, 0 )
; _SType = 1       ; time-of-pulse
; _EType = 0
; _EPulse = 8      ; number of markings
; _Min = 1000.0    ; minimum in RPM range
; _Max = 6000.0    ; maximum in RPM range
; _Delta = 100.0   ; bin-width of RPM classes
; _Res = 0.1       ; resolution of spectrum, meaning 1/10 order is visible,
                   ; computed over 10 revolutions
; _OMax = 6.0      ; lines in the spectrum up to this order are to be displayed
; _Mean = 0        ; 0 (arithmet. mean)
; OSpec = OtrOrderSpectrumP( vib, Inc01, _SType, _EType, _EPulse, _Min, _Max,
; _Delta, _Res, _OMax, _Mean )
; _OMax = 6.0 < 16 / ( 0.0002 * 6000 ) = 13.3
```

OtrPulseDuration

Pulse time measurement: Measurement of pulse duration for incremental encoders

Result = OtrPulseDuration(InputSignal, DxResult)

Result: Pulse duration	**DxResult**: Delta-X for the result channel, > 0
InputSignal: Signal from incremental encoder sensor	

The time that passes between two pulses of the input signal is determined with a precision of 1/32000 of the sample time. The input signal comes from an incremental encoder sensor that operates in pulse time mode.

An output occurs only when there is a pulse on the input, that is, the result may not be combined with other channels.

When multiple pulses fall into one sample interval, a 0 is returned.

Example

```
; 1 increment per pulse
pulse_dur = OtrPulseDuration( Ink_Geber_01, 1 )

; 10 increments per revolution
pulse_dur = OtrPulseDuration( Ink_Geber_01, 0.1 )

; 10 increments per 360 degrees angle
pulse_dur = OtrPulseDuration( Ink_Geber_01, 36 )
```

OtrResample

Angle sampling: Sampling of a vibration signal in reference to the rotation angle, given a pulse signal. A tracking, anti-aliasing filter is used.

AngleSignal = OtrResample(Vibration, Pulse_Signal, Signal_Type, EncoderType, EncoderPulse, OrderRef, Oversampling, Order3dB, FilterOrder, RPM_Min, RPM_Max, DelayTime)

AngleSignal: The signal sampled by angle

Vibration: Time-history of a vibration signal

Pulse_Signal: Time-history of the pulse signal

Signal_Type: Type of pulse signal

- 0: Number of events
- 1: Time of pulse
- 2: Sampled rectangle
- 3: Sine signal

EncoderType: Type of encoder, how many missing teeth?

- 0: Default
- 1: 1 tooth missing
- 2: 2 teeth missing

EncoderPulse: Pulses per revolution, ≥ 1

OrderRef: A visible order (line) in the result

Oversampling: Oversampling for OrderRef

Order3dB: Order, at which the low-pass filter dampens by 3dB

FilterOrder: The filter-order of the low-pass filter (1..10)

RPM_Min: Minimum occurring RPM-value

RPM_Max: Maximum occurring RPM-value

DelayTime: Delaytime, stated in seconds

The rotational angle is derived from the pulse signal by aggregating the angle increment of each pulse. The resulting signal trace over the angle is scaled to the x-coordinate, reflecting the number of revolutions made. The x-coordinate starts at 0, has the value of 0.5 after half a revolution, after a full revolution 1.0, after 2 whole revolutions 2.0 etc. The data rate for the result : 0.5 / (OrderRef * Oversampling) The absolute value for the RPMs is used. A Butterworth low-pass filter is used for anti-aliasing.

The function works properly for:

OrderMax < 24 / (SampleTime_Vibration * RPM_Max)

where SampleTime_Vibration is the sampling interval of the signal 'Vibration'.

Note that the highest frequency components of the resulting signal are already very heavily dampened.

Order3dB << OrderMax = OrderRef * Oversampling

<< means: much less than.

The RPM's should change only slowly. The RPM's should not fall far below 1% of capacity. The upper cutoff frequency of the low-pass filter must always be substantially below half of the vibration signal's sampling rate. Above about (0.4 * sampling rate), filtering cannot be performed.

If the current RPM-value rises above RPM_Max during measurement, whole oscillations of the order given by the parameter OrderRefare rejected.

The rotation angle is derived from the pulse signal by integration (aggregating the values). The function works with constant accuracy even for long-term measurements and is preferable to the function `OtrResampleFromRpm`.

Linear interpolation is used to find intermediate values. Between 2 pulses, the RPM-value is assumed to be constant. Interpolation is possible up to a maximum factor of 100 .

The function disregards the first 32 (approximately) measurement values in the signal.

Notes on signal type

0	Number of events. The pulse signal reflects the number of pulses counted in a sampling interval. The signal is received if the device's incremental counter input is set to "Event counting". The signal comprises a sequence of integers. Each number represents the number of pulses counted within the current sampling interval. If the signal consisted of the sequence of values { 0, 0, 1, 0, 1, 2, 1 }, this amounts to 5 pulses detected. This signal type comes with a certain amount of imprecision, since it doesn't reflect at what time within the sampling interval the pulses were counted.
1	Time of pulse. The signal is received if the device's incremental counter input is set to "Time of pulse". In this mode, an encoder pulse's exact position in time in the frame of the sampling interval is determined. This involves very precise capture of the pulse. Only this signal type allows such precise sampling in relation to the angle. In this mode it is also important that the measured RPM value be small enough so that there is never more than one encoder pulse per sampling interval. The sampling interval may have to be adjusted to a small enough value for this purpose. RPM_Max must then also be set to a correspondingly low value. Maximum possible RPM [revs/min] = 60 / (encoder pulses * sampling interval [s])
2	Sampled rectangle. One of the device's digital inputs is used to sample the time-history of the encoder's digital output. At every transition from zero to nonzero, the encoder is assumed to have turned by one increment. If the signal consisted of the sequence of values { 0, 0, 1, 1, 1, 0, 0, 1, 1, 0 }, this amounts to 2 pulses detected. If an analog voltage is sampled and there is no hardware comparator, the analog voltage data must be converted to a digital data sequence. If the signal is noisy, it may need to be smoothed beforehand and then have a Schmitt-Trigger performed. This signal type comes with a certain amount of imprecision, since it doesn't reflect at what time within the sampling interval the edge occurred.
3	Sine signal. One of the device's digital inputs is used to sample the time-history of the encoder's (sinusoid) analog output. Sinusoid or other signals with zero-crossing in a positive slope can be processed. If the signal is noisy, it may need to be smoothed beforehand and then have a Schmitt-Trigger performed. A zero-crossing in a positive slope is interpreted as a pulse from the encoder. If the signal consisted of the sequence of values { -3.0, -1.0, +1.5, +2.8, +1.3, +0.1, -0.6 }, a pulse is detected at the transition from -1.0 to +1.5.

Note — Notes on OrderRef and Oversampling

The maximum order in the result is OrderRef * Oversampling

For the maximum order, 2 points in the angle signal are allotted per period. OrderRef denotes the order line in the angle signal which can be clearly recognized. Also, this order should always be displayed in-phase as far as possible even if the RPM-value is so high that some signal components have to be skipped over. If the data rate of the angle signal becomes too high, whole periods of OrderRef are skipped. Oversampling may only be an integer factor ≥ 1. OrdRef can also be a fractional order.

Note — Notes on EncoderType

Special features of encoders with missing pulses:

The number of encoder pulses is always specified to include the missing tooth. E.g. for an encoder emitting a pulse every 10 degrees of rotation, which should then have 36 teeth, the number of encoder pulses to specify is then also 36. But the encoder only generates 35 pulses because one is missing. Also typical are encoders generating a pulse every 6 degrees. In this case, 60 teeth are to be specified although 2 are missing and only 58 actually present.

The first tooth after the gap is interpreted as the zero-pulse. This marks the start of sampling. Encoders with missing pulses can only be used with the signal type "Time of pulse". Recognizing the missing teeth is only possible if the RPM-value is relatively constant in the region around the gap. Particularly with extremely low RPM-values, this cannot be guaranteed. Since in that case, the gap cannot be clearly recognized, the minimum RPM must be set to a non-zero value. At higher RPM-values, the gap is usually clearly recognizable due to mechanical inertia. The function tries to re-synchronize itself after any error in the pulse sequence (or a presumably incorrectly interpreted pulse sequence). However, incorrect sampled values may occasionally occur.

Note Notes on the filter

If no low-pass filter is to be used, the following settings must be made:

```
Order3dB = 0
FilterOrder = 0
```

Always use a 1st or 2nd order filter.

Note Notes on the RPM-range

If a value for RPM_Min > 0 is set, then signal components up to that RPM-value are disregarded for the result, and whole periods of OrderRef can be skipped in the process. The minimum RPM must not be set too low since the tracking filter is dynamically limited.

RPM_Max may only be set so high that the data rate after sampling isn't greater than that of the input data. hen OrderRef or Oversampling have high values, this generally is accompanied by a high data rate. In this case, the maximum RPM-value must be set correspondingly low.

Note Notes on DelayTime

DelayTime = 0.0 Default ≥ 0.0

A delay refers to when the vibration signal lags behind the RPM-signal because of analog circuitry such as filters and amplifiers. A constant lag in the time signal results in a RPM-dependent delay in the angle signal. This can be compensated. The delay time is assumed to be constant. This isn't true exactly, but as a good approximation. The time lag is not a property of the device, but rather of the measurement chain and must be determined empirically by a control measurement.

Example

A vibration signal vib is sampled at a rate of 1 ms. The RPMs can go up to 4000 revs/min. The vibration is to be plotted over the rotation angle. There should be signal components up to the 5th order. The device's incremental counter input is set to "Time of pulse", the pulse signal is entitled Inc01. The encoder has 12 markings per revolution. RPM-values below 10 revs/min are to be ignored.

```
res = OtrResample( vib, Inc01, 1, 0, 12, 5.0, 1, 2.7, 8, 10.0, 4000.0, 0.0 )
```

The 8th-order anti-aliasing filter is designed to dampen the 2.7th order by 3 dB, th 5th order by 60 dB. For the 2.3th order, the amplitude error is already less than 5%.

The following holds: OrderMax = 5.0 * 1 ≤ 24 / (0.001 * 4000) = 6.0

Calculation of maximum order: The result has a resolution of 0.5 / OrderMax = 0.1 revolution.

It has 10 sample values per revolution.

OtrResampleAngle

Resampling of a periodic signal

Result = OtrResampleAngle(Vibration, Angle, MaxAngle, AmtPoints, Unit, IsSegmented)

Result: results channel

Vibration: Measurement signal

Angle: angle signal

MaxAngle: angle maximum

AmtPoints: number of points

Unit: unit of the results

IsSegmented: are the result data segmented?

0: no

1: yes

Two channels containing time-based data arrive simultaneously: One channel with the actual measurement signal returning periodic measured data, and one channel with associated angle data coming from an incremental encoder sensor operating in absolute angle mode.

Resampling generates a channel which represents the measured data as a function of the angle, where the angle-based data result from interpolation from the time data.

The magnitude of the angle maximum serves the purpose of scaling the results and states whether a revolution is interpreted as 360°, 720° or 2*pi.

The amount of points determines how many measurement points on the angle axis are to be generated by resampling. This number may not be so large that the data volume would be increased by resampling.

Example 1

The signal of the incremental encoder sensor, corrected by -0.1°, is passed to the function `OtrResampleAngle`, together with the periodic measurement signal on Channel_001. The result of resampling is a segmented record that maps the measurement signal to the angle in the range of 0°...360°. The amount of points for one revolution is 360, i.e. the angle resolution is 1°.

```
; angle offset for zero-point correction
angle = OtrAngleAdd(inc_encoder_01, -0.1)
; resampling
res = OtrResampleAngle(Channel_001, angle, 360.0, 360, "°", 1)
```

Example 2

In the following example, the result is displayed over the number of revolutions, i.e. a complete revolution gets a value of 1, and the unit is set to "Rev". The result is resolved with 100 points per revolution. The resulting record is non-segmented, so it counts the number of revolutions along the measurement.

```
res = OtrResampleAngle(Channel_001, inc_encoder_01, 1, 100, "Rev", 0)
```

OtrResampleFromRpm

Angle sampling: A vibration signal is referenced to the rotation angle. The RPM signal must be given.

AngleSignal = OtrResampleFromRpm(Vibration, RPM, OrderMax, RPM_Max)

AngleSignal: Result, the vibration signal plotted over rotation angle

Vibration: Time history of the vibration signal

RPM: Time history of the RPMs. Scaled in rotations/minute.

OrderMax: Maximum order (line) contained in result

RPM_Max: Maximum desired RPM-value in revs/min

The resulting signal trace over the angle is scaled to the x-coordinate, reflecting the number of revolutions made. The x-coordinate starts at 0, has the value of 0.5 after half a revolution, after a full revolution 1.0, after 2 whole revolutions 2.0 etc. The data rate for the result: 0.5 / (OrderRef * Oversampling). The absolute value for the RPMs is used.

The function doesn't come with anti-aliasing filtering. `OtrTrackingLowPass` should be called beforehand. Linear interpolation is used to determine intermediate values.

The function works properly for:

OrderMax ≤ 30 / (SampleTime_Vibration * RPM_Max)

where SampleTime_Vibration is the sampling interval of the signal 'Vibration'.

The RPM's should change only slowly. The RPM's should not fall far below 1% of the maximum value.

If the momentary RPM-value rises above RPM_Max during the measurement, whole revolutions are rejected. The rotation angle is computed from the RPM-values by means of integration (aggregation), therefore the RPM-values must be available and highly precise.

The signal processor works with 32 bit real numbers at a relative accuracy of 1e-7. In long-term measurements, a variable phase-shift tends to occur.

The same effect appears even more pronounced in the case of inexact RPM-values or values which cannot be integrated to a rotational angle.

Example

A vibration signal vib is sampled every 0.5 ms. The RPM-value rpm can rise to up to 3000 revs/min. The vibration is to be plotted over the angle. Signal components up to the 15th order are supposed to be included.

```
tlp = OtrTrackingLowPass( vib, rpm, 8.0, 4 )
res = OtrResampleFromRpm( tlp, rpm, 15.0, 3000.0 )
```

It is seen that: OrderMax = 15.0 ≤ 30 / (0.0005 * 3000) = 20.0

The resolution of the result is 0.5 / OrderMax = 0.0333 revolutions. There are 30 samples per revolution.

The anti-aliasing filter is configured to dampen the signal by 3dB in the 8th order, in the 14.4th order by 20 dB. At the 6.1th order the amplitude error is already less than 5%.

OtrRpmComplexOrder

Complex order line: Determines magnitude and phase of an order line referenced to the RPMs. The desired RPM range is divided into classes of equal width.

OtrRpmComplexOrder(Magnitude, Phase, Vibration, Pulse_Signal, Signal_Type, EncoderType, EncoderPulse, RPM_Min, RPM_Max, RPM_Bin_Width, OrderMed, WidthPercent, FilterOrder, Interpolation, DelayTime)

Magnitude: Result, magnitude referenced to RPMs

Phase: Result, phase referenced to RPMs

Vibration: Time-history of vibration signal

Pulse_Signal: Time-history of pulse signal

Signal_Type: Type of pulse signal

- 0: Number of events
- 1: Time of pulse
- 2: Sampled rectangle
- 3: Sine signal

EncoderType: Type of encoder, how many missing teeth?

- 0: Default
- 1: 1 tooth missing
- 2: 2 teeth missing

EncoderPulse: Pulses per revolution, ≥ 1

RPM_Min: Lower limit of desired RPM-range

RPM_Max: Upper limit of desired RPM-range

RPM_Bin_Width: Width (interval) of an RPM-class

OrderMed: The order (line) to be filtered out

WidthPercent: Width of the bandpass filter in percent, 10..100%

FilterOrder: The filter order of the bandpass filter

Interpolation: Specifies whether the result is interpolated

- 0: No interpolation
- 1: Constant interpolation (centered around reference values)
- 2: Linear interpolation

DelayTime: DelayTime, stated in seconds

The function finds whole periods for the specified order line. For each period, the magnitude, phase and mean RPM-value are determined. These value-pairs are filed in the result. If a value for the particular RPM-value was already filed, an average is taken. The function works with a signal plotted over rotation angle, to which a fixed-frequency bandpass filter is applied.

The root-mean-square value of the magnitude is determined. The phase is stated in the range -180 degrees ... +180 degrees. the value of the phase is 0 degrees for a cosine oscillation, -90 degrees for a sine oscillation. If a "0" is specified as the value for the Magnitude or Phase parameter, the respective quantity is not determined.

The RPM-range starts at RPM_Min, the resolution is always RPM_Bin_Width. The value for RPM_Max is used only to determine the number of result values. RPM_Min, RPM_Max and RPM_Bin_Width must all be scaled in revs/min.

For each RPM-class (of width RPM_Bin_Width) there should be enough periods in the vibration signal. If there are no values, the result for such a RPM-class is zero. If the interpolation parameter is not selected as zero, unfilled RPM-classes are filled by interpolating between adjacent values. If interpolation is selected, even unfilled outer bins (classes at the margins) are filled by means of extension.

The value range for the parameter WidthPercent is [10 ... 100.0]. E.g. at 30% width, the ratio of upper to lower cutoff frequency for the bandpass is 1.30 .

The parameter OrderMed is the order (line) at which the medium frequency of the bandpass is situated. The internally set medium frequency of the filter is:

Mediumfrequency = OrderMed * (Current_RPM/ 60)

The upper cutoff frequency is above the medium frequency and results for the filter width.

The function works properly for:

OrderMax << 10 / (SampleTime_Vibration * RPM_Max)

where SampleTime_Vibration is the sampling time of the signal 'Vibration'.

OrderMax = OrderMed * sqrt (1 + WidthPercent / 100)

<< means that the specifiable order should be substantially smaller.

The RPM's should change only slowly. The RPM's should not fall far below 1% of capacity.

Note that the bandpasses need some time for the transients to subside. This time increases strongly with the narrowness of the filter. A bandpass of width 10% is in this sense already extremely narrow. A width of 25% corresponds to a third-octave filter, a width of 100% to an octave filter.

During measurement, the results are transferred like a histogram, i.e., regular but infrequent transmission of intermediary results to the PC. Any interpolation procedure set will only take effect upon completion of the measurement.

Notes on signal type

0	Number of events. The pulse signal reflects the number of pulses counted in a sampling interval. The signal is received if the device's incremental counter input is set to "Event counting". The signal comprises a sequence of integers. Each number represents the number of pulses counted within the current sampling interval. If the signal consisted of the sequence of values { 0, 0, 1, 0, 1, 2, 1 }, this amounts to 5 pulses detected. This signal type comes with a certain amount of imprecision, since it doesn't reflect at what time within the sampling interval the pulses were counted.
1	Time of pulse. The signal is received if the device's incremental counter input is set to "Time of pulse". In this mode, an encoder pulse's exact position in time in the frame of the sampling interval is determined. This involves very precise capture of the pulse. Only this signal type allows such precise sampling in relation to the angle. In this mode it is also important that the measured RPM value be small enough so that there is never more than one encoder pulse per sampling interval. The sampling interval may have to be adjusted to a small enough value for this purpose. RPM_Max must then also be set to a correspondingly low value. Maximum possible RPM [revs/min] = 60 / (encoder pulses * sampling time [s])

2	Sampled rectangle. One of the device's digital inputs is used to sample the time-history of the encoder's digital output. At every transition from zero to nonzero, the encoder is assumed to have turned by one increment. If the signal consisted of the sequence of values { 0, 0, 1, 1, 1, 0, 0, 1, 1, 0 }, this amounts to 2 pulses detected. If an analog voltage is sampled and there is no hardware comparator, the analog voltage data must be converted to a digital data sequence. If the signal is noisy, it may need to be smoothed beforehand and then have a Schmitt-Trigger performed. This signal type comes with a certain amount of imprecision, since it doesn't reflect at what time within the sampling interval the edge occurred.
3	Sine signal. One of the device's digital inputs is used to sample the time-history of the encoder's (sinusoid) analog output. Sinusoid or other signals with zero-crossing in a positive slope can be processed. If the signal is noisy, it may need to be smoothed beforehand and then have a Schmitt-Trigger performed. A zero-crossing in a positive slope is interpreted as a pulse from the encoder. If the signal consisted of the sequence of values { -3.0, -1.0, +1.5, +2.8, +1.3, +0.1, -0.6 }, a pulse is detected at the transition from -1.0 to +1.5.

Note **Notes on EncoderType**

Special features of encoders with missing pulses:

The number of encoder pulses is always specified to include the missing tooth. E.g. for an encoder emitting a pulse every 10 degrees of rotation, which should then have 36 teeth, the number of encoder pulses to specify is then also 36. But the encoder only generates 35 pulses because one is missing. Also typical are encoders generating a pulse every 6 degrees. In this case, 60 teeth are to be specified although 2 are missing and only 58 actually present. The first tooth after the gap is interpreted as the zero-pulse. This marks the start of sampling. Encoders with missing pulses can only be used with the signal type "Time of pulse". Recognizing the missing teeth is only possible if the RPM-value is relatively constant in the region around the gap. Particularly with extremely low RPM-values, this cannot be guaranteed. Since in that case, the gap cannot be clearly recognized, the minimum RPM must be set to a non-zero value. At higher RPM-values, the gap is usually clearly recognizable due to mechanical inertia. The function tries to re-synchronize itself after any error in the pulse sequence (or a presumably incorrectly interpreted pulse sequence). However, incorrect sampled values may occasionally occur.

Note **Notes on DelayTime**

DelayTime = 0.0 Default ≥ 0.0

A delay refers to when the vibration signal lags behind the RPM-signal because of analog circuitry such as filters and amplifiers. A constant lag in the time signal results in a RPM-dependent delay in the angle signal. This can be compensated. The delay time is assumed to be constant. This isn't true exactly, but as a good approximation. The time lag is not a property of the device, but rather of the measurement chain and must be determined empirically by a control measurement.

Example

The magnitude and phase of the 1.5th order line, plotted over the RPMs, are to be determined from this order line's time-history.

Given: vibration signal vib sampled at a rate of 0.0005 ms and the pulse signal Inc01. The device's incremental counter input is set on "Time-of-pulse" mode. The encoder has 8 markings along its circumference.

```
OLine = OtrRpmOrder( vib, Inc01, 1, 0, 8, 1000, 6000, 100, 1.5, 30, 6, 0, 0 )
; _STyp = 1       ; Time-of-pulse
; _ETyp = 0
; _EPulse = 8     ; number of markings
; _Min = 1000.0   ; minimum of RPM-range
; _Max = 6000.0   ; maximum of RPM-range
; _Delta = 100.0  ; width of individual RPM-classes
; _om = 1.5       ; the 1.5th order is selected
; _width = 30     ; 30% of total width
; _fo = 6         ; a 6th-order bandpass filter is used
; _Ipl = 0        ; 0 default (no interpolation)
; _Delay = 0      ; no delay

; OLine = OtrRpmOrder( vib, Inc01, _STyp, _ETyp, _EPulse, _Min, _Max, _Delta, _om,
_width, _fo, _Ipl, _Delay )
```

Calculation of maximum order: OrderMax = 1.5 * sqrt(1 + 30 / 100) = 1.7

and thus: OrderMax = 1.7 << 10 / (0.0005 * 6000) = 3.33

OtrRpmOrder

OrderLine: Determines the effective (r.m.s.) value of an order line in relation to the RPMs. The desired RPM range is subdivided in classes of equal width.

RpmOrderLine = OtrRpmOrder(Vibration, RPM, RPM_Min, RPM_Max, RPM_ClassWidth, OrderMiddle, WidthPercent, FilterOrder, Interpolation)

RpmOrderLine: Representation in relationship to RPMs

Vibration: Time history of the vibration signal

RPM: Time history of the RPMs. Scaled in rotations/minute

RPM_Min: Lower end of desired RPM-range

RPM_Max: Upper end of desired RPM-range

RPM_ClassWidth: Bin spacing (class width) of RPM classes

OrderMiddle: The order (line) to be isolated

WidthPercent: Bandpass filter bandwidth in percent, 10% ... 100%

FilterOrder: The filter order of the bandpass filter

Interpolation: Specifies whether the results are interpolated?

0: No interpolation

1: Constant interpolation (centered around auxiliary values)

2: Linear interpolation

The RPM-range always begins with RPM_Min, the resolution is always RPM_ClassWidth. The parameter RPM_Max is only used to specify the number of values in the result. RPM_Min, RPM_Max and RPM_ClassWidth must also be scaled in RPMs.

There should be an adequate number of measurement values in the vibration signal for each RPM-class (of bin spacing RPMClassWidth). Where no values are present, the result for the RPM-class is zero. Only if a nonzero value is selected for the interpolation parameter, the unfilled RPM-classes are filled by means of interpolation of adjacent values. If a form of interpolation is selected, even unfilled classes at the margins are filled by extending the curve at a constant level.

The value range for the parameter WidthPercent is [10 ... 100.0]. For instance, at 30% width, the relationship of upper to lower cutoff frequency of the bandpass is 1.30.

The parameter OrderMiddle is the order (line) in which the medium frequency of the bandpass is located. The internally selected medium frequency of the filter is:

MediumFrequency = OrderMiddle * (Momentary_RPM / 60)

The upper cutoff frequency is located above the medium frequency and is derived from the filter bandwidth.

The function works properly for:

OrderMax << 10 / (SampleTime_Vibration * RPM_Max)

where SampleTime_Vibration is the sampling interval of the signal 'Vibration'.

OrderMax = OrderCenter * sqrt (1 + WidthPercent / 100)

<< which indicates that the order to be specified is substantially smaller.

The RPM's should change only slowly. The RPM's should not fall far below 1% of capacity.

Note that bandpasses need a certain amount of time for transients to subside. This time increases markedly as the filter bandpass bandwidth becomes more narrow. In this context, a bandpass bandwidth of 1% is extremely narrow. A width of 25% corresponds to a third-octave filter, a width of 100% to an octave filter.

During measurement, the results are transferred like a histogram, i.e., regular but infrequent transfer of intermediary results to the PC. Any interpolation specified takes effect only after completion of the measurement.

The function works on a signal which is sampled at intervals of rotational angle and to which a fixed frequency bandpass filter is applied, and from which an r-m-s value is then derived for each RPM-class.

Example

The effective (r.m.s.) value of the 1.5th order line in relationship to the RPMs. The input data are the vibration signal vib with the sampling interval of 0.0005 ms and the RPM signal rpm.

```
_RPM_Min = 1000.0     ; RPM-range minimum
_RPM_Max = 6000.0     ; RPM-range maximum
_RPM_Delta = 100.0    ; width of individual RPM-classes
_om = 1.5             ; the 1.5th order is selected
_width = 30           ; 30% of total width
_fo = 6               ; a 6th order bandpass filter is used
_Interpolation = 0    ; 0 default no interpolation)
OLine = OtrRpmOrder( vib, rpm, _RPM_Min, _RPM_Max, _RPM_Delta, _om, _width, _fo,
_Interpolation )
```

Calculation of maximum order: OrderMax = 1.5 * sqrt(1 + 30 / 100) = 1.7

and thus: OrderMax = 1.7 << 10 / (0.0005 * 6000) = 3.33

OtrRpmPresentation

RPM-based signal representation: The time histories of a vibration signal and of the RPM are evaluated to determine the vibration behavior in reference to the RPMs. The desired RPM-range is subdivided into classes of equal width.

RPMRepresentation = OtrRpmPresentation(Vibration, RPM, RPM_Min, RPM_Max, RPM_ClassWidth, Calculation, Interpolation)

RPMRepresentation: Signal recorded in reference to momentary RPM-value

Vibration: Time history of the vibration signal

RPM: Time history of the RPMs. Scaled in rotations/minute.

RPM_Min: Lower end of desired RPM-range

RPM_Max: Upper end of desired RPM-range

RPM_ClassWidth: Bin spacing (class width) of RPM classes

Calculation: How are the values of an RPM-class represented mathematically?

0: Root-mean square (default)

1: Arithmetic mean

2: Minimum

3: Maximum

Interpolation: Is the result interpolated?

0: No interpolation

1: Constant interpolation

2: Linear interpolation

Vibration values corresponding to RPM-values outside of the desired range are ignored (closed outer bins).

The RPM-range always begins with RPM_Min, the resolution is always RPM_ClassWidth.

The parameter RPM_Max is only used to specify the number of values in the result. The results are similar to a histogram in nature, and it is often appropriate to represent them in the form of bar or step graphs.

"Vibration" and "RPM" can either both be in the time domain or in the domain of rotational angle. The RPMs need not be scaled in revs/min. But the RPM-signal, and the parameters RPM_Min, RPM_Max and RPM_ClassWidth must all be scaled to a common y-unit.

There should be an adequate number of measurement values in the vibration signal for each RPM-class (of bin spacing RPMClassWidth). Where no values are present, the result for the RPM-class is zero. Only if a nonzero value is selected for the interpolation parameter, the unfilled RPM-classes are filled by means of interpolation of adjacent values.

If a form of interpolation is selected, even unfilled classes at the margins are filled by extending the curve at a constant level.

During measurement, the results are transferred like a histogram, i.e., regular but infrequent transfer of intermediary results to the PC.

Any interpolation specified takes effect only after completion of the measurement.

Example

The time history of a vibration is to be used to derive a representation of the rms-value of the vibration in reference to the RPMs. Given: vibration signal vib and RPM-signal rpm.

```
rms_rpm = OtrRpmPresentation( vib, rpm, 1000, 6000, 100, 0, 0 )
```

OtrRpmPresentVector

RPM-representation of a spectrum: The time-history of a spectrum and corresponding RPM-signal is determined. The desired RPM-range is divided into bins of equal width.

RPMRepresentation = OtrRpmPresentVector(Spectrum_Sequence, RPM, RPM_Min, RPM_Max, RPM_Bin_Width, Processing)

RPMRepresentation: Representation of the spectrum in reference to the RPMs **Spectrum_Sequence**: Time-history of a spectrum **RPM**: Time-history of RPM signal **RPM_Min**: Lower limit of desired RPM-range **RPM_Max**: Upper limit of desired RPM-range	**RPM_Bin_Width**: Width (interval) of an RPM-class **Processing**: How are the values from a single RPM-class processed? 0: Arithmetic mean 1: Maximum 2: Minimum

This function takes a sequence of spectra as its input data. This is the spectrum's time-history, determined, for instance with the function `FFT`. The function also requires the corresponding set of RPM-values. This is the RPM's time-history. There is, then, a corresponding spectrum for each RPM value. Each spectrum-RPM pair is written to the result-matrix. The result-matrix contains a spectrum for each RPM range.

The RPM range always begins at RPM_Min, the resolution is RPM_Bin_Width. The specification for RPM_Max determines the number of values for the result.

If an RPM value is outside of the specified range, the corresponding spectrum is disregarded. (closed outer bins).

There should be enough spectra for each RPM-class (of width RPM_Bin_Width). If there is no spectrum, the result in this RPM-class is zero. During the measurement, the results are transmitted like a histogram, i.e. regular but infrequent transfer of intermediate results to the PC.

Example

Given: a vibration signal channel "Vibration" and a RPM-signal channel "Revolutions". Both have the sampling interval 1ms. The spectrum is computed on the basis of the vibrations. This spectrum is to be represented as a function of the RPMs.

```
Spectra     = fft( Vibration, 0, 1024 )
_Revs       = mean( Revolutions, 1024, 1024 )
Spectrum_N = OtrRpmPresentVector( Spectra, _Revs, 1000, 6000, 100, 0 )
;_Min = 1000 ; Minimum of RPM-range
;_Max = 6000 ; Maximum dof RPM-range
;_Delta = 100  ; bin-width of RPm-classes
;_Calc = 0 ; 0 = Arithmetic mean
; Spectrum_N = OtrRpmPresentVector( Spectra, _Revs, _Min, _Max, _Delta, _Calc )
```

Every 1024 values of the vibration channel, a spectrum is determined. In other words, after 1024 ms a corresponding value for the RPMs is needed. The channel Revs is averaged so as to return one RPM-value per spectrum.

OtrRpmSpectrum

Spectrum: The FFT-spectrum (rms-values!), determined from the time histories of Vibration and RPM in relationship to the RPM's. The desired RPM range is subdivided in classes of equal width.

RPMSpectrum = OtrRpmSpectrum(Vibration, RPM, RPM_Min, RPM_Max, RPM_ClassWidth, WindowWidth, WindowType, AveragingType)

RPMSpectrum: Spectrum in relation to RPM

Vibration: Time history of the vibration signal

RPM: Time history of the RPMs. Scaled in rotations/minute

RPM_Min: Lower end of desired RPM-range

RPM_Max: Upper end of desired RPM-range

RPM_ClassWidth: Bin spacing (class width) of RPM classes

WindowWidth: Width of the time-window in data points, 100 ... 8192

WindowType: Window function for the FFT used

0: Rectangular

1: Hamming

2: Hanning

3: Blackman

4: Blackman / Harris

5: Flat Top

AveragingType: How are the multiple spectrum computations of each RPM-class averaged?

0: Arithmetic mean

1: Maximum

2: Minimum

RPM_Min, RPM_Max and RPM_ClassWidth are scaled to revs/minute just as RPM. The window width need not be a power of 2. Window widths of, say, 500 or 1000 samples also produce "pretty" frequency line distances. The spectral lines are given as rms-values.

The averaging works on the magnitude spectrum. The mean RPM value during a spectrum determines the RPM-class. Therefore, the RPMs should change only slowly.

If no spectrum is computed in a certain RPM class, this spectrum is filled with zeroes. During measurement, the results are transferred like a histogram, i.e., regular but infrequent transmission of intermediary results to the PC. The result is a segmented waveform. Each segment is a spectrum.

Example

From the time-history of a vibration vib and of the RPMs rpm, the spectrum is to be determined in relation to the RPMs.

```
_RPM_Min = 1000.0    ; minimum of RPM range
_RPM_Max = 6000.0    ; maximum of RPM range
_RPM_Delta = 100.0   ; bin spacing of RPM classes
_WindowWidth = 1000  ; width of window for the FFT, in number of measurement values
_WindowType = 3      ; 0 Rectangular, 3 Blackman
_AveragingType = 0   ; 0 (arithmet. mean)
FFTSpectrum = OtrRpmSpectrum( vib, rpm, _RPM_Min, _RPM_Max, _RPM_Delta, _WindowWidth,
_WindowType, _AveragingType )
```

A spectrum is computed which has a frequency line spacing of 2 Hz, a sampling interval of 0.5ms and a window width of 1000 points.

OtrSynthSin

Internal function for imc Online FRAME.

OtrTrackingLowPass

Smoothing: Tracking low-pass filter. A vibration signal is low-pass filtered, and the filter cutoff frequency depends on the RPM-value.

Filtered = OtrTrackingLowPass(Vibration, RPM, Order3dB, FilterOrder)

Filtered: Low-pass filtered signal	**Order3dB**: Order at which the low-pas filter dampens the signal by 3dB
Vibration: Time history of vibration signal	**FilterOrder**: The low-pass filter order
RPM: Time history of RPMs. Scaled in revs/min	

The absolute value for the RPMs is used. The internally set cutoff frequency of the filter is CutoffFrequency = Order3dB * (Current_RPM / 60)

The cutoff frequency must always be substantially less than half the sampling rate of the vibration signal in order for the filtering effect to be achieved.

The function works properly for:

Order3dB << 24 / (SampleTime_Vibration * max (RPM))

where SampleTime_Vibration is the sampling interval of the signal 'Vibration' and max (RPM) is the maximum RPM-value occurring.

<< means that the specifiable order should be substantially smaller.

Conversely, the maximum RPM-value may not be too high. The RPM's should change only slowly. The RPM's should not fall far below 1% of the maximum value.

The upper cutoff frequency of the low-pass filter must always be substantially less than half the sampling rate of the vibration signal. Above about 0.4 times the sampling rate, no filtering can be performed.

Example

A vibration signal vib is is sampled every 0.2 ms. The RPM-signal rpm can get up to 6000 revs/min. Components above the 10th order are to be suppressed.

```
tlp = OtrTrackingLowPass( vib, rpm, 10.0, 6 )
```

A 6th order low-pass filter is used. It dampens by 3dB at the 10th order. It is seen that:

OrderMax = 10.0 << 24 / (0.0002 * 6000) = 20.0

The low-pass filter is configured to dampen the signal by 3dB in the 10th order, in the 22nd order by 40 dB. Below th 8.3th order the amplitude error is already less than 5%.

11.2.9.2.14 P

Poll

POLL-Operator: A single value is generated from a DIO-bit in the acquisition mode. This single vlaue can be used like a DIO-bit in the input mode.

DIOBitInput = POLL DIOBitAcquisition

The generated single value can also be subjected to computation operations without the trigger having been released. With control commands, for instance, the single value can be subjected to calculations in the segment `OnTimer`.

Example

```
LED_01 = LogNot( POLL DIO_Bit01 )
```

If DIO_Bit01 = 0, the first LED in the device lights up although no triggers were released.

Power1

Computes a single-phase power measurement

Power1(InstPower, P, S, Q, PowerFactor, uRMS, iRMS, Time, u, i)

InstPower: Name of the virtual channel with instantaneous power

P - TruePower: Name of the virtual channel with true power data

S - AppPower: Name of the virtual channel with apparent power data

Q - ReacPower: Name of the virtual channel with reactive power data

PowerFactor: Name of the virtual channel with power factor data

uRMS: Channel with RMS voltage data

iRMS: Channel with RMS current data

Time: Variable containing averaging time in seconds (single value)

u: Channel with voltage data

i: Channel with current data

If a channel (e.g. reactive power in example above) is not needed, then enter 0 for its parameter.

Example

```
Power1( IP, P, S, 0, PF, Urms, Irms, 5, Voltage, Current )
```

Reference

A general description is found in the documentation on imc Online FAMOS: "*Information and Tips*" > "*Power Measurement* 884".

Power2

Computes a two-phase power measurement

Power2(InstPower, P, S, Q, PowerFactor, uRMS1, iRMS1, uRMS2, iRMS2, Time, u1, i1, u2, i2)

InstPower: Name of the virtual channel with instantaneous power

P - TruePower: Name of the virtual channel with true power data

S - AppPower: Name of the virtual channel with apparent power data

Q - ReacPower: Name of the virtual channel with reactive power data

PowerFactor: Name of the virtual channel with power factor data

uRMS1: Channel with 1st phase RMS voltage data (u1)

iRMS1: Channel with 1st phase RMS current data (i1)

uRMS2: Channel with 2nd phase RMS voltage data (u2)

iRMS2: Channel with 2nd phase RMS current data (i2)

Time: Variable containing averaging time in seconds (single value)

u1: Channel with 1st phase voltage data

i1: Channel with 1st phase current data

u2: Channel with 2nd phase voltage data

i2: Channel with 2nd phase current data

If a channel (e.g. reactive power in example above) is not needed, then enter 0 for its parameter.

Example

```
Power2( IP, P, S, 0, PF, Urms1, Irms1, Urms2, Irms2, 5, Voltage1, Current1, Voltage2,
Current2 )
```

Reference

A general description is found in the documentation on imc Online FAMOS: "*Information and Tips*" > "*Power Measurement* 884".

Power3

Computes a three-phase power measurement

Power3(InstPower, P, S, Q, PowerFactor, uRMS1, iRMS1, uRMS2, iRMS2, uRMS3, iRMS3, Time, u1, i1, u2, i2, u3, i3)

InstPower: Name of the virtual channel with instantaneous power

P - TruePower: Name of the virtual channel with true power data

S - AppPower: Name of the virtual channel with apparent power data

Q - ReacPower: Name of the virtual channel with reactive power data

PowerFactor: Name of the virtual channel with power factor data

uRMS1: Channel with 1st phase RMS voltage data (u1)

iRMS1: Channel with 1st phase RMS current data (i1)

uRMS2: Channel with 2nd phase RMS voltage data (u2)

iRMS2: Channel with 2nd phase RMS current data (i2)

uRMS3: Channel with 3nd phase RMS voltage data (u3)

iRMS3: Channel with 3nd phase RMS current data (i3)

Time: Variable containing averaging time in seconds (single value)

u1: Channel with 1st phase voltage data

i1: Channel with 1st phase current data

u2: Channel with 2nd phase voltage data

i2: Channel with 2nd phase current data

u1: Channel with 3nd phase voltage data

i1: Channel with 3nd phase current data

If a channel (e.g. reactive power in example above) is not needed, then enter 0 for its parameter.

Example

```
Power3( IP, P, S, 0, PF, Urms1, Irms1, Urms2, Irms2, Urms3, Irms3, 5, Voltage1, Current1,
Voltage2, Current2, Voltage3, Current3)
```

Reference

A general description is found in the documentation on imc Online FAMOS: "*Information and Tips*" > "*Power Measurement* 884".

PulseDuration

A signal's average pulse duration within a window, with resampling.

a = PulseDuration(b, RF)

a: Result

b: Input channel

RF: Reduction factor; in samples; ≥ 1

The function works in a window of RF values and generates a result value after every RF values.

The ratio is the number of samples not equal to zero divided by the number of samples equal to zero, for all complete pulses. A complete pulse is a sequence of three edges, an edge being a transition from zero to a non-zero value or vice versa.

Analog signals should be preprocessed (STri 1017).

If no pulse is completed within a window, the last pulse duration is returned or, if the window size multiplied by the number of pulses windows is greater than the last pulse frequency, the product of window size and the number of windows.

PulseFrequency

A signal's average pulse duration within a window, with reduction.

a = PulseFrequency(b, RF)

a: Result	**RF**: Reduction factor; in samples; ≥ 1
b: Input channel	

The function works in a window of RF values and generates a result value after every RF values.

The average pulse frequency is calculated on the basis of all complete pulses in the window. A 'complete impulse' is defined as a sequence of three edges, where 'edge' denotes a transition from zero to non-zero or vice-versa.

Analog signals should be preprocessed (STri 1017).

If no pulse is completed within a window, the last pulse frequency is returned or, if the sampling rate of the result divided by the number of pulses windows is lesser than the last pulse frequency, the quotient of sampling rate and number of windows.

PulsePhase

The average phase difference between two signals in a window, with resampling.

a = PulsePhase(b, c, RF)

a: Result	**c**: Input channel 2
b: Input channel 1	**RF**: Reduction factor; in samples; ≥ 1

The function works in a window of RF values and generates a result value after every RF values.

The phase difference is calculated on the basis of two corresponding positive slopes, and the difference is stated in seconds. A positive slope denotes a transition from zero to a non-zero value.

Analog signals should be preprocessed (STri 1017).

11.2.9.2.15 R

RangeMax

Returns the input channel's **upper measurement range limit**.

a = RangeMax(b)

a: Upper limit of input channel measurement range (single value)	**b**: input channel

Note Balancing

After channel balancing, this function only returns the **correct value**, if "**Prepare**" **was performed**.
Background: Balancing can be performed either during measurement or before a measurement. Afterwards, "Prepare" is not always necessary. In these cases, the measurement range is not determined correctly. Instead, the same measurement rage which had been displayed before the balancing continues to be displayed.

RangeMin

Returns the input channel's **lower measurement range limit**.

a = RangeMin(b)

a: Lower limit of input channel measurement range (single value)	**b**: input channel

Note Balancing

After channel balancing, this function only returns the **correct value**, if "**Prepare**" **was performed**.
Background: Balancing can be performed either during measurement or before a measurement. Afterwards, "Prepare" is not always necessary. In these cases, the measurement range is not determined correctly. Instead, the same measurement rage which had been displayed before the balancing continues to be displayed.

ReadyForPowerOff

Device ready for deactivation: The device can be turned off now.

Prerequisite:
Only available in: imc Online FAMOS!
Only usable with control commands!
The function `ReadyForPowerOff` may only be called in the control command `OnPowerOff`.

ReadyForPowerOff( )

This function is called in the section `OnPowerOff` which is activated when the device is turned off. The device's internal battery then provides the power for a maximum of 8 s. Within this time, the program can be closed correctly in the section `OnPowerOff`. When the function `ReadyForPowerOff` is called, the device is ready to be turned off. Before the device is really turned off, it may be necessary to conclude other processes, e.g. saving to the hard drive.

Example

```
OnSyncTask( 0.01 ) ; cycle time of 0.01s
    DisplayVar_01 = pv.Channel_001
End

OnPowerOff( 0.01 ) ; cycle time of 0.01s
    DisplayVar_01 = 0
    DOut01_Bit01  = 0
    DOut01_Bit02  = 0
    ReadyForPowerOff()
End
```

RecordEvent

Record event: Records a transition from zero to not zero as an event

Prerequisite:
There must be at least one channel assigned to the "*BaseTrigger*". Texts can be recorded only while channels assigned to the "*BaseTrigger*" are measuring.

RecordEvent(b, "c")

b: Event channel	**c**: Event text

As a result a TimeStampAscii channel (OfaEvents) will be created. The event messages can be displayed with a curve window as table or graphical over time.

For each device/imc Inline FAMOS-task, only **one log channel is created**. The function can be called multiple times, and all entries it makes are deposited in this channel.

Example

```
RecordEvent( Greater( Channel_001, 9 ), "High water mark passed" )
```

If the level on Channel_001 exceeds 9, this is recorded as an event.

Note

Modifying the name

Tip: Assign an appropriately distinct name to the channel. The log channel has a default name which cannot be changed by means of the function (e.g. "*OFA_Events*"). To do this, open the OFA/IFA-Editor and there, open "*Properties*" (F5). Here, you can change the log-channel's name.

RecordText

Return text: The specified text is output.

Prerequisite:

For imc Online FAMOS, observe the following: There must be at least one channel assigned to the "*BaseTrigger*". Texts can be recorded only while channels assigned to the "*BaseTrigger*" are measuring.

RecordText ("Text")

Text: Text to output

This function should only be used as a message following certain events, for example when a certain value in the control command Switch is reached. If the function is called repeatedly, the output memory for texts can quickly overflow.

Example

```
OnInitAll
    Value = 1
    StartTimerPeriodic( 5, 0.1, 0.1 )
End

OnTimer( 5 )
    Switch Value
        Case 1
            RecordText( "Value = 1" )
            Value = 2
        End
        Case 2
            RecordText( "Value = 2" )
            Value = 3
        End
        Default
            Value = Value + 1
        End
    End
End
```

In contrast to the `RecordEvent`-function, the `RecordText` function always outputs the text (in the `RecordEvent` function, a transition of the channel parameter from 0 to non-zero is necessary for output of text).

For each device/imc Inline FAMOS-task, only **one log channel is created**. The function can be called multiple times, and all entries it makes are deposited in this channel.

Note — Modifying the name

Tip: Assign an appropriately distinct name to the channel. The log channel has a default name which cannot be changed by means of the function (e.g. "*OFA_Events*"). To do this, open the OFA/IFA-Editor and there, open "*Properties*" (F5). Here, you can change the log-channel's name.

Red

Reduction: reduces the number of values in the input channel

a = Red(b, RF)

a: Result	**RF**: Reduction factor in samples; ≥ 1
b: Input channel	

The function reduces the amount of values by the factor RF. RF must be an integer. To avoid aliasing effects, the input channel should be previously low-pass filtered.

Example

```
Red = Red( FiltLP( Channel_001, 0, 0, 4, 125 ), 4 )
```

A channel sampled at 1 kHz is reduced by a factor of 4. It is previously filtered with a low-pass which has half the resulting sampling frequency, in other words 125 Hz, as its cutoff frequency.

ReduceDataRate

Reducing the data rate: A channel's data rate is reduced to the specified data rate.

Prerequisite:
Only available in: imc Online FAMOS!

a = ReduceDataRate(b, Datarate)

a: Result	**b:** Channel with measured values
	Datarate: Data rate in Hz, ≥ 2

This function is applied to Fieldbus channels which were sampled at a slightly higher data rate than necessary. Additionally, the channel contains some duplicate measured values. This function eliminates these duplicate measured values as far as possible while still achieving the target data rate.

Both the input channel's data rate and the target data rate must be integer numbers. The ratio of the input channel's data rate to the target data rate may not exceed a factor of 2.5.

Example

```
res = ReduceDataRate( Signal_500Hz, 300)
; Reduces the data rate of a 500Hz-channel Signal_500Hz to 300 Hz.
```

ReplaceFirstValues0

Replaces signal values with the value 0.0: During the phase of signal transients due to filtering at the measurement's start, the initial signal values are replaced with 0.0.

Prerequisite:
Only available in: imc Inline FAMOS!

a = ReplaceFirstValues0(Signal, Count)

a: Result	**Count**: Number of values to replace
Signal: Input channel	

This function is used to suppress the undesired signal fluctuations due to filtering, which occur at the measurement's start. For this purpose, as many initial values are replaced with 0.0 as specified in the parameter Count.

The signal parameter may be either a channel or a single value (e.g. a pv-variable).

In contrast to the function `SkipFirstValues`, the values during the transients' phase are replaced instead of skipped.

Example

```
Signal_Filtered = FiltHP( Signal_01, 1, 0, 4, 100 )
Signal_Corrected = ReplaceFirstValues0( Signal_Filtered, 1000 )
```

A high-pass filter is applied to the sine signal Signal_01 (the sampling rate of Signal_01 is 1 kHz).
When the measurement is started, a signal spike occurs (whose height depends on the phasing).
This signal spike would not occur once the filter's transients had subsided.
The transients are suppressed for the duration of one second.
For this purpose, the first 1000 values of the filtered signal are replaced with the value 0.0.

ReplaceFirstValuesN

Replaces signal values with n-th value: During the filter's transients phase at the measurement's start, the initial signal values are replaced with the n-th value.

Prerequisite:
Only available in: imc Inline FAMOS!

a = ReplaceFirstValuesN(Signal, Count)

a: Result	**Count**: Number of values to replace
Signal: Input channel	

This function is used to suppress the undesired signal fluctuations due to filtering, which occur at the measurement's start. At the start of the measurement, this function does not return any results. After the first n values accumulated, all previous values are retroactively replaced with the n-th value.

Only channels are permitted as the signal parameter.

In contrast to the function `SkipFirstValues`, the values during the transients' phase are replaced instead of skipped.

Example

```
Signal_Filtered = FiltHP( Signal_01, 1, 0, 4, 100 )
Signal_Corrected = ReplaceFirstValuesN( Signal_Filtered, 1000 )
```

A high-pass filter is applied to the sine signal Signal_01 (the sampling rate of Signal_01 is 1 kHz).
When the measurement is started, a signal spike occurs (whose height depends on the phasing).
This signal spike would not occur once the filter's transients had subsided.
The transients are suppressed for the duration of one second.
For this purpose, the first 1000 values of the filtered signal are replaced with the 1000th value.

ReSample

Sampling rate conversion: Re-sampling of vectorized data.

a = ReSample(b, VectorLen)

a: result	**VectorLen**: Length of result vectors
b: Vectorized input channel	

The vectors of a vectorized input channel are modified to a new vector length by means of resampling. The new vector elements are determined by interpolating from the existing vector elements. The curve shape remains unchanged as a result of the resampling

Example

```
FFT513 = FFT( Signal, 0, 1024 )
Res500 = ReSample( FFT513, 500 )
```

Vectorization of a signal with a vector length of 1024, and calculation of the FFT. Subsequent alteration of the result's vector length from 513 to 500 by means of resampling.

Restore

Restore pv.x = 0

With this attribute, the value of a pv-variable can be restored in the device automatically if the device is prepared for a measurement.

For instance, a pv-variable's last value can be restored in the device, if the measurement will be restarted, also if the device has to be prepared again.

Reference

For more information see in the documentation on imc Online FAMOS: "*Types of Variables*" > "*Process Vector Variables*" > "*Saving/Restoring pv-variables* 1141".

Example

```
OnInitAll
    Int Restore pv.x = 0   ; for Integer pv-variables
    Restore pv.y = 0.0     ; for Float pv-variables
End
```

With the syntax `Restore pv.x = 0`, the pv-variable is loaded with the last value upon each start, even if "= 0" seems to imply that the variable is set to 0.

RMS

Root-mean-square value of a window, with reduction

a = RMS(b, WS, RF)

a: Result

b: Input channel

WS: Window size in samples;

In imc Online FAMOS:

1 * RF, 2 * RF, ... ,10 * RF

In imc Inline FAMOS:

1 * RF, 2 * RF, ... ,1000 * RF

RF: Reduction factor in samples; ≥ 1

The RMS-function calculates the RMS-value for the last FG values (window) and generates one result value after every RF values.

For RF > 1, then, there is a reduction of data.

The window size must be an integer multiple (1 to 10/1000) of the reduction factor.

Example

```
eff = RMS( Channel_001, 50, 10 )
```

Returns for every tenth input value the RMS-value of the last 50 input values.

Note

If the window size is not 1*RF, then there aren't enough values available initially. In that case, the available values are taken as the calculation basis.

Rosette1

Rosette calculation: Analysis of strain measurements with rosettes, Eps1 ≥ Eps2.

Rosette1(Eps1, Eps2, Sig1, Sig2, SigV, Phi, Grid_A, Grid_B, Grid_C, Type, Poisson, EModul, AngleR)

Eps1: Result Principal strain ε_1 [µm/m]

Eps2: Result Principal strain ε_2 [µm/m]

Sig1: Result Principal stress σ_1 [N/mm^2]

Sig2: Result Principal stress σ_2 [N/mm^2]

SigV: Result reference stress per Mises, σ_υ [N/mm^2]

Phi: Result Grid A angle $\phi_{P,Q}$ [°]

Grid_A, _B, _C: Channel for Grid A, B, C (ε_a, ε_b, ε_c) with measured strains in µm/m

Type: Rosette type

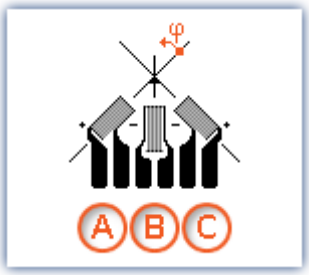
1: Rectangular rosette (0°/45°/90°)

2: Delta rosette (0°/60°/120°)

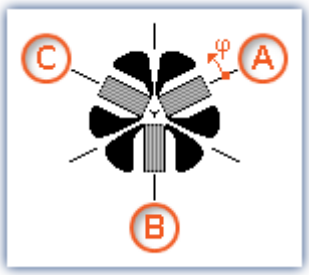
Poisson: Poisson's ratio υ

EModul: Modulus of elasticity in GPa υ

AngleR: Angle rectification below, in µm/m



Rosette type 0°/45°/90°



Rosette type 0°/60°/120°

Angle rectification below: For determining the Grid A angle, the arctan of a fraction is found. If both the numerator and denominator of this fraction are very small, small changes of the values could cause large angle changes.

If the sum of the absolute amounts of the numerator and denominator are less than or equal to the specified angle rectification, the angle is internally assigned to the value 0°.

Numbering of the measurement grid: In order to obtain correct values from measurement with 3-element rosettes, the measurement grid must be numbered in a very particular way. The numbering of the measurement grid must proceed as in the illustrations in the Formula Assistant.

Angle measurements in counterclockwise direction are assigned positive sign here, while clockwise measurements get negative sign.

The input channels for Grids A, B and C must be stated in µm/m, the angle rectification must also be stated in µm/m.

The results Eps1 and Eps2 are then also stated in µm/m, the principal stress and the reference stress according to Mises in N/mm^2 and the Grid A angle in ° (degrees).

For the principal strains the relationship Eps1 ≥ Eps2 applies, i.e. Principal strain 1 is the greater of the two principal strains.

If no virtual channel is needed, instead of the channel name, a 0 can be entered.

For parameterization of the Rosette1-function the formula-assistant is recommended.

Example

```
Rosette1( Eps1, Eps2, Sig1, Sig2, SigV, Phi, Grid_A, Grid_B, Grid_C, 1, 0.3, 210, 5.0 )
```

Formulas used	
Principle strain at 45°	$\varepsilon_{1,2} = \frac{\varepsilon_a + \varepsilon_c}{2} \pm \frac{1}{\sqrt{2}} \sqrt{(\varepsilon_a - \varepsilon_b)^2 + (\varepsilon_c - \varepsilon_b)^2}$
Principle strain at 60°	$\varepsilon_{1,2} = \frac{\varepsilon_a + \varepsilon_b + \varepsilon_c}{3} \pm \sqrt{(\frac{2\varepsilon_a - \varepsilon_b - \varepsilon_c}{3})^2 + \frac{1}{3}(\varepsilon_b - \varepsilon_c)^2}$
Principle stress 1	$\sigma_1 = \frac{E}{1 - v^2}(\varepsilon_1 + v\varepsilon_2)$
Principle stress 2	$\sigma_2 = \frac{E}{1 - v^2}(\varepsilon_2 + v\varepsilon_1)$
Reference stress as per Mises	${\sigma_v}^2 = {\sigma_1}^2 + {\sigma_2}^2 - \sigma_1\sigma_2$
Angle grid at 45°	$\phi_{P,Q} = \frac{1}{2}\arctan(\frac{2\varepsilon_b - \varepsilon_a - \varepsilon_c}{\varepsilon_a - \varepsilon_c})$
Angle grid at 60°	$\phi_{P,Q} = \frac{1}{2}\arctan(\frac{\sqrt{3}(\varepsilon_b - \varepsilon_c)}{2\varepsilon_a - \varepsilon_b - \varepsilon_c})$
Angle rectification	(Magnitude(Numerator)+Magnitude(Nominator)) > Angle rectification, the compute; else 0.

Rosette2

Rosette calculation: Analysis of strain measurements with rosettes, |Eps1| ≥ |Eps2|.

Description: see Rosette1 1005

Difference: For the principal strains the relationship **|Eps1| ≥ |Eps2|** applies, i.e. the **absolute value** of Principal strain 1 is the greater of the two principal strains.

Round

Rounds the argument to the nearest integer

a = Round(b)

This function returns the **nearest integer** for a real number input.

Example

The function returns

Input channel value	Result value
1,2	1
1,5	2
1,8	2
-1,2	-1
-1,5	-1
-1,8	-2

RSFlipflop

RS flip-flop function

a = RSFlipFlop(R, S)

a: Result	**R**: Reset
	S: Set

Returns a 1 for HIGH (H) state or 0 for LOW (L) state.

A state change from L to H occurs if S is not equal to zero and R is equal to zero.

Likewise, a state change from H to L occurs if S is equal to zero and R is not equal to zero.

Otherwise, the state does not change.

The function may also be called with a single value (e.g. virtual bit or Display variable) as its first parameter. In order to obtain useful results in such a case, calls of the function should occur at constant time intervals, e.g. in a Timer or a synchronous task.

Example

An LED lights up when the signal exits the range from 0 to 8, and goes off again once the signal is in the range from 0.5 to 6.

```
_R= LogAnd( Less( Channel_001, 6 ), Greater( Channel_001, 0.5 ))
_S=  LogOr( Greater( Channel_001, 8), Less( Channel_001, 0))
LED_01 = RSFlipFlop( _R, _S)
```

RunAutoBalance

Function for calling the automatic balancing for the selected channels.

RunAutoBalance()

The function does not require any parameters.

The channels to be balanced are selected before calling the function.

Activate the property "*Balance at device startup*" (Setup page: "*Channel balance*" or "*Analog channels*") for the desired channels. If you wish, you can also display the parameter of the same name as an additional column by using the "*Column chooser* 248".

Parameter	**Description**		
Category: Channel	*long name*	*short name*	*column ID*
Balance at device startup	*Balance at device startup*	*Balance at startup*	*eBalanceAtDeviceStart*
	This parameter has two purposes: • Before a Diskstart/Autostart measurement starts, **balancing** of the selected channels is performed. • The channels selected can be balanced using the **imc Online FAMOS function** `RunAutoBalance` 1008. This balancing operation starts automatically before every Diskstart-measurement. The type of balancing performed is set on the following page: Setup page: "*Channel balance*" > "*Balance*". Available options: Taring or bridge balancing (depending on the hardware).		

`RunAutoBalance` should not be called at each run of the program. One way to ensure this is by means of a virtual bit.

Example

```
; Executed during a running measurement
OnTriggerMeasure( BaseTrigger )
If Virt_Bit01 = 1
   RunAutoBalance()
   Virt_Bit01 = 0
   End
End
```

RunAutoShuntCalibration

Function for calling the automatic shunt calibration of the selected channels

RunAutoShuntCalibration()

11.2.9.2.16 S

SamplesGate

Gate for values: The input channel's values are only transferred, if the current value of the control channel is equal to the reference value.

a = SamplesGate(b, c, d)

a: result	**c**: control channel
b: input channel	**d**: reference value

The function sometimes yields output values at irregular intervals. Therefore, the result is generally difficult to combine with other channels in calculations.

Sawtooth

Converts input channel to a saw tooth waveform.

a = SawTooth(b, y0, dy, P)

a: Result	**dy**: Increment
b: Input channel	**P**: Cycle count in samples
y0: Start value	

An input is used to set the sampling rate of the sawtooth.

SendMessage

A message (e.g. "Message_001") is filled and sent with the specified channels.

Prerequisite:
Only available in: imc Online FAMOS!
This function will appear only, if a send message has been defined at the CAN-assistant!

SendMessage_[Message_001](Mode, Parameter, CAN_001, [CAN_002], [...])

Mode: Send cyclically, conditionally or always? 0: Cyclic sending 1: Conditional sending 2: Send always (only for Control commands)	**Parameter**: Depending on Mode: Pulse rate, condition or reserved If mode = 0: Pulse rate in s If mode = 1: Condition If mode = 2: Reserved, 0	**CAN_001**: 1st channel to send [**CAN_002**], [**...**]: If 2 channels, for instance, are defined in the CAN-Assistant

Operation Mode 2 can only be used with control commands.

Configuring a message to transmit is the condition for applying this function (see chapter "*Fieldbusses*" > "*CAN-Bus Interface*" > "*CAN-Bus Assistant*" > ... > "*Sending messages* 542"). In the CAN-Assistant, a formal framework of messages and channels is constructed. The function SendMessage is used to fill this framework and send it. The channels and single values in imc Online FAMOS are sent in the format set in the CAN-Assistant. The SendMessage function can be used, for instance, to assign a virtual channel created in imc Online FAMOS to the channel defined in the CAN-Assistant and send it in the data format defined.

The SendMessage function is automatically set up and listed in the Functions directory tree in response to the CAN-Assistant data. The actual function name SendMessage is extended with a suffix (e.g. `SendMessage_Message_001`).

The first two parameters (Mode and Parameter) are contained in each SendMessage-function. The other channel parameters are dependent on the configuration in the CAN-Assistant: The channels defined in the CAN-Assistant for this message are themselves the additional parameters of the function. If 2 channels, for instance, are defined in the CAN-Assistant, the SendMessage function has 4 parameters.

In cyclical message-sending, one message containing the current channel data is sent periodically at each sending-pulse interval. With conditional sending, exactly one message is sent if the specified condition is fulfilled (the condition is fulfilled by a transition from 0 to non-zero). Conditions must be fulfilled by channels; single values don't count. Other permissible parameters (the data to be sent) are channels and single values.

Example

The following parameter description of the function SendMessage is based on a message with 2 channels.

```
_Co = Greater( Test1, 500 )
SendMessage_Message_001( 1, _Co, Channel_001, Channel_002 )
```

or

```
SendMessage_Message _001( 0, 0.1, Channel_001, Channel_002 )
```

Sin

Sinus: Sine of the input channel

a = Sin(b)

a: Result	**b**: Input channel; in radians

SingleValueChannel

A virtual channel is created from a single value.

Prerequisite:
Only usable with control commands!

VirtualChannel = SingleValueChannel (Trigger, DataRate)

VirtualChannel: Result **Trigger**: Trigger to be assigned to the channel	**DataRate**: Average ... max. number of readings in a second; allowed: 1 ... 10000

The function `SingleValueChannel` is used to declare a virtual channel. The declaration is made as an `OnInitAll` sequence. In the curve window, this virtual channel is also represented as a virtual channel, in imc Online FAMOS or imc Inline FAMOS it is regarded as single value. In `OnTriggerStart`, `OnTriggerMeasure` and `OnTriggerEnd` the single value can be redefined as desired. Every time a value is assigned to the single-value variable, it is written to the virtual channel as a sample. If a value is continuously assigned, the output memory for the virtual channel can overflow.

This virtual channel cannot be retrieved but may only appear on the left side of an assignment statement.

Example 1

Virtual channels can be created with integers or with real numbers.

```
OnInitAll
    VChanReell = SingleValueChannel( BaseTrigger, 1000 )
    int VChanInt = SingleValueChannel( BaseTrigger, 1000 )
End

OnTriggerMeasure( BaseTrigger )
    If Virt_Bit01 > 0
        VChanReell = 10.5
        Virt_Bit01 = 0
    End
    If Virt_Bit02 > 0
        VChanInt = 10
        Virt_Bit02 = 0
    End
End
```

Example 2

```
OnInitAll
    MinMax = SingleValueChannel( BaseTrigger, 1000 )
End

OnTriggerMeasure( BaseTrigger )
    If Channel_001 > 100
        MinMax = 100
    End
    If Channel_001 < 5
        MinMax = 5
    End
End
```

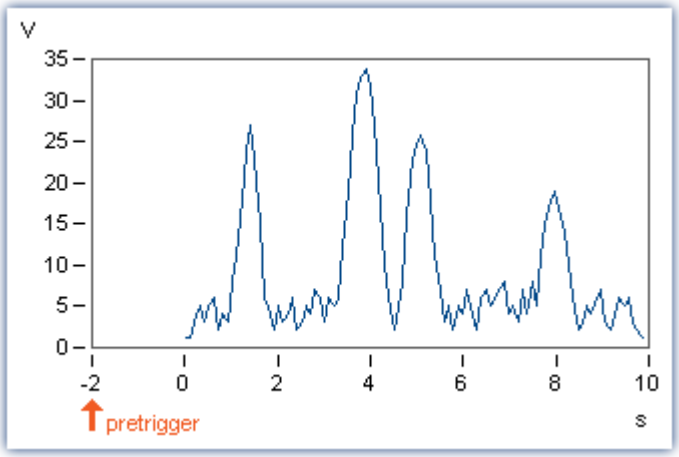
The virtual channel created is filled according to assignments of single values. Whenever an assignment is carried out, this single value is added to the virtual channel as a sample value. Thus it can happen that the virtual channel contains no samples. On the other hand, a large number of samples can accumulate in a short time. The data rate should be dimensioned correspondingly.

The unit and the clock rate (Delta-X) of this virtual channel can be set in the Properties dialog. The X-offset is fixed at 0, pretrigger time is not allowed for.

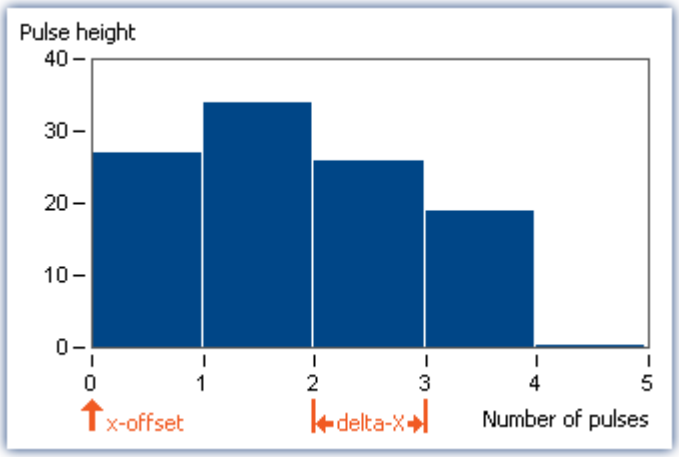
By default, the clock rate is set to 1s and no units are displayed in the curve window.

Example

As an example, voltage data (with a 2 s pretrigger) are displayed below in a curve window.



The voltage channel's big impulses are extracted using the function `SingleValueChannel` (here, there is one value representing each impulse, for a total of 4). The virtual channel "Pulse" is the result.



The x-offset is 0, delta-X in this example is 1.

SkipFirstValues

Skip signal values: During the filter's transients phase at the measurement's start, the initial signal values are skipped.

Prerequisite:
Only available in: imc Inline FAMOS!

a = SkipFirstValues(Signal, Count)

a: Result	**Count**: Number of values to skip
Signal: Input channel	

This function is used to suppress the undesired signal fluctuations due to filtering, which occur at the measurement's start. For this purpose, as many values are skipped as specified in the parameter Count.

The results of the function can only be combined with channels having the same number of values to be skipped.

In contrast to the functions `ReplaceFirstValues0` and `ReplaceFirstValuesN`, the values during the transients' phase are skipped instead of replaced.

Example

```
Signal_Filtered = FiltHP( Signal_01, 1, 0, 4, 100 )
Signal_Corrected = SkipFirstValues( Signal_Filtered, 1000 )
```

A high-pass filter is applied to the sine signal Signal_01 (the sampling rate of Signal_01 is 1 kHz).

When the measurement is started, a signal spike occurs (whose height depends on the phasing).

This signal spike would not occur once the filter's transients had subsided.

The transients are suppressed for the duration of one second.

For this purpose, the first 1000 values of the filtered signal are skipped.

SlopeClip

The maximum slope dy/dx between two adjacent values is clipped

a = SlopeClip(b, Max)

a: Result	**Max**: Maximum slope
b: Input channel	

Smo3

Three-point smoothing: The input channel is smoothed by the weighted average of the three last samples

a = Smo3(b)

a: Result	**b**: Input channel

Smo5

Five-point smoothing: The input channel is smoothed by the weighted average of the five last samples

a = Smo5(b)

a: Result	**b**: Input channel

SoundPressureLevel

Sound pressure level: The function finds the course of the sound pressure level over time.

a = SoundPressureLevel(Signal, FrequencyRating, TimeConstant, ReductionFactor)

a: Result - The signal's sound pressure level	**TimeConstant**: Time constant for averaging
Signal: signal to be evaluated	**In imc Online FAMOS:**
FrequencyRating: signal's frequency rating	≥ 0.0: Time constant is defined by the user in s
0: no rating	e.g. 0.125 for FAST-weighting, 1.0 for SLOW-weighting
1: A-rating	**In imc Inline FAMOS:**
2: B-rating	-1: Fast (0.125s)
3: C-rating	-2: Slow (1s)
	-3: Pulse
	-4: Peak
	-5: RMS in interval
	-6: RMS from start
	≥ 0.0: Time constant is defined by the user in s
	ReductionFactor: Factor for resampling; ≥ 1
	For a ReductionFactor = 1, no resampling

The function requires a sound signal expressed in the unit Pa (Pascal). The signal is first frequency rated, e.g. with an A-rating according to DIN IEC 651. Next, time-weighting (moving RMS with exponential averaging) and resampling are performed.

The result is referenced to $20*10^{-6}$ Pa and expressed in dB.

Example

```
SPL = SoundPressureLevel ( Signal, 1, 0.125, 1000 )
```

The signal is subjected to A-rating.

The frequency-weighted signal is then time-weighted with a time constant of 0.125 s (FAST) and resampled with a factor of 1000.

The signal originally has a sampling frequency of 20 kHz; the A-rated result has a sampling frequency of only 20 Hz.

The result SPL is the sound pressure level over time and expressed in dB.

SpecThirds

1/3 octave spectrum in reference to time: From the time-histories of a vibration, the 1/3 octave spectrum referenced to time is determined.

Prerequisite:
Only available in: imc Inline FAMOS!

a = SpecThirds(Vibration, f1, f2, Frequency weighting, Time rating, Output interval)

a: Result - 1/3 octave spectrum referenced to time

Vibration: Time-history of vibration signal, the time scaled in seconds

f1: Center frequency of lowest 1/3 octave in Hz

10, 12.5, 16, 20, 25, 31.5, 40, 50, 63, 80, 100, ...

f2: Center frequency of highest 1/3 octave in Hz

1000, 1250, 1600, 2000, 2500, 3150, 4000, 5000, 6300, 8000, 10000, 12500, 16000, ...

Frequency weighting: Frequency weighting for the result

0: linear

1: A-weighting

2: B-weighting

3: C-weighting

4: D-weighting

Time rating: Averaging time, time rating of filtered data

-1: Fast (0.125s)

-2: Slow (1s)

-3: Pulse

-4: Peak

-5: RMS in interval

-6: RMS from start

> 0.0: Time constant is defined by the user, given in seconds

Output interval: 1/3 octave spectra are calculated using this time increment (delta-time, dt). Integer multiple of the vibration's sampling time. Specified in seconds.

Remarks:

The two frequency limits f1 and f2 are to be given as the 1/3-octave center frequencies, e.g. f1 = 8 Hz and f2 = 12500 Hz. f1 < f2.The upper 1/3-3-octave with its frequency band must lie entirely within half of the sampling frequency.

The individual 1/3-octave values are stated as root mean square (RMS) values.

While the transients in the individual 1/3-octave (band-pass) filters are subsiding at the start of the measurement, the input signal's values are ignored.The transient time for the 1 kHz 1/3-octave is is assumed to be 20 ms.This time interval is inversely proportional to the 1/3-octave frequency.For very low 1/3-octaves, this time interval becomes considerable.A correspondingly long measurement must then be expected.

During the settling phase the RMS is assumed to be 0.0.The result is a segmented data set. Each segment contains one 1/3-octave spectrum.The x-coordinate of the result counts the 1/3-octave bands (just like the Famos-function OctA).For good representation in the curve window, 1/3-octave labeling should be selected.

The z-coordinate of the result represents time. The first spectrum is taken where time is equal to the output interval.

The 1/3-octave filter and analyses are in accordance with DIN IEC 651 (sound level measurement), DIN 45652 (1/3-octave filter for electro-acoustic measurements) and DIN EN 61260 or IEC 1260 (band-pass filter for octaves and fractions of octaves, filter Class 0).

Time rating "Pulse": For increasing amplitudes the time constant is 35 ms, for decreasing amplitudes 1.5 s.Thus impulse-shaped signals are captured quickly, the response decays slowly.

Time rating "Peak": Extreme response for very short impulses; ensuring capture of the peak value.Time constant is zero during increasing amplitude (can be performed exactly by computer, by analog operation only in approximation);during decreasing amplitude 3 s.

Example

The 1/3-octave spectrum is to be calculated from the time-history of the vibration every 0.1 s.

The signal is sampled every 0.025 ms.

```
fEval = 1 ; 0 (linear) 1 (A-weighting)
f1 = 10
f2 = 12500
tEval = -1 ; -1 (Fast)
tInterval = 0.1 ; [s]
Thirds = SpecThirds( vib, f1, f2, fEval, tEval, tInterval )
```

Sqrt

Square root of argument

a = Sqrt(b)

a: Result	**b**: Input channel

StDev

Standard deviation of a window, with reduction.

a = StDev(b, WS, RF)

a: Result	**WS**: Window size in samples;
b: Input channel	**In imc Online FAMOS:**
	1 * RF, 2 * RF, ... ,10 * RF
	In imc Inline FAMOS:
	1 * RF, 2 * RF, ... ,1000 * RF
	RF: Reduction factor in samples; ≥ 2

The standard deviation function gives the standard deviation of a window with WS samples for every RF input samples.

The window size specified must be an integer multiple of RF up to 10/1000*RF.

Example

```
erg = StDev( Channel_001, 10, 2 )
```

Returns for every second input sample the standard deviation of the last 10 input samples.

Note

If the windows size is not 1*RF, there are not enough input samples in the beginning. The result is then computed with the available input samples.

STri

Schmitt trigger with upper and lower threshold

a = STri(b, S_upper, S_lower)

a: Result	**S_upper**: Upper threshold
b: Input channel	**S_lower**: Lower threshold

If the previous result was 0, the new result is:

0, if the current value is less than S_upper
otherwise 1

else (if the previous result was 1) the new result is:

1, if the current value is greater than S_lower
otherwise 0

For computing the first value, the previous result is assumed to be 0.

Example

Computes a clean TTL signal

```
res = STri( Channel_001, 4, 1 )*5
```

Sum

Sum of input channel with reduction

a = Sum(b, RF)

a: Result	**RF**: Reduction factor in samples; ≥ 1
b: Input channel	

The summation function reduces the number of values by the factor RF and computes the moving sum of the input values.

Example

Converts the distance increments in a total distance.

```
distance = Sum( Inc_01, 1 )
```

Note — Adding small numbers to proportionally much larger numbers

Observe the limitations of the resolution of numbers in 32-bit Float format.

Sum2

Summation with reset to 0. Finds the sum of all input channel values since the last reset to 0 since the start of the measurement with reduction.

Result = Sum2(InputChannel, Reduction, Reset)

Result: Sum	**Reduction**: Reduction factor
InputChannel: Channels whose values are to be summated.	**Reset**: If unequal to 0, reset sum to 0

The function `Sum2` reduces the number of values by the factor "Reduction". In contrast to the function `Sum`, the function `Sum2` offers the option to reset the accumulated sum to 0 during measurement. Otherwise, the behavior of the two functions is the same.

Example

```
Result = Sum2( Channel_001, 10, VrtBit_01 )
```

The parameter 'Reset' may be either a single value (e.g. virtual bit) or a channel.

If the parameter 'Reset' is a single value, and resetting is executed, the input channel samples accumulating during this processing step are ignored in the summation. Multiple samples in one processing step can accumulate especially in the case of high sampling rates or where there are extensive calculations (e.g. use of FFTs). The summation result is afterwards zero.

If the parameter 'Reset' is a channel and resetting is executed, then any input channel samples accumulating after the reset are captured correctly. Even if there arrive during the same processing step.

SyncOverload

Synchronous task is overloaded: Sets the specified value if the synchronous task is overloaded.

Prerequisite:
Only available in: imc Online FAMOS!
Only usable with control commands!
The function SyncOverload() may only be called in the synchronous task.

Overloaded = SyncOverload(Value)

Overloaded: Result	**Value**: Fixed value to which the result is set

If the synchronous task is overloaded, the result variable is set to the value specified in the function `SyncOverload`. The function can be called at any point in the lowest level of a synchronous task. No combinations with other functions are permitted. The parameter Value must take a fixed value (e.g. 1, 2 or 3).

As results, Display variables, virtual bits, Ethernet bits, digital outputs, pv-variables and vector elements generated with the function `VectorChannel` are all permitted. If the result is a vector element, then if the synchronous task is constantly overloaded, no channels and thus no vectors can be outputted any longer (in the curve window).

If multiple synchronous tasks are going and data overload is detected in one of them, the overload will be indicated for this synchronous task or possibly for a faster synchronous task.

Example

```
OnSyncTask( 0.0001 )
    Virt_Bit01 = SyncOverload( 1 )
    pv.SyncOverloaded = SyncOverload( 10 )
    ...
End

OnSyncTask( 0.001 )
    ...
    DisplayVar_32 = SyncOverload( 20 )
End
```

11.2.9.2.17 T

Tan

Tangent of input channel

a = Tan(b)

a: Result	**b**: Input channel; in Radians

TextAdd

Join texts. Two texts are conjoined. Text2 is appended to Text1.

Text = TextAdd(Text1, Text2)

Text: Result	**Text1**: Text, to which to append Text2
	Text2: Text, to which Text1 is appended

Instead of the function `TextAdd`, the operator '+' can be used, see example.

Text-functions require large amounts of processing time and therefore should only be used for isolated outputs of text, rather than continually.

Example

```
OnInitAll
    Text1 = "Number"
    Text3 = ""
    IsText = 0
    Number = 1000
End

OnTriggerMeasure ( BaseTrigger )
    ...
    If IsText
        Text3 = Text1 + TextFormatH( Number, 8 )
        RecordText( Text3 )
        ; The text "Number = 0x000003E8" is returned
        IsText = 0
    End
    RecordEvent( Channel_001, "Index = " + TextFormatI( 250 ) )
    ; If the event occurs, the text "Index = 250" is returned
End
```

TextFormatE

Converts a real number to text. A real number is converted to text in floating point notation.

Text = TextFormatE(Number, DecPlace)

Text: Result	**DecPlace**: Number of digits after the decimal
Number: Real number to be converted to text	

The text contains as many significant digits as the parameter supplied to the function.

E.g. with four decimal places, the number 35 is represented as "3.5000E01" and the number 32767 as "3.2770E04".

Text-functions require large amounts of processing time and therefore should only be used for isolated outputs of text, rather than continually.

Example

```
OnInitAll
    Text1 = "Number"
    Text3 = ""
    IsText = 0
    Number = 1000
End

OnTriggerMeasure ( BaseTrigger )
    ...
    If IsText
        Text3 = Text1 + TextFormatE( Number, 6 )
        RecordText( Text3 )
        ; The text "Number = 1.000000E03" is returned
        IsText = 0
    End
    RecordEvent( Channel_001, "Index = " + TextFormatE( 250, 4 ) )
    ; If the event occurs, the text "Index = 2.5000E02" is returned
End
```

TextFormatF

Translates a real number to text. A real number is converted to text with fixed point notation.

Text = TextFormatF(Number, DecPlace)

Text: Result	**DecPlace**: Number of digits after the decimal
Number: Real number to be converted to text	

The text contains as many decimal places as the corresponding parameter function contains.

E.g., with four decimal places, the number 35 is represented as "3.5000" and the number 32767 as "32767.0000".

Text-functions require large amounts of processing time and therefore should only be used for isolated outputs of text, rather than continually.

Example

```
OnInitAll
    Text1 = "Number"
    Text3 = ""
    IsText = 0
    Number = 1000
End

OnTriggerMeasure ( BaseTrigger )
    ...
    If IsText
        Text3 = Text1 + TextFormatF( Number, 6 )
        RecordText( Text3 )
        ; The text "Number = 1.000000" is returned
        IsText = 0
    End
    RecordEvent( Channel_001, "Index = " + TextFormatF( 250, 4 ) )
    ; If the event occurs, the text "Index = 250.0000" is returned
End
```

TextFormatH

Converts a number to text. A number is converted to text with hexadecimal notation.

Text = TextFormatH(Number, Places)

Text: Result	**Places**: Number of decimal places
Number: Number to be converted to text	

The text contains as many digital places as as the parameter supplied to the function.

E.g. with four significant digits, the number 35 is represented as "0023" and the number 32767 as "7FFF".

Text-functions require large amounts of processing time and therefore should only be used for isolated outputs of text, rather than continually.

Example

```
OnInitAll
    Text1 = "Number"
    Text3 = ""
    IsText = 0
    Number = 1000
End

OnTriggerMeasure ( BaseTrigger )
    ...
    If IsText
        Text3 = Text1 + TextFormatH( Number, 8 )
        RecordText( Text3 )
        ; The text "Number = 0x000003E8" is returned
        IsText = 0
    End
    RecordEvent( Channel_001, "Index = " + TextFormatH( 250, 4 ) )
    ; If the event occurs, the text "Index = 0x00FA" is returned
End
```

TextFormatI

Converts a number to text in integer notation.

Text = TextFormatI(Number)

Text: Result	**Number**: Number to be converted to text

The text only contains as many digits as needed to represent the integer.

E.g. the number 35 appears as "35" and the number 32767 as "32767".

Text-functions require large amounts of processing time and therefore should only be used for isolated outputs of text, rather than continually.

Example

```
OnInitAll
    Text1 = "Number"
    Text3 = ""
    IsText = 0
    Number = 1000
End

OnTriggerMeasure ( BaseTrigger )
    ...
    If IsText
        Text3 = Text1 + TextFormatI( Number )
        RecordText( Text3 )
        ; The text "Number = 1000" is returned
        IsText = 0
    End
    RecordEvent( Channel_001, "Index = " + TextFormatI( 250 ) )
    ; If the event occurs, the text "Index = 250" is returned
End
```

Tolerance

Function that returns logical value of whether each input channel value is within the given tolerance.

a = Tolerance(b, min, max)

a: Result	**min**: Minimum
b: Input (channel or single values)	**max**: Maximum

Example

```
Res1 = Tolerance( Signal_01, pv.Min1, pv.Max1 )
Res2 = Tolerance( Signal_02, 2.5, 6.5 )
```

For example, Res2 takes the value 1 if Signal_02 is 4.1.

For example, Res2 takes the value 0, if Signal_02 is 2.1.

TransRec

Transitional Recording: Data reduction for analog data

Prerequisite:
Only available in: imc Online FAMOS!

a = TransRec(b, Precision)

a: result	**Precision**: Precision in units of the argument
b: input channel	

To perform long-term measurements at high resolution, a special computational algorithm has been designed. The principle on which it operates is to only save such data (data points) which are necessary for reconstructing the original signal at a given level of precision.

This means that rapid signal movements are sampled at a high sampling rate. However slow-paced signal movements are sampled at as few points as will still allow the signal to be reconstructed.

Whether a measured value is saved depends on the signal and on the tolerance threshold set. Data are only saved if the measured signal can not be represented by a substitute curve within the tolerance demanded. The data-reduced channel never deviates from the input signal by more that the amplitude difference demanded.

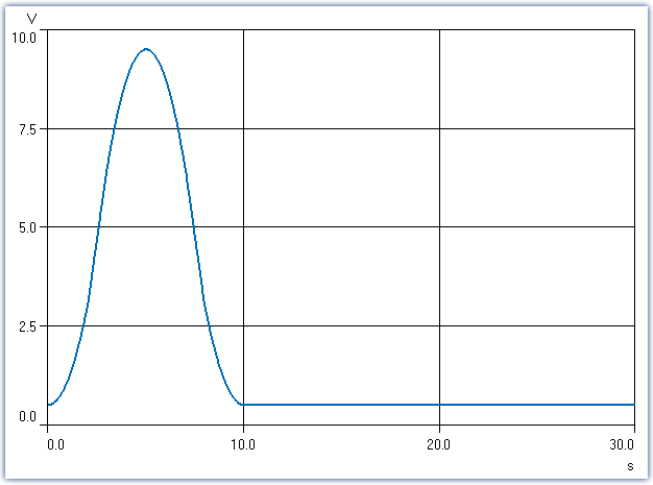
Data reduction factors of up to 254 can be achieved by means of this procedure.

Note Transfer to PC is not possible

Results of the `TransRec` function and DI-port compressed channels are blocked from transfer to the PC.

Example 1

The following input signal, consisting over a period of 10s of a long period of constancy and then a peak at 9.5V, is to be measured.

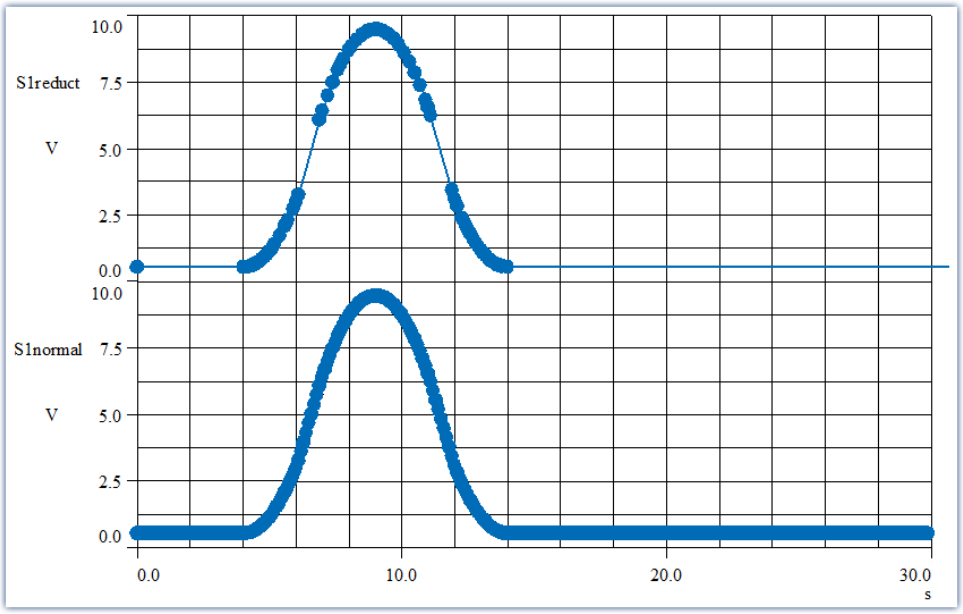


In this example, the signal is recorded once with and once without `TransRec` and saved.

As tolerance threshold set for the data reduction is 10mV.

S1reduct is measured with data reduction

S1normal is measured without data reduction



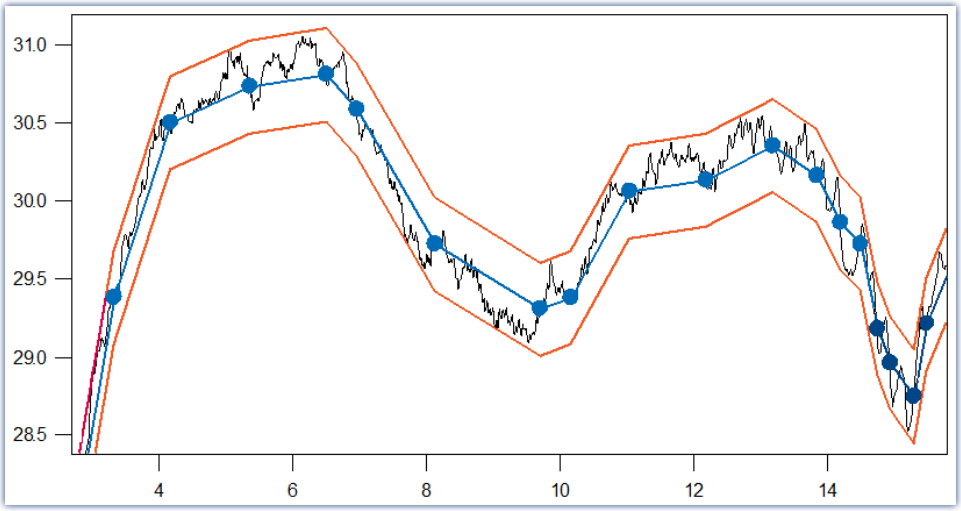
The curves are displayed on top of each other. The filled circles represent sample values, the lines between them are interpolations

The data reduction only adds measured points wherever the curve features significant curvature. Straight-line signal plots of any slope are recognized and summarized as a straight line between two measured points.

Example 2

In the illustration below, the results from a temperature measurement are shown. The original data (zigzag line) was recorded with a tolerance of 0.3°C. The reduced data are shown using solid circles, joined by straight lines.

To demonstrate the effectiveness of the algorithm, a tolerance band of ±3°C has been parallel to the interpolated line. As you can see, the original signal never exceeds the tolerance bounds.



The filled circles are derived by Transitional Recording.
The original signal never leaves the tolerance band.

For what sort of signals is Transitional Recording used?

Signals which feature movement only rarely or at constant slope, but in which major changes need to be sampled at a high rate. If a certain amount of imprecision (e.g. 1% of the input range) is acceptable, then this procedure is especially effective. Since temperature measurement often meets these criteria, the use of this procedure is very useful for this case.

`TransRec` is particularly suited to long-term measurements which would probably result in data overload in the absence of data reduction.

Precision

If you have defined a scaling for the reduced channel, then the precision should be selected accordingly.

Example

Scaling factor: 0.03 A/V
Input range: 0.3 A
Precision: 10 mA
Setting to make: 0.01

The unit of your signal is "A", so the tolerance (precision) is also specified in "A". 10 mA takes the notation 0.01 A, but the unit "A" must not be stated in the function.

Post calculations

The calculation results of the function `TransRec` can not be subject to further calculation in imc Online FAMOS or imc Inline FAMOS. Therefore it is best to first have all other calculations with data from a virtual channel performed and have `TransRec` executed last.

Example

```
_Difference_tmp = Channel_001 - Channel_002
Difference = TransRec (_Difference_tmp, 0.1)
```

In this example, the difference between two channels is first calculated.

The underline at the beginning indicates that the signal "_Difference_tmp" is not to be used as a virtual channel. Only after performing data reduction, we obtain the virtual channel with its correct name "Difference".

Maximum Reduction

The `TransRec` function produces resulting data sets with reduction factors between 1 and 254. Even with a constant signal, every 254th data point is recorded, thereby limiting the maximum reduction factor to 254.

Digital channels

The function `TransRec` can be applied to digital signals. However, this is not recommended.

Wherever possible, use the pre-processing for digital signals: **Reduction**. This option is available in the Setup dialog: **Sampling & Preprocessing**.

File format / Data type

The result of the `TransRec`-function takes the data type: **XY-waveform**. Like the other data types, it can be processed by other imc software packages (e.g. imc FAMOS).

Limitations

Buffer memory size in the device is limited to 128 kBytes. This means that the maximum buffer duration may be somewhat limited, depending on the sampling rate. However, if the technique is used properly and a significant reduction factor is achieved, that will not matter any more.

Note

Storage of measurement data always automatically generates sets of 64 samples. Therefore, when a measurement (or file) is closed after a fixed space of time, it usually happens that data points from before or after the fixed time interval are also saved in the file. If there is no data value which would normally be saved at the moment that the file is to be closed, the file is closed anyway. This means that the last data point in a file may be from a point in time significantly preceding the stopping time specified.

TSAGate

Gate for values in the case of time-stamped data: The input channel's values are only passed on as long as the control channel's value is non-zero.

Prerequisite:
Only available in: imc Inline FAMOS!

a = TSAGate(b, c, p)

a: Result	**c**: Control channel
b: TSA input channel	**p**: Pretrigger in bytes

The function sometimes yields output values at irregular intervals. Therefore, the result is generally difficult to combine with other channels in calculations.

The parameter p (pretrigger in Bytes) may be 0. Otherwise, p must be set in accordance with the size of the TSA-sample.

To do this, the size of a TSA-sample must be known.A TSA-sample fundamentally consists of a header and the data bytes.The header is always 8 Bytes long.The size of the data bytes may be a few Bytes (e.g. in the CAN-Bus up to 8 Bytes), or also several 1000 Bytes. Additionally, an overhead (e.g. for synchronization markers, etc.) can be taken into account:For smaller TSA-samples, the overhead should be only a few Bytes, while for larger ones, 10 % of a TSA-sample's data bytes, for example.

How the size of the parameters p is determined is illustrated by the example of a CAN message containing 8 data bytes:

The CAN message is sent every 100 ms. A pretrigger time of 2 s is to be taken into account. With 8 header bytes, 8 data bytes and an overhead of 4 in this example, the following results are obtained:

p = (8 Bytes + 8 Bytes + 4 Bytes) * 2 s / 100 ms = 400 Bytes.

Example

```
g = TSAGate ( TSA_CAN_001, c, 400 )
```

The channel g is a collection of all CAN messages appearing when the control channel c is non-zero. This takes a pretrigger of 2 s into account. For the calculation of the pretrigger in Bytes, see above.

11.2.9.2.18 U

UnEqual

Checks whether the first argument is not equal to the second argument

a = UnEqual(b, c)

a: Result	b: 1. argument
	c: 2. argument

a = 1 if b is not equal to c, otherwise 0.

Upper

Returns the greater value of the arguments

a = Upper(b, c)

a: Result	b: 1. argument
	c: 2. argument

Example

```
sound = Upper( dB( Channel_001 ), -100 )
```

The input channel is converted into decibel and restricted to a lower border of -100 dB.

11.2.9.2.19 V

General information about the VectorChannel functions

Prerequisite:
Only usable with control commands!

A virtual channel is declared with the function `VectorChannel`. All elements are initialized as zero. Declaration is performed in the OnInitAll-Block. In the curve window, this virtual channel is also represented as a virtual channel, but in imc Online FAMOS or imc Inline FAMOS it's treated as a static vector. The vector elements can be assigned and queried like single values.

Whenever the function `VectorChannelSet` is called, the complete vector is transferred to the virtual channel. The function `VectorChannelSet` can be called in `OnTriggerStart`, `OnTriggerMeasure` and `OnTriggerEnd` as well as in blocks which don't depend on triggers, e.g. in `OnSyncTask`. If the function `VectorChannelSet` is called too quickly and too often in succession, the memory allocation for the virtual channel can overflow.

Virtual channels can be generated from vectors with numbers and from vectors with real numbers (see example).

From version imc STUDIO 5.0 vector-handling has been enhanced with local vectors. Now, functions with 2 results vectors are supported, e.g. `FFTAmplitudePhase`(AmlSpec_Local, PhasSpec_Local, ...), where AmlSpec_Local and PhasSpec_Local must be declared as local vectors in the `OnInitAll`-block.

Example Example, generating virtual channels from vectors

```
OnInitAll
    int v1 = VectorChannel( BaseTrigger, 1000, 3 ) ; vector with integers
    ; v1[1] = 0, v1[2] = 0, v1[3] = 0  happens automatically
    v1[3] = 2

    v2 = VectorChannel( BaseTrigger, 1000, 3 ) ; vector with real numbers
    ; v2[1] = 0, v2[2] = 0, v2[3] = 0  happens automatically
    v2[3] = 2.5
End

OnSyncTask( 0.5 )
    v1[1] = v1[1] + 1
    v1[2] = VirtualBit_01
    v1[3] = 1
    VectorChannelSet( v1 ) ; the vector is transferred to the virtual channel

    v2[1] = v2[1] + 1
    v2[2] = VirtualBit_01
    v2[3] = 1.5
    VectorChannelSet( v2 ) ; the vector is transferred to the virtual channel
End
```

VectorChannel

Virtual channel from vector: A segmented virtual channel is generated from a vector.

Prerequisite:
Only usable with control commands!

VirtualChannel = VectorChannel(Trigger, DataRate, ElementCount)

VirtualChannel: Result **Trigger**: Trigger to which the channel is to be assigned.	**DataRate**: Average... max. count of vectors in a second; Allowed: 1 ... 10000 **ElementCount**: Number of elements in the vector; Allowed: 1 ... 100000

A virtual channel is declared with the function `VectorChannel`. All elements are initialized as zero. The declaration is performed in the `OnInitAll`-Block. In the curve window, this virtual channel is also represented as a virtual channel, but in imc Online FAMOS it's treated as a static vector. The vector elements can be assigned and queried like single values.

Whenever the function `VectorChannelSet` 1033 is called, the complete vector is transferred to the virtual channel. The function `VectorChannelSet` can be called in `OnTriggerStart`, `OnTriggerMeasure` and `OnTriggerEnd` as well as in blocks which don't depend on triggers, e.g. in `OnSyncTask`. If the function `VectorChannelSet` is called too quickly and too often in succession, the memory allocation for the virtual channel can overflow.

Virtual channels can be generated from vectors with numbers and from vectors with real numbers (see example).

Example 1

```
; Generating a virtual channel from a vector with integers:
OnInitAll
   int v1 = VectorChannel( BaseTrigger, 1000, 3 )
   ; v2[1] = 0, v2[2] = 0, v2[3] = 0  performed automatically
   v1[3] = 2
End

OnSyncTask( 0.5 )
   v1[1] = v1[1] + 1
   v1[2] = Virt_Bit01
   v1[3] = 1
   VectorChannelSet( v1 )
End
```

Example 2

```
; Generating a virtual channel from a vector with real numbers:
OnInitAll
   v2 = VectorChannel( BaseTrigger, 1000, 3 )
   ; v2[1] = 0, v2[2] = 0, v2[3] = 0  performed automatically
   v2[3] = 2.5
End

OnSyncTask( 0.1 )
   v2[1] = v2[1] + 1
   v2[2] = Virt_Bit02
   v2[3] = 1.5
   VectorChannelSet( v2 )
End
```

VectorChannelSet

Writing a vector: A vector is written to a segmented virtual channel.

Prerequisite:
Only usable with control commands!

VectorChannelSet(Vector)

Vector: A segmented virtual channel generated with the function `VectorChannel`	

VectorFromFile

Read vector from file

Vector = VectorFromFile("Filename")

Vector: Resulting vector	**"Filename"**: Name of a file containing the vector elements

Load the file

In imc STUDIO, the file of the vector data is permanently integrated in the experiment. Import this file using the ribbon.

Ribbon	View
Setup-Configuration > Supplemental Files	Complete

Notes

- The vector data is limited to 50000 values.
- Data formats (Float, Integer, Bool) can under certain circumstances be converted by imc Online FAMOS during loading. See also the notes on Boolean variables in IF conditions. 877

Example

```
Vector = VectorFromFile( "Vector_01.DAT" )
```

VectorizeAndSkip

Create vector series: groups input samples into a series of vectors

Vector = VectorizeAndSkip(b, VLength, SkipLength)

Vector: Output vector series	**VLength**: Number of samples per vector; > 1
b: Input channel	**SkipLength**: Number of samples to be skipped; ≥ 0

The input samples are grouped into a series of vectors.

A fixed number of samples can be skipped between successive vectors.

VectorizeOverlapped

Creates an overlapping sequence of vectors from the input channel

Vector = VectorizeOverlapped(b, VLength, Increment)

Vector: Result vector sequence	**VLength**: Length of a vector; > 1
b: input channel	**Increment**: Distance to the next vector; ≥ 1, ≤ VLength

The input channel is converted to a vector sequence.

The vectors can overlap each other, i.e. values from the input channel are entered multiple times in the vector sequence.

After every 'Increment' (e.g. 100) input channel values the subsequent 'Vlength' (e.g. 1024) input channel values are converted to a vector.

At a sampling rate of 1kHz, a vector of the length 1024 is created from these samples every 100ms.

Example

The vector a returns a vector with 1024 values for every 100 input values.

This results in an overlap ratio of (1024-100) / 1024 * 100% = 90.23%.

```
a = VectorizeOverlapped( Channel_001, 1024, 100 )
b = FFT( a, 0, 1024 )
```

VectorStatic

Creates a static vector. A static vector whose elements can be changed during the measurement is created.

Prerequisite:
Only usable with control commands!

VectStat = VectorStatic(Trigger, Element Count)

VectStat: Result	**Element count**: Number of the vector's elements; Allowed: 1 ... 100000
Trigger: Trigger to which the vector is to be assigned	

A static vector is declared using the function `VectorStatic`. All elements are initialized at "0". The declaration is made in the `OnInitAll` block.

The vector elements are treated as single values. Whenever an assignment is made to a vector element, the vector element's current value is transferred to the result vector. Assignments to vector elements can be called in `OnTriggerStart`, `OnTriggerMeasure` and `OnTriggerEnd` as well as in the blocks which do not depend on triggers, such as `OnTimer` or `OnSyncTask`.

Static vectors can be generated with integers and with real numbers (see examples).

Example

```
OnInitAll
    int vs1 = VectorStatic( BaseTrigger, 3 ) ; vector with integers
    ; vs1[1] = 0, vs1[2] = 0, vs1[3] = 0  performed automatically

    vs2 = VectorStatic( BaseTrigger, 3 ) ; vector with real numbers
    ; vs2[1] = 0.0, vs2[2] = 0.0, vs2[3] = 0.0  performed automatically
End

OnSyncTask( 0.5 )
    If Virt_Bit01 > 0
        vs1[1] = 4
        vs1[2] = 5
        vs1[3] = 6
    Else
        vs1[1] = 1
        vs1[2] = 1
        vs1[3] = 2
    End

    vs2[1] = pv.Channel_001
    vs2[2] = pv.Channel_002
    vs2[3] = pv.Channel_003
End
```

Access to vectors via variables

It is possible to access the elements of the vector with the help of variable. Thus, different values can be read according to the variable's value.

Example

In the following example, the value is **written** to the vector by means of the "pointer"-variable: "DisplayVar_01":

```
; writing to a vector at a variable position
Vektor[DisplayVar_01] = DisplayVar_02+10
```

In the following example, the value is **read** from the vector by means of the "pointer"-variable: "DisplayVar_01":

```
; reading from the vector at a variable position
DisplayVar_03 = Vector[DisplayVar_01]
```

Note

Notes on assignment

The first element is addressed with the value "1". The second with "2", etc.

Size of vector

The maximum number of values is limited to 100 000 values.

VibrationFilter

Vibration-weighting: A signal is filtered according to the frequency weighting set. Additionally, retroactive time weighting (moving RMS-value with exponential averaging) and resampling can be performed.

VibSignal = VibrationFilter(Signal, FrequencyWeighting, TimeConstant, ReductionFactor)

VibSignal: Filtered signal **Signal**: Input signal to be weighted; the time scaled in seconds. **FrequencyWeighting**: Frequency weighting for the signal	**TimeConstant**: Time constant for the exponential RMS-calculation in seconds, ≥ 0. e.g. 0.125 with FAST-weighting 1.0 with SLOW-weighting	**Reduction** factor: Factor for resampling, ≥ 1. With ReductionFactor = 1, no resampling. This enables the data volume of the result to be conveniently reduced when the time constant is large.

For a time constant = 0, no subsequent time weighting is performed. The reduction factor must in this case be exactly 1.

For numerical reasons, there are constraints on the input signals' sampling intervals which depend on the frequency weighting (see above).

Frequency weighting for the signal

per ISO 2631-1, 2nd edition, 1997:	
10	Wk, z direction and for vertical recumbent direction, except head. Recommended signal's sample time: ≥ 0.5ms and ≤ 5ms or ≥ 200Hz and ≤ 2kHz
11	Wd, x and y directions and for horizontal recumbent direction. Recommended signal's sample time: ≥ 0.5ms and ≤ 2ms or ≥ 500Hz and ≤ 2kHz
12	Wf, motion sickness. Recommended signal's sample time: ≥ 5ms and ≤ 500ms or ≥ 2Hz and ≤ 200Hz
13	Wc, seat-back measurement. Recommended signal's sample time: ≥ 0.5ms and ≤ 10ms or ≥ 100Hz and ≤ 2kHz
14	We, measurement of rotational vibration. Recommended signal's sample time: ≥ 0.5ms and ≤ 2ms or ≥ 500Hz and ≤ 2kHz
15	Wj, vibration under the head of a recumbent person. Recommended signal's sample time: ≥ 0.5ms and ≤ 2ms or ≥ 500Hz and ≤ 2kHz

per DIN 45671 Part 1, Sept. 1990:	
16	Hx, Whole-body vibrations, standing, seated position, measurement direction: x, y, Recumbent position, measurement direction: y, z. Recommended signal's sample time: ≥ 0.2ms and ≤ 50ms or ≥ 20Hz and ≤ 5kHz
17	Hz, Whole-body vibrations, standing, seated position, measurement direction: z Recommended signal's sample time: ≥ 0.5ms and ≤ 10ms or ≥ 100Hz and ≤ 2kHz
18	Hxl, Whole-body vibrations, recumbent position, measurement direction: x Recommended signal's sample time: ≥ 0.5ms and ≤ 2ms or ≥ 500Hz and ≤ 2kHz
19	Hb, Whole-body vibrations, body position not specified Recommended signal's sample time: ≥ 0.5ms and ≤ 10ms or ≥ 100Hz and ≤ 2kHz
20	Hh, Hand transmitted vibration, for all measurement directions Recommended signal's sample time: ≥ 0.02ms and ≤ 2ms or ≥ 500Hz and ≤ 50kHz
per ISO 7505, 1st edition, 1986-05-01:	
20	hand transmitted vibration, weighting filter. Recommended signal's sample time: ≥ 0.02ms and ≤ 2ms or ≥ 500Hz and ≤ 50kHz
per ISO 2631-1, 1st edition, 1985:	
21	Weighting factors for transverse (x, y) vibrations, see table 3. Recommended signal's sample time: ≥ 0.5ms and ≤ 100ms or ≥ 10Hz and ≤ 2kHz
22	Weighting factors for longitudinal (z) vibrations, see table 3. Recommended signal's sample time: ≥ 0.5ms and ≤ 50ms or ≥ 20Hz and ≤ 2kHz
per ISO 2631-4, 2001:	
23	Wb, passenger and crew comfort in fixed-guideway transport systems. Recommended signal's sample time: ≥ 0.5ms and ≤ 5ms or ≥ 200Hz and ≤ 2kHz
per ISO 2631-2, 2003:	
24	Wm, human exposure to vibration in buildings. Recommended signal's sample time: ≥ 0.2ms and ≤ 5ms or ≥ 200Hz and ≤ 5kHz
per ISO 6954, 2000:	
25	Acceleration input. Recommended signal's sample time: ≥ 0.2ms and ≤ 5ms or ≥ 200Hz and ≤ 5kHz
26	Velocity input. Recommended signal's sample time: ≥ 0.5ms and ≤ 2ms or ≥ 500Hz and ≤ 2kHz
per ISO 5349-1, 2001:	
27	Hand transmitted vibration, weighting filter. Recommended signal's sample time: ≥ 0.02ms and ≤ 2ms or ≥ 500Hz and ≤ 50kHz

per ISO 8041, 2005:	
28	Wb, passenger and crew comfort in fixed-guideway transport systems. Recommended signal's sample time: ≥ 0.5ms and ≤ 5ms or ≥ 200Hz and ≤ 2kHz
29	Wc, seat-back measurement. Recommended signal's sample time: ≥ 0.5ms and ≤ 10ms or ≥ 100Hz and ≤ 2kHz
30	Wd, x and y directions and for horizontal recumbent direction. Recommended signal's sample time: ≥ 0.5ms and ≤ 2ms or ≥ 500Hz and ≤ 2kHz
31	We, measurement of rotational vibration. Recommended signal's sample time: ≥ 0.5ms and ≤ 2ms or ≥ 500Hz and ≤ 2kHz
32	Wf, whole body low-frequency vibrations, motion sickness. Recommended signal's sample time: ≥ 5ms and ≤ 500ms or ≥ 2Hz and ≤ 200Hz
33	Wh, hand transmitted vibration. Recommended signal's sample time: ≥ 0.02ms and ≤ 2ms or ≥ 500Hz and ≤ 50kHz
34	Wj, vibration under the head of a recumbent person. Recommended signal's sample time: ≥ 0.5ms and ≤ 2ms or ≥ 500Hz and ≤ 2kHz
35	Wk, z direction and for vertical recumbent direction, except head. Recommended signal's sample time: ≥ 0.5ms and ≤ 5ms or ≥ 200Hz and ≤ 2kHz
36	Wm, human exposure to vibration in buildings. Recommended signal's sample time: ≥ 0.2ms and ≤ 5ms or ≥ 200Hz and ≤ 5kHz

Example

```
SignalVib = VibrationFilter( Signal, 10, 0.125, 1000 )
```

Wk Weighting per ISO 2631-1

The signal is subjected to vibration weighting. The frequency-weighted signal is then time-weighted at a time constant of 0.125 s (FAST) and resampled with a factor of 1000.

The signal originally has a sampling frequency of 20 kHz, the vibration-weighted result has a sampling frequency of only 20 Hz.

VIsAnyGreater

Return value is 1 if a vector element of b is larger than the corresponding element in c. Otherwise, the return value is 0.

a = VIsAnyGreater(b, c)

a: Return value	**b**: Input vector series **c**: Input vector series

Example

```
_tmpV1 = VectorizeAndSkip( Channel_001, 512, 0 )
_tmpV2 = VectorizeAndSkip( Channel_002, 512, 0 )
a = VIsAnyGreater(_tmpV1, _tmpV2 )
```

a: 1 when a sample of Channel_001 larger than Channel_002

VMax

Maximum from vector: returns the maximum value from each input vector

a = VMax(b)

a: Output value	b: Input vector series

VMaxV

Maximum from vector series: maximum values from a reduced vector series

a = VMaxV(b, WindowOption, RF)

a: Output vector series b: Input vector series	WindowOption: Compute maximum vector from 1: RF input vectors 2: All current input vectors	RF: Reduction factor (RF) in vectors; ≥ 1

For each element of the vector, the maximum value of the element for a vector sequence is determined. The result is another vector and over time a vector sequence.

For a window option of 1, the maxima over RF vectors are found and one output vector is generated after each RF input vectors.

For a window option of 2, the maxima of all previous input vectors are found. After every RF input vectors, one result vector is found.

VMean

Mean from vector: returns the mean value from each input vector

a = VMean(b)

a: Output value	b: Input vector series

VMeanV

Mean from vector series: mean values from a reduced vector series

a = VMeanV(b, WindowOption, RF)

a: Output vector series b: Input vector series	WindowOption: Compute mean vector from 1: RF input vectors 2: All current input vectors	RF: Reduction factor (RF) in vectors; ≥ 1

For each element of the vector, the mean value of the element for a vector sequence is determined. The result is another vector and over time a vector sequence.

For a window option of 1, the means over RF vectors are found and one output vector is generated after each RF input vectors.

For a window option of 2, the means of all previous input vectors are found. After every RF input vectors, one result vector is found.

© 2023 imc Test & Measurement GmbH

VMin

Minimum from vector: returns the minimum value from each input vector

a = VMin(b)

a: Output value	b: Input vector series

VMinV

Minimum from vector series: minimum values from a reduced vector series

a = VMinV(b, WindowOption, RF)

a: Output vector series b: Input vector series	WindowOption: Compute minimum vector from 1: RF input vectors 2: All current input vectors	RF: Reduction factor (RF) in vectors; ≥ 1

For each element of the vector, the minimum value of the element for a vector sequence is determined. The result is another vector and over time a vector sequence.

For a window option of 1, the minima over RF vectors are found and one output vector is generated after each RF input vectors.

For a window option of 2, the minima of all previous input vectors are found. After every RF input vectors, one result vector is found

VRedV

Reduction from vector series: reduces a vector series

a = VRedV(b, RF)

a: Output vector series b: Input vector series	RF: Reduction factor as vectors; ≥ 1

Every RF input vectors are reduced to a single output vector.

VRMS

RMS from vector: returns the RMS value of each input vector

a = VRMS(b)

a: Output value	b: Input vector series

VSum

Sum from vector: returns the sum of the elements in each input vector

a = VSum(b)

a: Output value	b: Input vector series

VValueAtXValue

Y-value at x-value: returns the y-value for a specified x-value

a = VValueAtXValue(b, XValue)

a: output value	**XValue**: X-value
b: Input vector series	range: x-offset up to delta-x * vector_length - x-offset

Example

```
frqPos = VXValueOfMax( SPEK_PosA ) ; frqPos = Display variable
phaPos = VValueAtXValue( SPEK_PosP, frqPos )
```

VXValueOfMax

X-value of maximum: returns x-value of the maximum element in a vector

a = VXValueOfMax(b)

a: Output value	**b**: Input vector series

VXValueOfMin

X-value of minimum: returns x-value of the minimum element in a vector

a = VXValueOfMin(b)

a: Output value	**b**: Input vector series

VXValueWithYValue

X-value with y-value: Returns x-value for a specified y-value

a = VXValueWithYValue(b, YValue, ErrorValue)

a: Output value	**YValue**: Y-value to find
b: Input vector series	**ErrorValue**: Error value to return if y-value not found

11.2.9.2.20 W

WindRoseCorr

To Values less than zero 360 is added and from values greater than 360, 360 is subtracted.

a = WindRoseCorr(b)

a: result	**b**: argument

The function `NorthCorrection` 967 corrects the discontinuity at due North according to the addition method. It prevents skipping from 360° to 0° in the averaging interval.

For values fluctuating around 360°, an average of 180° is avoided.

The result of averaging could lie outside of the wind rose range 0°..360°, e.g 365°.

The function `WindRoseCorr` returns the result to the wind rose range (0°..360°), e.g. changes 365° to 5°.

This procedure is conditional on the wind direction not changing more than 100 degrees.

The only sensible way to use the functions `NorthCorrection` 967 and `WindRoseCorr` is in a combination as illustrated in the example below. Instead of averaging, the standard deviation, i.e. the function `StDev` 1016, can be used.

Example

```
NC = NorthCorrection( Channel, 10 )
NC_Mean = Mean( NC, 10, 10 )
Result = WindRoseCorr( NC_Mean )
```

11.2.9.3 Functions with Control commands

Prerequisite:
Only usable with control commands!

Function Groups	Operators and Symbols	Additional functions
Measurement Process States 1043	(=) Equal Operator (Comparison) 898	AND 906
Loops, Conditions 1050	(<>) Inequality operator 899	CloseSaveInterval 913
Timer Functions 1054	(<) Less than operator 899	CurrentValue 922
CanMsg 1055	(<=) Less than or equal operator 899	DelayBuffer* 923
Controller Functions 1060	(>) Greater than operator 900	ECU* 932
	(>=) Greater than or equal operator 900	GetDateTime 942
		NOT 968
		OnECUCmdReturn_ECU_* 933
		OR 969
		ReadyForPowerOff 998
		SyncOverload 1019
		VectorChannel 1032
		VectorChannelSet 1033
		VectorStatic 1034

11.2.9.3.1 Measurement process states

Operations can be carried out in response to different states of the measurement process. Calculations of channel data can only be performed in `OnTriggerMeasure`. In the other states, only operations with single values (state variables (single values), bits, DACs, LEDs etc.) are allowed.

Variables for single values must be initialized in the `OnInitAll`-block if they are used in a formula. Each measurement state must be concluded with an `End` command.

OnInitAll

Executed once at the start

OnInitAll

All commands within this section are executed once at the start. This section is executed exactly one time after the configuration is loaded to the device. This makes this section suitable for initializations. Global variables (single values) in particular are initialized here for later use. Digital bits also can be initialized here.

Loops and conditions are supported within the `OnInitAll` routine.

All global variables later used for calculations in imc Online FAMOS or imc Inline FAMOS must first be initialized in the `OnInitAll` block.

Example

```
OnInitAll
    Virt_Bit01 = 1
    Voltage    = 10
    Counter    = 0
    LED_01     = 0
    int v[8]
    int i      = 0

    For i = 1 Till 8 Step 1
        v[i] = 1
    End
End
```

Note Prepare and Reconfigure

The "*OnInitAll*" block runs when the device needs to be prepared again. But this is not applicable to repeated measurements, unless the device configuration has been changed between measurements.

When changes are made to the Panel pages or to PC-side analyses, such as imc Inline FAMOS, Bus Decoder, the device configuration remains unaffected. For this reason, no preparation is performed.

Even **manually clicking the button "*Prepare*"** does **not cause preparation** to be performed. In this case, the system checks whether any changes had been made to the device configuration. Then preparation is performed if this applies.

Preparing the device **can be forced** by selecting the menu item **"*Reconfiguration*"**.

See also: "*Prepare and Reconfigure* 195"

OnAlways

Executed constantly

OnAlways

The command lines in this section are constantly executed after the commands in `OnInitAll`.

This is not a one-off event, but is constantly repeated whether or not measurements are in progress.

This section works with global variables (single values) and digital bits. If calculations are performed on global variables (single values), they must first be initialized in the section `OnInitAll`.

Example

```
OnInitAll
    Reset = 0
End

OnAlways
    Virt_Bit10 = 1
    If DisplayVar_01 > 0
       DisplayVar_02 = 0
       Reset = 1
    Else
       DisplayVar_02 = 1
       Reset = 0
    End
End
```

Note

Writing to the device like"LED_01 = 1" should be avoided in `OnAlways`. It could cause an bus-overload.

OnMeasureStart

Executed at the push of the Start-button

OnMeasureStart

The command lines in this section are carried out one time only if the measurement is started, meaning that the button for starting the measurement is pressed.

This section works with global variables (single values) and digital bits. If calculations are performed on global variables (single values), they must first be initialized in the section `OnInitAll`.

Example

```
OnInitAll
    Reset = 0
End

OnMeasureStart
    Virt_Bit10 = 1
    If DisplayVar_01 > 0
       DisplayVar_02 = 0
       Reset = 1
    Else
       DisplayVar_02 = 1
       Reset = 0
    End
End
```

OnMeasureEnd

Executed at the push of the Stop-button

OnMeasureEnd

The command lines in this section are carried out one time only if the measurement is completed, meaning that the button for ending the measurement is pressed or the test carried out finished independently.

This section works with global variables (single values) and digital bits. If calculations are performed on global variables (single values), they must first be initialized in the section `OnInitAll`.

Example

```
OnInitAll
    Reset = 0
End

OnMeasureEnd
    Virt_Bit10 = 1
    If DisplayVar_01 > 0
       DisplayVar_02 = 0
       Reset = 1
    Else
       DisplayVar_02 = 1
       Reset = 0
    End
End
```

OnSyncTask

Run exactly upon elapse of the time interval set.

Prerequisite:

Only available in: imc Online FAMOS Professional!

OnSyncTask(CycleTime)

This segment forms the synchronous task, which is activated cyclically in synchronization with the device's AD converter. The command lines contained in this segment are performed exactly one time once the specified cycle time has elapsed again. A maximum of 4 synchronous tasks are possible.

The parameter Time is stated in s.

The following entries are allowed: 0.0001s, 0.0002s, 0.0005s, ..., 0.1s, 0.2s, 0.5s, 1s.

Example

```
OnSyncTask( 0.1 ) ; Cycle time of 0.1s
    If VirtBit_01 <> 0
       VirtBit_02 = 1
    Else
        VirtBit_02 = 0
    End
End
```

If the cycle time is insufficient for performing the commands, the beeper and LED_06 are activated. Alternatively, this can be monitored using the command `SyncOverload` 1019.

Reference

For more information on **controller** see in the documentation on imc Online FAMOS: "*imc Online FAMOS Professional*" > "*PID-Controller* 886".

OnTriggerStart

Run upon start of the measurement

OnTriggerStart(TriggerName)

The command lines in this section are executed a single time at the start of every measurement, at the exact moment the trigger is released.

This section works with global variables (single values) and digital bits. If calculations are performed on global variables (single values), they must first be initialized in the section `OnInitAll`.

Example

```
OnInitAll
    Counter = 0
    VirtBit_02 = 1
End

OnTriggerStart( BaseTrigger )
    Counter = Counter + 1
    If VirtBit_01 > 0
        VirtBit_02 = 0
    End
End
```

OnTriggerEnd

Executed at the end of every measurement

OnTriggerEnd(TriggerName)

The command lines in this section are carried out one single time at the end of every measurement, at the moment that no more data are arriving due to the trigger.

This section works with global variables (single values) and digital bits. If calculations are performed on global variables (single values), they must first be initialized in the section `OnInitAll`.

Example

```
OnInitAll
    Counter = 0
    VirtBit_02 = 0
End

OnTriggerEnd( BaseTrigger )
    Counter = Counter + 1
    If VirtBit_01 < 1
        VirtBit_02 = 1
    End
End
```

OnTriggerMeasure

Executed during a running measurement

OnTriggerMeasure(TriggerName)

The command lines in this section are executed during a running measurement. This is not a one-off event, instead, the execution of this section is constantly repeated between the trigger start and the trigger end.

This section works with channels and virtual channels, but also with single values and digital bits.

If calculations are performed on global variables (single values), they must first be initialized in the section `OnInitAll`.

Example

```
OnTriggerMeasure( BaseTrigger )
    VirtChannel_001 = Channel_001 + 5
    If VirtBit_01 > 0
        VirtChannel_002 = Channel_002 + 10
    Else
        VirtChannel_002 = Channel_002 + 5
    End
End
```

OnTimer

Run upon elapse of the time interval set

OnTimer(Timer_ID)

The command lines in this section are executed once for each time the appropriate timer is released. A timer can be created using `StartTimerPeriodic` or `StartTimerSingle` (see Timer functions 1054).

If calculations are performed on global variables (single values), they must first be initialized in the section `OnInitAll`.

The specified timer-ID must exactly match the one for the corresponding timer start function; the values permitted are 1 ... 8.

Example

```
OnTriggerStart( BaseTrigger )
    StartTimerPeriodic( 5, 0.1, 0.1 )
End

OnTimer( 5 )
    If VirtBit_01 > 0
        VirtBit_02 = 1
    Else
        VirtBit_02 = 0
    End
End
```

OnPowerOff

Run upon switch off the device.

Prerequisite:
Only available in: imc Online FAMOS!

OnPowerOff(cycle time)

This section performs the power off procedure which comes into effect when the device switched off. The battery in the device then takes over supply for a maximum of 8 s. Within this time, the application can be closed properly in the section `OnPowerOff`. When calling the function `ReadyForPowerOff`, device supply via the internal battery is ready to disabled.

One `OnPowerOff` section per synchronous task is allowed. The `OnPowerOff` section associated with the synchronous task is identified by the stated cycle time.

The cycle time parameter is stated in seconds. Any of the following cylcle times may be specified: 0.0001 s, 0.0002 s, 0.0005 s, ..., 0.1 s, 0.2 s, 0.5 s, 1 s.

Example

```
OnSyncTask( 0.01 ) ; Cycle time: 0.01s
    DisplayVar_01 = pv.Channel_001
End

OnPowerOff( 0.01 ) ; Cycle time: 0.01s
    DisplayVar_01 = 0
    DOut01_Bit01  = 0
    DOut01_Bit02  = 0
    ReadyForPowerOff()
End
```

11.2.9.3.2 Loops, conditions

If condition, Else

This control command sets the operations appearing underneath to be carried out conditionally.

The operations are only executed if the condition is fulfilled. If the single value passed to it is not equal to 0, the condition is fulfilled. The operations must be concluded by End or Else.

Else denotes the start of operations to be performed if the prior If-condition is not fulfilled. The operations must be concluded by End.

Example

```
OnTriggerMeasure( BaseTrigger )
    If VirtBit_01 > 0
        VirtChannel_001 = Channel_001 + 1
    End
    If VirtBit_05 = 0
        VirtChannel_002 = Channel_002 + 5
    Else
        VirtChannel_002 = Channel_002 + 10
    End
End
```

A condition is usually employed in the manner shown in the example: two operands are compared according to a comparison operator. Naturally, a condition in the form of "If a" is also possible. In such a case, a must be a single value of type BOOL, or the Boolean result of a calculation.

The comparison operators "<", ">", "=" etc. also return variables of type BOOL. Instead of "`If VirtBit_01 > 0`", the command lines "`a = VirtBit_01 > 0`" and "`If a`" can be coded. Combinations of different conditions are also allowed, e.g. "`If VirtBit_01 > 0 AND VirtBit_02 = 0`".

Example Functions and Bit-variables in conditions

Functions in conditions

```
If Greater(Current_RPMs, 2200) = 1
```

or the following alternative:

```
If Greater(Current_RPMs, 2200)
```

Bit-variables in conditions

```
If Virt_Bit01 = 1
```

or the following alternative:

```
If Virt_Bit01
```

Example Calculation results in conditions

If the results of a calculation are certain to be in Boolean format, the variable can be used in the subsequent conditions without any assignment, as if it were a bit-variable.

```
a = b > c
If a = 1
```

or the following alternative:

```
a = b > c
If a
```

The variable "a" is a single value variable which is created in `OnInitAll`. "`b > c`" always returns either 1 or 0 as its result. Even if the variable contains a non-boolean result at a different location, it can be used in this case.

This behavior only applies to single value variables and not to device-variables.

Note Boolean variables from files

See also the notes when using imported variables in If conditions 877.

Switch

This control command can be used to perform operations in dependence upon the value of the parameter b. In the list of case-blocks, different values for the parameter b are treated.

A Case-block is executed each time the numerical value of the parameter b is equal to the Case-command's numerical value. Any numerical values for the parameter b not explicitly listed are treated in the default case.

The Switch command is concluded with an End-command.

The parameter for a Case-command must be an integer ≥ 0. Real numbers are not allowed as Case-command parameters.

Example

```
OnTriggerMeasure( BaseTrigger )
    Value = CurrentValue( Channel_001, 0, 0.0 )
    Switch Value
        Case 1
            VirtChannel_001 = Channel_001 + 1
            VirtChannel_002 = Channel_002 + 1
        End
        Case 2
            VirtChannel_001 = Channel_001 + 10
            VirtChannel_002 = Channel_002 + 2
        End
        Default
            VirtChannel_001 = Channel_001 + 100
            VirtChannel_002 = Channel_002 + 3
        End
    End
End
```

For

For Counter = Start Till End Step Iteration

A For-loop enables repeated execution of operations. The number of loop runs is always fixed in For-loops. The loop is carried out, starting when Counter = Start, until Counter is smaller than or equal to the value "End". After each run of the loop, the Counter is iterated upward by the number "Iteration".

The counter must be defined as an **integer single value** in the `OnInitAll`-block. The **maximum loop count** can also be specified by a **variable**. In that case it must be declared in an the `OnInitAll`-block as an integer and can be changed during the measurement.

Start, End and Iteration must be integers. The For-block is concluded by an "End" command. Single values and digital bits can be processed in loops, channel definitions are not permitted here.

Example

```
OnInitAll
    int i = 0
    HistoSum = 0
    int MaxLoops= 15
End

OnTriggerMeasure( BaseTrigger )
    Histo = Histogramm( Channel_001,  -10, 10, 16, 1, 1 )
    HistoSum = 0
    For i = 0 Till MaxLoops Step 1
        HistoSum = HistoSum + GetHistoValue( Histo, i )
    End
    VrtSum = Channel_001*0 + HistoSum
End
```

While condition

While condition

A While-loop enables repeated execution of operations. The number of loop runs for "While" depends on the condition, so the number isn't necessarily known when the loop call comes. The operations in the While-block are carried out until the condition is met, i.e. the parameter of the While instruction is not equal to 0. The While-block is concluded by an "End" command.

Single values and digital bits can be processed in loops, channel definitions are not permitted here.

Example

```
OnInitAll
    int i = 0
    HistoSum = 0
End

OnTriggerMeasure( BaseTrigger )
    Histo = ClHistogram( Channel_001,-10,10,16,1,1)
    HistoSum = 0
    i = 0
    While i <= 15
        HistoSum = HistoSum + GetHistoValue( Histo, i )
        i = i + 1
    End
    VrtSum = Channel_001*0 + HistoSum
End
```

The While-loop is carried out until the condition is fulfilled. If the condition is always fulfilled, the While-loop is never exited. While-loops are commonly used in the manner illustrated by the example: two operands are compared using a comparison operator. Combinations of conditions are also allowed: e.g "While i ≤ 15 AND HistoSum ≤ 5000".

Default

The Case-command states the particular value of the variable specified by a prior switch command. In the list of case-blocks, different values for the parameter of the switch command are treated. A Case-block is executed each time the numerical value of the parameter of the switch command is equal to the Case-command's numerical value. Any numerical values for the parameter of the switch command not explicitly listed are treated in the default case. The Default command is concluded with an End-command. The parameter for a Case-command must be an integer ≥ 0. Real numbers are not allowed as Case-command parameters.

Example

```
OnTriggerMeasure( BaseTrigger )
    Value = CurrentValue( Channel_001, 0, 0.0 )
    Switch Value
        Case 1
            VirtChannel_001 = Channel_001 + 1
            VirtChannel_002 = Channel_002 + 1
        End
        Case 2
            VirtChannel_001 = Channel_001 + 10
            VirtChannel_002 = Channel_002 + 2
        End
        Default
            VirtChannel_001 = Channel_001 + 100
            VirtChannel_002 = Channel_002 + 3
        End
    End
End
```

11.2.9.3.3 Timer functions

Up to eight different timers can be implemented. Each timer must be started by a function. Both periodic timers (the timer is released periodically at fixed intervals) and one-off timers (the timer is released after a specified interval) can be realized. To stop a periodic timer, the function StopTimer must be used. Timers can be started and stopped in any state of the measurement process.

In the `OnTimer` block, only single values (e.g. single value variables, bits, LEDs, DACs, ...) can be subjected to calculations.

StartTimerPeriodic

Periodic timer. Creates a periodic timer.

StartTimerPeriodic (Timer_ID, TimeInterval, TimeInterval_1)

Timer_ID: Identifies the timer (1..8)	**TimeInterval**: Release upon elapse of time interval, in seconds **TimeInterval_1**: Time in seconds, after which release occurs

The time-count begins when this function is called. When the time TimeInterval_1 has elapsed, the timer is released one single time. After that, the timer is released repeatedly after each interval of duration TimeInverval. When the timer is released, the `OnTimer` section of the corresponding Timer_ID is executed.

The time intervals can be specified to a precision of 0.0001 s (0.1 ms). TimeInterval must be at least 0.0001 s; TimeInterval_1 can also be 0 s.

The permitted maximum values for the parameters TimeInterval and TimeInterval_1 depend on the device types used.

Example

```
OnTriggerStart( BaseTrigger )
    ; Create periodic timer
    StartTimerPeriodic ( 5, 0.5, 0.0 )
End

OnTimer( 5 )
    If VirtBit_01 > 0
        LED_01 = 1
    Else
        LED_01 = 0
    End
End
```

StartTimerSingle

Single-use timer. Creates a timer for one-off use.

StartTimerSingle (Timer_ID, TimeInterval)

Timer_ID: Identifies the timer (1..8)	**TimeInterval**: Time in seconds, after which release occurs

The time-count begins when this function is called. When the time TimeInterval has elapsed, the timer is released one single time. Then, the `OnTimer` section of the corresponding Timer_ID is executed.

The time interval can be stated to a precision of 0.0001 s (0.1 ms) and must be at least 0.0001 s.

The permitted maximum value for the parameter TimeInterval depends on the device type currently in use.

Example

```
OnTriggerStart( BaseTrigger )
    ; Create a one-off timer
    StartTimerSingle( 1, 5.0 )
End

OnTimer( 1 )
    If VirtBit_01 > 0
        LED_01 = 1
    Else
        LED_01 = 0
    End
End
```

StopTimer

Deletes a timer. The timer cannot be released. It no longer exists.

StopTimer (Timer_ID)

Timer_ID: Identifies the timer (1..8)

Example

```
OnTriggerMeasure( BaseTrigger )
    If VirtBit_01 > 0
        ; Create periodic timer
        StartTimerSingle( 5, 0.0 )
    End
    If VirtBit_02 > 0
        StopTimer( 5 ) ; Delete timer
    End
End

OnTimer( 5 )
    If VirtBit_05 > 0
        LED_01 = 1
    Else
        LED_01 = 0
    End
End
```

11.2.9.3.4 CanMsg

Prerequisite:

Only available in: imc Online FAMOS!

CanMsg

CanMsg-initialization: Initialization of the CanMsg-structure with default values.

CanMsg1 = CanMsg()

CanMsg1: CanMsg-structure

To send CAN- messages with CanMsg-structures, the function `CanMsg` must be called in the `OnInitAll` section. In the process, a `CanMsg`-structure is created and initialized. MessageID, message-length and message-content must be set before the CAN-message is sent. The message-ID and message-length are set with the variables `.ID` and `.Len`. The individual message bytes are set with the variables `.Byte0` ... `.Byte7` or with the function `.SetData`. The CAN-message thus created is sent by means of the `.Transmit1_S1`.

Example sending a CAN-message

```
OnInitAll
    CanMsg1 = CanMsg()
End

OnAlways
    If VirtBit_01 > 0
        CanMsg1.ID = 100
        CanMsg1.Len   = 2
        CanMsg1.Byte0 = 10
        CanMsg1.Byte1 = 20
        CanMsg1.Transmit1_S1()
        VirtBit_01    = 0
    End
End
```

Example receiving a CAN-message

```
OnInitAll
    CanID    = 0
    CanLen   = 0
    CanByte0 = 0
    CanByte1 = 0
End

OnCanMessageReceive1_S1( CanMsg1 )
    CanID    = CanMsg1.ID
    CanLen   = CanMsg1.Len
    CanByte0 = CanMsg1.Byte0
    CanByte1 = CanMsg1.Byte1
End

OnTriggerMeasure( BaseTrigger )
    VID    = Channel_001*0 + CanID
    VLen   = Channel_001*0 + CanLen
    VByte0 = Channel_001*0 + CanByte0
    VByte1 = Channel_001*0 + CanByte1
End
```

Note

The `CanMsg` functions only appear if the CAN-Assistant had previously been opened. No changes to or operations in the CAN-Assistant need to have happened, but only once it has been opened does imc Online FAMOS learn that there is a CAN Bus interface.

CanMsg.GetData

Value from CAN-message: A value is extracted from a CAN-message.

Result = CanMsg.GetData(StartByte, StartBit, BitCount, BitSequence, Format)

Result: Value extracted from the CAN-message

StartByte: Initial byte in the message, 0..7

StartBit: Initial bite in the message, 0..7

BitCount: Number of bits in the value to be sent, 1..32

ByteSequence: Intel- or Motorola-format

1: Intel

2: Motorola

Format: Numerical format of the value to be sent,

1: signed integer

2: unsigned integer

3: real number (32 bit)

4: digital bit

To use this function, it's necessary to generate a `OnCanMessageReceive1_S1`-section, in which the function `.GetData` is called. The content of the received CAN-messages is extracted from the message with the variables `.Byte0` ... `.Byte7` or with the function `.GetData`.

Example

```
OnInitAll
    CanID    = 0
    CanLen   = 0
    CanByte0 = 0
    CanByte1 = 0
End

OnCanMessageReceive1_S1( CanMsg1 )
    CanID    = CanMsg1.ID
    CanLen   = CanMsg1.Len
    CanByte0 = CanMsg1.GetData( 0, 4, 16, 1, 1 )
    CanByte1 = CanMsg1.GetData( 2, 4, 16, 1, 1 )
End

OnTriggerMeasure( BaseTrigger )
    VID    = Channel_001*0 + CanID
    VLen   = Channel_001*0 + CanLen
    VByte0 = Channel_001*0 + CanByte0
    VByte1 = Channel_001*0 + CanByte1
End
```

CanMsg.SetData

Value in message: A value is inserted into a CAN-message.

CanMsg.SetData(Value, StartByte, StartBit, BitCount, ByteSequence, Format, Factor, Offset)

Value: Value to be sent

StartByte: Initial byte in the message, 0..7

StartBit: Initial bit in the message, 0..7

BitCount: Number of bits in the value to be sent, 1..32

ByteSequence: Intel- or Motorola-Format,

1: Intel

2: Motorola

Format: Numerical format of the value to send,

1: signed integer

2: unsigned integer

3: real number (32 bit)

4: digital bit

Factor: Weighting of the value to be sent with a factor

Offset: Weighting of the value to be sent with an offset

To use the function, a `CanMsg`-structure must be defined in the `OnInitAll` section.

Message-ID, message-length and message-content must be set before the CAN-message is sent. The message-ID and message-length are set with the variables `.ID` and `.Len`. The individual message bytes are set with the variables `.Byte0` ... `.Byte7` or with the function `.SetData`. The CAN-message thus created is sent by means of the function `.Transmit1_S1`.

Example

```
OnInitAll
    CanMsg1 = CanMsg()
End

OnAlways
    If VirtBit_01 > 0
        CanMsg1.ID = 100
        CanMsg1.Len = 4
        CanMsg1.SetData( 1000, 0, 4, 16, 1, 1, 1.0, 0.0 )
        CanMsg1.SetData( 2000, 2, 4, 16, 1, 1, 1.0, 0.0 )
        CanMsg1.Transmit1_S1()
        VirtBit_01 = 0
    End
End
```

CanMsg.Transmit1_S1

Send a CAN-message: A CAN-message is sent to Node 1.

CanMsg.Transmit1_S1( )

To use the function, a `CanMsg`-structure must be defined in the `OnInitAll` section.

Message-ID, message-length and message-content must be set before the CAN-message is sent. The message-ID and message-length are set with the variables `.ID` and `.Len`. The individual message bytes are set with the variables `.Byte0` ... `.Byte7` or with the function `.SetData`. The CAN-message thus created is sent by means of the function `.Transmit1_S1`.

Example 1

```
OnInitAll
    CanMsg1 = CanMsg()
End

OnAlways
    If VirtBit_01 > 0
        CanMsg1.ID = 100
        CanMsg1.Len   = 2
        CanMsg1.Byte0 = 10
        CanMsg1.Byte1 = 20
        CanMsg1.Transmit1_S1()
        VirtBit_01    = 0
    End
End
```

Example 2

```
OnInitAll
    CanMsg1 = CanMsg()
End

OnAlways
    If VirtBit_01 > 0
        CanMsg1.ID = 100
        CanMsg1.Len = 4
        CanMsg1.SetData( 1000, 0, 4, 16, 1, 1, 1.0, 0.0 )
        CanMsg1.SetData( 2000, 2, 4, 16, 1, 1, 1.0, 0.0 )
        CanMsg1.Transmit1_S1()
        VirtBit_01  = 0
    End
End
```

OnCanMessageReceive1_S1

This section deals with CAN-messages which are received from the device and can't be permanently set in the CAN-Assistant.

OnCanMessageReceive1_S1(CanMsg1)

CanMsg1: CanMsg-structure to be newly created

The `CanMsg`-structure passed in is created and initialized for this section. The elements of the `CanMsg`-structure are treated like local single value variables in the section `OnCANMessageReceive_S1` and may also be used only within this section. These variables are filled when the command for reading CAN-messages is run, and can be used and queried in the section `OnCANMessageReceive_S1` on the right side of formulas. They are single values.

The content of the CAN-message can be extracted from the message either with the variables `.Byte0` ... `.Byte7` or with the function `.GetData`.

Example

```
OnInitAll
    CanID    = 0
    CanLen   = 0
    CanByte0 = 0
    CanByte1 = 0
End

OnCanMessageReceive1_S1( CanMsg1 )
    CanID    = CanMsg1.ID
    CanLen   = CanMsg1.Len
    CanByte0 = CanMsg1.Byte0
    CanByte1 = CanMsg1.Byte1
End

OnTriggerMeasure( BaseTrigger )
    VID    = Channel_001*0 + CanID
    VLen   = Channel_001*0 + CanLen
    VByte0 = Channel_001*0 + CanByte0
    VByte1 = Channel_001*0 + CanByte1
End
```

11.2.9.3.5 Controller functions

Prerequisite:

Only available in: imc Online FAMOS Professional!

Reference

For more information on **controller** see in the documentation on "*imc Online FAMOS and imc Inline FAMOS*" > "*Calculation examples*" > "*PID-Controller* 886".

CtPID

Create and initialize a PID-controller: Initializes a PID-controller with the specified values.

Controller = CtPID(P_component, I_component, D_component)

Controller: the controller-structure to be created	**I_component**: I-component
P_component: P-component	**D_component**: D-component

This function is called once for each controller used in the section `OnInitAll`. It creates a controller-structure. The constants passed are stated for a continuous (non-discrete) controller. They serve to initialize the elements `.KP`, `.KI`, `.KD`. All other elements are initialized as 0.0.

Except for `.DCutOff`, which is set to approx. 0.1 / [controller's cycle time].

The PID-controller's structure has been enhanced: In particular, input control, controller output limiting, and band limiting of the D-component are all possible.

Computation of the controller is performed in the `OnSyncTask`-block with the function `.Calc`. The elements of the structure can be queried and changed at any time, and the set point `.SetPoint` as well.

Example

```
OnInitAll
    EngineController = CtPID( 20.0, 0.5, 0 ) ; P, I, D
    EngineController.SetPoint = 6000
End

OnSyncTask( 0.1 )
    DAC_VoltageEngine = EngineController.Calc( pv.Speed )
    If VirtBit_01 <> 0
        EngineController.Reset = 1
    End
    If VirtBit_02 > 0
        EngineController.SetPoint = 2000
    End
End
```

CtPID.Calc

Compute PID-controller: calculation of the PID-controller.

ControllerOutput = CtPID.Calc(ProcessValue)

ControllerOutput: the controller's output	**ProcessValue**: instantaneous measurement value, process value for the controller

The function calculates exactly one step of the controller. For the currently passed process value, the new controller output is determined as the return value. The return value is then typically passed to a DAC. The controller output is subsequently available in `.PV`; the controller output also in `.CO`.

When the controller's parameters are changed, these changes only take effect upon the next call of the function `.Calc`, because that is the first (and only) time a new controller output is computed.

The function `.Calc` is only allowed in synchronous tasks.

Example

```
OnInitAll
    EngineController = CtPID( 20.0, 0.5, 0 ) ; P, I, D
    EngineController.SetPoint = 6000
End

OnSyncTask( 0.1 )
    DAC_VoltageEngine = EngineController.Calc( pv.Speed )
    If VirtBit_01 <> 0
        EngineController.Reset = 1
    End
    If VirtBit_02 > 0
        EngineController.SetPoint = 2000
    End
End
```

CtTwoPos

Create and initialize a two-point controller: Initializes a two-position controller with the specified values.

Controller = CtTwoPos(Hysteresis, Output_Inversion)

Controller: The newly created controller structure	**Output_Inversion**: Should the output be inverted?
Hysteresis: The distance between two reversal points	0: Default 1: Invert output

This function is called once for each two-position controller used in the section `OnInitAll`. It creates a controller structure.

The controller structure contains a number of elements which are initialized by this function. Thus, the controller's set point `.SetPoint` is set = 0, and the controller output `.CO` is set = 0. With the output inverted, `.CO` is set = 1.

The hysteresis is the distance between the upper and the lower reversal point. It should be > 0.

The two reversal points lie symmetrically around the setpoint. If the setpoint is = 20, for instance, and the hysteresis = 2, then the resulting reversal points are 21 and 19.

The controller's method of functioning: If the process value is greater than the upper reversal point, then the controller output = 1 is generated and returned by the function `.Calc`. If the process value is lower than the lower reversal point, then the resulting controller output is = 0. If the process value is between the two reversal points, the controller output remains unchanged.

By means of the parameter Output_Inversion, the controller output can normally be inverted, so that when the process value is too high, the resulting controller output is = 0; and when the process value is too low, the resulting CO is = 1.

Computation of the controller is performed in the `OnSyncTask`-block with the function `.Calc`. The elements of the structure can be queried and changed at any time, and the set point `.SetPoint` as well.

Example

```
OnInitAll
    Thermostat = CtTwoPos( 2, 0 ) ; Hysteresis, invert output
    Thermostat.SetPoint = 20 ; optional: set setpoint
    DigitalOut_01 = 0 ; optional: output in rest state
End

OnSyncTask( 0.1 )
    DigitalOut_01 = Thermostat.Calc( pv.Temperature_01 ) ; set setpoint
    ...
    If VirtBit_01 <> 0
        Thermostat.SetPoint = 22.0 ; change setpoint
    End
End
```

CtTwoPos.Calc

Calculate two-point controller: computation of a two-point controller.

ControllerOutput = CtTwoPos.Calc(ProcessValue)

ControllerOutput: the controller's output	**ProcessValue**: instantaneous measurement value, process value for the controller

The function computes exactly one controller step. For the currently passed process value, the new controller output is determined as the return value. The return value is then typically passed to a digital output. The process value is subsequently available in `.PV`; the controller output in `.CO`.

The controller output can take values between 0 and 1.

When the controller's parameters are changed, these changes only take effect upon the next call of the function `.Calc`, because that is the first (and only) time a new controller output is computed. The function `.Calc` is only permitted in synchronous tasks.

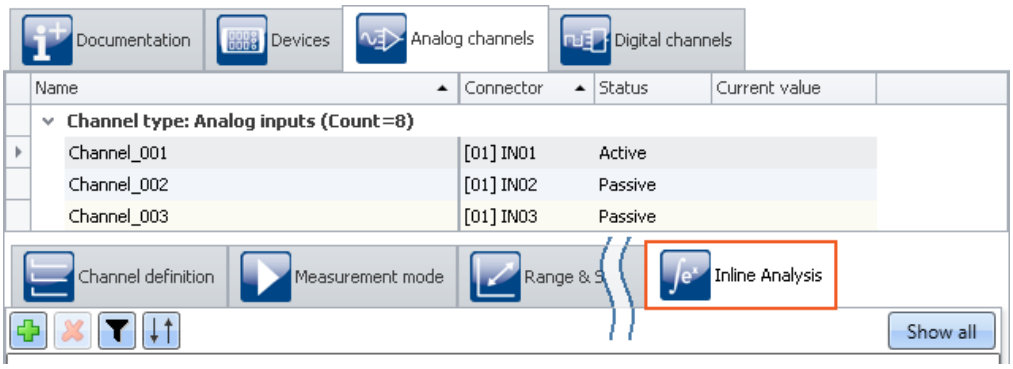
Example

```
OnInitAll
    Thermostat = CtTwoPos( 2, 0 ) ; Hysteresis, invert output
    Thermostat.SetPoint = 20 ; optional: set setpoint
    DigitalOut_01 = 0 ; optional: output in rest state
End

OnSyncTask( 0.1 )
    DigitalOut_01 = Thermostat.Calc( pv.Temperature_01 ) ; set setpoint
    ...
    If VirtBit_01 <> 0
        Thermostat.SetPoint = 22.0 ; change setpoint
    End
End
```

11.3 Inline Analysis

By using the following dialog, it is possible to compute a derivative or calculated channel from a input channel.

Setup page: "*Analog channels*"
Dialog: "*Inline Analysis*"



Note Which Inline Functions are visible

Which Inline Functions are available depends on the product configuration and licensing.
(see "*Product Configuration / Licensing* 29" or Technical Data Sheet)

11.3.1 Inline Functions - operation

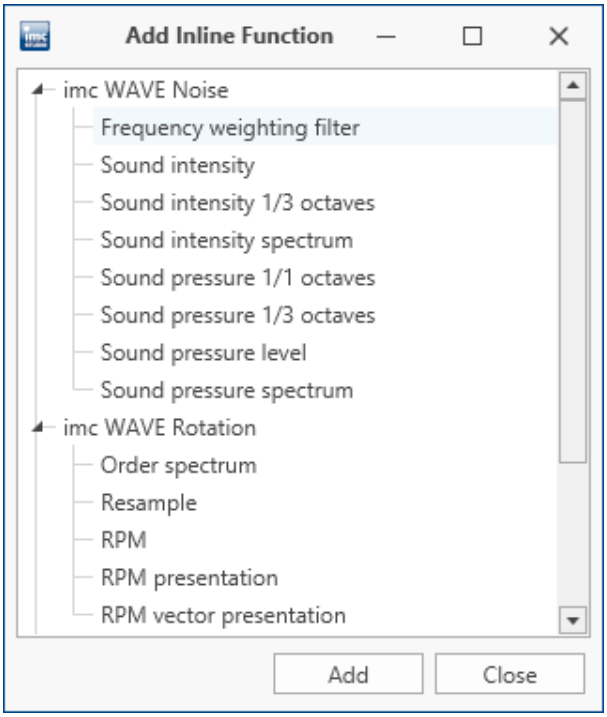
Display in the dialog

The dialog always displays all Inline Functions pertaining to the channels selected. If any selected channel has no Inline Functions, that area is empty. Clicking on the button "*Show all*" opens a tool window in which all Inline Functions for all channels are displayed.

Note Applying changes

Changes are only applied once you perform the action "*Process configuration* 190" (✔). This involves adding and deleting Inline Function as well as modifying the parameters. If the last device has been removed, then configuration processing is no longer possible, which means that the functions can no longer be cleared away.

Adding Inline Functions



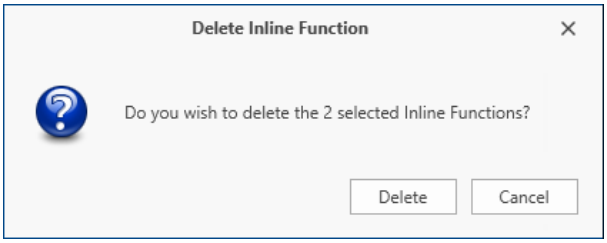
Adding new Inline Functions

- Select the channel from the table of channels or multiple channels if you wish to add a Inline Function to multiple channels at the same time.
- Click on the plus-sign ✚. A dialog appears in which all possible Inline Functions for the channel are presented.
- Select a Inline Function and click on "*Add*". The Inline Function is added to the list. Alternatively, you can add the Inline Function by double-clicking it.

In this way you can add more Inline Functions, or close the dialog.

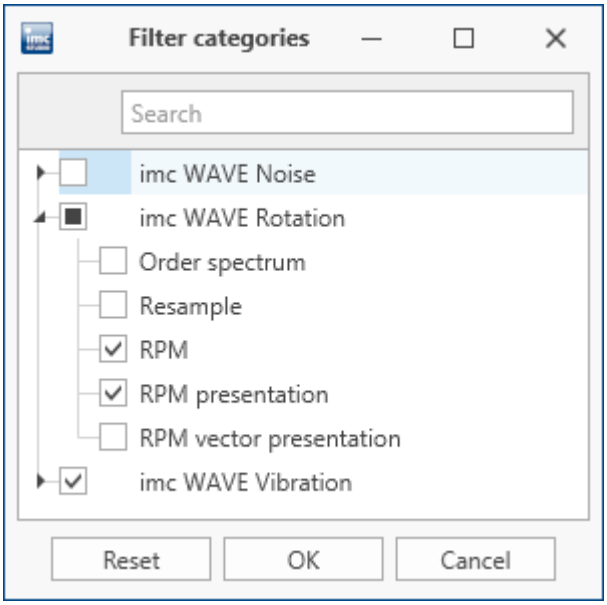
Deleting Inline Functions

- Select the derived channels to be deleted.
- Click on the X-symbol ( ). Ensure that the amount of Inline Functions to be deleted matches your intentions:



Deleting Inline Functions

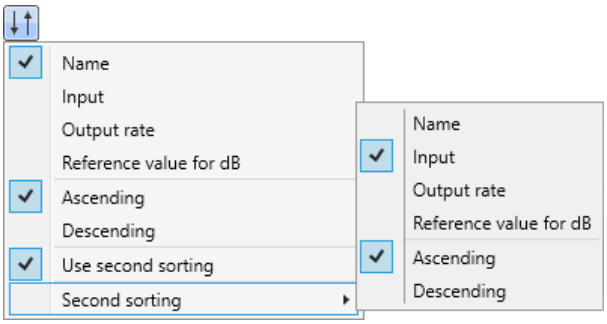
Filtering Inline Functions



Filtering Inline Functions

- Large numbers of Inline Functions and large channel counts can cause the list to become unwieldy. For better ease of managing, , you can filter the view:
- Click on the Filter-symbol ( ). In the dialog, select which categories to include.

Sorting Inline Functions



Sorting Inline Functions

- Within a single Function group, you can sort the entries in ether ascending or descending order according to the name, input, output rate or reference value.
- Click on the Sorting symbol ( ). By means of the "*Second sorting*", you can refine the sorting order to the next level.

Parameters - Modifying the Inline Function

You can modify the parameters for the Inline Function. Depending on the type of Inline Function, a variety of input boxes or selection lists are available for this purpose.

Parameter	Description
Name	The parameter: "*Name*" defines the output-variable's name. This appears in the Data Browser. Once you set up a Inline Function, the "*Name*" reflects the name of the input channel (e.g. "*Channel_001_Leq_01*"). But when you subsequently change the input channel, the "*Name*" does not also change.
Input	The parameter: "*Input*" defines the input channel on which the calculation is performed. This, too, can be changed subsequently. Along with the inputs captured by the device (analog-, incremental counter-, 3PDI-), calculated channels such as imc Online FAMOS can be used. The result of some Inline Function calculations can be used as input for another calculation (first level). However, a further calculation (second level) is not supported. Some inline functions have additional input channels, e.g. when determining the sound intensity from two microphone signals, the order time plot of the RPMs is also required for the order tracking analysis.

Additional parameters may also be available depending on the Inline Function type.

11.3.2 Introduction to the imc WAVE functions

The Inline Analyses "imc WAVE Noise 1068", "imc WAVE Vibration 1078" and "imc WAVE Rotation 1074" offer numerous capabilities for spectral examinations in the fields of **acoustics**, **vibration analysis** and **order-tracking analysis**. What this means is taking signals measured over the time domain, typically generated by microphones and accelerometers, and performing spectral frequency analyses, such as Fast Fourier Transformation or 1/3- and octave spectrum analysis, on them in real time. For this purpose, a comprehensive variety of settings options are available in order to allow parameterization of the analyses according to established standards. This includes time- and frequency-weighting with standard procedures and filters, window functions, etc.

The Inline Analysis "imc WAVE Structure 1084" offers functions for 2 channel **spectrum analysis**, e.g. frequency response function or coherence.

Note Calibration

If needed, perform calibration of the microphone: See "*Calibration of IEPE sensors* 202"

11.3.3 imc WAVE Noise

Analyzer functionalities

Functions for noise analysis, e.g. sound pressure level, Leq, sound intensity, FFT-spectra, 1/3-octave and 1/1-octave spectrum.

- Sound level meter according to Standard IEC 61672-1
- 1/3- and 1/1-octave analysis as 3D or averaged according to Standard IEC 61260-1
- FFT analysis as 3D or averaged
- Determining the sound intensity (time plot, narrow band spectrum and 1/3-octave spectrum) for measurements with an intensity probe

Functions	Description
Basic analysis functions	• FFT analysis (up to 131072 data points) as 3D or averaged • simultaneous calculation of octaves and 1/3-octaves in real time • calculation of spectra with constant absolute and relative bandwidth • selectable frequency range for the spectrum analysis • A-, B-, C- and linear frequency weighting selectable • time-weighting: Fast, Slow • window functions: rectangular, Hamming, Hanning, Flattop, Blackman and Blackman-Harris • frequency -weighted sound level with exponential time weighting • calculation of frequency-weighted averaging levels

Typical applications

- acoustics measurements, fully conformant to applicable industry standards
- acceptance and certification measurements with respect to noise emission and sound levels
- product qualification tests
- product optimization in the R&D field
- in-vehicle noise levels: tests for quality assessment and optimization
- comprehensive and wholistic investigations: root cause analysis of origins, propagation paths and effects of noise and vibration
- incorporation of acoustics expertise into general application fields of physical measurement engineering

Calculations

Z, A, B, C are standards-compliant designations for the frequency weighting (frequency filtering); standardized (acoustic) filters with specific frequency responses. "L" stands for (sound-) level.

Calculation	Description
Frequency weighting filter	Weighted time signal • **Frequency-weighting**: Z (without) or A, B or C. The result is also a time-domain signal.

Calculation	Description

Sound intensity level	Determination of sound intensity by means of a sound intensity probe according to the two-microphone technique.
	• **Frequency-weighting**: Z (without) or A, B or C
	• **Distance (spacer)**: distance between microphones
	• **Density**: air density
	• **Averaging**:
	  • *Interval*: The averaging is performed over the defined output interval (1 / output rate) as the arithmetic mean of the intensity plot.
	  • *from start*: The averaging is performed over the entire plot over time, from the start of the measurement. The last value is the arithmetical mean of the plot of the intensity up until this point in time.
	• **Output rate**: for the averaging type "Interval", the averaging interval = 1 / output rate.

Sound intensity 1/3 octaves	Determination of sound intensity as 1/3-octaves by means of measurement by a sound intensity probe according to the two-microphone technique.
	• **Frequency-weighting**: Z (without) or A, B or C
	• **Distance (Spacer)**: distance between microphones, length of spacer
	• **Density**: air density
	• **Averaging**:
	  • *Interval*: The averaging is performed over the defined output interval (1 / output rate) as the arithmetic mean of each 1/3-octave band.
	  • *from start*: The averaging is performed over all 1/3-octave bands from the start of the measurement as the arithmetic mean of each 1/3-octave band.
	• **Lower/upper center frequencies**: center frequencies for 1/3-octaves
	• **Output rate**: for the averaging type "Interval", the averaging interval = 1 / output rate.

Sound intensity spectrum (FFT)	Determination of the sound intensity as a narrow band spectrum by means of measurement with a sound intensity probe according to the two-microphone technique.
	• **Frequency-weighting**: Z (without) or A, B or C
	• **Distance (spacer)**: Distance between microphones, length of spacer
	• **Density**: air density
	• **Averaging**:
	  • *none*: no averaging
	  • *Count*: Instead of *none* and *from start*, a number can be entered. The averaging is performed over the specified count of spectra as the arithmetic mean of the intensity spectrum for each frequency line.
	  • *from start*: The averaging is performed over all spectra from the start of the measurement, as the arithmetic mean for each frequency line.
	• **Samples** (number): Count of input signal samples used to determine a spectrum. Sample value counts which are not powers of two are allowed.
	• **Window** (function): Rectangle, Hamming, Hanning, Blackman, Blackman-Harris, Flattop
	• **Overlap**: 0%, 10%, 25%, 33.33%, 50%, 66.67%, 75%, 90%
	• **Log. axis**: Generates an X-axis scaled for display with 1/3-octaves.

Calculation	Description
Sound pressure 1/1 octaves	Leq or LeqT 1/1-octaves: time-averaged sound level, equivalent continuous sound level over the measurement duration or with time interval T • for **Time-weighting**: "*Leq from start*" or "*Leq in interval*" • **Frequency-weighting**: Z (without) or A, B or C Lτ 1/1-octaves: time weighted sound level • for **Time-weighting**: F (Fast), S (Slow), I (Impulse) or Peak • **Frequency-weighting**: Z (without) or A, B or C
Sound pressure 1/3 octaves	Leq or LeqT 1/3-octaves: time-averaged sound level, equivalent continuous sound level over the measurement duration or with time interval T • for **Time-weighting**: "*Leq from start*" or "*Leq in interval*" • **Frequency-weighting**: Z (without) or A, B or C Lτ 1/3-octaves: time weighted sound level • for **Time-weighting**: F (Fast), S (Slow), I (Impulse) or Peak • **Frequency-weighting**: Z (without) or A, B or C
Sound pressure level	Mean level, equivalent continuous sound level of the frequency-weighted sound pressure signal over the measurement duration or with time interval T • for **Time-weighting**: "*Leq from start*" or "*Leq in interval*" • **Frequency-weighting**: Z (without) or A, B or C Lτ and Lτ max: frequency- and time-weighted sound level • for **Time-weighting**: F (Fast), S (Slow), I (Impulse) or Peak • **Frequency-weighting**: Z (without) or A, B or C • **Maximum**: Maximum of Lτ

Calculation	Description

Sound pressure spectrum (FFT)	Fast Fourier Transform: Amplitude spectrum
	• **Frequency-weighting**: Z (without) or A, B or C
	• **Result**: Output of the results values as:
	  • *RMS*: harmonic as RMS-value
	  • *RMS²*: harmonic as squared RMS-value
	  • *Peak*: harmonic as peak value/amplitude
	• **Averaging:**
	  • *none*: no averaging
	  • *Number*: averaging performed according to the "*Avg.-type*" selected, over the specified count of spectra
	  • *from start*: averaging performed according to the "*Avg.-type*" selected, over all spectra as of the measurement start
	• **Avg.-type**: RMS (root mean square), Arithmetic Mean, Maximum (of all spectra), Minimum (of all spectra)
	• **Samples** (number): Count of input signal samples used to determine a spectrum. Sample value counts which are not powers of two are allowed.
	• **Window** (function): Rectangle, Hamming, Hanning, Blackman, Blackman-Harris, Flattop
	• **Overlap**: 0%, 10%, 25%, 33.33%, 50%, 66.67%, 75%, 90%
	• **Log. axis**: Creates an X-axis for the representation scaled with thirds.
	• **Diff./Int.**:
	  • *without*: The time signal's amplitude spectrum is generated **without** differentiation or integration.
	  • *differentiation*: The derivative of the amplitude spectrum is found. This is performed by **multiplication** of the spectrum by the angular frequency ω (2*π*f).
	  • *double differentiation*: The second derivative of the amplitude spectrum is found. This is performed by **multiplication** of the spectrum by the **square** of the angular frequency ω (2*π*f).
	  • *integration*: The integral of the amplitude spectrum is found. This is performed by **division** of the spectrum by the angular frequency ω (2*π*f).
	  • *double integration*: The second integral of the amplitude spectrum is found. This is performed by **division** of the spectrum by **square** of the angular frequency ω (2*π*f).
	Note: In all derivation and integration procedures, the result for the DC component (frequency: 0 Hz) is 0 and thus has no significance.

Notes on Sound intensity

Sound intensity is determined by a two-microphone-based technique. For this purpose, an intensity probe is used in which two microphones are mounted at a small and fixed distance from each other.

The course of the sound pressure over time at each microphone is provided as a signal.

The calculation is performed as the multiplicative product of the sound pressure and the sound particle velocity.

The sound pressure used is the average of the sound pressures at the two microphones.

The sound particle velocity is determined by integration of the pressure differential. The pressure differential is an approximation of the derivative of the sound pressure appearing in the Euler equation.

To prevent the calculated integral from drifting away due to initial values and (small) offset errors, and thereby (strongly) distorting the result, a high-pass filter is applied.

For the purpose of the output, an average is taken, for which the parameter Time Weighting sets the averaging region.

All result values are intensities and stated in W/m^2 if the input data are scaled in Pa over s.

Sign

The calculated sound intensity is an average with an arithmetical sign.

The sign expresses the direction.

The sign is positive when the energy first hits the first microphone and next the second. This means it is positive when Microphone 1 is aimed at the source.

Even when the result is expressed in dB, the sign is included in the result. In order for this to be possible and remain clearly interpretable, the unsigned dB-result is limited to 0 dB. Additionally, for each of the three calculations, an extra sign-channel is generated.

- For the "overall" sound intensity, a time-domain channel with the current sign.
- For the 1/3-octave sound intensity, a segmented channel with a sign for each 1/3-octave.
- For the sound intensity spectrum, a segmented channel with a sign for each frequency reference point.

Limitations of the procedure

The microphone distance determines what frequency range is possible for the analysis:

When frequency components become very low, they are no longer possible to evaluate precisely because their phase differential is either very difficult or impossible to determine.

When frequency components become very high, they are no longer possible to evaluate precisely because their wavelength approaches the size of the distance between microphones.

The approximation of the derivative by a pressure differential, mentioned above, only works well when the frequency components are significantly less than 1/4 of the sampling frequency.

1/3-octave-based analysis

The calculation is performed for each 1/3-octave: First the 1/3-octave filtering is performed, next the calculation of the sound intensity.

The two frequency boundaries f1 and f2 should be specified as 1/3-octave midband frequencies, e.g. f1 = 8 Hz and f2 = 12500 Hz. f1 < f2. The top 1/3-octave along with its frequency band must lie completely within one-half of the sampling frequency.

The top 1/3-octave should lie clearly within 1/4 of the sampling frequency.

The duration of transient oscillations for the 1kHz 1/3-octave is assumed to be 20ms. This duration is inversely proportional to the 1/3-octave frequency. For very low 1/3-octaves, this duration becomes substantial. The duration of the measurement must be set accordingly.

The result is a segmented data set. Each segment contains one 1/3-octave spectrum. The result's x-coordinate counts the 1/3-octaves (just like the FAMOS-function OctA). To achieve a useful display in the curve window, 1/3-octave labeling should be selected there.

The z-coordinate of the result represents the time. The time difference between the segments, delta-z, corresponds to the parameter "Output-Interval".

The 1/3-octave filter and weightings are in accordance with IEC 651 (sound level meter), DIN 45652 (1/3-octave filter for electroacoustic measurements) and EN61260-1:2014 or IEC61260-1:2014 (band filter for octaves and fractions of octaves, filter Class 1).

In addition to the signal transients due to the 1/3-octave filter, the high-pass filter used also causes transient oscillations.

Total sound intensity

The input signals sound intensity is determined. There is no 1/3-octave filtering. The cutoff frequency of the high-pass filter used internally is 50 Hz.

The calculation is possible either without frequency weighting (Z) or with A, B or C weighting.

The signal transients are determined by the high-pass filter. Their duration is inversely proportional to the filter's cutoff frequency.

The result is a time-domain channel. The output interval specified is its sampling interval.

Sound intensity spectrum

The sound intensity spectrum is calculated via the cross-power spectrum from the complex spectra of the input signals. The integration is performed over the frequency domain.

Notes

The elapse time between the two microphones is of decisive importance for the analysis. All filters and pre-processing applied to the microphone signals, which affect the signal phase, must always proceed identically for both channels.

The transient oscillation procedure depends on what filters are used.

During the transient oscillation procedure, the intensities are assumed to be 0.0.

11.3.4 imc WAVE Rotation

Analyzer functionalities

Functions for analysis of rotating machinery

- Resampling of time-domain channels
- Order-tracking analysis as a 3D or averaged
- FFT analysis as a 3D or averaged
- Calculation of an input signal over time to a channel over RPMs (RPM-presentation)
- Calculation of spectrum over time or revolutions, to a 3D-channel over RPMs (RPM-vector-presentation)
- Calculation of 3D cross sections

Functions	Description
Basic analysis functions	• FFT analysis (up to 131072 data points) as 3D or averaged • Calculation of order-tracking spectra • Determining of RPMs, and Resample-functions • Window functions: rectangular, Hamming, Hanning, Flattop, Blackman and Blackman-Harris • Class-counting and representation of time data and spectra plotted over the RPMs

Typical applications

- product qualification tests
- product optimization in the R&D field
- End-Of-Line Quality Assurance
- comprehensive investigations of rotating machinery

Calculations

Calculation	Description

Frequency spectrum (FFT)	Fast Fourier Transform: Amplitude spectrum
	• **Result**: Output of the results values as:
	  • *RMS*: harmonic as RMS-value
	  • *RMS*^{2}: harmonic as squared RMS-value
	  • *Peak*: harmonic as peak value/amplitude
	• **Averaging:**
	  • *none*: no averaging
	  • *Number*: Averaging is performed according to the selected "*Avg.-type*" over the specified count of spectra
	  • *from start*: Averaging is performed according to the selected "*Avg.-type*" over all spectra as of the measurement start
	• **Avg.-type**: RMS (root mean square), Arithmetic Mean, Maximum (of all spectra), Minimum (of all spectra)
	• **Samples** (number): Count of input signal samples used to determine a spectrum. Sample value counts which are not powers of two are allowed.
	• **Window** (function): Rectangle, Hamming, Hanning, Blackman, Blackman-Harris, Flattop
	• **Overlap**: 0%, 10%, 25%, 33.33%, 50%, 66.67%, 75%, 90%
	• **Log. axis**: Creates an X-axis for the representation scaled with thirds.
	• **Diff./Int.**:
	  • *without*: The time signal's amplitude spectrum is generated **without** differentiation or integration.
	  • *differentiation*: The derivative of the amplitude spectrum is found. This is performed by **multiplication** of the spectrum by the angular frequency ω (2*π*f).
	  • *double differentiation*: The second derivative of the amplitude spectrum is found. This is performed by **multiplication** of the spectrum by the **square** of the angular frequency ω (2*π*f).
	  • *integration*: The integral of the amplitude spectrum is found. This is performed by **division** of the spectrum by the angular frequency ω (2*π*f).
	  • *double integration*: The second integral of the amplitude spectrum is found. This is performed by **division** of the spectrum by **square** of the angular frequency ω (2*π*f).
	Note: In all derivation and integration procedures, the result for the DC component (frequency: 0 Hz) is 0 and thus has no significance..

RPM	Determining the RPMs from, for example, squarewave, sinusoidal or incremental encoder signals, with the measurement mode "Impulse Time" • **Schmitt trigger**: upper and lower threshold • **Counter**: encoder, impulse time, sampled squarewave signal, sinusoidal signal, 1 missing tooth, 2 missing teeth • **Counter pulses**: tooth/impulse count, including "missing teeth" • **without zero pulse**: When this selection is deactivated, RPM-values are only determined as of the first impulse (before this, the RPM-value = 0 RPM). • **Minimum RPM**: bottom boundary of RPM-range

Calculation	Description
RPM presentation	The plot of a signal over the RPM-values is generated from the signal's time plot and the RPM-value's time plot. The desired RPM-range is subdivided among classes of equal width. • **RPM**: Time plot of RPMs • **Averaging**: RMS-value, arithmetic mean, minimum, maximum Specifies how values within the same RPM-class are subjected to calculations. • **Interpolation**: Tooth/impulse count, including "missing teeth" • **Minimum RPM**: bottom boundary of RPM-range • **Maximum RPM**: top boundary of RPM-range • **RPM class width**: width of one RPM-class
RPM vector presentation	The plot of a spectrum over the RPM-values is generated from the spectrum's time plot and the RPM-value's time plot. The desired RPM-range is subdivided among classes of equal width. • **RPM**: Time plot of RPMs • **Averaging**: Arithmetic mean, minimum, maximum, RMS-value Specifies how values within the same RPM-class are subjected to calculations. • **Interpolation**: Tooth/impulse count, including "missing teeth" • **Minimum RPM**: bottom boundary of RPM-range • **Maximum RPM**: top boundary of RPM-range • **RPM class width**: width of one RPM-class
Order spectrum (FFT)	Determination of the order spectrum • **RPM**: channel containing the RPM information for purpose of re-sampling • **Filter type**: low-pass (fixed) - Type of RPM-tracking filter serving as the anti-aliasing filter for resampling • **Characteristic**: Butterworth (fixed) • **Order**: integer between 1 and 10; filter order of the tracking filter • **3dB order**: integer ≤ maximum order; order (line), at which the tracking low-pass filter dampens by 3dB • **Maximum RPM**: top boundary of desired RPM-range • **Window** (function): Rectangle, Hamming, Hanning, Blackman, Blackman-Harris, Flattop • **Overlap**: 0%, 10%, 25%, 33,33%, 50%, 66,67%, 75%, 90% • **Maximum order**: the highest order (line) displayed in the order spectrum • **Samples** (number): Count of sampled values of the resampled input signal used to determine a spectrum. Sample counts which are not powers of two are allowed. Derived as 2 * "*Maximum Order*" divided by "*Resolution*" • **Resolution**: distance between lines in the order spectrum

Calculation	Description
Resample	Sampling of the input channel over the angle. The X-unit of the result is "revolutions". The values in the result are to be interpreted as RMS-values. • **RPM**: channel containing the RPM information for purpose of re-sampling • **Filter type**: low-pass (fixed) - Type of RPM-tracking filter serving as the anti-aliasing filter for resampling • **Characteristic**: Butterworth (fixed) • **Order**: integer between 1 and 10; filter order of the tracking filter • **3dB order**: integer ≤ maximum order; order (line), at which the tracking low-pass filter dampens by 3dB • **Maximum RPM**: top boundary of desired RPM-range • **Window** (function): Rectangle, Hamming, Hanning, Blackman, Blackman-Harris, Flattop • **Maximum order**: the highest order contained in the sampled signal
Spectrum cut	By means of this function, it is possible to make a vertical cut in the spectrum at a given X-position. Thus for example, it is possible to determine the signal plot in the z-direction at a specific frequency or a sampling point. The input can be a frequency spectrum or an order tracking spectrum, as well as 1/3-octave and octave spectra. • **X position**: Position of the vertical cross section; or for a frequency spectrum, a frequency in Hz; or for an order spectrum, an order. In the case of a 1/3-octave or octave spectrum, the desired center frequency is selected.

11.3.5 imc WAVE Vibration

Analyzer functionalities

Functions for vibration analysis of human vibration and machine diagnostics

- Filters (LP, HP, BP, BS, simple/double integration, simple/double differentiation, envelope curve)
- Machine diagnostics according to Standard ISO 10816 and ISO 20816
- Human vibration filters according to Standard ISO 2631-1, ISO 8041, DIN EN 12299
- 1/3- and 1/1-octave analysis of vibration as 3D or averaged according to Standard IEC 61260-1
- FFT analysis as 3D or averaged

Functions	Description
Basic analysis functions	• FFT analysis (up to 131072 data points) as 3D or averaged • simultaneous calculation of octaves and 1/3-octaves in real time • calculation of spectra with constant absolute and relative bandwidth • selectable frequency range for the spectrum analysis • Human vibration filter, high-, low-, bandpass and band rejection • time-weighting: Fast, Slow • window functions: rectangular, Hamming, Hanning, Flattop, Blackman and Blackman-Harris • frequency -weighted sound level with exponential time weighting

Typical applications

- standards-conformant measurements for human vibration and machine diagnostics
- acceptance and certification measurements in conjunction with impact of vibration on humans
- in-vehicle seating comfort: Measurements for qualification and optimization
- hand-arm vibration: handheld machinery
- acceptance and certification measurements in conjunction with impact of vibration on machinery
- product qualification tests
- product optimization in the R&D field
- comprehensive and wholistic investigations: root cause analysis of origins, propagation paths and effects of vibration

Calculations

"L" stands for (vibration-) level.

Calculation	Description

Filter	• **Filter type**: without, Low-pass filter, High-pass filter, Band-pass filter, Band-stop filter
	• **Characteristic**: Butterworth, Bessel, Chebychev, Critical damping
	• **Order**: integer <= 100
	• **Cutoff frequency**: up to one-half of the sampling rate
	• **Diff./Int.**:
	  • *without*: Filtering performed without subsequent differentiation or integration of the time-domain signal.
	  • *differentiation*: The filtered signal is differentiated in the time domain.
	  • *double differentiation*: The filtered signal is differentiated twice in the time domain.
	  • *integration*: The filtered signal is integrated in the time domain.
	  • *double integration*: The filtered signal is integrated twice in the time domain.
	  • *envelope*: The absolute value of the filtered signal is determined.

Filter rms	The filtering is performed in accordance with a specified frequency-weighting. Subsequently, calculation of the moving exponential RMS value (time weighting) is performed on the results. By means of re-sampling, the result can be reduced by an integer factor which is determined by the output rate. • **Frequency-weighting**: without or see "*Possible filters* 1082" • **Time-weighting**: F (Fast = 0.125 s) or S (Slow = 1 s), I (Impulse), Peak, "*Leq from start*" or "*Leq in interval*" • **Averaging time interval**: Duration of interval with "*Leq in interval*" time weighting • **Maximum**: Determines the maximum among all values since the measurement start (Peak-Hold-Max) • **Characteristic**: For Frequency-weighting LP, HP, BP, BS - See "Filter" • **Order**: For Frequency-weighting LP, HP, BP, BS - See "Filter" • **Cutoff frequency**: For Frequency-weighting LP, HP, BP, BS - See "Filter" • **Diff./Int.**: See "Filter" • **Output rate**: same as input channel (maximum), or reduced by integer factor

Calculation	Description

Machine diagnosis	In accordance with ISO 10816/20816 - The function has two results channels.
	1. Result (with specified name) in accordance with the settings for the filtering, time weighting, etc.
	2. "Alert"-channel, containing the zone values (A = 0, B = 1, C = 2, D = 3), according to which zone the result (1.) lies in
	• **Filter type**: without, Low-pass filter, High-pass filter, Band-pass filter, Band-stop filter
	• **Characteristic**: Butterworth, Bessel, Chebychev, Critical damping
	• **Order**: integer <= 100
	• **(Lower/Upper) cutoff frequency**: up to half the sampling rate
	• **Time-weighting**:
	  • *RMS from start*: RMS (root mean square) values from start of measurement
	  • *RMS in interval*: RMS (root mean square) values in the interval
	  • *Peak value absolute in interval*: absolute value of the amplitude
	  • *Peak value positive in interval*: absolute value of the positive values
	  • *Peak value negative in interval*: absolute value of the negative values
	  • *Peak-to-peak value from start*: positive - negative values as of start of measurement
	  • *Peak-to-peak value in interval*: positive - negative values within the interval
	  • *Crest factor from start*
	  • *Crest factor in interval*
	  $Crest\ factor = \frac{\lvert Input\ signal \rvert}{Input\ signal_{RMS}}$
	• **Averaging time interval**: Duration of interval with "in interval" time weighting
	• **Maximum**: Determines the maximum among all values since the measurement start (Peak-Hold-Max)
	• **Diff./Int.**:
	  • *without*: The time signal is filtered without subsequent differentiation or integration.
	  • *differentiation*: The derivative of the filtered signal is taken.
	  • *double differentiation*: The second derivative of the filtered signal over the time domain is taken.
	  • *integration*: The integral of the filtered signal over the time domain is taken.
	  • *double integration*: The second integral of the filtered signal over the time domain is taken.
	  • *envelope*: *The absolute value* of the filtered signal is determined.
	• **Zone**: A - very good (Value = 0), B - suitable (Value = 1), C - not suitable (Value = 2), and D - critical (Value = 3)
	Boundary values can be specified for the zones. The results channels values are classified according to these boundary values and allocated to the corresponding zones. The result of the classification is represented in the "Alert"-channel by the values 0 (Zone A) through 3 (Zone D) and thus offers the ability to perform zone monitoring.
	• **Output rate**: max. like input channel, or reduced by integer factor

Calculation	Description
Vibration 1/1 octaves	Leq or LeqT 1/1-octaves: time-averaged vibration level, equivalent continuous vibration level over the measurement duration or with time interval T • for **Time-weighting**: "*Leq from start*" or "*Leq in interval*" • **Frequency-weighting**: without or human vibration filter (e.g. Hx, DIN 45671, ...) Lτ 1/1-octaves: time weighted vibration level • for **Time-weighting**: F (Fast), S (Slow), I (Impulse) or Peak • **Frequency-weighting**: without or human vibration filter (e.g. Hx, DIN 45671, ...)
Vibration 1/3 octaves	Leq or LeqT 1/3-octaves: time-averaged vibration level, equivalent continuous vibration level over the measurement duration or with time interval T • for **Time-weighting**: "*Leq from start*" or "*Leq in interval*" • **Frequency-weighting**: without or human vibration filter (e.g. Hx, DIN 45671, ...) Lτ 1/3-octaves: time weighted vibration level • for **Time-weighting**: F (Fast), S (Slow), I (Impulse) or Peak • **Frequency-weighting**: without or human vibration filter (e.g. Hx, DIN 45671, ...)

Calculation	Description

Vibration spectrum (FFT)	Fast Fourier Transform: Amplitude spectrum
	• **Frequency-weighting**: without or human vibration filter (e.g. Hx, DIN 45671, ...)
	• **Result**: Output of result values as:
	  • *RMS*: harmonic as RMS-value
	  • *RMS²*: harmonic as squared RMS-value
	  • *Peak*: harmonic as peak value/amplitude
	• **Averaging:**
	  • *none*: no averaging
	  • *Number*: Averaging is performed according to the selected "*Avg.-type*" over the specified count of spectra
	  • *from start*: Averaging is performed according to the selected "*Avg.-type*" over all spectra as of the measurement start
	• **Avg.-type**: RMS (root mean square), Arithmetic Mean, Maximum (of all spectra), Minimum (of all spectra)
	• **Samples** (number): Count of input signal samples used to determine a spectrum. Sample value counts which are not powers of two are allowed.
	• **Window** (function): Rectangle, Hamming, Hanning, Blackman, Blackman-Harris, Flattop
	• **Overlap**: 0%, 10%, 25%, 33.33%, 50%, 66.67%, 75%, 90%
	• **Log. axis**: Creates an X-axis for the representation scaled with thirds.
	• **Diff./Int.**:
	  • *without*: The time signal's amplitude spectrum is generated **without** differentiation or integration.
	  • *differentiation*: The derivative of the amplitude spectrum is found. This is performed by **multiplication** of the spectrum by the angular frequency ω (2*π*f).
	  • *double differentiation*: The second derivative of the amplitude spectrum is found. This is performed by **multiplication** of the spectrum by the **square** of the angular frequency ω (2*π*f).
	  • *integration*: The integral of the amplitude spectrum is found. This is performed by **division** of the spectrum by the angular frequency ω (2*π*f).
	  • *double integration*: The second integral of the amplitude spectrum is found. This is performed by **division** of the spectrum by **square** of the angular frequency ω (2*π*f).
	Note: In all derivation and integration procedures, the result for the DC component (frequency: 0 Hz) is 0 and thus has no significance.

Possible filters

- Low-pass, High-pass, Band-pass or Band-stop
- Wk, z direction and for vertical recumbent direction, except head. As per ISO 2631-1:1997
- Wd, x and y directions and for horizontal recumbent direction. As per ISO 2631-1:1997
- Wf, motion sickness. As per ISO 2631-1:1997
- Wc, seat-back measurement. As per ISO 2631-1:1997
- We, measurement of rotational vibration. As per ISO 2631-1:1997
- Wj, vibration under the head of a recumbent person. As per ISO 2631-1:1997
- Hx, whole body vibration, standing, sitting position: x- and y-directions. Recumbent position: y- and z-directions. As per DIN 45671-1:1990-09
- Hz, whole body vibration, standing, sitting position: z-direction. As per DIN 45671-1:1990-09

- Hxl, whole body vibration, recumbent position: x-direction. As per DIN 45671-1:1990-09
- Hb, whole body vibration, body posture not specified. As per DIN 45671-1:1990-09
- Hh, hand-arm-vibration, for all directions. As per DIN 45671-1:1990-09
- hand transmitted vibration, weighting filter. Nach ISO 7505:1986-05
- Weighting factors for transverse (x, y) vibrations. As per withdrawn ISO 2631-1:1985
- Weighting factors for longitudinal (z) vibrations. As per withdrawn ISO 2631-1:1985
- Wb (passenger and crew comfort in fixed-guideway transport systems). As per ISO 2631-4:2001
- Wm (human exposure to vibration in buildings). As per ISO 2631-2:2003
- Acceleration input. Nach ISO 6954:2000
- Velocity input. Nach ISO 6954:2000
- Wh (hand transmitted vibration, weighting filter). As per ISO 5349-1:2001
- Wb (passenger and crew comfort in fixed-guideway transport systems). As per ISO 8041:2005
- Wc (seat-back measurement). As per ISO 8041:2005
- Wd (x and y directions and for horizontal recumbent direction). As per ISO 8041:2005
- We (measurement of rotational vibration). As per ISO 8041:2005
- Wf (Whole body low frequency, motion sickness). As per ISO 8041:2005
- Wh (hand transmitted vibration). As per ISO 8041:2005
- Wj (vibration under the head of a recumbent person). As per ISO 8041:2005
- Wk (z direction and for vertical recumbent direction, except head). As per ISO 8041:2005
- Wm (human exposure to vibration in buildings). As per ISO 8041:2005
- Wb (railway applications, passenger traveling comfort, Z ground, Z seat shell). As per EN 12299:2009
- Wc (railway applications, passenger traveling comfort, X seat backrest). As per EN 12299:2009
- Wd (railway applications, passenger traveling comfort, X ground, Y ground, Y seat shell). As per EN 12299:2009
- Wp (railway applications, passenger traveling comfort, Y ground, phi ground). As per EN 12299:2009

11.3.6 imc WAVE Structure

Analyzer functionalities

Functions for 2-channel analysis in the frequency domain

- Generation of output signals for the purpose of subsequent modal analysis
- Calculation of transfer functions with noisy input- and/or output signals
- Calculation of the coherence as a quality indicator
- Power rating by means of the cross-power-spectrum and spectral power density
- FFT analysis as 3D or averaged

Functions	Description
Basic analysis functions	• FFT analysis (up to 131072 data points) as 3D or averaged • Power density spectrum (up to 131072 points) as 3D or averaged • Coherence (up to 131072 points) averaged • Cross power density (up to 131072 points) spectrum as 3D or averaged • Frequency response (up to 131072 points) function as 3D or averaged

Typical applications

- product qualification tests
- product optimization
- evaluation of signal power and structural behavior

Calculations

Calculation	Description

Frequency spectrum (FFT)	Fast Fourier Transform: Amplitude spectrum
	• **Result**: output of the results values as:
	  • *RMS*: harmonic as RMS-value
	  • *RMS*^{2}: harmonic as squared RMS-value
	  • *Peak*: harmonic as peak/amplitude value
	• **Averaging:**
	  • *none*: no averaging
	  • *Number*: Averaging is performed according to the "*Avg.-type*" selected, over the specified count of spectra
	  • *from start*: Averaging is performed according to the "*Avg.-type*" selected, over all spectra since the measurement start
	• **Avg.-type**: RMS (root mean square), Arithmetic Mean, Maximum (of all spectra), Minimum (of all spectra)
	• **Samples** (number): Count of input signal samples used to determine a spectrum. Sample value counts which are not powers of two are allowed.
	• **Window** (function): Rectangle, Hamming, Hanning, Blackman, Blackman-Harris, Flattop
	• **Overlap**: 0%, 10%, 25%, 33.33%, 50%, 66.67%, 75%, 90%
	• **Log. axis**: Creates an X-axis for the representation scaled with thirds.
	• **Diff./Int.**:
	  • *without*: The time signal's amplitude spectrum is generated **without** differentiation or integration.
	  • *differentiation*: The derivative of the amplitude spectrum is found. This is performed by **multiplication** of the spectrum by the angular frequency ω (2*π*f).
	  • *double differentiation*: The second derivative of the amplitude spectrum is found. This is performed by **multiplication** of the spectrum by the **square** of the angular frequency ω (2*π*f).
	  • *integration*: The integral of the amplitude spectrum is found. This is performed by **division** of the spectrum by the angular frequency ω (2*π*f).
	  • *double integration*: The second integral of the amplitude spectrum is found. This is performed by **division** of the spectrum by **square** of the angular frequency ω (2*π*f).
	Note: In all derivation and integration procedures, the result for the DC component (frequency: 0 Hz) is 0 and thus has no significance..

Coherence	Determines the coherence between the input and the output time plots, over the frequency. • **Input**: time plot of input (1st input signal) • **Output**: time plot of output (2nd input signal) • **Samples** (number): Sample count of the input signals which are used to determine a spectrum. Sample value counts which are not powers of two are allowed. • **Window** (function): Rectangle, Hamming, Hanning, Blackman, Blackman-Harris, Flattop • **Overlap**: 0%, 10%, 25%, 33.33%, 50%, 66.67%, 75%, 90% • **Log. axis**: Generates an X-axis scaled for display with 1/3-octaves.

Calculation	Description

Cross power spectrum	Determines the cross-power spectrum of the input and output signals over the frequency. The function has two results channels:
	1. Amplitude spectrum
	2. Phase spectrum
	• **Input**: time plot of input (1st input signal)
	• **Output**: time plot of output (2nd input signal)
	• **Averaging**:
	  • *none*: no averaging
	  • *from start*: Averaging is performed separately for the Real- and Imaginary parts over all spectra for each frequency line, beginning at the measurement start.
	• **Samples** (number): Sample count of the input signals which are used to determine a spectrum. Sample value counts which are not powers of two are allowed.
	• **Window** (function): Rectangle, Hamming, Hanning, Blackman, Blackman-Harris, Flattop
	• **Overlap**: 0%, 10%, 25%, 33.33%, 50%, 66.67%, 75%, 90%
	• **Log. axis**: Generates an X-axis scaled for display with 1/3-octaves.

Power spectrum density (PSD)	Determines the power density spectrum (normalized power spectrum with frequency resolution) of the input and output signals over the frequency. The function has two results channels:
	1. Amplitude spectrum
	2. Phase spectrum
	• **Input**: time plot of input (1st input signal)
	• **Output**: time plot of output (2nd input signal)
	• **Averaging**:
	  • *none*: no averaging
	  • *from start*: Averaging is performed separately for the Real- and Imaginary parts over all spectra for each frequency line, beginning at the measurement start.
	• **Samples** (number): Sample count of the input signals which are used to determine a spectrum. Sample value counts which are not powers of two are allowed.
	• **Window** (function): Rectangle, Hamming, Hanning, Blackman, Blackman-Harris, Flattop
	• **Overlap**: 0%, 10%, 25%, 33.33%, 50%, 66.67%, 75%, 90%
	• **Log. axis**: Generates an X-axis scaled for display with 1/3-octaves.

Calculation	Description

Frequency response function (FRF)	Determines the transfer function (Frequency response function) from the input time domain signal to the output time domain signal over the frequency. The function has two results channels:
	1. Amplitude spectrum
	2. Phase spectrum
	• **Input**: time plot of input (1st input signal)
	• **Output**: time plot of output (2nd input signal)
	• **Averaging**:
	  • *none*: no averaging
	  • *from start*: Averaging is performed separately for the Real- and Imaginary parts over all spectra for each frequency line, beginning at the measurement start.
	• **Type**:
	  • *H1: Gxy / Gxx*: Measurement with noisy output signal, calculated from the auto- and cross-spectrum
	  • *H2: Gyy / Gyx*: Measurement with noisy input signal, calculated from the auto- and cross-spectrum
	  • *HV: (Gxy / |Gxy|) * sqrt (Gyy / Gxx)*: Geometric mean of H1 and H2. The input and output are noisy
	• **Samples** (number): Sample count of the input signals which are used to determine a spectrum. Sample value counts which are not powers of two are allowed.
	• **Window** (function): Rectangle, Hamming, Hanning, Blackman, Blackman-Harris, Flattop
	• **Overlap**: 0%, 10%, 25%, 33.33%, 50%, 66.67%, 75%, 90%
	• **Log. axis**: Generates an X-axis scaled for display with 1/3-octaves.

11.3.7 Monitoring the overmodulation and under range

Overmodulation of "analog channels"

Monitoring the overmodulation of "analog channels" (violations of the specified measurement range) as per the standard: "*DIN EN 61672-1*".

For purposes of display and evaluation of the overmodulation a variable is provided: "*WAVE_OverloadVariable*". It indicates that one measurement channel is reporting overmodulation but not which channel.

When overmodulation is detected, then for **one second** the variable "*WAVE_OverloadVariable*" takes the value "1". After that one second the system checks whether the overmodulation is still happening.

- if "no", the variable reverts to "0".
- if "yes", the value "1" is retained until there is no more overmodulation. Then the variable reverts to "0".

The following channels are monitored: active analog channels belonging to the firmware groups A and B (imc DEVICES and imc DEVICEcore), which are subjected **to calculations in an Inline Analysis** function. No other channels such as incremental counters or 3rd-party device channels.

Sound pressure level under range (imc WAVE Noise)

Fall of the calculated sound pressure level under range as per the standard: "*DIN_EN_61672*".

In order to determine the level linearity according to the standard, it is important that the under range is observed and evaluated taking into account the measuring range of the input channels. For this purpose, a supplemental variable, "*WAVE_UnderrangeVariable*", is available (in parallel) for all existing sound level calculations in a measurement. This variable indicates that a calculated sound pressure level reports under range, but not which calculation.

If an under range is detected, the variable "*WAVE_UnderrangeVariable*" is set to "1" at least for one second. After elapse of that one second, the system checks whether any calculation of sound pressure level is still reporting under range.

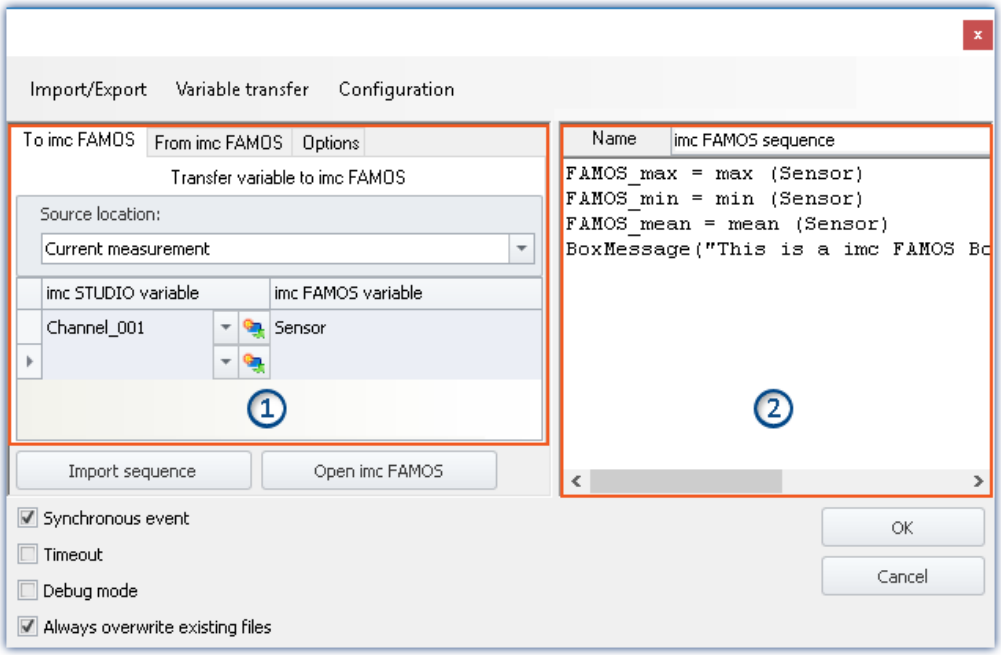
- If "no", the variable is set back to "0".
- If "yes", the value "1" is retained until the level is no longer under range. Then the variable is set back to "0".

Only such sound pressure level calculations are monitored for under range whose input channels originate from devices belonging to firmware groups A or B (imc DEVICES and imc DEVICEcore).

Sound pressure level calculations whose input channels have no measurement range such as 3rd-party device channels are not monitored.

11.4 imc FAMOS dialog

The dialog consists of multiple areas:



imc FAMOS Sequence Editor and transfer table of imc STUDIO variables to imc FAMOS

- Transfer table 1091 (1): **Transfer of the imc STUDIO-variables** is accomplished by means of this table.
- imc FAMOS Sequence Editor (2): Here, the **imc FAMOS sequence in use** is displayed and can be edited.

Procedure

- The **measured variables** set in the Transfer Table 1091 under "*To imc FAMOS*" **are transferred to imc FAMOS**.
- The **calculations** belonging to the sequence are **performed**.
- The **result variables** set in the Transfer Table 1091 under "*From imc FAMOS*" **are transferred to imc STUDIO**.

Editing the imc FAMOS sequence

In order to create a imc FAMOS sequence, there are a variety of techniques:

- By editing in the imc FAMOS **Sequence-Editor**.
- Using the **Start command for imc FAMOS** via the button "*Open imc FAMOS*". (Preferable when not knowing the imc FAMOS functions and their parameters)
- By **importing** an existing **sequence file** using the button "*Import Sequence*".

To edit or test a sequence in imc FAMOS, use the button "*Open imc FAMOS*".

Dialog interface

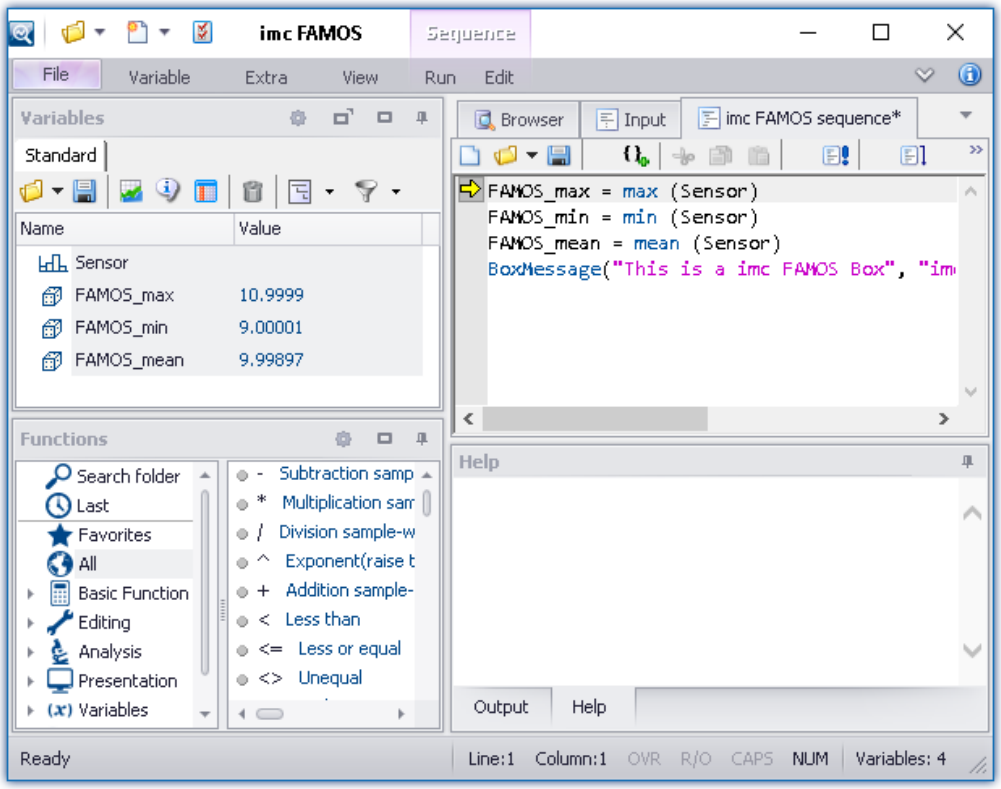
Checkbox	Description
Synchronous event	"*Synchronous event*" means that the **source waits until** imc FAMOS has concluded the sequence. This means: • Sequencer: The next command line is only started when imc FAMOS has concluded the sequence. • Automation: The next step only starts when imc FAMOS has concluded the sequence. If "*Synchronous event*" is deactivated, the new command line or step runs immediately. Meanwhile, the sequence continues to run in parallel. See also the notes on "*Parallel processing of sequences* 1097".
Timeout	The option "*Timeout*" (only available if the dialog is called from a command) only appears if "*Synchronous Event*" is activated. • At the latest, after a specified timeout time, the Sequencer runs the next line.
Debug Mode	If the command is executed while the option "*Debug Mode*" is activated, then imc FAMOS is opened. In this case, the sequence is not run automatically. In Debug Mode, you can **run the sequence step-by-step**, and make changes which are saved in the command. If imc FAMOS is closed, the results are transferred to imc STUDIO, if they are available in the table "*From imc FAMOS*". The command is only concluded as of this moment in time. Please note that in Debug Mode, the command needs longer to execute: • With **the option: "*Synchronized event*" active**: The subsequent commands wait until the imc FAMOS-command has concluded. • With **the option: "*Synchronized event*" deactivated**: Required results may not exist in time since the command is still working. As well, the **option: "*Timeout*"** should not be activated, since it can cause premature closing of the command, although imc FAMOS has not yet been ended.
Always overwrite existing files	If this option is activated, files having the same name in the destination folder will be overwritten without a confirmation prompt. This option is useful for automated routines.

Open imc FAMOS

Alternatively, **imc FAMOS** can be started using the button "*Open imc FAMOS*". In this case, the imc FAMOS Editor is used for making entries. You have the entire scope of the Function Assistant's functionality, help texts, etc., at your disposal.

In imc FAMOS, you can run the sequence directly for testing purposes. To this end, the variables in the table: "*To imc FAMOS*" are automatically set up with the current values. They appear in imc FAMOS's Variables list. If the variables do not yet exist in imc STUDIO at that moment, an empty variable is used in substitute (see note 1091 below).

The results of the test evaluation are not transferred back to imc STUDIO.



imc FAMOS as editor

Save the sequence if you wish to use the changes in imc STUDIO.

Reference

An example of Sequencer working in conjunction with imc FAMOS is found in the Sequencer-Tutorial 1777.

Note

Variable not present

If a variable does not exist in imc STUDIO when it is to be transferred to imc FAMOS, an "empty" variable is created in imc FAMOS as a substitute.

- Data type: Normal data set
- X-Delta: 1
- Total size: 0

Using the imc FAMOS-paths

The default paths set in imc FAMOS also apply to the command.

E.g. "`SEQ MySequence`" is premised on "MySequence.seq" being located in the default sequence folder of the installed imc FAMOS program. Alternatively, it is possible to specify the absolute path, e.g. "`SEQ "D:\SEQ\MySequence.seq"`".

Transferring the files used when exporting to other computers

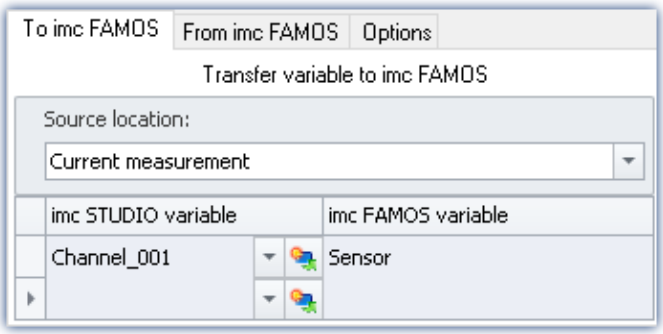
When exporting an experiment, the external files called in the sequence are not automatically included! Copy these files separately.

The exception is the files located in the experiment's subfolder "Meta". There, you are able to save your own files about the experiment, such as your own metadata, imc FAMOS-sequences, info-files, ... (for more information, see the section "*Experiments, projects and the database* 113").

11.4.1 Transfer table

Transfer of the imc STUDIO variables or the imc FAMOS variables is accomplished by means of the transfer table. It is possible to rename the variables. In this way, existing sequences can be adopted directly.

Passing variables to the sequence ("To imc FAMOS")



The variables entered in the column: "*imc STUDIO variable*" are passed to the imc FAMOS sequence. There, they are assigned the name of the associated cell in the column: "*imc FAMOS variable*".

The variable does not need to be already registered in imc STUDIO. In this case, it is created as an "empty" variable in imc FAMOS (see note 1091).

In the example shown, the measured variable "*Channel_001*" is passed to the imc FAMOS sequence as the variable "*Sensor*".

You are able to transfer placeholders

Where is the measurement "x" saved? Which test object is used? Such information can be transferred directly to imc FAMOS by means of placeholders ( ). When the command is run, the placeholder is resolved and transferred.

Source: Selecting the source (measurement)

All variables defined are transferred from the source selected. (not editable for "*imc FAMOS Automation (Datenschneiden)* 1607")

Source	Description
Current measurement	The current measured data are transferred. Note that in this case, only the amount of circular buffer memory set for the curve window display is available for use. This may be only a fraction of the entire measurement.
Last completed measurement	If the measured data are saved, the last measurement saved is loaded automatically. The associated variables are passed to the imc FAMOS sequence. More detailed information on the "Last measurement" is presented in the chapter: "*Storage options and directory structure*" > "*The last measurement* 441"
Measurement number Measurement#<No>	The variables belonging to the measurement with the respective measurement number are passed.

Note — Variables from varying measurements

In some cases, variables from varying measurements need to be transferred to imc FAMOS. Independent of what "Source" is set, you can define in the Variables list whether a variable is to be transferred from a different measurement.

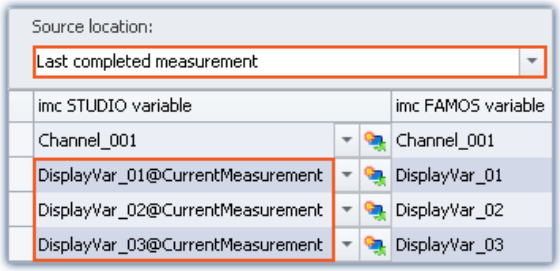
In this case, use the following syntax for the imc STUDIO variable:

Syntax	Example
@<MeasurementName>	<VariableName>@<MeasurementName> e.g. Channel_001@2010-11-15 15-44-03 (1)
@Measurement#<MeasurementNumber>	<VariableName>@Measurement#<MeasurementNumber> e.g. Channel_001@Measurement#1
@LastMeasurement	corresponds to: "*Last completed measurement*"
@CurrentMeasurement	corresponds to: "*Current Measurement*"

For measurement names, an **input support tool** is available: in the input box behind the channel name, click on the keyboard combination: <CTRL> + <SPACE>. You are then provided with a list of various input options. Select one and modify the results if appropriate.

Sample application: Variables from a saved measurement in combination with variables from "*Current Measurement*"

The analysis of measured data often requires the use of additional parameters which are saved with variables under "*Current Measurement*". With this imc FAMOS command, you are able to transfer variables both from a saved measurement and from "*Current Measurement*".

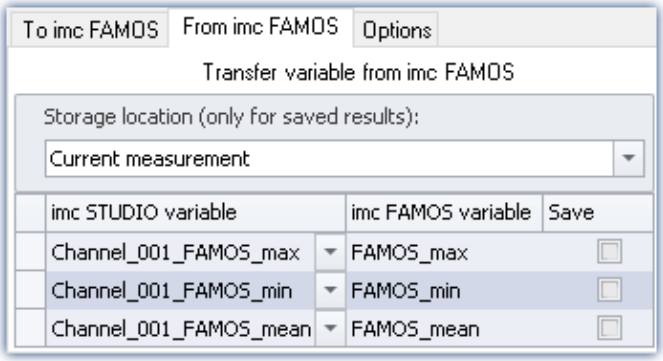


In this example, a particular measurement is selected. Additionally, variables from "*Current Measurement*" are also transferred. In this case, use the following syntax for the imc STUDIO variable:

<VariableName>@CurrentMeasurement

Example: *DisplayVar_01@CurrentMeasurement*

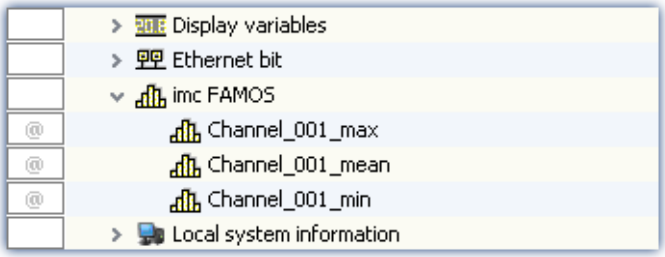
Receiving the sequence's results ("From imc FAMOS")



The variables calculated by imc FAMOS are transferred to imc STUDIO along with the transfer table. On this occasion they can also be renamed.

Note: A channel can not be transferred back to a "*User-defined Variable*" of the type "*Numeric*" (single value). For this purpose, please modify the type in imc FAMOS if you wish to obtain a single value. If you require a channel, use as the target a variable created using imc FAMOS. This will then always be assigned the appropriate data type.

The results appear in the Data Browser:



Storage location without saving: Selection of the target (measurement)

The result appears in the Data Browser, but is not saved.

Storage location	Description
Current measurement	If the target variable already exists (e.g.: a user-defined variable), the result is copied to the existing variable, provided that the variable's type is correct. If the target variable does not yet exist, the result is created under the category "*imc FAMOS*" and is assigned the scope (Validity range): "*Temporary*". The variable's type is selected on the basis of the content.
Last completed measurement	The memory location is ignored. The result is saved under "*Current Measurement*" and treated the exact same way as for the selection "*Current Measurement*"
Measurement#<No>	

Storage location with saving: Selection of the target (measurement)

(Not possible with "*imc FAMOS automation (data cutting)* 1607")

The result appears in the Data Browser and is stored as a.dat-file.

Speicherort	Beschreibung
Last concluded measurement	(recommended) If the option **Save** is selected, use **Last concluded measurement** or **Measurement#<No>**. The result is saved to the folder with the last concluded measurement respectively to the folder of measurement with selected number #<No.>. After loading the measurement, the result is always available. **Please note that at least one channel must be saved on PC. Otherwise the measurement folder does not exist.** If in Data Browser the selected measurement number is assigned to "Current measurement" , the system acts as if "*Current Measurement*" is selected.
Measurement#<No>	
Current measurement	(not recommended) If **Current measurement** is selected as target, no saved measurement is selected. A subfolder reflecting the current time stamp (e.g. "*imcFAMOSResult_2014-07-31 12-50-43*") is set up in the experiment folder for the results.

"Options"

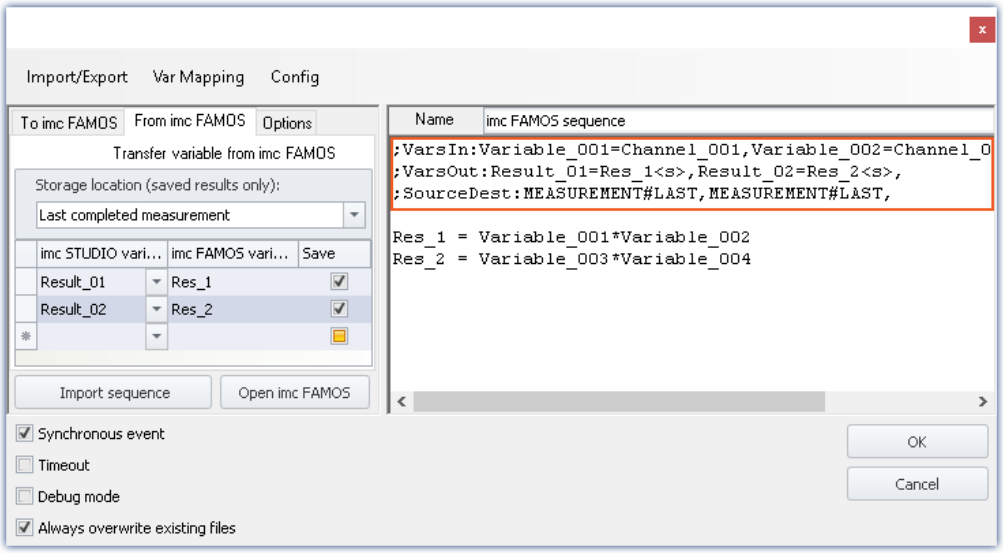
On this page, you can define fixed parameters which are used in the sequence.

Passing (assignment in the Transfer table) of variables can be incorporated in the sequence

Passing (assignment in the Transfer table) of variables can be incorporated in the sequence. Once the assignment is in a sequence, it can easily be passed (copied) to other imc FAMOS-commands.

The assignment is expected/entered as a "Header" in the first three lines of the sequence.

By means of the menu, the assignment can be passed from the transfer table into the sequence, or conversely from the sequence to the table.



Below, the header is set up (with sample names as per the picture):

Example

Variables: To imc FAMOS:

```
;VarsIn:Variable_001=Channel_001,Variable_002=Channel_002,Variable_003=Channel_003,
```

Variables: From imc FAMOS:

```
;VarsOut:Result_01=Res_1<s>,Result_02=Res_2<s>,
```

Source and data storage location:

```
;SourceDest:MEASUREMENT#LAST,MEASUREMENT#LAST,
```

Variables

Description	Variables: To imc FAMOS:	Variables: From imc FAMOS:
Start	;VarsIn:	;VarsOut:
First variable name	Name in imc FAMOS	Name in imc STUDIO
Assignment character	=	=
Second variable name	Name in imc STUDIO	Name in imc FAMOS
Activation of data storage (optional)		<s>
Separator from next assignment	,	,

Source and data storage location:

Description	Syntax
Start	;SourceDest:
First name	Source of the page "To imc FAMOS"
Second name	Storage location of the page "From imc FAMOS"
Separator	,

Possible syntax:

Source or data storage location	Syntax
Last concluded measurement	MEASUREMENT#LAST
Measurement number 3	Measurement#3
Current measurement	empty, so only ","
Fixed measurement name (as in the Data Browser)	2017-02-08 16-42-41 (1)

Example

Examples	Description
;SourceDest:MEASUREMENT#LAST,Measurement#1,	Source: Last concluded measurement Data storage location: Measurement number 1
;SourceDest:MEASUREMENT#LAST,2017-02-08 16-42-41 (1),	Source: Last concluded measurement Data storage location: Measurement with the name 2017-02-08 16-42-41 (1)
;SourceDest:,MEASUREMENT#LAST,	Source: Current measurement Data storage location: Last concluded measurement

11.4.2 Information and tips

Parallel processing of sequences

Only one sequence can ever be processed at the same time. If multiple sequences are started in parallel, they are processed one at a time in succession. This can cause "congestion" if there are very many requests.

The data for the sequence (which are valid at its starting time) are saved temporarily one single time and are available at the start of the analysis.

Closing imc STUDIO

Upon closing imc STUDIO, the sequence currently running is not interrupted. However, any return values will not be processed. Any additional sequences in the queue for processing are canceled. A corresponding notification is posted and you have a choice whether to cancel closing the program.

imc FAMOS automation (data cutting)

This note only pertains to the "*imc FAMOS automation (data cutting)*" and not to the command.

Data cutting and execution of the subsequent imc FAMOS-sequence are separated.

- For each imc FAMOS-cutting interaction there is a separate mechanism cutting the measured data. This makes it possible for multiple parallel cutting interactions to be performed, or very many such interactions to be performed in rapid succession. It is even possible for cutting interaction to be performed while the associated imc FAMOS-sequence for the previous cut is still running. On this topic, please observe the note above on "*Parallel processing of sequences*"
- The data which have been cut are temporarily saved on the hard drive. Note: When there are very many channels, this can cause reduce performance due to pressure on system resources.

Breakpoints and Bookmarks

If the imc FAMOS-sequence contains any defined breakpoints or bookmarks, these are stated along with the sequence. When you close imc FAMOS, comments on the bookmarks/breakpoints appear in the imc FAMOS Sequence-Editor (on the right-hand side).

The following is an example

```
;@# imc FAMOS Sequence
;@# VERSION:     V4.0
;@# DESCRIPTION:
;@# BREAKPOINTS: 20 94
;@# BOOKMARKS:   87 0 0 112 0 0 0 0 0 0
```

Here, breakpoints have been defined for lines 20 and 94, and bookmarks for lines 87 (Bookmark 0) and 112 (Bookmark 3). Upon opening imc FAMOS, comment lines are recognized and removed. Additionally, the corresponding lines are supplied with bookmarks and breakpoints accordingly.

Note Error source

When changes are not made in imc FAMOS, but are instead made directly via the imc FAMOS Sequence-Editor (on the right-hand side), the line numbers do not automatically change accordingly. Even if, for instance, the transfer of variables 1095 (assignment in the transfer table) is saved with the sequence.

In this case, the data points need to be corrected manually.

To the extent possible, only make any changes using imc FAMOS whenever using any breakpoints/bookmarks in the sequence.

11.5 Data Processing (general)

imc STUDIO DataProcessing is a imc STUDIO plug-in by means of which it is possible during a running measurement to outsource various data processing and computational tasks, which previously were performed by the measurement device, to the PC. This means that the measurement device's computational resources are freed up, and the high computing power of the PC is exploited.

A number of functions packages for Data Processing are available:

Functions package	Description
imc Inline FAMOS 851	Processing and analysis of measured data during running measurement
Inline Analysis 1064	Analysis of measured data during running measurement. It is possible to compute a derivative or calculated channel from a input channel. For this purpose, a variety of functions packages are available.
imc STUDIO BusDecoder 1101	Expansion package for decoding Fieldbus log channels
Powertrain Monitoring 1106	The component imc STUDIO Powertrain Monitoring is developed for the diagnosis of powertrains. The powertrains can consist of motors, shift gearboxes and engines as well as devices for braking. The diagnosis can be used in field scenarios, test plants or end of line tests in a production.

Opening Data Processing

The plug-in can be opened via the Navigation Pane: .

Tasks

For each activated functions package, a **Task**, represented by a card tab, is opened. In consequence, you can jump from one functions package to another. All Tasks are processed in parallel. How many tasks can be performed depends on the demands made by the functions used and on the PC's capacities.

For each functions package, you can set up additional tasks. Here again, the amount possible depends on the capacities of the PC used.

Configuring results channels in the Setup

For quick and clearly organized configuration of the results channels belonging to Data Processing-tasks, all results channels appear in the channel table in the main window: Setup. Here, the channels can be configured like the virtual channels belonging to imc Online FAMOS

Note

Wherever the documentation refers to a device, the statements also apply analogously for multiple devices.

11.5.1 Ribbon - Data Processing

Task

Menu item	Description
New	Adds a new Task. Select from which **Functions Package** the new Task is to be generated and enter an appropriate name for the task.
Edit	Opens a dialog in which, for example, the selected task's name can be changed.
Delete	Deletes the selected Task. The configuration is discarded and can no longer be restored.

Configuration

Menu item	Description
Restore	Restores the most recently **applied** state of the selected task's configuration. The current configuration is discarded and can no longer be restored.
Check	Checks the configuration of the selected Task. If an error occurs, a notification will appear in the status bar! Special feature of imc Inline FAMOS: The faulty portion will appear highlighted in the Editor. If the syntax-check concludes without an error, all newly defined virtual channels and variables will be adopted in the Channels list.
Apply	Applies the configuration of the selected Task. As soon as the configuration has been **applied**, the Task starts and the calculations are performed. In the table of channels in the Setup, the newly defined channels appear as additional **virtual channels**.

Clipboard

Menu item	Description
Cut	Removes the selection (the highlighted portion in the Editor) to the Clipboard.
Copy	Copies the selection (the highlighted portion in the Editor) to the Clipboard.
Paste	Inserts the Clipboard contents at the position highlighted.

Edit

Menu item	Description
Undo	Reverses the effect of the last change (in the Editor).
Redo	Restores the previously reversed change.

Search

Menu item	Description
Find	Find specific text
Find and Replace	Find and replace specific text

Im-/Export

Menu item	Description
Import	Imports Data Processing Task configurations.
Export	Exports the configuration of the selected Data Processing Task.

11.6 Bus Decoder - Decoding fieldbus log channels

There is a limit on the count of active fieldbus channels decoded in the device. You have the option of outsourcing the decoding of individual channels from the device to the PC. There are two methods of doing this:

- using imc FAMOS (post-processing - after the measurement), and
- using the imc STUDIO component: Bus Decoder (during the measurement)

The decoding is performed on the fieldbus log channel. By means of the Fieldbus-Assistant, you can specify whether the particular channel is to be decoded either in the device or via the Bus Decoder, for example.

- By means of the Bus Decoder, you can select which among the selected fieldbus channels to decode from the log channel during the measurement.
- When using imc FAMOS, all fieldbus channels are always decoded (not only specifically highlighted ones).

Bus Decoder

The decoding is performed on the basis of the data streams of the measurement currently running on the PC. This utilizes the PC's computation resources.

The decoding information which is usually located in separate configuration files (e.g. with CAN in *.dcb) is instead in embedded in the log channel. Thus, the log channel contains all information necessary for decoding.

The following functions are available:

- Decoding of either all or individual channels from a log-channel
- Resampling of the channels
- Saving of the result channels

The results generated can be displayed on Panel pages and saved with the associated measurement data. Subsequent processing by means of imc Inline FAMOS is also possible.

Note — Notes on activing

Bus Decoder is a package of functions for Data Processing.
Activate the Bus Decoder and the "imc STUDIO DataProcessing Editor" using the Product Configurator 29, so that it will appear in the software. You will find the Bus Decoder in the group: "imc STUDIO DataProcessing".

Supported fieldbus types (Bus Decoder)

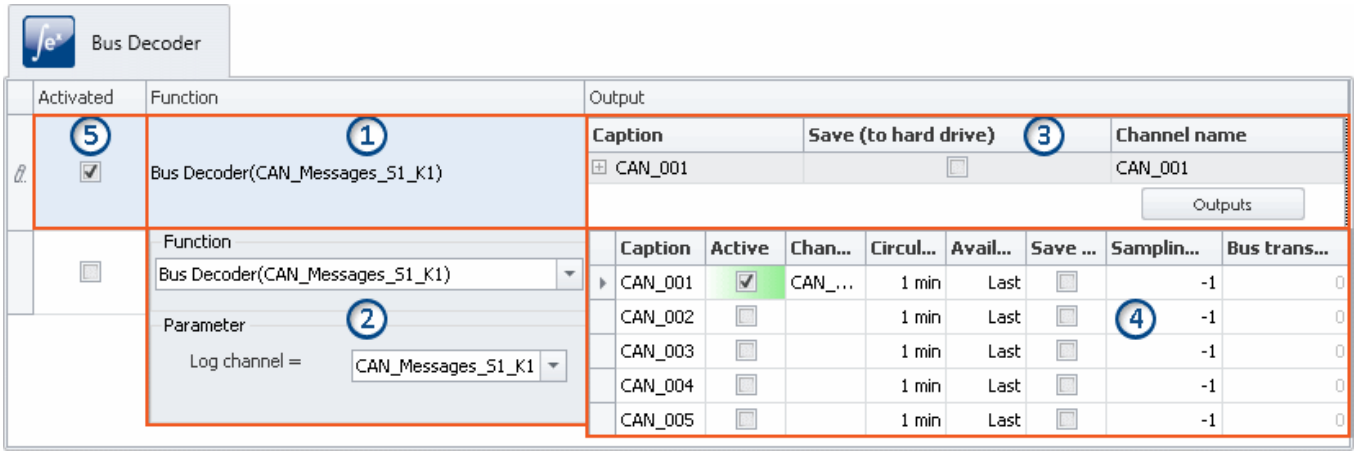
Decoding of the log channels is possible for multiple fieldbusses. Some fieldbusses need to be configured separately for this purpose.

Fieldbus type	FW group A	FW group B	Note
CAN	●	---	Per-channel adoption in the blob. Channel settings: See "*Blob - Decoding with the Bus Decoder* 539" Logging of the data must be activated.
SPI	●	---	All channels are adopted in the blob. No selection is possible.
MVB	●	---	All channels are adopted in the blob. No selection is possible. Logging of the data must be activated.
FlexRay	●	---	Per-channel adoption in the blob. Channel settings: See "*Properties of Signals* 628" Logging of the data must be activated.

11.6.1 Opening Bus Decoder

Open the plug-in Data Processing via the Navigation pane. In the plug-in's main window, there will be a tab (Task:) for the functions package Bus Decoder. If it is not selected already, left-click the mouse on the Task name. The Bus Decoder Editor will appear in the main window.

11.6.2 User interface



The window is subdivided into five regions:

1. Function: Lists any functions whose parameters have already been set (decoding of a log channel)
2. Function: Selection of the log channel
3. Output: Displays the activated outputs (the decoded channels)
4. Output: Activate and parameterize the outputs here (the decoded channels)
5. Activate/deactivate functions

Region 1: Display of parameterized functions

Until the settings have been made for any function, this line appears empty. Once the settings for any functions have been made, these functions are displayed here.

Decoding a log channel (setting up a function)

In order to decode a desired log channel, click in the column "**Function**" in the first empty row. In consequence, Region 2 opens. There, select the log channel.

Changing a function

To alter a function which has already been set up, click on it corresponding cell.

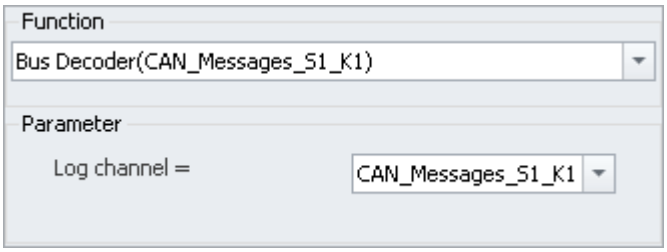
Deleting a function

To delete a function which has already been set up, open the context menu in the corresponding cell and select "*Delete function*".

Region 2: Selecting the log channel

Here you select the "**Log channel**".

In the bottom input box having the arrow button (), select the desired log channel.



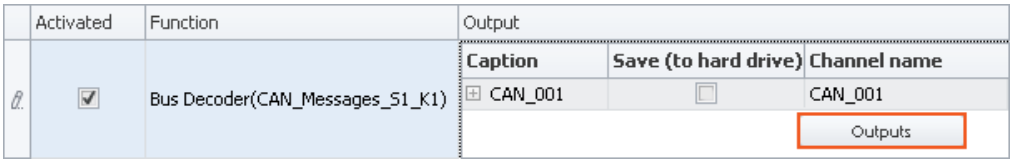
When you click the mouse anywhere outside of the dialog, the settings are concluded.

Region 3: Displaying the activated outputs (the decoded channels)

Under "**Output**", all channels belonging to the log channel, for which decoding is configured in the Bus Decoder, are listed. Each output quantity generates a channel (virtual channel), which is available in the Setup and in the Panel and can be linked with a Widget.

Activating (decoding) and making settings for outputs

In order to activate and parameterize outputs (channels), click in the column "*Output*" on the button "**Outputs**". Immediately, the parameterizing window (Region 4) opens. There, select and configure the channels.



For each function, the selected channels are set separately, for which reason there is an "*Outputs*" button for each function. For each function, any number of channels can be activated. Here, too, as with the functions, the amount is limited only by the performance power of your PC.

Region 4: Window for parameterizing and activating the channels

Here, you can activate and parameterize the channels. The channels available are listed in a table.

	Caption	Active	Chan...	Circul...	Avail...	Save ...	Samplin...	Bus trans...
▸	CAN_001	☑	CAN_...	1 min	Last	☐	-1	0
	CAN_002	☐		1 min	Last	☐	-1	0
	CAN_003	☐		1 min	Last	☐	-1	0
	CAN_004	☐		1 min	Last	☐	-1	0
	CAN_005	☐		1 min	Last	☐	-1	0

Activating channels

Put a check mark in the column "**Active**" for the channel desired.

Checking this box activates the decoding for the channel. The channel name is set automatically, but can be edited as desired subsequently. Activated channels appear as "**Virtual channels**" in the Channels table (Setup) and in the Data Browser (Panel).

Parameterizing channels

Each channel has parameters which can be adapted to the respective requirements.

E.g., storage of channels on the PC can be activated. To do this, put a checkmark in the box for the column "*Save (HD)*". If necessary, adjust the other settings as well, by editing the inputs.

In the channel table (Setup), the outputs defined appear as "**Virtual channels**". Here, the channel can be parameterized like any other virtual channels (e.g. the data storage). The changes made in the Setup have a direct effect on the Bus Decoder settings, since the same settings, in part, can be made here.

For more information on the individual parameters, see the section: "*Output Parameters* 1104".

Region 5: Activating/deactivating functions

Here, the function can be activated/deactivated.

Only such log channels are decoded which have a checkmark in the column "**Activated**". All newly added functions are activated by default.

11.6.3 Output parameters

Every output can be parameterized. The parameters affect the results returned. The table contains the following columns, in which settings can be made:

- Parameters denoted by ● can also be set in the channel table in the Setup.

Parameter	Description
- **Caption**	Name of the output quantity (permanently assigned)
- **Active**	Activates/deactivates decoding of the channel • **Deactivated:** the channel is not decoded • **Activated:** the channel is decoded
- **Channel name**	Name of the virtual channel on which the output quantity is to be outputted. This name appears in the channel table (Setup) and in the Data Browser (Panel).

	Parameter	Description
•	**Circular buffer duration (curve window)**	For the time period specified, the data are retained on the PC for the purpose of display in the Panel. The Widgets (e.g. the curve window) only contain data for the duration of the buffer period set by the user.
•	**Available events (curve window)**	With this setting, you determine whether all events triggered in the Panel are to be available, or only the last one. If "*Circular buffer duration (curve window)*" is set, then only "*Last*" is available. For a detailed explanation, see "Storage options 414".
•	**Save (to hard drive)**	Saves the output quantities to the PC
-	**Averaging time**	>0 Sampling of the decoded channel at a defined sampling time -1 The original channel's setting (Feldbus-Assistenten) is applied. 0 Recorded with a time stamp
	Bus transmission cycle [s]	Defined by the Assistant >0 Sampling interval/transfer cycle of the channel 0 Recorded with a time stamp

11.7 Powertrain Monitoring - Machine and bearing diagnostics

General

The component imc STUDIO Powertrain Monitoring has been developed in cooperation with the company **GfM** (**Gesellschaft für Maschinendiagnose mbH**) for power train diagnostics.

It is dedicated to diagnosis of powertrains. The powertrains can consist of motors, shift gearboxes and engines as well as devices for braking. The diagnosis can be used in field scenarios, test rigs or end of line tests in production.

Powertrain Monitoring offers two different kinds of diagnosis of vibrations: a **Base diagnosis** 1110 on the basis of characteristic values and a **Advanced Diagnosis** 1117 on the basis of a frequency selective search of kinematic pattern. A configuration for a specific gear type will be created via the **Powertrain Monitoring Assistant**. This configuration will be saved in the project of imc STUDIO and can be distributed via this project on different test locations. The configuration is used by the **Editor** by applying configured inputs to physical measurement channels. This system allows to use the same configuration on different measurement systems if multiple test locations are driven with the same powertrain type.

Note

Notes on activing

Powertrain Monitoring is a package of functions for Data Processing.

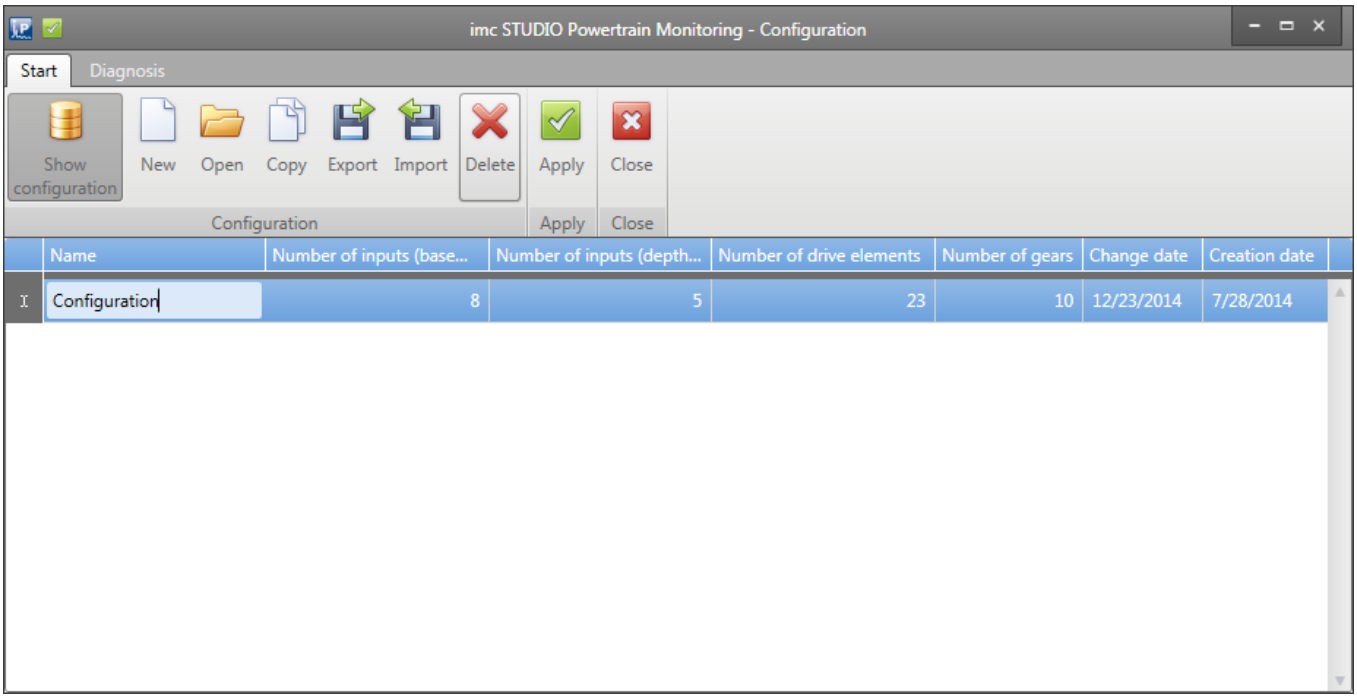
Activate the Powertrain Monitoring and the "imc STUDIO DataProcessing Editor" using the Product Configurator 29, so that it will appear in the software. You will find the Powertrain Monitoring in the group: "imc STUDIO DataProcessing".

Licensing

For Powertrain Monitoring, an extra license is required.

11.7.1 Assistant

11.7.1.1 Configuration

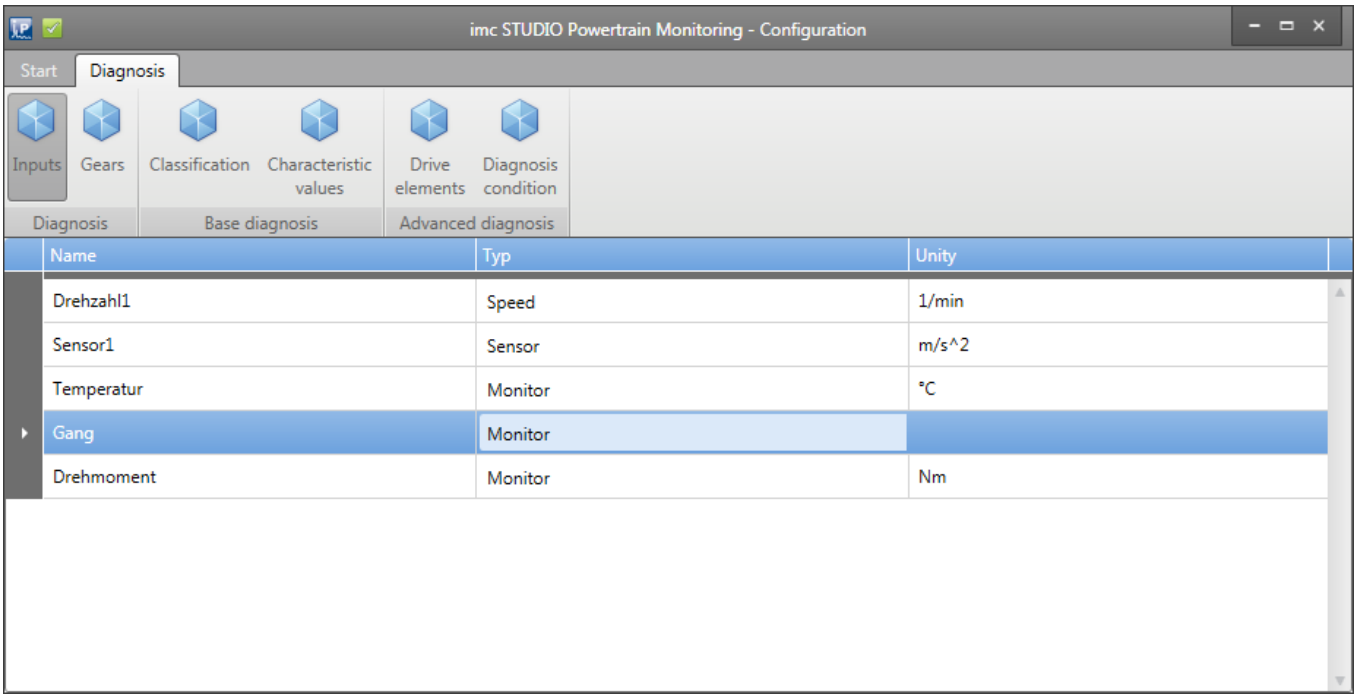


Picture 1: Manage Configurations

Via the menu point show configurations, all configurations that are saved in the active imc STUDIO project can be managed. That means creating new, open, copy, export, import and delete configurations.

One configuration can be applied to the active project by the menu point apply. **To have this persistent the imc STUDIO project have to be saved**.

11.7.1.2 Inputs



Picture 2: Assistant Inputs

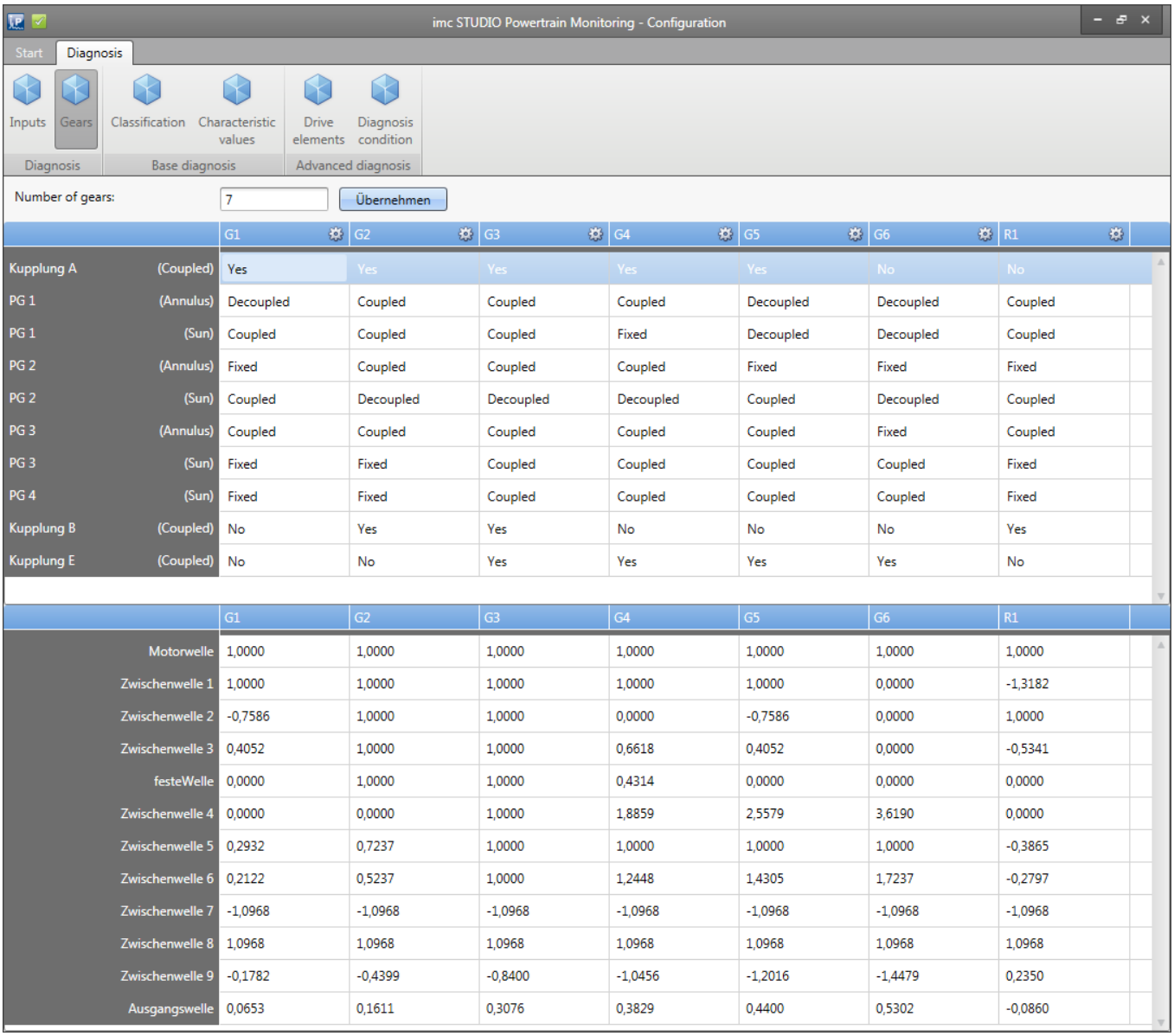
An essential part of both diagnosis procedures are the measurement of structure-borne sound of the test item via accelerometers. These will be added as sensors in the assistant.

Particularly for the advanced analysis the accurate measurement of the speed is required. This speed input will be assigned to sensors as reference speed.

Furthermore up to 12 process information can be measured with the Powertrain Monitoring. These serve for the classification of the operational state in the base diagnosis and thus a description of steady states of the system. In the advanced diagnosis the process information can be used to limit the analysis to certain operational states.

In the menu point inputs, the accelerometer, speed and process inputs can be configured. For a speed input the maximum occurring speed must be set in the properties. For a sensor the reference speed must be chosen from the already configured speed channels. For the advanced analysis 1117 the measurement time for this sensor must also be set.

11.7.1.3 Gears



		G1	G2	G3	G4	G5	G6	R1
Kupplung A	(Coupled)	Yes	Yes	Yes	Yes	Yes	No	No
PG 1	(Annulus)	Decoupled	Coupled	Coupled	Coupled	Decoupled	Decoupled	Coupled
PG 1	(Sun)	Coupled	Coupled	Coupled	Fixed	Decoupled	Decoupled	Coupled
PG 2	(Annulus)	Fixed	Coupled	Coupled	Coupled	Fixed	Fixed	Fixed
PG 2	(Sun)	Coupled	Decoupled	Decoupled	Decoupled	Coupled	Decoupled	Coupled
PG 3	(Annulus)	Coupled	Coupled	Coupled	Coupled	Coupled	Fixed	Coupled
PG 3	(Sun)	Fixed	Fixed	Coupled	Coupled	Coupled	Coupled	Fixed
PG 4	(Sun)	Fixed	Fixed	Coupled	Coupled	Coupled	Coupled	Fixed
Kupplung B	(Coupled)	No	Yes	Yes	No	No	No	Yes
Kupplung E	(Coupled)	No	No	Yes	Yes	Yes	Yes	No

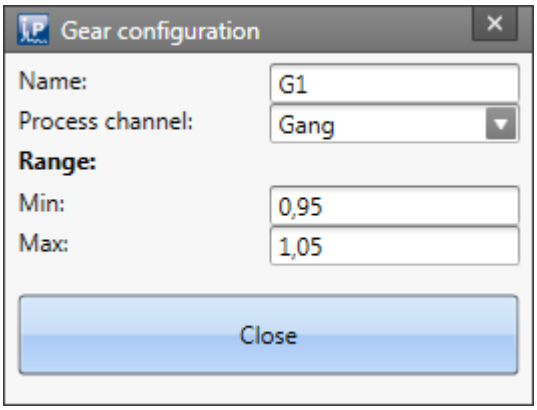
	G1	G2	G3	G4	G5	G6	R1
Motorwelle	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000
Zwischenwelle 1	1,0000	1,0000	1,0000	1,0000	1,0000	0,0000	-1,3182
Zwischenwelle 2	-0,7586	1,0000	1,0000	0,0000	-0,7586	0,0000	1,0000
Zwischenwelle 3	0,4052	1,0000	1,0000	0,6618	0,4052	0,0000	-0,5341
festeWelle	0,0000	1,0000	1,0000	0,4314	0,0000	0,0000	0,0000
Zwischenwelle 4	0,0000	0,0000	1,0000	1,8859	2,5579	3,6190	0,0000
Zwischenwelle 5	0,2932	0,7237	1,0000	1,0000	1,0000	1,0000	-0,3865
Zwischenwelle 6	0,2122	0,5237	1,0000	1,2448	1,4305	1,7237	-0,2797
Zwischenwelle 7	-1,0968	-1,0968	-1,0968	-1,0968	-1,0968	-1,0968	-1,0968
Zwischenwelle 8	1,0968	1,0968	1,0968	1,0968	1,0968	1,0968	1,0968
Zwischenwelle 9	-0,1782	-0,4399	-0,8400	-1,0456	-1,2016	-1,4479	0,2350
Ausgangswelle	0,0653	0,1611	0,3076	0,3829	0,4400	0,5302	-0,0860

Picture 3: Assistant Gears

A special process information, always required by Powertrain Monitoring, is the gear information. This process information is also used in the classification like all other process information. It also serves as output of status information for the base diagnosis. Thus a status information is written whether a certain characteristic value is exceeded in a certain gear. This allows to control the test procedure for example skipping a certain gear.

For the advanced analysis the gear information serves as information for calculating the transmission of the individual drive elements correctly and therefore allows a frequency selective search of kinematic patterns.

The gears can be configured in the menu point gears (see Picture 3).



Picture 4: Gear configuration

To identify the current gear in the analysis, a unique range of one process information must be defined. The same process input must be used for all gears. This can be configured by clicking the button besides the gear name (see Picture 4).

11.7.2 Editor

To apply the configuration built with the assistant to the current experiment, the Powertrain Monitoring Editor 1114 has to be used.

One of the configurations that are saved in the current project must be chosen in the **configuration selection box**. The executing function should be set depending on the chosen configuration to base or advanced diagnosis. Thus all sensor, speed and process inputs are read from the configuration and are provided as inputs. These must be assigned to measurement channels that are available in the current experiment.

The measurement channels must met the following requirements:

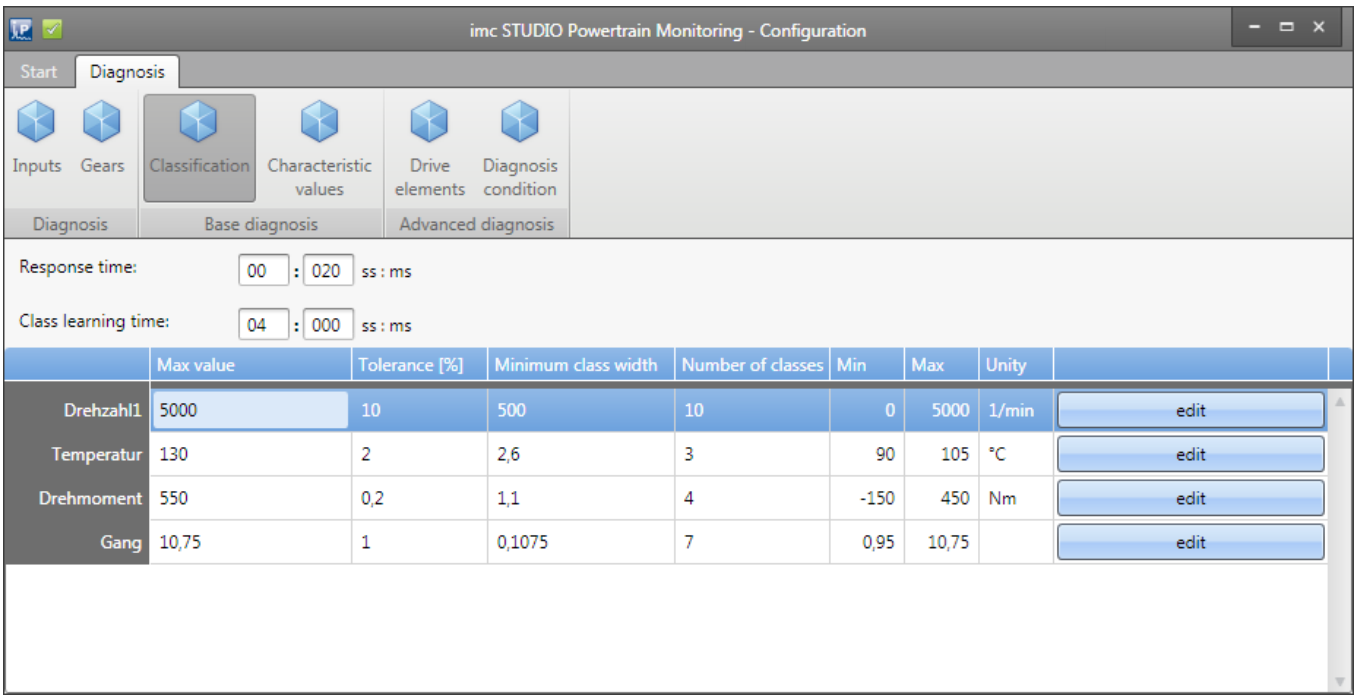
- Accelerometers and speed channels must be measured with 50 kHz.
- Sampling frequencies of process channels must be an integer divider of 50 kHz (e.g. 100 Hz, 1 kHz, …)
- All channels must be set to undefined measurement time.

11.7.3 Base diagnosis

The base diagnosis consists of a characteristic value calculation with self-learning thresholds. For the learning phase a classification is used where an individual warn- and alarm threshold is learned for each class. The classification 1111 can be configured with the Powertrain Monitoring Assistant.

11.7.3.1 Assistant

11.7.3.1.1 Classification



Picture 5: Assistant Classification

The following general settings must be configured in the menu point **Classification**:

Settings	Description
Reaction Time	If a certain class is active continuously over this time it counts as stable. The learning time for unlearned classes starts counting from this time point. For already learned classes, the learned warn- and alarm threshold will be used from this time point. The user defined threshold will be used between the actual class change and the reaching of the reaction time.
Class Learning Time	A class will be considered as learned if the system was in this state over the defined class learning time. The class learning time can also be reached by multiple time parts that sum up to the class learning time. During the learning phase the maximum occurred characteristic values will be saved. At the end of the learning phase warn- and alarm thresholds are calculated by multiplying the occurred maximum with a percent factor.

Class details of the channel: Drehzahl1

	Min	Max	Unity	Ignore
Class 1	0,00	1.000,00	1/min	☐
Class 2	1.000,00	1.300,00	1/min	☐
Class 3	1.300,00	1.600,00	1/min	☐
Class 4	1.600,00	2.000,00	1/min	☐
Class 5	2.000,00	2.500,00	1/min	☐
Class 6	2.500,00	3.000,00	1/min	☐
Class 7	3.000,00	3.500,00	1/min	☐
Class 8	3.500,00	4.000,00	1/min	☐
Class 9	4.000,00	4.500,00	1/min	☐
Class 10	4.500,00	5.000,00	1/min	☐

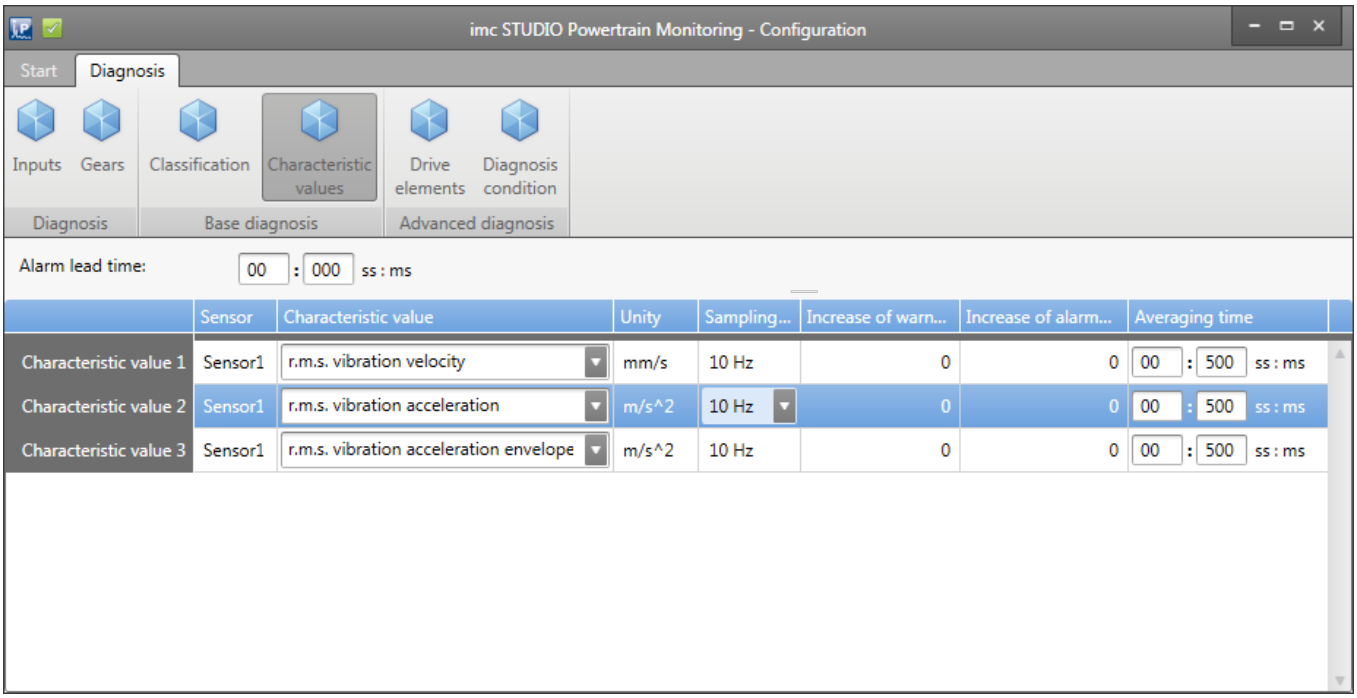
Schließen

Picture 6: Assistant Classification Details

For the classification each process information is used that were defined (see chapter "*Configuration* 1107"). Each process information can be divided in up to 1000 classes. Thus the class borders can also be learned automatically by only configuring the class width of each process information. Furthermore the classes can be configured individually by clicking the Edit button (see Picture 6).

An exception is the **gear** information. This process information can be configured as described in chapter "*Gears* 1109" and is not modifiable in this section.

11.7.3.1.2 Characteristic values



Picture 7: Assistant Characteristic Values

The characteristic values can be configured in the corresponding menu point (see Picture 7).

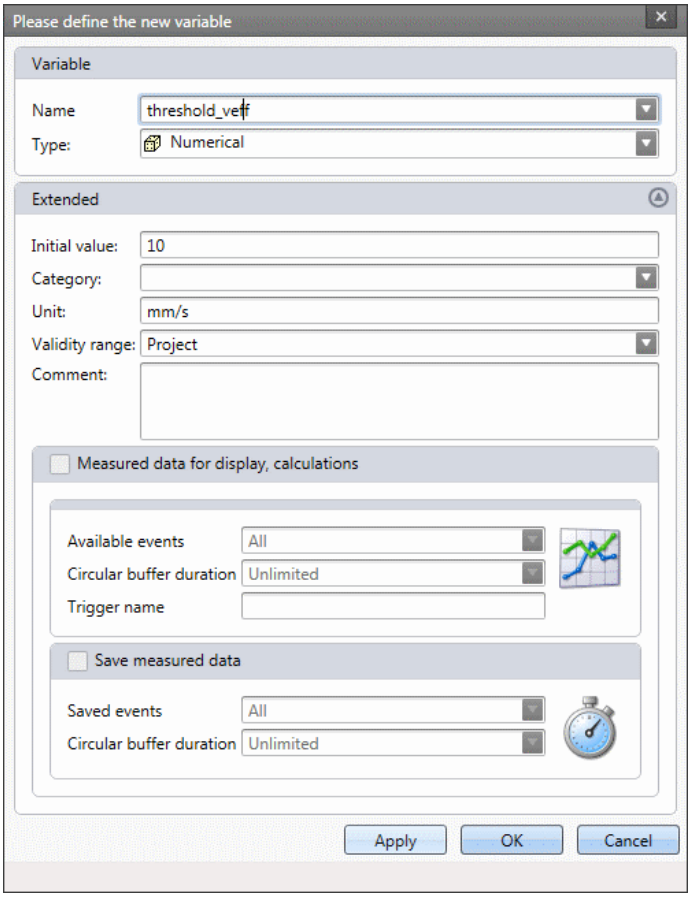
The following characteristic values are available for the base diagnosis for each configured accelerometer:

- R.M.S. of vibration velocity based on DIN-ISO 10816-3 in the frequency range 10 Hz - 1 kHz
- R.M.S. of vibration acceleration in the frequency range 0 Hz - 25 kHz
- R.M.S. of vibration acceleration of the envelope in the frequency range 0 Hz - 25 kHz

The characteristic value will be calculated from the measured signal of the accelerometer with the given average time. A prerequisite is a configured Anti-Aliasing filter for the sampling frequency of 50 kHz for the accelerometers in imc STUDIO.

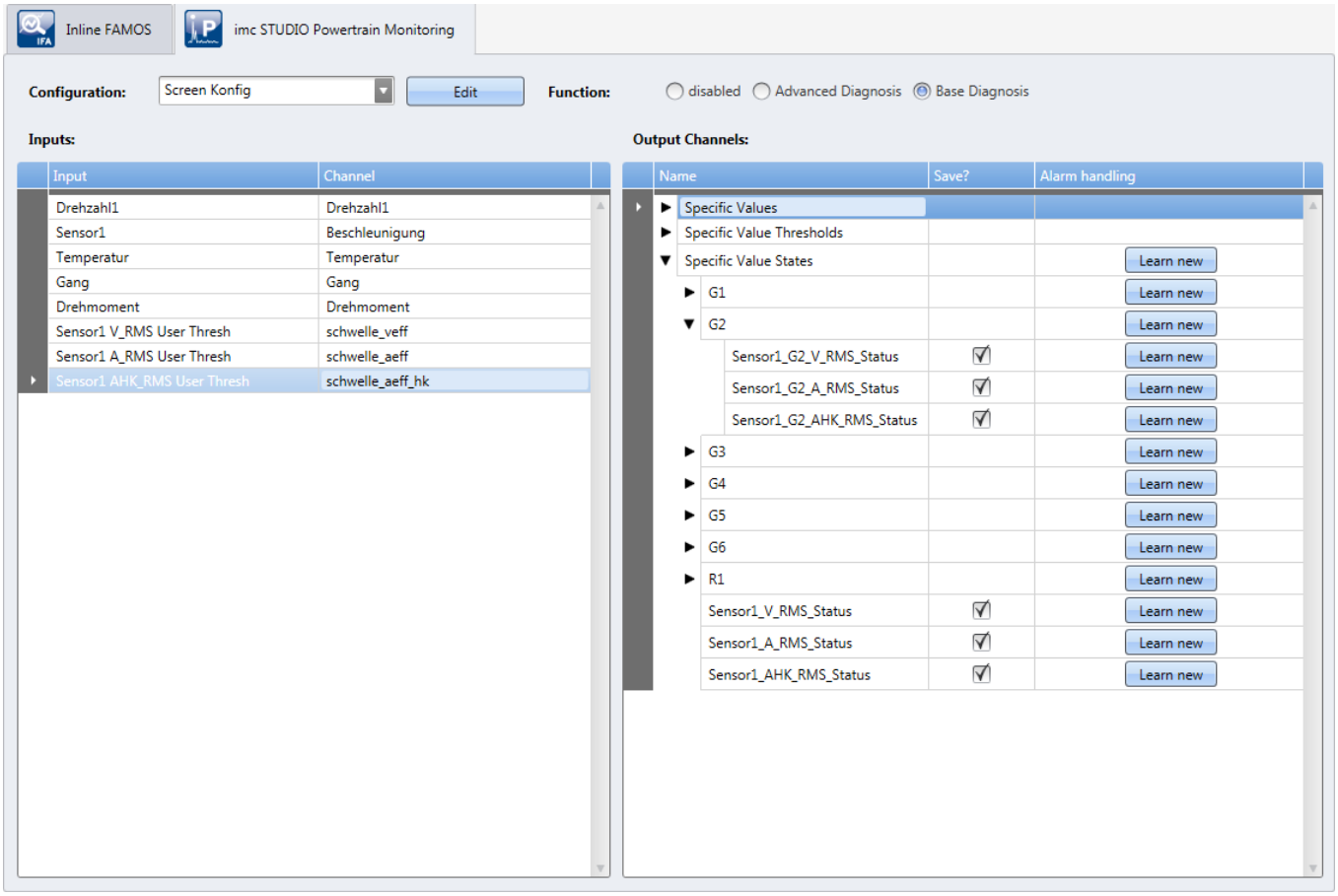
The configured percent increase of warn- and alarm thresholds are multiplied to the maximum characteristic value at the end of the learning phase of the class.

11.7.3.2 Editor



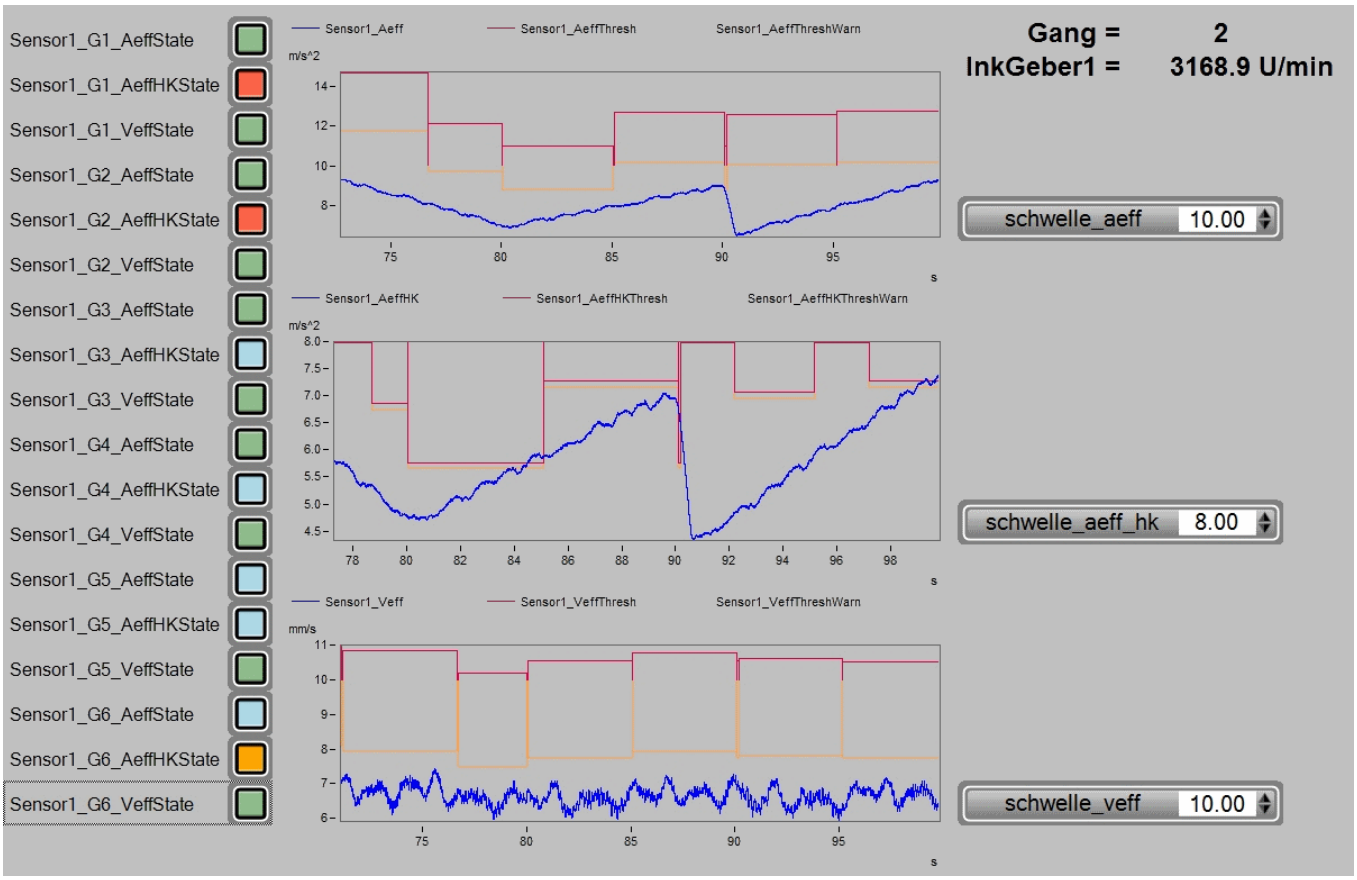
Picture 8: Create User defined variables

The inputs of the configuration must be assigned to the physical measurement channels for the base diagnosis (see also general description of the editor 1110). Furthermore each characteristic value has a user defined threshold that must be assigned. These are assigned to user defined variables that can be configured in the Data Browser by right clicking in the Data Browser (see Picture 8).



Picture 9: Base Diagnosis Editor

11.7.3.3 Measurement data



Picture 10: Example: Base Diagnosis

For each configured characteristic value a warn- and alarm threshold channel is provided.

If a certain class is learned, the corresponding warn- and alarm threshold is put out to the channel. If the class is not yet learned or the reaction time is not yet elapsed, the user defined threshold will be put out.

Additionally to the characteristic value, warn threshold and alarm threshold, state information are provided. These signalize an exceedance of a threshold in the following manner:

- No Exceedance of any threshold = 0
- Exceedance of the user defined threshold = 1
- Exceedance of the warn threshold = 2
- Exceedance of the alarm threshold = 3

The state information are written for each characteristic value and for each individual gear.

By using existing visualization tools the state information can be visualized for example as traffic light.

If the user defined threshold is changed during the measurement, the user defined thresholds will be reevaluated and reset if the new threshold does not exceed the characteristic value. The warn- and alarm thresholds are based on learned classes and will not be reset due to a change of the user defined threshold. They must be reinitialized by resetting and relearning the classes. This can be achieved in the Powertrain Monitoring Editor 1114 on the right side at the various state channels.

Picture 10 shows a possible way of visualizing the data of the base diagnosis. On the left side the state information are shown for each characteristic value and each gear. The curve windows contain the characteristic values and corresponding warn- and alarm thresholds. The user defined thresholds are placed on the right side.

11.7.3.4 Summary of function

- Calculating the characteristic value of one accelerometer with the configured average time and bandwidth.
- Determining the current class
  - A class counts as stable if it is active the whole reaction time.
  - A class is learned if the learning time is reached. This can also be reached in multiple time parts.
  - At the end of the learning phase the maximum characteristic value during the learning phase is used to calculate the warn- and alarm thresholds.
- If the class is not yet learned or not stable the user defined threshold will be written to the threshold channels.
- If the class is learned and stable the learned warn- and alarm threshold will be written to the corresponding threshold channels.
- Depending on the exceedance of a threshold the state is written to the state channels for the characteristic value.

11.7.4 Advanced diagnosis

The advanced diagnosis of machine monitoring evaluates abnormal states in spectra and envelope spectra. It provides early evidence of evolving damages. The methods are well known and accepted. They are partly described in the VDI 3832. If an abnormal state of a drive element is recognized it is possible to view the damage evolution long before an actual failure. So it is possible to evolve knowledge about approaches of improvement.

Basis for the advanced diagnosis are the order and envelope order spectra. These spectra are calculated from the vibration signal and the speed signal if defined conditions are fulfilled for a certain measurement time. Possible conditions are speed or process ranges to allow an analysis in well-defined states.

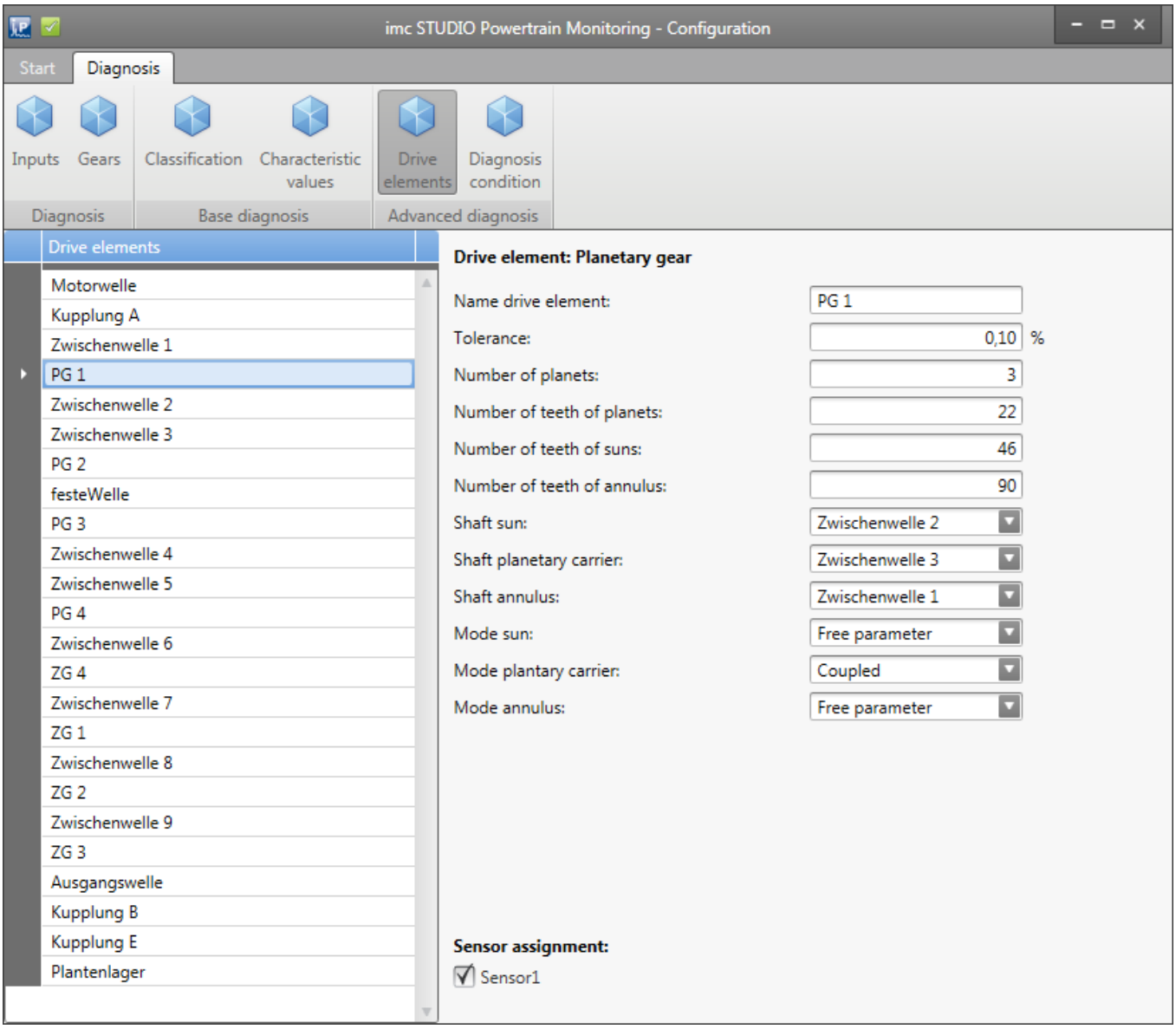
The calculated spectra are then analyzed according to a significance analysis. This is done by first calculating a moving significance threshold that represents the average amplitude over the abscissa. The spectrum is then divided by this threshold and all values below one are set to zero. The remaining peaks are significant and thus with high possibility a consequence of a deterministic mechanic cause, in contrast to random excitations. If this significant spectral components fit to kinematic pattern of a drive element, they are called diagnostic attributes below.

For further diagnosis the spectra can be visualized as single spectrum, color map or waterfall diagram.

11.7.4.1 Assistant

The advanced diagnosis is based on a cyclic evaluation of the measured vibration signal. In order that an analysis can take place, the predefined diagnosis conditions 1122 must be fulfilled over a configured measurement time. The measurement time is configured in the menu point **inputs**, as described in chapter "*Configuration* 1107". The longer the measurement time the better the resolution of the calculated spectra. This increases the diagnosis quality. However the maximum order or frequency is lower. This can possibly lead to not examining all kinematic patterns.

11.7.4.1.1 Drive elements



Picture 11: Assistant Drive Elements

The monitoring gear has to be configured in the Powertrain Monitoring Assistant. Thus the complete power train must be represented under the menu point Drive Elements (see Picture 11). Thereby at some drive elements free parameter can be chosen. These free parameters must then be set in the menu point Gears. This allows an individual configuration for each gear. For example a clutch can coupled in one gear and decoupled in another and thus influencing the transmission of the shafts. That a certain drive element can be monitored by the advanced diagnosis, it has to be assigned to one ore more sensors.

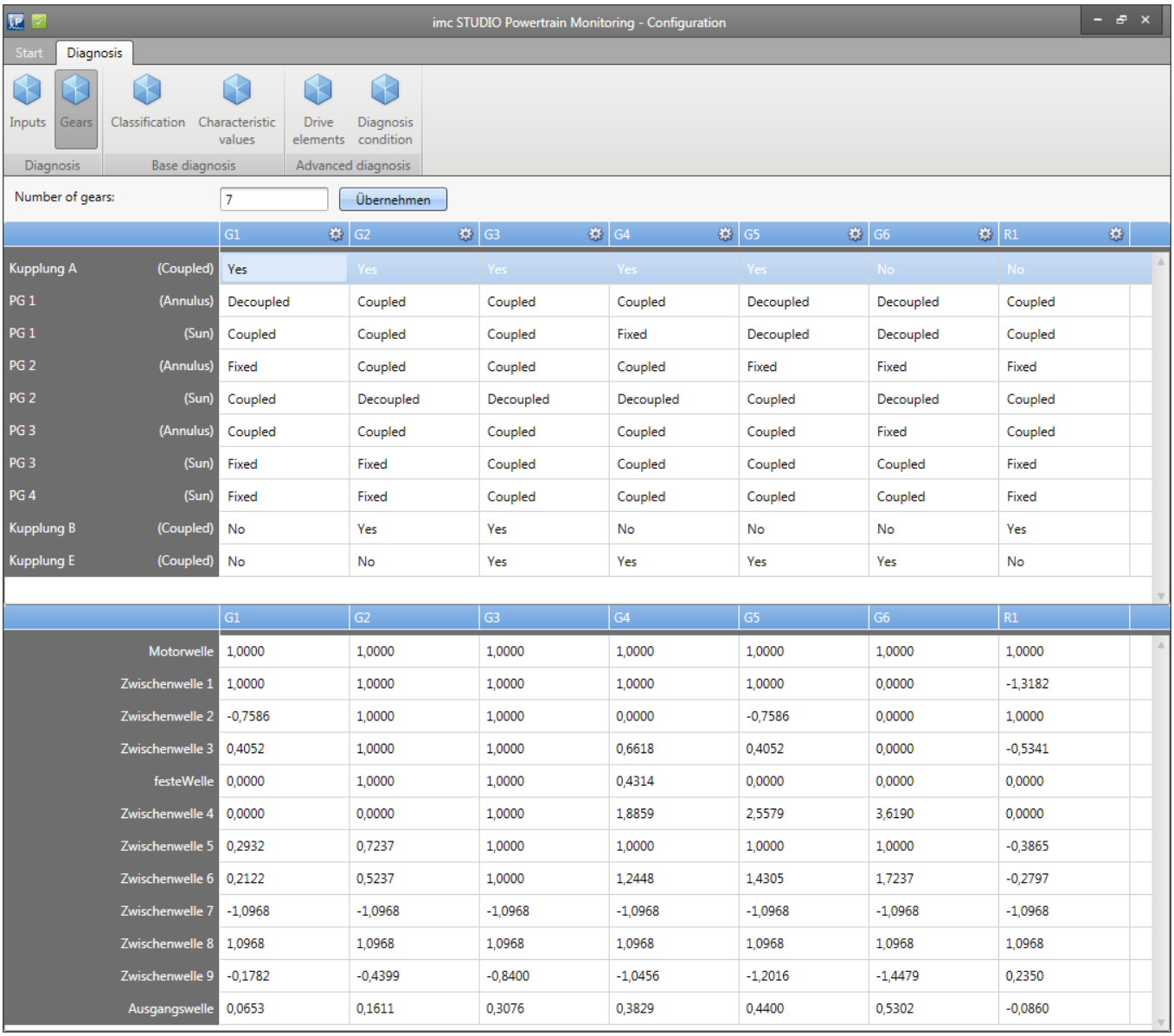
The following drive elements can be configured:

Drive elements	Description
Shaft	Exactly one shaft in the system must be configured as reference shaft. This is also the shaft where the speed is measured. It has always the transmission of 1. The transmission of all other shafts in the machine are in relation to this reference shaft. Normally a tolerance of 0.1 % is suitable. In the analysis it is searched for loose particles and unbalance on the configured shafts.

Drive elements	Description
Drive Shaft	The drive shaft can be used as in the configuration like a normal shaft. This can also be used as reference shaft for speed measurement. Normally a tolerance of 0.1 % is suitable. In the analysis it is searched for loose particles, unbalance and link damage or misalignment on the configured drive shafts.
Clutch	A clutch can be configured as fixed connection or as shifting clutch between two shafts. Therefore the input and output shaft must be set to one shaft of the already configured shafts. If this clutch is fixed the option coupled has to be set to yes. If the clutch is decoupled due to gear switching the option must be set as free parameter. Normally a tolerance of 0.1 % is suitable. In the analysis it is searched for misalignment on the configured clutches.
Gear Drive	The gear drive describes a spur gear and bevel gear. It must be chosen a shaft for the pinion and wheel respectively. It has to be set whether the pinion and wheel are coupled with the shaft or whether the coupling between pinion and shaft or wheel and shaft is decoupled due to gear switching. Furthermore the number of tooth for the pinion and wheel has to be entered. Normally a tolerance of 0.1 % is suitable. In the analysis it is searched for revolving deviation of flank shapes and local deviation of flank shapes at the pinion and wheel on the configured gear drives.
Belt Gearing	For the belt gearing the shaft of the driven and driving wheel. It has to be configured whether the driven and driving wheel is fixed coupled with the shaft or whether the connection is modified due to gear switching. Furthermore the diameter of the driven and driving wheel as well as the length of the belt must be defined. Normally a tolerance of 0.1 % is suitable. In the analysis it is searched for local irregularities of the belt on the configured belt gearings.
Planetary Gear	The planetary gear covers planetary gears with single planets (in contrast to stage planets). The number of planets, number of tooth of the planets, sun and annulus must be configured. Furthermore the shafts must be selected for the annulus, sun and planetary carrier. The mode of the planetary gear must be configured. One of the following combinations can be used: • sun and planetary carrier rotate and annulus is fixed • sun and annulus rotate and planetary carrier is fixed • planetary carrier and annulus rotate and sun is fixed • all shafts rotate The modes can also be set as free parameter if the state changes due to gear shifting. Normally a tolerance of 0.1 % is suitable. In the analysis it is searched for local irregularities of the sun, annulus and planets as well as for revolving deviation of flank shapes on the configured planetary gears.

Drive elements	Description
Rolling Bearing	For the rolling bearing the kinematic parameters must be configured. These are the rotational cage order, the ball spin order, the double ball spin order and the ball pass order of the outer and inner ring. These data can be obtained from the manufacturer of the rolling bearing. The input of the rotational orders can be done with respect to order 1 or with respect to the relative speed of the rolling bearing. Furthermore the shafts for the outer and inner ring must be chosen. If one of these is fixed the shaft can be left empty. If the rolling bearing is configured as planetary rolling bearing no shafts must be set. Normally a tolerance of 1 % is suitable to compensate slippage of the rolling bearing. In the analysis it is searched for outer and inner ring damage, cage damage and rolling element damage on the configured rolling bearings.
Three-Phase Motor	For the three-phase motor the net frequency must be configured. Normally a tolerance of 0.1 % is suitable. In the analysis it is searched for magnetic imbalance on the configured three-phase motors.
Steady Frequency	The elements steady frequency and steady frequency envelope can be used to monitor individual frequencies in the frequency spectrum and envelope frequency spectrum respectively. The frequency must be configured.
Steady Order	The elements steady order and steady order envelope can be used to monitor individual orders in the order spectrum and envelope order spectrum respectively. The order must be configured.

11.7.4.1.2 Gears



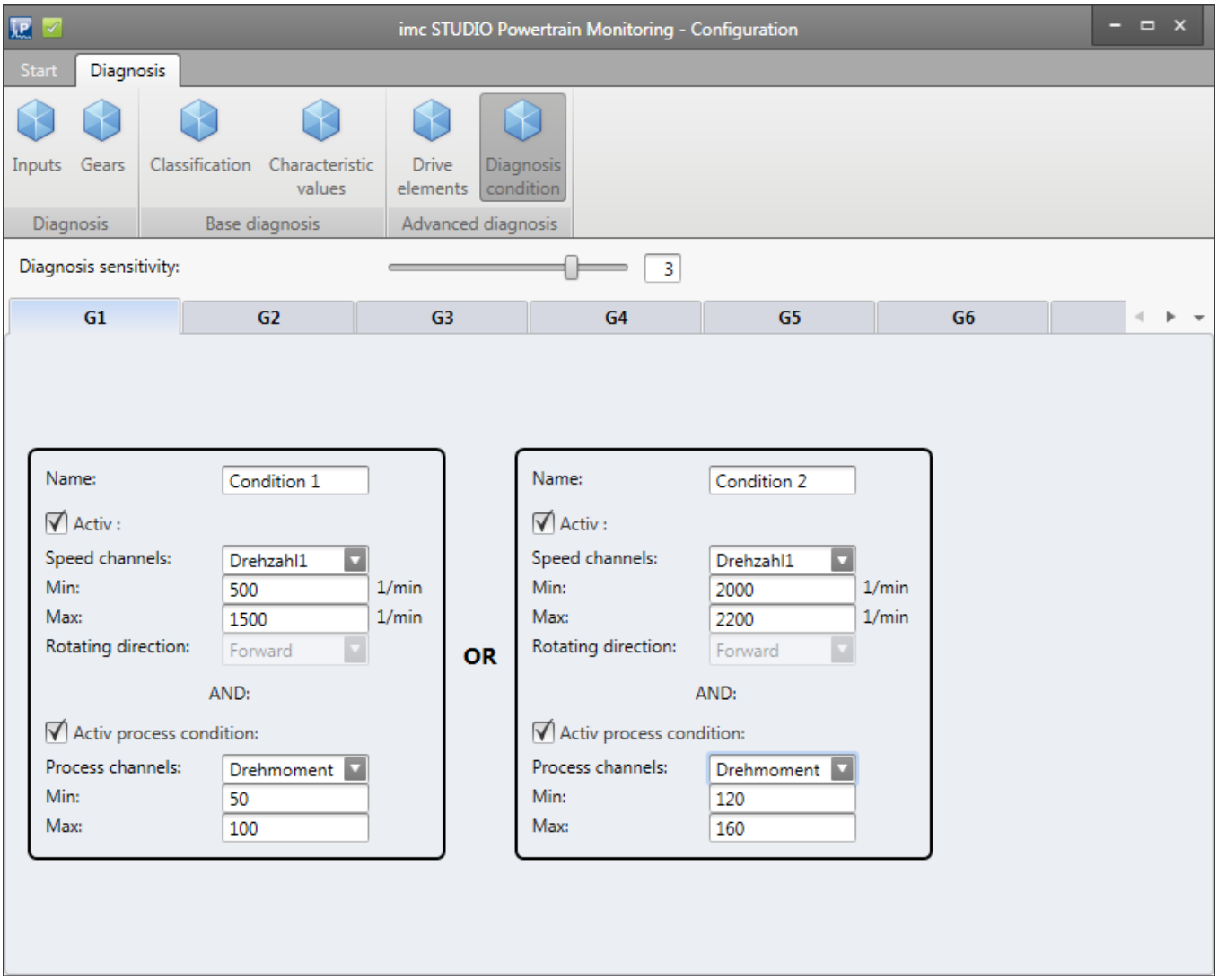
		G1	G2	G3	G4	G5	G6	R1
Kupplung A	(Coupled)	Yes	Yes	Yes	Yes	Yes	No	No
PG 1	(Annulus)	Decoupled	Coupled	Coupled	Coupled	Decoupled	Decoupled	Coupled
PG 1	(Sun)	Coupled	Coupled	Coupled	Fixed	Decoupled	Decoupled	Coupled
PG 2	(Annulus)	Fixed	Coupled	Coupled	Coupled	Fixed	Fixed	Fixed
PG 2	(Sun)	Coupled	Decoupled	Decoupled	Decoupled	Coupled	Decoupled	Coupled
PG 3	(Annulus)	Coupled	Coupled	Coupled	Coupled	Coupled	Fixed	Coupled
PG 3	(Sun)	Fixed	Fixed	Coupled	Coupled	Coupled	Coupled	Fixed
PG 4	(Sun)	Fixed	Fixed	Coupled	Coupled	Coupled	Coupled	Fixed
Kupplung B	(Coupled)	No	Yes	Yes	No	No	No	Yes
Kupplung E	(Coupled)	No	No	Yes	Yes	Yes	Yes	No

	G1	G2	G3	G4	G5	G6	R1
Motorwelle	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000
Zwischenwelle 1	1,0000	1,0000	1,0000	1,0000	1,0000	0,0000	-1,3182
Zwischenwelle 2	-0,7586	1,0000	1,0000	0,0000	-0,7586	0,0000	1,0000
Zwischenwelle 3	0,4052	1,0000	1,0000	0,6618	0,4052	0,0000	-0,5341
festeWelle	0,0000	1,0000	1,0000	0,4314	0,0000	0,0000	0,0000
Zwischenwelle 4	0,0000	0,0000	1,0000	1,8859	2,5579	3,6190	0,0000
Zwischenwelle 5	0,2932	0,7237	1,0000	1,0000	1,0000	1,0000	-0,3865
Zwischenwelle 6	0,2122	0,5237	1,0000	1,2448	1,4305	1,7237	-0,2797
Zwischenwelle 7	-1,0968	-1,0968	-1,0968	-1,0968	-1,0968	-1,0968	-1,0968
Zwischenwelle 8	1,0968	1,0968	1,0968	1,0968	1,0968	1,0968	1,0968
Zwischenwelle 9	-0,1782	-0,4399	-0,8400	-1,0456	-1,2016	-1,4479	0,2350
Ausgangswelle	0,0653	0,1611	0,3076	0,3829	0,4400	0,5302	-0,0860

Picture 12: Assistant Gear configuration

In the menu point Gears the free parameter can be set for each configured gear as described in chapter "Gears 1109". The resulting transmission of each shaft is shown in the table below as an overview. Thus it can be traced whether the complete power train was entered correctly. Picture 12 shows an example configuration of a power train with 10 gears.

11.7.4.1.3 Diagnosis conditions

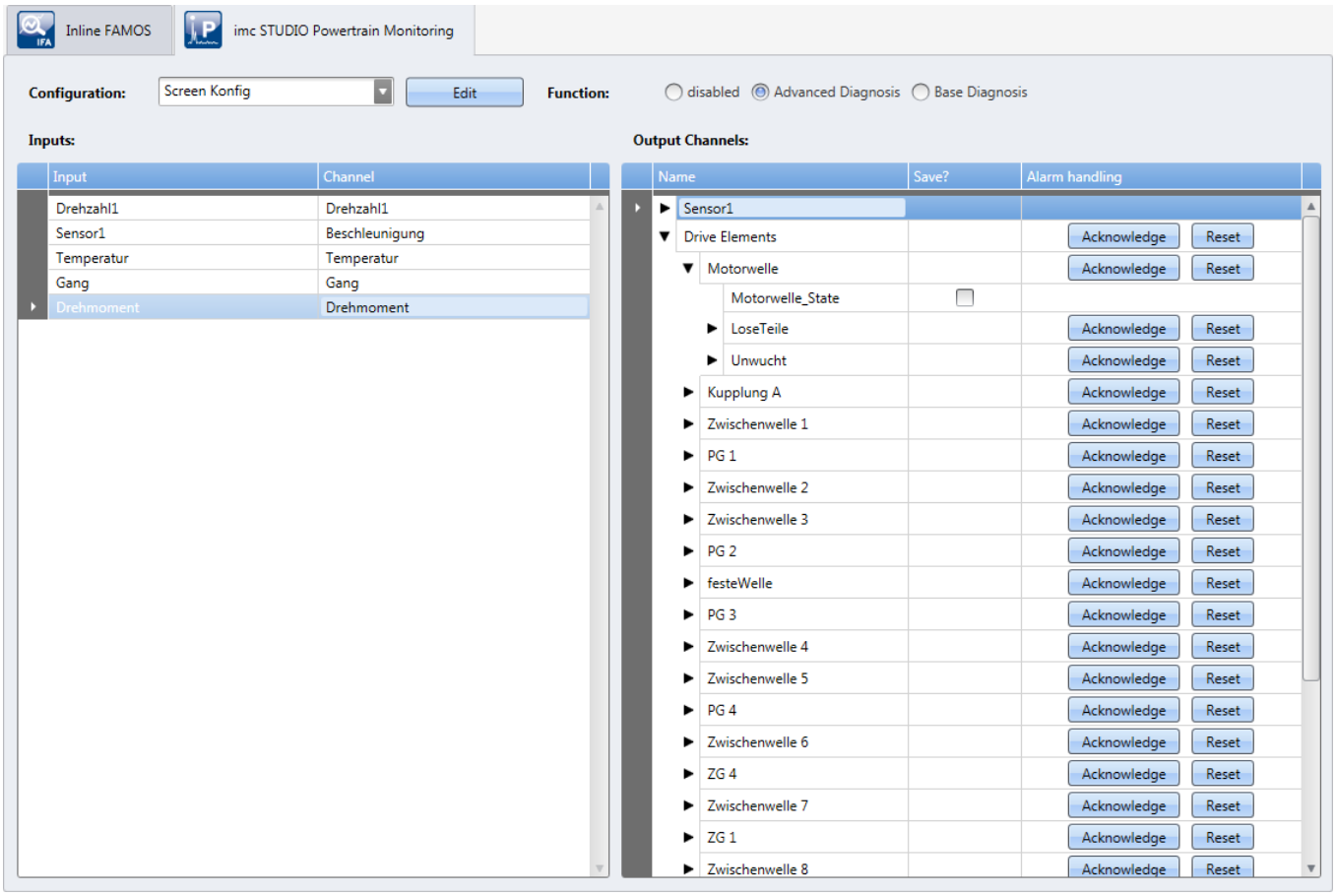


Picture 13: Assistant Diagnosis Conditions

In the menu point diagnosis condition the sensitivity of the advanced analysis must be set. This parameter corresponds to the number of detections before an irregularity is signalized.

Furthermore 5 different diagnosis conditions can be set for each gear individually. One diagnosis condition can consist of a speed range and one process range (see Picture 13). If one of the conditions are fulfilled over the complete measurement time, the analysis of kinematic patterns will be started.

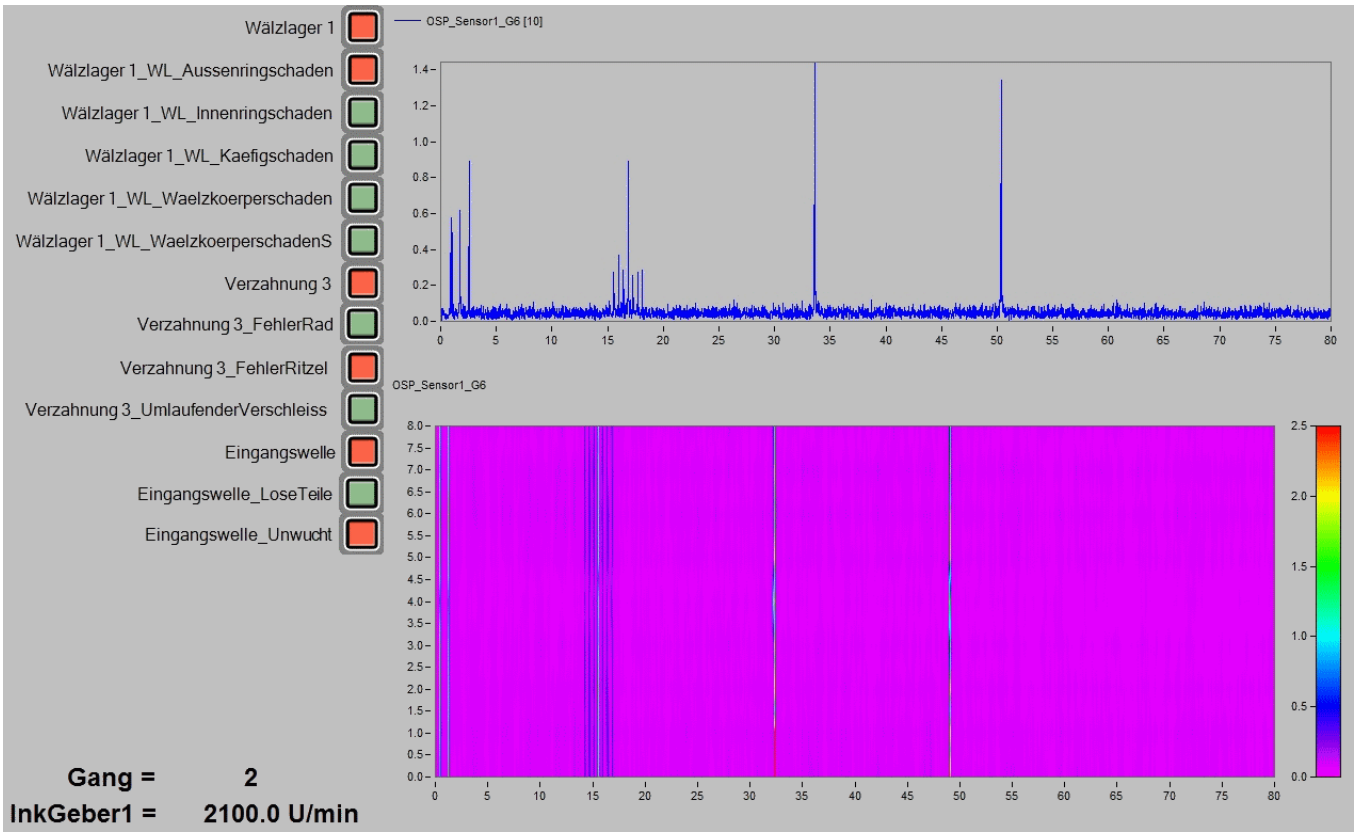
11.7.4.2 Editor



Picture 14: Editor Advanced Diagnosis

For the configuration of the advanced analysis the inputs of the configuration must be assigned to the physical measurement channels (see also general description of the editor 1110).

11.7.4.3 Measurement data



Picture 15: Example: Advanced Diagnosis

The following output channels are provided for each configured sensor in the advanced analysis:

- Order spectra for each configured gear
- Envelope order spectra for each configured gear
- Frequency spectra for each configured gear
- Envelope frequency spectra for each configured gear

Furthermore the following output channels are provided for each configured drive element:

- For each possible irregularity of one drive element 1118 **a state channel** whether this irregularity is detected (0 or 1)
- **One state channel** for each drive element whether one or multiple irregularities are detected (0 or 1)
- For each diagnostic attribute **three channels** with the **order**, **amplitude** and **significance** of the kinematic pattern

All these channels are triggered channels. Thus the time point of the measurement is saved in the channel.

The signalized irregularities can be acknowledged or reset in the Powertrain Monitoring Editor (see picture Editor 1123). An Acknowledgement increases the significance threshold for the searched orders by the given percent value and resets the alarm state. Thus in the following analysis this irregularity is signalized only if the new threshold is exceeded. A reset of an alarm resets the significance threshold to its origin value.

Panel pages offer various possibilities to visualize the input and processed data. Picture 15 shows one Panel page exemplary. The last calculated order spectrum of Sensor1 in the 6th gear is visualized. Furthermore all events of the same order spectrum are visualized in a color map. Irregularities of the drive elements are visualized on the left part of the Panel page with Widgets. On the left bottom corner the current gear and the current speed is shown.

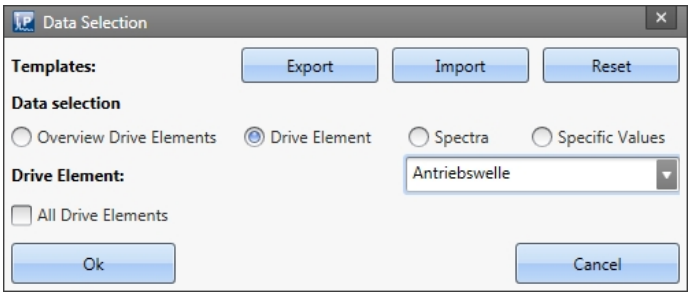
11.7.4.4 Summary of function

- Evaluating whether one of the diagnosis condition is fulfilled over the configured measurement time.
- If one diagnosis condition is fulfilled an evaluation of the vibration signal is done.
  - Calculation of all frequency, envelope frequency, order and envelope order spectra of all vibration signals.
  - Calculation of diagnostic attributes of all kinematic frequencies or orders.
  - Determining whether an irregularity is detected with respect to the preceding analysis.
- Output of the spectra, diagnostic attributes and alarm states in the corresponding channels.

11.7.5 Visualization

11.7.5.1 Create Panel pages via menu action

In imc STUDIO an additional action "Powertrain Monitoring Panel" can be added to a Ribbon via the function "Customize". If this new action is executed, Panel pages with selected data from the Powertrain Monitoring can be generated automatically (Picture 16).



Picture 16: Powertrain Monitoring Panel

The dialogue directly accesses the tasks of the Powertrain Monitoring of the current experiment. Thus all data of the configured tasks are available in the selected category. It can be chosen between the following categories:

- **Overview Drive Elements:** Generates an overview page with the alarm state of all configured drive elements.
- **Drive Element:** A Panel page fort he selected drive element is generated with state information for each damage pattern.
  - If the option "All Drive Elements" is chosen, this Panel page is created for each drive element in the list. This can lead to lot of Panel pages, depending on the configuration.
- **Spectra:** For the selected sensor and gear one Panel page with order spectrum, envelope order spectrum, frequency spectrum and envelope frequency spectrum is generated. These are visualized as color map.
- **Specific Values:** A Panel page with all configured specific values at the chosen sensor is created. For each specific value the alarm state for each gear, an aggregated alarm state as well as a curve window with the specific value and the corresponding warn and alarm thresholds are visualized.

In order that the channel assignment is working properly, the default channel names in the Powertrain Monitoring Tasks must be used. These are automatically generated when a task is applied. Additionally the used Panel pages can be exported, imported and reset via the "Powertrain Monitoring Panel". Further information can be found in chapter Manage Panel Templates 1126.

11.7.5.2 Create Panel pages via scripting

The described functionality in "Create Panel Pages via Menu Action 1125" is also accessible via the imc STUDIO Scripting Editor. Thus the generated Panel pages can be individualist e.g. by adding further channels that do not come from the Powertrain Monitoring tasks.

To have the Powertrain Monitoring accessible from one Script, a reference to the .NET Assembly imc.Studio.PlugIns.GfM.PowertrainMonitoring.Scripting.dll must be added to the project. This can be done via the menu Project > AddReference > GAC.

A short example for creating a Panel page for the first drive element that can be found in the Powertrain Monitoring configuration follows.

```
using System.Linq;
using System.Collections.Generic;
using imc.Studio.Interfaces.V2.Core;
using imc.Studio.PlugIns.GfM.PowertrainMonitoring.Scripting;
```

```
// get reference to PowertrainMonitoring component
ISharedComponentBase shared_comp;
SharedComponents.TryGetValue("PowertrainMonitoring", out shared_comp);
var pm_ref = shared_comp as IPowertrainMonitoringBase;

if (pm_ref == null)
{
    //no reference to Powertrain Monitoring found
    return;
}

//use template directory of current project
string template_dir = pm_ref.GetTemplateDir();
//get first element of DriveElement List
var element = pm_ref.GetDriveElementList().FirstOrDefault();

if (element != null)
{
    //get Panel template file and insert Panel
    string panel_file = template_dir + pm_ref.GetDriveElementType(element) + ".dbv";
    string page_name = Panel.InsertPageBefore(0, panel_file, "", element);

    //get reference to Panel and apply Powertrain Monitoring variables
    PanelPage pp = Panel.Pages.FirstOrDefault(p=>p.Name == page_name);
    pm_ref.ApplyVariablesDriveElement(pp.Base, element);

    //e.g. add additional channels to Panel page
}
```

11.7.5.3 Manage Panel templates

With the installation of the Powertrain Monitoring there are also Panel templates for the different data types and their visualization installed. These can be redesigned as needed. In the menu action "Powertrain Monitoring Panel" there is a function available to export all templates to a directory. The exported Panel pages can be loaded in imc STUDIO like normal Panel pages. After redesigning the pages the new Panel pages can be activated via the import function of the menu action.

These are then saved with the current project. Thus they are also exported if the imc STUDIO project is exported. If the project is changed the templates may be imported again by the user. Via the function reset the user designed templates are overwritten with the default ones.

When designing the Panel templates it is crucial to maintain the exact file names.

Table 1: Panel templates

Template File	Description
AntriebOverview.dbv	Overview Drive Elements
Welle.dbv	Diagnosis Data Shaft
KardanWelle.dbv	Diagnosis Data Drive Shaft
Kupplung.dbv	Diagnosis Data Clutch
Waelzlager.dbv	Diagnosis Data Rolling Bearing
Zahnradgetriebe.dbv	Diagnosis Data Gear drive
Planetengetriebe.dbv	Diagnosis Data Planetary gearing
Riemengetriebe.dbv	Diagnosis Data Belt gearing
Drehstrommotor.dbv	Diagnosis Data Three-phase motor
FesteFrequenz.dbv	Diagnosis Data Steady Frequency
FesteFrequenzHK.dbv	Diagnosis Data Steady Frequency Envelope
FesteOrdnung.dbv	Diagnosis Data Steady Order
FesteOrdnungHK.dbv	Diagnosis Data Steady Order Envelope
Spektren.dbv	Frequency and order spectra of one sensor in one gear
Grunddiagnose.dbv	R.M.S. of Base diagnosis of one sensor

The Panel templates contain Widgets with specific names. On the basis of these names it will be determined which channels are assigned to the Widgets. Furthermore there are placeholder that are replaced with the instantiated data types. Generally it is crucial to take care of the exact spelling of the placeholders and Widget names. Otherwise a correct assignment of the channels to the Widgets is not possible.

In the following tables an overview of all names and their channel assignment is shown.

Table 2: Used placeholders

Placeholder	Description	Usage
{Sensor}	Name of the sensor in the configuration	Specific values
{Antrieb}	Name of the drive element in the configuration	Diagnosis data of one drive element
{Antrieb1} .. {Antrieb30}	Name of the chosen drive element	Overview of drive elements

Table 3: Widget names for spectra

Widget name	Channel assignment	Example Channel name
Signal_OSP	Order spectrum of one sensor in one gear	OSP#Sensor1#G1
Signal_HKOSP	Envelope order spectrum of one sensor in one gear	HKOSP#Sensor1#G2
Signal_FSP	Frequency spectrum of one sensor in one gear	FSP#Sensor1#G3
Signal_HKFSP	Envelope frequency spectrum of one sensor in one gear	HKFSP#Sensor1#G4

© 2023 imc Test & Measurement GmbH

Table 4: Widget names for Base Diagnosis

Widget name	Channel assignment	Example Channel name
Status_VEff	Gear dependent status channel of r.m.s. of vibration velocity	Sensor1#G1#VEff#Status
Signal_VEff	r.m.s. of vibration velocity and corresponding warn and alarm thresholds	Sensor1#VEff#Alarmschwelle
Status_AEff	Gear dependent status channel of r.m.s. of vibration acceleration	Sensor1#G2#AEff#Status
Signal_AEff	r.m.s. of vibration acceleration and corresponding warn and alarm thresholds	Sensor1#AEff#Warnschwelle
Status_AEffHK	Gear dependent status channel of r.m.s. of vibration acceleration envelope	Sensor1#G3#AEffHK#Status
Signal_AEffHK	r.m.s. of vibration acceleration envelope and corresponding warn and alarm thresholds	Sensor1#AEffHK

Table 5: Widget names for Diagnosis data

Widget name	Channel assignment	Example Channel name
Signifikanz_{Damage}	All significance values of the chosen drive element at the current damage pattern	Antriebswelle#LoseTeile#S2#Grundordnung_OSP#Signifikanz
Amplitude_{Damage}	All amplitude values of the chosen drive element at the current damage pattern	Antriebswelle#LoseTeile#S2#Grundordnung_OSP#Amplitude
Ordnung_{Damage}	All order values of the chosen drive element at the current damage pattern	Antriebswelle#LoseTeile#S2#Grundordnung_OSP#Ordnung

The following listing shows all drive elements and their available damage pattern that are used in the diagnosis data. It must be used the exact given spelling of the name for these damage pattern.

Table 6: Names for damage pattern

Drive Element	Damage Pattern	Name
Shaft	Imbalance	Unwucht
	Loose particles	LoseTeile
Drive shaft	Imbalance	Unwucht
	Loose particles	LoseTeile
	Link damage or misalignment	Gelenkfehler
Clutch	Misalignment	Ausrichtfehler
Rolling Bearing	Cage damage	WL_Kaefigschaden
	Rolling element damage	WL_Waelzkoerperschaden
	Rolling element damage spin	WL_WaelzkoerperschadenS
	Outer ring damage	WL_Aussenringschaden
	Inner ring damage	WL_Innenringschaden
Gear Drive	Revolving tooth damage	UmlaufenderVerschleiss
	Local irregularity wheel	FehlerRad
	Local irregularity pinion	FehlerRitzel
Planteray Gear	Revolving tooth damage	UmlaufenderSchaden
	Local irregularity planet	FehlerPlanet
	Local irregularity sun	FehlerSonne
	Local irregularity annulus	FehlerHohlrad
Belt Gear	Local irregularity belt	FehlerRiemen
Three-Phase Motor	Magnetical imbalance	MagnetischeUnwucht
Steady Frequency	Steady frequency	FesteFrequenzSchaden
Steady Frequency Envelope	Steady frequency envelope	FesteFrequenzHKSchaden
Steady Order	Steady order	FesteOrdnungSchaden
Steady Order Envelope	Steady order envelope	FesteOrdnungHKSchaden

12 Channel types, variable types, data types and data storage formats

In imc STUDIO, the various components work with variables, and some work with files as well. Variables can take a variety of formats and types.

This chapter provides an overview of the various possibilities and applications.

Overview

Synopsis	Section
What kinds of data types are available? What are the differences between them?	• Data types 1130 • Structured data sets - Event-based or sequenced channel 1131 • Time-stamped channels - Setting the RAM-buffer duration, circular buffer duration and pretrigger duration 1132
What variable types and channel types are there? Where do the variables come from?	• Monitor channels 1135 and GPS variables 1146 • Trigger, devices and system variables 1138 • Process vector variables 1141 and Ethernet bits 1137 • User-defined variables 1149
In what format are the variables saved? How to export/import configurations?	• Data file format - Saving measured data during measurement 1153 • Memory requirements for measurement data 1155 • Parameter set 1159

12.1 Data types

The main characteristic of a variable is its data type. The following list presents a select number of data types. Additional, personally designed types are also possible. Some can be generated by means of Scripting, others by using imc FAMOS.

Data type	Description
Normal data set	A normal channel (equidistant) is a one-dimensional field (vector) of real numbers with units, names etc. A sequence of sampled values of a measured quantity with a fixed sampling rate represents a normal channel. These include for example the analog channels, equidistantly captured fieldbus channels, ...
Single values	A single value is an individual real number having a name and in some cases a unit as well. These include, for example: Display-, Process vector variables, user-defined numeric variables, ...
Bits	A variable having the values 0 or 1. These include, for example: Ethernet-bits, virtual bits
Text-variables	A user-defined text. These include, for example: user-defined text-variables

Data type	Description
XY-data sets	Time-stamped channel with numerical values. XY-data sets arise where two values must be stored for each data point, namely, the (amplitude) value and additionally the time- or x-value. Example: CAN-channel with the setting "x-axis": "*Time-stamp*" Observe the notes on making the settings for RAM-buffer duration, circular buffer duration and pretrigger duration 1132.
TimeStamp-ASCII (TSA)	Time-stamped channel with ASCII-values, also known as TSA. Such data sets contain a list of texts (or generally, blocks of ASCII-code), each of which comes with its own time stamp (similar to XY data sets). TSA-channels are used for the message protocols of the various fieldbusses. Example: RecordEvent/RecordText, various fieldbus protocols, ... Observe the notes on making the settings for RAM-buffer duration, circular buffer duration and pretrigger duration 1132.
Vectors	A defined set of single values. These include, for example: various vectors from imc Online FAMOS or imc Inline FAMOS or Automation 1580.
Text array	Text arrays from imc FAMOS can't be read and evaluated in imc STUDIO by ordinary means. Widgets are not able to handle text arrays. It is possible to access variables via a script. A text array can be imported from a file. Exchange in both directions can be accomplished by means of an imc FAMOS-command. Thus, text arrays can be created using imc FAMOS, transferred to imc STUDIO and then passed back to imc FAMOS at a later time, in order to be filled by imc FAMOS with additional entries.

12.1.1 Structured data sets - Eventierter oder sequentierter Kanal

Normally a data set consists of a series of measurement values sampled at a fixed sampling rate. Various applications require options for concatenating data sets.

Segments

Example: The imc Online FAMOS function "FFT" - This function generates one new segment for each time a user-specified number of points accumulates.

A data set may be segmented, with each segment having the same length and the entire data set's length being a whole number multiple of the segment length. The segmented data sets also include matrices.

Events

Example: Multiple trigger-events in one data set.

A data set can also be structured as several events, which, unlike segments, can take different lengths. Thus, an additional list containing the beginning and length of each event must be managed along with the data set. Various data set characteristics, such as trigger time and sampling rate, are also managed in the capacity of characteristics of particular events.

Reference See also

Storage options and directory structure > "*Events and circular buffer* 422"

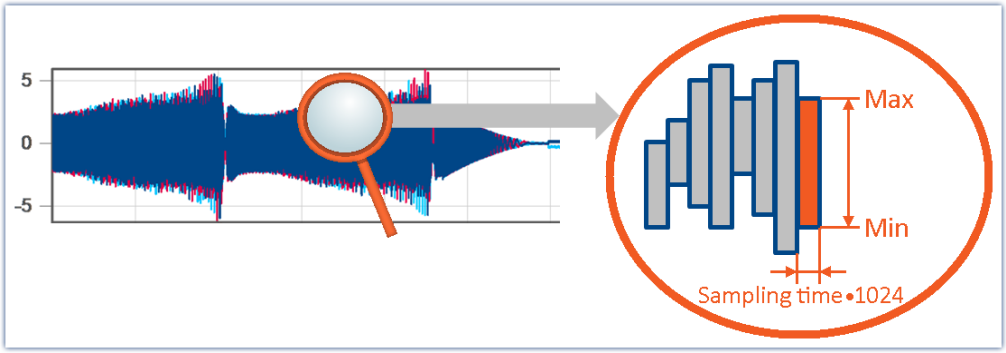
12.1.2 Envelope type monitor channel (min-/max-channel)

Some analog channels can generate **a reduced** envelope type monitor channel (min-/max-channel). The envelope ensures a complete overview of the measurement and smooth changing of the display even at a high data rate, without burdening the network with an excessive data rate.

The system cyclically condenses **x data points** down to **two new values**: the respective **minimum** and **maximum values** of the x points. The channel's data rate is thus reduced by a factor of x/2.

This reduced count of values is displayed as an envelope curve:

- With imc EOS monitor channels, the factor x is permanently set as 1024.
- With imc ARGUS*fit* monitor channels, the factor x can be freely specified in the *preprocessing* 366.



Envelope type monitor channels
Example of reduction by 1024 points
Resulting reduction of the data rate: 512

Envelope type monitor channels are available on the Setup page: "*Analog Inputs*" under the channel type: "*Monitor Channels*".

- The imc EOS monitor channels are by default envelope type monitor channels.
- For imc AGRUS*fit* monitor channels, the preprocessing must be set to "*Envelope*".

Note Signal processing

The envelope type monitor channels are not suitable for signal processing computations in, for example, imc Online FAMOS/imc Inline FAMOS!

12.1.3 Time-stamped channels - Setting the RAM-buffer duration, circular buffer duration and pretrigger duration

For time-stamped numeric channels (XY-data sets)

The RAM-buffer duration, circular buffer duration and pretrigger duration are specified as a time duration on the Setup pages. For time-stamped channels, the imc STUDIO software assumes an average sampling rate $S_{assumed}$ of 100 samples per second. This is necessary in order for the software to convert the specified time duration into a buffer size expressed in samples.

If the user knows the actual average sample rate S_{real}, then the duration to set can be calculated according to the following equation:

$$t_{toset} = t_{real} \cdot \frac{S_{real}}{S_{assumed}}$$

t_{toset}	the RAM-buffer duration, circular buffer duration or pretrigger duration [s]
t_{real}	the desired actual RAM-buffer duration, circular buffer duration or pretrigger duration [s]
$S_{assumed}$	the average sample rate assumed by the software $S_{assumed} = 100\ \mathrm{Samples/s}$
S_{real}	the actual average sample rate in Samples/s

Example

The actual sample rate S_{real} is 1200 Samples/s
The desired buffer duration t_{real} is 1min.

$$t_{toset} = t_{real} \cdot \frac{S_{real}}{S_{assumed}}$$

$$t_{toset} = 60\ \mathrm{s} \cdot \frac{1200\ \mathrm{Samples/s}}{100\ \mathrm{Samples/s}}$$

$$t_{toset} = 720\ \mathrm{s}$$

For time-stamped text channels (TSA)

Time-stamped text channels (TSA-channels) are used for, among other things, message protocols of the various fieldbusses. A sample in such a channel consists of a time stamp and text or payload byte of arbitrary length, plus multiple bytes specific to the TSA format. The RAM-buffer duration, circular buffer duration and pretrigger duration for these channels is specified in the dialogs *Configuration or Storage* as a time duration. The program assumes an average sampling rate $S_{assumed}$ of 100 samples per second, where for each sample a size $L_{assumed}$ of 20 bytes per sample is assumed. These 20 bytes correspond to about the size of one sample (or entry) in the fieldbusses message protocol. These presumed values are necessary for the program to be able to convert the specified buffer duration to a buffer size in bytes.

In order to determine the RAM-buffer duration, circular buffer duration and pretrigger duration t_{toset} to set, the actual average data rate D_{real} must be calculated. This data rate depends on two estimated values.

$$D_{real} = S_{real} \cdot (L_{real} + 12\ \mathrm{Bytes/Sample})$$

D_{real}	the actual average data rate [Bytes/s]
S_{real}	the actual average sample rate [Sample/s]
L_{real}	the actual average number of payload bytes per sample [Bytes/Sample]

If the user knows the actual average data rate D_{real}, then the duration to set can be calculate according to the following equation:

$t_{toset} = t_{real} * D_{real} / D_{assumed}$

t_{toset}	the RAM-buffer duration, circular buffer duration or pretrigger duration [s]
t_{real}	the desired actual RAM-buffer duration, circular buffer duration or pretrigger duration [s]
$D_{assumed}$	the average data rate assumed by the program [Bytes/s] $D_{assumed} = S_{assumed} \cdot L_{assumed}$ $D_{assumed} = 100\ \text{Sample/s} \cdot 20\ \text{Bytes/Sample}$ $D_{assumed} = 2000\ \text{Bytes/s}$
D_{real}	the actual average data rate [Bytes/s]

Example

The channel is a TSA channel.
The actual average data rate S_{real} is 150 Samples/s
On average, 48 payload bytes per sample are contained: $L_{real} = 48\ \text{Bytes/Sample}$

$$D_{real} = S_{real} \cdot (L_{real} + 12\ \text{Bytes/Sample})$$

$$D_{real} = 150\ \text{Samples/s} \cdot (48\ \text{Bytes/Sample} + 12\ \text{Bytes/Sample})$$

$$D_{real} = 9000\ \text{Bytes/s}$$

The desired buffer duration t_{real} is 1 min.

$$t_{toset} = t_{real} \cdot \frac{D_{real}}{D_{assumed}}$$

$$t_{toset} = 60\ \text{s} \cdot \frac{9000\ \text{Bytes/Sample}}{2000\ \text{Bytes/s}}$$

$$t_{toset} = 279\ \text{s}$$

12.2 Various variable types - Channel type

Along with the data type, the variable's source impacts how the variables can be used.

When the source is a device, one usually speaks of channel types. These variables are configured in the Channel-table on the respective Setup-page. This applies to analog channels, Monitor channels, Ethernet-bits, ... A list of all channel types is presented in the section "*Channel table* 340"

There are also local variables on the PC. These are generated using the dialog "*User-defined variables* 1149".

In the next few section you will be introduced to a variety of variable types which require a detailed look and which can not be described adequately just in terms of their data type.

- Monitor channels 1135
- Ethernet bits 1137
- Trigger, devices and system variables 1138
- Process vector variables (pv-variables) 1141
- GPS 1146
- User-defined variables 1149

12.2.1 Monitor channels

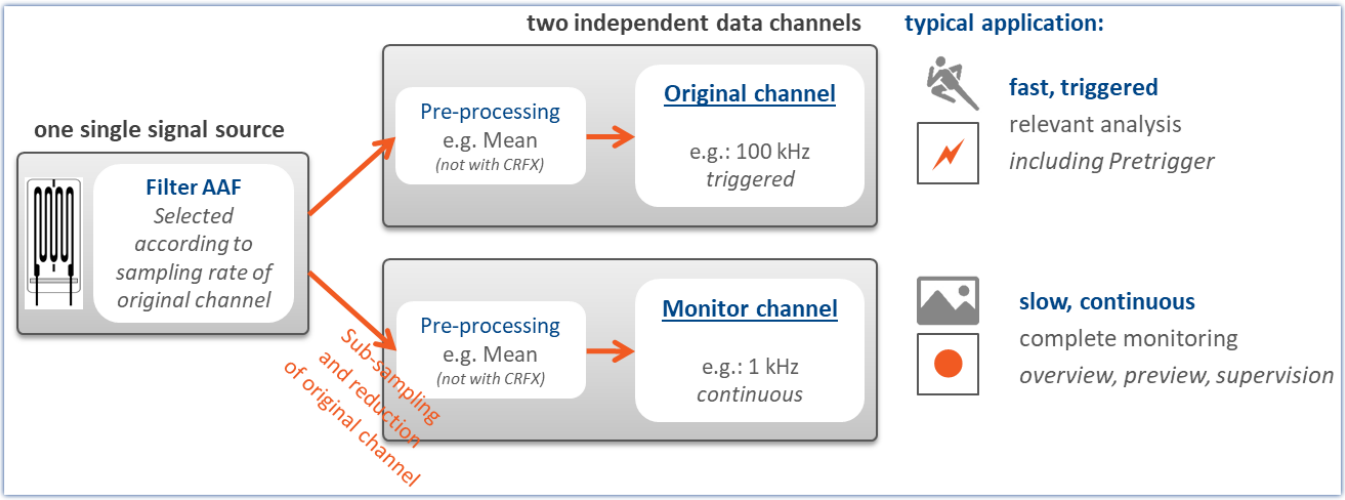
Monitor channels - For devices belonging to the firmware group A 187 (imc DEVICES)

Monitor channels are digital copies of the input signals and serve the purpose of continuous monitoring. Their settings for triggering, data storage and the **sampling rate** do not need to match those of the respective original channel. By means of monitor channels, it is possible to display the plot of captured signals before occurrence of the trigger event, at a low sampling rate.

Their **maximum sampling rate** is that of the input channel. The monitor channel does not have its own independent low-pass filter. For this reason, at low resampling rates, **aliasing-effects can occur**.

When an input channel is configured for "*preprocessing* 366", the monitor channel receives the unprocessed raw data.

The following channel types can have Monitor channels: Analog inputs, Digital inputs and Incremental counter inputs. Most fieldbus interfaces also offer an option for creating monitor channels.



Concept for monitor channels

Monitor channels - For devices belonging to the firmware group B 187 (imc DEVICEcore) - (from B11 187 - e.g. imc ARGUS*fit*)

Digital copies of the input signals of the same name for slow, continuous monitoring. Their settings for triggering, data storage and the **preprocessing** do not need to match those of the respective original channel. By means of monitor channels, it is possible to display the plot of captured signals before occurrence of the trigger event, at a low sampling rate.

The **data reduction** is configured by the "*preprocessing* 366". The sampling rate itself is **not changeable** and matches that of the original channel.

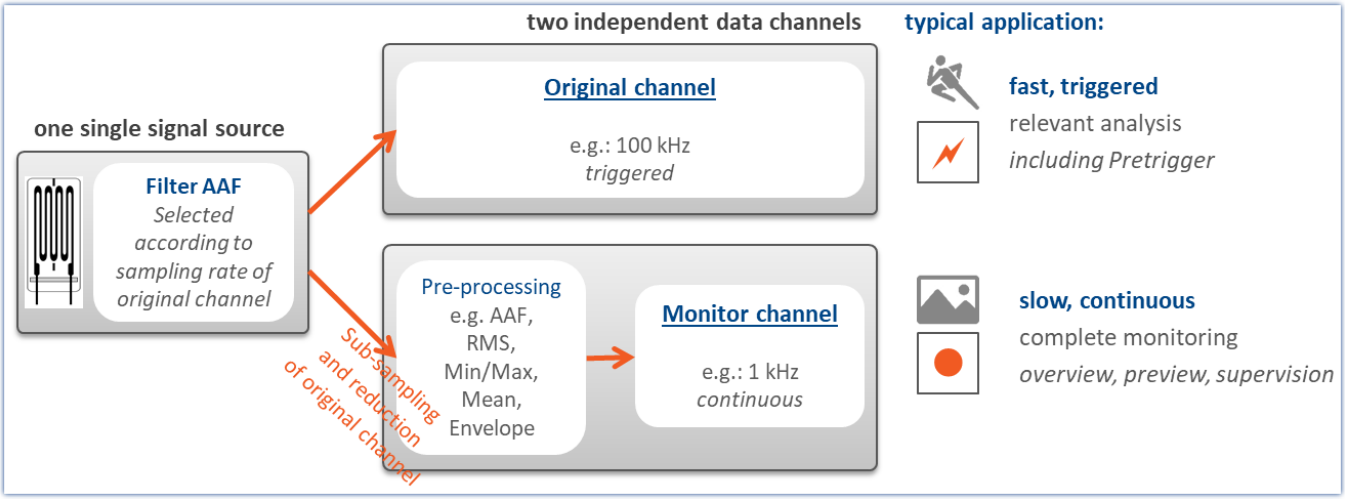
In order to **avoid aliasing effects**, the Monitor channel can be provided with its own AAF in the "*preprocessing* 368". Additionally, **further data reduction** such as RMS and envelopes are possible by means of the preprocessing.

Note — Other advantages of preprocessing

- Reduction of the data rate by means of **filtering, RMS-value, or min/max-statistics**.
- **Trigger conditions** can be defined based on the RMS-values or on the filtered values. In this way, the sound intensity, for example, can serve as the trigger source.
- **Envelope representation** allows reduction of the data rate without loss of visual information.

The following channel types can have Monitor channels: Analog inputs.

Each Monitor channel comes with its own process vector variable (pv-variable), which reflects the instantaneous value. This is performed in an analogous way to the original channel. The value is read out after preprocessing of the Monitor channel.



Concept for monitor channels

Envelope type monitor channels - applicable to imc EOS devices - device group B10 187

Every analog channel can **additionally return a data-reduced** envelope type monitor channel (min-/max-channel) - See "*Envelope type monitor channel (min-/max-channel)* 1132"

12.2.2 Ethernet bits

imc STUDIO can derive trigger conditions from its own analog and digital channels as well as from AND/ OR combinations of these. When working with multiple devices, it's often necessary to announce a trigger event to all devices.

When working with multiple channels, it is often necessary to announce a trigger event to all the devices. It is often difficult to lay digital lines in a widely dispersed network. One solution is to use an Ethernet-connection, if available. This is accomplished using the so-called "*Ethernet bits*". One way to think of them is as 32 hard-wired digital lines running parallel to the Ethernet, which connect all devices mutually. Thus the device has an Ethernet-DIO port, in addition to the physical DIO. These digital inputs and outputs can be used for triggering just like the unit's physical DIO-bits.

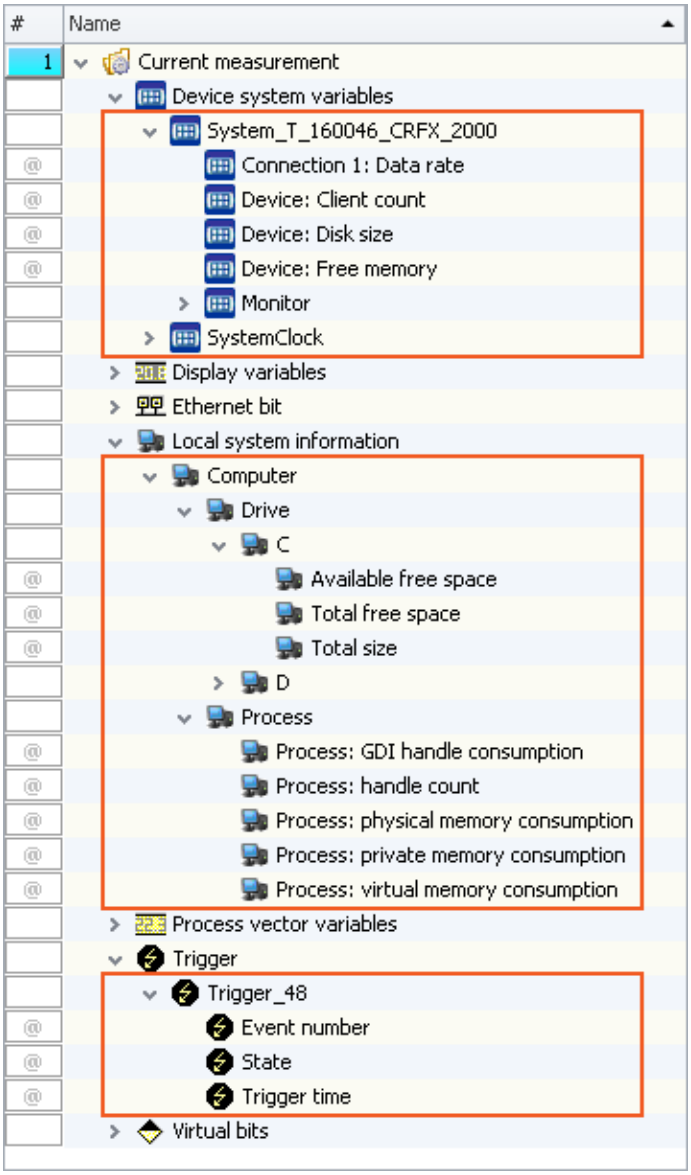
In this context, note the following:

- Ethernet bits only take effect within the scope of one protocol.
- The ethernet bits of TCP/IP devices only take effect within their Subnet.
- Another analogy to the physical DIO also applies: just as in the Ethernet, there is a delay between the setting of a bit and the recognition by another device of the new state.

Note

When a measurement is prepared, all bits, including the Ethernet bits, are reset. This also affects devices on the network which are currently performing measurements. Any Ethernet bits which had been set in the process of the running measurements will be reset when any device in the network is prepared

12.2.3 Trigger, devices and system variables



In an expanded tree diagram, the following variables are displayed:

Variables	Description
Device system variables	Utilization and info on device status, storage space, measurement status, connection status and synchronization status, as well as time information
Local system information	User information, system load and hard disk status
Trigger	Status and information about the triggers

These variables contain various kinds of information which are each displayed separately and which can easily be moved to the Panel by means of Drag&Drop.

Trigger		Description
Trigger Name e.g. Trigger_01 or BaseTrigger	Event number	With multi-triggering: count of previous trigger releases
	State	Current state of the trigger: armed, released, stopped
	Trigger time	Time when the state last changed

Device system variable <Device name>	Description
Free memory	Available memory in internal data storage medium
Disk size	Capacity of internal data storage medium
Client count	Count of active connections to the device. E.g. imc STUDIO, imc STUDIO Monitor or imc REMOTE connections or a data storage medium used. If imc REMOTE WebServer is enabled in the device properties, a connection is permanently occupied.
Connection 1	First detected connection with the device (imc STUDIO or imc WAVE).
Data rate	Currently transmitted Bytes/s at the receiver (instantaneous value)
Critical channel fill level	Proportional (in percent) fill level of the "critical channel's" FIFO. The "critical channel" is the channel with the highest fill level. The fill level is allowed to rise temporarily and have high peaks. However, it may never reach 100%, or a data overflow occurs in consequence. If the fill level rises not temporarily but constantly, this indicates an overflow will occur later. To prevent data overflows, see the notes in these chapters/sections: "*Setup pages - Configuring device*" > "*Information and Tips*" > "*Avoidance of Data Overflow* 755".
Critical channel name	Name of "critical channel", see "*Critical channel fill level*".
Connection 2 Monitor 1	First connection with imc STUDIO Monitor (corresponds to "*Connection 1*").
Connection x Monitor n	Additional connections

Device system variable System	Description
Experiment	
Measurement status	Returns the state of the measurement: measurement running (1), measurement stopped (0), unclear (-1) • 1: Measurement running applies when **at least one device** is performing a measurement • 0: Measurement stopped applies when **all devices** are known not to be performing measurement • -1: Unclear applies when **at least one device** is not connected, and **no connected device** is running
Synchronization status	Returns the device's synchronization status: Synchronized (1), not synchronized (0) • 1: Synchronized, when **all devices** are synchronized • 0: Not synchronized, when **at least one device** is not synchronized Devices which are not intended to be synchronized are not included in the status check.
Connection status	Returns the status of the device's connection with imc STUDIO: connected (1), not connected (0) • 1: Connected, when **all devices** are connected • 0: Not connected when **at least one device** is not connected
SystemClock	
PC time	Current PC clock time
System time	Current clock time of imc STUDIO ("VRTC 307"). This time is used for all components of imc STUDIO.

Local system information	Description
User	Using this variable, you can automatically include an indication in your report of which user is logged-in.
Name	Name of the logged-in user.
Role	Allocated role of the logged-in user.
Computer	
Drive n	Information about the local data carrier. Pay attention to the free memory space if you are trying to save large volumes of data.
Total free space	Free memory on the local data carrier. This property reflects the total amount of free memory on the drive, not just the portion available to the current Windows user. In most cases, the parameter "Available free space" is to be used.
Total size	Data storage capacity of the local data carrier.
Available free space	Free memory space on the local data carrier. This property reflects the amount of free memory space available to the current Windows user.
Process	Various kinds of system information. For information and value limits, see the associated Internet forums. Monitoring these values can be of interest in long-term testing.

12.2.4 Process vector variables (pv-variables)

The "*Process vector*" is a collection of **single-value variables** (process vector variables or pv-variables). **For all measurement channels** of the device, these single-value variables are already **pre-defined**.

Note Values are available immediately

Updating of the device pv-variables occurs immediately after preparing the measurement device. Thus, the pv-variables always display the current value, even if the measurement is not yet running, or has already concluded, or the channel-trigger has not yet released.

imc Online FAMOS works independently of the trigger mechanism and can thus access the measured values currently being captured by means of the variables. These can then be used, in conjunction with the "*Synchronous Tasks*", for monitoring and control purposes.

Variants of the pv-variables

Variants	Description
pv-variables of the data acquisition channels	pv-variables of the data acquisition channels (e.g. analog channels, CAN, ...) always show the currently captured measurement value starting from the action "*Prepare*".
User-defined pv-variable	pv-variables personally defined by the user, e.g. by means of imc Online FAMOS. Prerequisite: A firmware group A 187 device.
Display GPS Information 1146	GPS signal are made available as pv-variable and fieldbus channels (with GPS interface)
State of the external power "*pv.State.ExternalPower*"	The pv-variable takes the following value: 0 = state unknown/not yet found (initialization value) 1 = external current supply present ("power ok") 2 = external current supply not present/deactivated ("power fail")
n of system time "*pv.State.SyncTimeDeviviation*"	Only in conjunction with NTP- and PTP-synchronization Current time deviation of devices from the NTP-or PTP-Server in seconds. The return value is exactly 0 if the server is ether not activated or cannot be reached; otherwise it is in the range of a few milliseconds.

User-defined pv-variable

Prerequisite: A firmware group A 187 device.

In parallel to the already existing device-pv-variables, you can create your own pv-variables. This is conditional on the following requirements:

Component | Rights	read	create	write
imc Online FAMOS	yes	no	no
imc Online FAMOS Professional	yes	yes	yes
imc Inline FAMOS	yes	no	yes (if they were created using imc Online FAMOS Professional)

Creating a user-defined pv-variable

You can create user-defined pv-variables from a variety of locations (prerequisite: "*imc Online FAMOS Professional* 856"):

- imc Online FAMOS 880 with Control Commands
- Automation 1580
- Setup page: "*Variables*" via the menu ribbon: "*Setup-configuration*" > "*Create Process vector variable* 205"

Properties of the pv-variables

Properties	Description
Trigger independence	Updating of the device pv-variables occurs immediately after preparing the measurement device. Thus, the pv-variables always display the current value, even if the measurement is not yet running, or has already concluded, or the channel-trigger has not yet released.
Disallow read-/write-access	By means of the variable's "*Status*" (Setup page "*Variables*"), you are able to set whether other additional components are able to see the variable.

Status	Description
Active	The variable exists in the device and is transferred to the PC. The following rules apply to user-defined pv-variables: **No write-permission on the PC**: The value can only be written via imc Online FAMOS. Other components such as the Panel can only read the value but not write it. If the initialization in imc Online FAMOS is removed, the variable remains in the list. The measurement can only be started if the variable is initialized in at least one location (e.g. imc Online FAMOS or Automation). Otherwise this error message appears: Error number: 6408 "*There is at least one process vector variable that is not being used. Make sure that it has write permission (Setup) or is being used (e.g. imc Online FAMOS, Automation, ...)*"
Passive	The variable exists in the device, but is not transferred to the PC. It is only possible to access the value via imc Online FAMOS. For user-defined pv-variables, the following applies: If the initialization in imc Online FAMOS is removed, the variable is deleted from the list.
read/write	Like "*active*", but **with write-permission on the PC**; e.g. the value of the variable can be changed by means of the Panel.
write	*not recommended* - The status has no significance and usually behaves like "*read/write*"

Direct read-access via some fieldbusses	Appropriate variants of the CAN-Bus and LIN-Bus are able to read the current values of the pv-variables and to output these on the bus. More detailed information including the prerequisites are presented in the descriptions for the respective fieldbusses.

Deactivation of pv-variables for data acquisition channels

For devices belonging to the firmware group A 187 (imc DEVICES)

By default, pv-variables are activated for the data acquisition channels (e.g. the analog channels).

Even if they are not used, the make great demands on the device system resources. Therefore, your device's entire aggregate sampling rate 369 is only available if the pv-variables are deactivated. Otherwise you receive messages indicating that the aggregate sampling rate has been exceeded, although this is not true mathematically.

How to **deactivate the pv-variables** for data acquisition channels is described in the section: Setup > "*Measurement options* 338".

For devices belonging to the firmware group B 187 (imc DEVICEcore)

Deactivation is not possible.

Names for device-pv-variables

If multiple devices are used in an experiment, a pv-variable's name is automatically extended with the device name (complete name). The name supplement is applied to the pre-defined pv-variables.

E.g. "pv.GPS.longitude_imcDev__05123456": pre-defined pv-variable for the GPS position of device "imcDev__05123456".

Data type of the pv-variables

A pv-variable has a defined data type. You can create the following data types by means of, for instance, imc Online FAMOS / Setup or Automation:

For devices belonging to the firmware group A 187 (imc DEVICES)

Data type	Description	imc Online FAMOS	Other
32 bit integer	Signed integer	Initialization: `int pv.myPV_Int = 123`	**Setup**: 32 bit integer **Automation**: Integer

TI-float	Real number (floating point)
	Optimized for imc Online FAMOS-processing
		Initialization:
		`pv.myPV_float = 123.45`
		`float pv.myPV_float = 123.45`
			Setup: TI-float
			Automation: Floating point

Float	Real number (floating point); IEEE-Float e.g. from CRFX/CRXT- or CAN channels	Initialization: --- Only for application in calculations	**Setup**: Float **Automation**: ---

12.2.4.1 Saving/Restoring pv-variables

Prerequisite: A firmware group A 187 device.

When a measurement is concluded or the device is deactivated, the current values of pv-variables can be saved in a way so that they are restored as the initial values upon restarting the measurement.

The current values are saved to a file in the device. This storage file is in non-volatile memory and remains intact even when the device is off.

Note

- There is only one storage file in the device; separate files are not created for the various experiments!
- The decision whether a pv-variable is restored with the same name depends on the restoration strategy used.
- A fully functional UPS is required.

Declaration

To save and restore pv-variables in the device, the desired pv-variables are provided with an attribute.

In an imc Online FAMOS program, this attribute is set using the keyword `restore` with a pv-variable is declared:
`restore pv.Counter = 1`

Save/Restore times

The current pv-variable values are saved at the following times:

- before (re-)configuration (configuration by user PC, Diskstart/Autostart); more precisely: after data capture is stopped
- when the device is deactivated (event POWER SHUTDOWN)

The pv-variable values are restored at the following times:

- after (re-)configuration (configuration by user PC, Diskstart/Autostart); more precisely: after opening and starting the online task and before starting the measurement

Restoration strategies

The attribute has a parameter for specifying the restoration strategy.

A variety of restoration strategies can be specified:

- Matching name, data type, scaling and comment
  By specifying a string (as the comment), it is possible to distinguish the same pv-variable name in different tasks (e.g. different experiments).
  Note: This restoration strategy is used by the operator software "imc DEVICES" (imc Online FAMOS).
- Matching name, data type, Idx, system address, and scaling (exact matching)
  Note: If the original experiment is changed, this usually results in a change of the Idx or system address, so that the values of the pv-variables are not restored!
- Matching name and comment
  By specifying a string (as the comment), it is possible to distinguish the same pv-variable name in different tasks (e.g. different experiments).
  Note: This restoration strategy is used by the imc STUDIO plug-in Automation.

Deleting the process vector backup file

To delete the saving file, proceed as follows:

- Open the plug-in **Setup**
- Open the page **Devices**

By means of the parameter **Process vector backup file**, you can delete the file. Add the parameter as a column in the table or use the **Details** tool window (hidden parameters must be made visible).

- Select the devices desired in the Device Table
- Click on the button *Delete* for the parameter **Process vector backup file**

What happens when you change experiments?

- The pv-variable remains intact in the device if an imc Online FAMOS variable of the same name and of the same type (Integer or Float) exists.
- If an experiment is run which does not include the variable, the variable is deleted from the process vector saving file and is thus no longer present.
- If the variable is used in the context of the imc STUDIO Automation, it remains intact as long as the Automation Task of the same name is present.

12.2.4.2 Update rate of pv-variables

The current measurement values for the Process Vector are updated at the respective **channel's clock rate**, but not higher than 10kHz. The filter that has been set for the channel is also used, exactly as for the channel's triggered acquisition.

Data from **imc CRONOS*flex* modules** are transmitted at a **maximum transfer rate** of 5 kHz (200 µs). Data recorded at higher sampling rates are transferred in blocks. This means that pv-variables of imc CRONOS*flex* modules are **updated at a maximum rate of 5 kHz**. The pv-variables are thus subject to the following constraints:

pv-variables from channels with sampling rates < 5 kHz can only be transferred at multiples of 200 µs:

Channel sampling rate	Update rate of the PVV
5 kHz	5 kHz
2 kHz	**1 kHz**
1 kHz	1 kHz

12.2.4.3 pv-variables as monitor channels

In the plug-in Panel, it is possible to link Widgets with the pv-variables. Since these are already updated following the preparation, in other words before release of a trigger, this technique is a simple alternative to monitor channels.

12.2.5 GPS

At the GPS socket it is possible to connect a GPS-receiver. This makes it possible to achieve absolute **synchronization to GPS time**. If the GPS-mouse has reception, the measurement system synchronizes itself automatically. **Synchronization with a NMEA source** is possible. The precondition for this is that the clock must return the GPRMC-string along with the one-second-interval clock signal.

All **GPS information** can be **evaluated** and subjected to **subsequent processing** by OFA/IFA.

GPS signals are **available as**: process vector variables and fieldbus channels.

GPS information	Description
pv.GPS.course	Course in °
pv.GPS.course_variation	Magnetic declination in °
pv.GPS.hdop	Dilution of precision for horizontal
pv.GPS.height	Height over sea level (over geoid) in meter
pv.GPS.height_geoidal	Height geoid minus height ellipsoid (WGS84) in meter
pv.GPS.latitude pv.GPS.longitude	Latitude and longitude in degree (Scaled with 1E-7)

GPS information	Description
pv.GPS.pdop	Dilution of precision for position
pv.GPS.quality	GPS quality indicator 0 Invalid position or position not available 1 GPS standard mode, fix valid 2 differential GPS, fix valid ...
pv.GPS.satellites	Number of used satellites.
pv.GPS.speed	Speed in km/h
pv.GPS.time.sec	The number of seconds since 01.01.1970 00:00 hours UTC. For this reason, it is no longer possible to assign the value to a Float-format channel without loss of data. This count of seconds can be transformed to absolute time under Windows and Linux.To do this, use the function below. `MySeconds = CreateVChannelInt( Channel_001, pv.GPS.time.sec)`
pv.GPS.vdop	Dilution of precision for vertical see e.g. www.iota-es.de/federspiel/gps_artikel.html (German)

Note

Scaling of the latitude and longitude

pv.GPS.latitude and pv.GPS.longitude are **INT32 values**, **scaled with 1E-7**. They must be **treated as Integer channels**, otherwise the **precision is diminished**.

By means of imc Online FAMOS, you are able to generate virtual channels from them. However, due to the reversal of the scaling, precision is lost:

```
latitude = Channel_001*0+pv.GPS.latitude *1E-7
```

Recommendation: Use the corresponding fieldbus channel: "*GPS.latitude*" or "*GPS.longitude*". Here, no scaling is required, so that the precision is preserved.

Sampling rate

Due to system limitations, GPS channels for determining the fastest sampling rate in the system are not taken into account. For an working configuration, at least **one other channel** (fieldbus, digital or analog) must be sampled at either the **same** sampling rate as the GPS-channel, or a **faster** one.

Internal variables; do not use

- pv.GPS.counter
- pv.GPS.test
- pv.GPS.time.rel
- pv.GPS.time.usec

GPS-Receiver

The **GARMIN GPS receivers** supplied by imc are set ready for operation and provide a 1 Hz or 5 Hz pulse, depending on the model.

The following conditions must be met in order to use other GPS receivers from imc devices:

- **RS232 port settings**
  - **Baud rate**: Possible values are 4800, 9600, 19200, 38400, 57600 or 115200
  - 8 bit, 1 stop bit, no flow control
- The following **NMEA strings** must be sent: ***GPRMC, GPGGA, GPGSA***. The order of the strings must be adhered to.
  Additional strings should be deactivated. If this is not possible, all other strings must be **before** the GPGSA string!
- The receiver must deliver a **1 Hz clock**.
- The rising edge of the clock must mark the second specified in the next GPRMC string.
- All three strings should be sent as soon as possible after the 1 Hz clock, so that there is sufficient time for processing between the last string and the next 1 Hz clock.

NMEA-Talker IDs

Supported NMEA-Talker IDs:

- GA: Galileo Positioning System
- GB: BeiDou (BDS) (China)
- GI: NavIC (IRNSS) (India)
- GL: GLONASS, according to IEIC 61162-1
- GN: Combination of multiple satellite systems (GNSS) (NMEA 1083)
- GP: Global Positioning System (GPS)
- GQ: QZSS regional GPS augmentation system (Japan)

12.2.6 User-defined variables

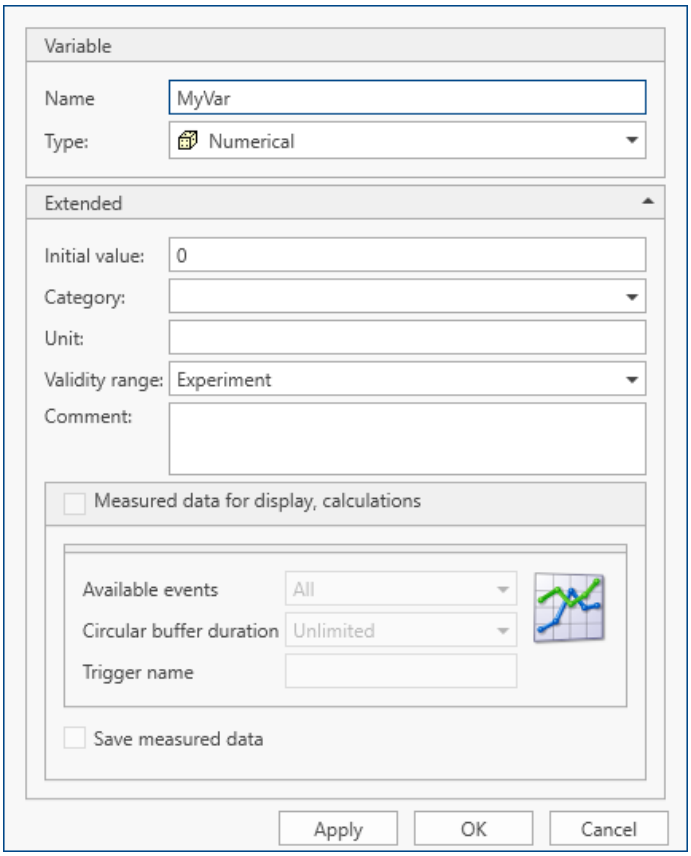
imc STUDIO provides the user with the ability to **create** and use **user-defined variables**.

Note Variables are not available in the device

User-defined variables only exist on the PC. It is not possible to access them from the device.

Creating variables

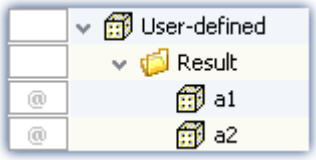
You can create user-defined variables in the Data Browser by clicking on the symbol . Furthermore, at certain locations where variables can be used, you will find a button with the caption "*Create*", for example in the command "*Set Variables* 1764". A dialog opens which helps you in setting up the variable.



Create User-defined Variable dialog

The dialog contains two regions. "*Variable*" contains an input box for the name and a selection box for the type. In the lower region "*Extended*", you can specify additional properties such as the "*Initial value*" and "*Unit*".

Variable

Parameter	Description
Name	Here, enter a name for your variable. All alphanumeric characters are permitted. If you use the name of an already existing variable, the existing one is deleted and a new one with the specified properties is created.  *Display in the Data Browser* To join multiple variables together in a group, you can use the character '.' in the variable's name. E.g. *Result.a1*, *Result.a2* results in this display in the Data Browser.
Type	For your user-defined variable, you can choose among various types: **Typ** — **Description** Numerical — Saves a **number**; this type suitable both for integer values and for floating points numbers. Under *Initial value*, you can set the value with which to initialize the variable (see the section *Extended*). The default for this value is 0.0. Text — Saves an **arbitrary text**. Under *Initial value*, you can set the value with which to initialize the variable (see the section *Extended*). The default for this value is an empty string, represented by the readout *Empty Value* in gray letters.

Extended

Parameter	Description
Initial value	Your variable is initialized with this value. This means that when creating the variables, when loading an experiment and upon resetting of the variables ("*Set Variables* 1764"), the value entered here is written in the variable. In this box, it is only possible to enter anything if the variable's type is **Numerical** or **Text**. With all other types, this box is disabled and the variable is empty when created.

Parameter	Description
Category	Assigning a **category** gives you the ability to structure the variables in the Data Browser. If there are already categories of user-defined variables, then to display them, click on the symbol in the box *Category* after entering the name. This expands a selection list of all previously existing categories. The categories only serve structuring purposes and do not influence what values the variables can take. **Custom variables created in custom main category** When you create a variable having no category, it is automatically placed among the "*User-defined variables*". When you create a variable having a category, this variable is not created under "*User-defined variables*" but parallel with that category. This allows you to create your own custom folder structures. If you wish to create a category within the category "*User-defined variables*", do the following: write a "\" before the category name. Category: "MeasurementPoint_1". A category called "MeasurementPoint_1" is created in **parallel with** the other categories, such as "*Analog Inputs*". It contains the variable. Category "\MeasurementPoint_1". This creates a category "MeasurementPoint_1" **within the category** "*User-defined variables*". It contains the variable. This applies, for example, to variables created by means of the command: "*Load variable* 1760 " or to generating a user-defined variable by means of the Data Browser.
Unit	You can assign a unit to your variables.

Parameter	Description
Validity range	The setting **Validity Range** determines the availability of your variable. What is saved is always only the **variable itself**, not the current value! **Validity Range** — **Description** Temporary — Temporary variables are associated with the session, which means they only exist until imc STUDIO is closed. Upon the next start they are no longer available. Experiment — Variables with this validity range are associated with the current experiment. They are saved along with it and are available again after loading. Sequencer — Variables with this validity range are associated with the Sequencer. You should use this validity range if you intend to load other experiments via the Sequencer. Your variables remain intact even if the new experiment has its own variables. Project — Variables with this validity range are associated with the current project. They are thus available for all of a project's experiments. Persistent — Persistent variables are project-dependent and globally available for all projects and experiments. Any "higher-level" variables are not overwritten if a variable of the same name but from a different validity range is loaded. Example: Suppose a variable exists with the validity range "*Project*" or "*Sequencer*". An experiment is loaded by means of the command: "*Open experiment*". A variable having the same name exists in this experiment and has the validity range "*Experiment*". This variable is not loaded from the experiment when the command is run. Please also observe the notes on the action: "*Load/Refill variable* 1223" in regard to the "Validity range".
Comment	It is possible to add a comment to a variable. It is seen in the Data Browser in the column *Comment*, or can be displayed in a Widget with the help of placeholders 166, for example.
Measured data for display, calculations	This option currently has no function.
Save measured data	This option currently has no function.

12.3 Data file format - Saving measured data and export/import formats

With imc STUDIO, you are able to save a variety of elements (variables, configurations) in various formats. You can either save the measured data directly during measurement, or export them subsequently in a different format. You can save device configurations and later import them again.

Next, a selection of available formats will be presented. Significantly more formats are available for export of variables than can be mentioned here. Please refer to the information on the respective "export template", if provided, for more details.

12.3.1 Data file format - Saving measured data during the measurement

By default, the measured data are saved in the file format imc FAMOS. The desired file format can be set in the file format selection list:

Format	Description
imc FAMOS	Default format for all device variables.
imc FAMOS ZIP 1153	Compressed format for many fieldbus variables.
CANAlyse-MDF 1154	CANAlzser-MDF format for CAN-Log data.

Note

"*imc FAMOS ZIP*" and "*CANAlyse-MDF*": Applicable with devices of the firmware group A 187 (imc DEVICES).

imc FAMOS

Each channel is saved in a separate file with the extension *.raw. raw stands for raw data, meaning the original data returned by the measurement device.

Reference

Detailed documentation on this file format is presented in the manual:
"imcSharedComponents.pdf" > File Assistant > File Format.
There you will find descriptions of the modules which jointly use imc FAMOS and imc STUDIO.

Measured data saved by your device can be read directly by imc FAMOS (required imc FAMOS version: see "*Technical data sheet*").

With which version and which device were the measured data generated?

The saved measurement files record with which version of imc STUDIO and with which firmware/3rd-party device/DataProcessing and which device the measurement was performed. For this purpose, the *.raw-file can be opened with a text editor. This info is readable whichever data format (imc2 or imc3) is used.

Note Conversion to EXCEL or ASCII

If you want to convert the imc measurement data into a EXCEL table or a ASCII file, use the component: "*imc Format Converter* 1191".

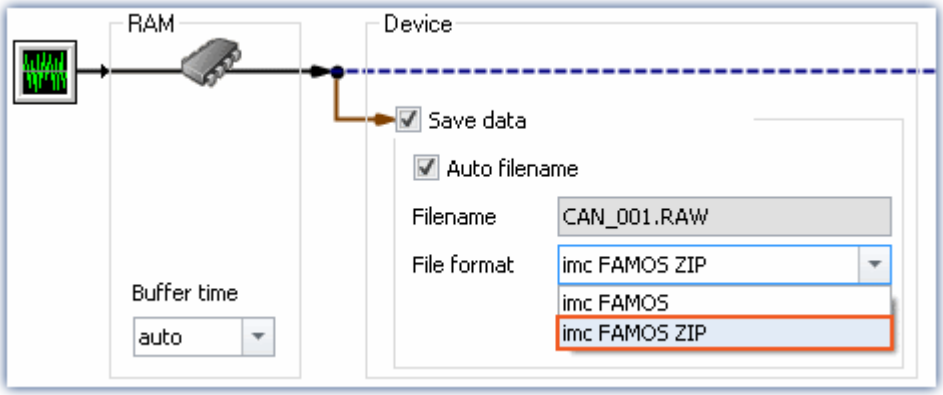
imc FAMOS ZIP

For certain channel types, it is possible to **reduce the disk space requirements** (file format: "*imc FAMOS ZIP*"). In the background, the file size is minimized by means of zip-compression. The result depends accordingly on the nature of the signal.

The filename remains unchanged. As well, the handling of the file, for instance with imc FAMOS is still the same (required imc FAMOS version: see "*Technical data sheet*").

Data compression is possible for the following channel types:

- analog/digital fieldbus channels
- log channels of the CAN-fieldbus
- digital input-ports



Note limitation: computing power of the basic system

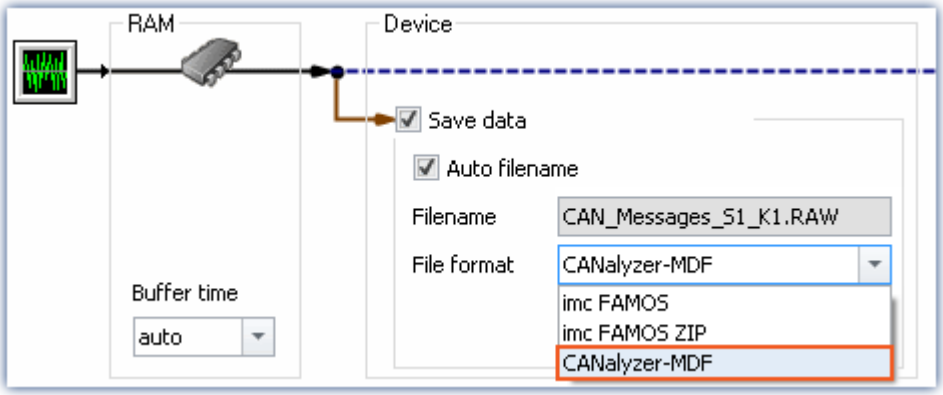
The compression of the data is limited by the computing power of the basic system. An **overload due to too many individually compressed channels can lead to a restart of the system!** The **number of channels** is the most important factor. Typical measurements with an imc BUSDAQ-4 showed that up to **60 channels** can be successfully compressed permanently.

If the compression is limited to the log channels, an overload does not lead to a restart. However, it is possible that compression may take so long that the data cannot be retrieved in time. In this case, the system reports a data overflow.

CAN-Log Data in CANAlyser-MDF format

Logs of CAN messages can be saved either in imc FAMOS format or in the CANAlyse-MDF file format.

This option is only available for the device.



Note

- The maximum possible measurement duration is 11 hours, 55 minutes and 49.67 seconds.
- This format is only available if "Log channel" is activated for the node in the CAN Assistant.

12.3.2 Memory requirements for measurement data

For measurements saved to the **internal storage medium**, note:

If the available memory is exhausted during measurement, the measurement continues anyway (display and additional saving to PC-hard disk, in some cases).

In order to estimate the memory requirement for a measurement of a particular length, one must first distinguish among the various data types. There are distinctions between **analog** and **digital**, **reduced** and **equidistant** data, as well as between **untriggered** and **triggered measurement**. Additionally, the "*cluster size*" of the storage medium used plays a role.

Data type:

- Analog data require 16 bits per sample (or for the data type: Float 4 Byte pro Sample (whether your device supports the data type: Float is stated in the respective device manual)
- Digital data require 16 bits per 16 bit port
- Virtual channels in imc Online FAMOS require 32 Bit per sample
- The same applies to channels reduced using Transitional Recording (only possible on the device).
- When estimating the memory requirements for virtual waveforms, it is necessary to take the function used into consideration. For example, the simple RMS-value of 1000 readings reduces the data mount by that factor.

Trigger:

- For each triggering event, a file header is saved, which provides additional information on the stored data (imc FAMOS file format).
- The size of this header depends on the particular settings (e.g. header of the first Event: 1536 Byte, header of the subsequent Events: 512 Bytes)

Note

- The calculated memory requirement is the minimum memory space requirement.
- For each file on the hard drive, there is a series of entries in the FAT (file allocation table) which determine which clusters are occupied by the file, i.e., where the file contents are physically located on the disk.
- The folders, subfolders and files they contain are administered separately and occupy at least one cluster.

Data Types

Data Type	Memory Requirement	Comment
Analog channels	2(4) byte / sample *	4 Byte for the data type: float
Digital channels (16 bit DIO port)	2 byte / sample *	
Virtual channels (Data computed in imc Online FAMOS)	4 byte / sample *	
Analog channels reduced using Transitional Recording	4 byte / saved sample *	depends also on imc Online FAMOS functions
Digital channels reduced using Transitional Recording	4 byte / saved sample *	2 byte / saved sample + 2 byte for the time information

* + any file header for trigger events (size is variable)

Cluster Size

With **FAT16** formatting of the **storage medium**, the cluster size depends on the size of the storage medium. It is clear that small clusters make better use of the memory space, since less space is wasted at the ends of files by partially empty clusters.

The **PC-Hard disk** (e.g. WIN98), by contrast, has **FAT32**-formatting, in which the cluster size is constant at **4k**.

Cluster Size (FAT16)	Maximum Logical Drive Size
2 k = 2048 byte = 4 sectors	128 MB
4 k = 4096 byte = 8 sectors	256 MB
8 k = 8192 byte = 16 sectors	512 MB

DOS configures the cluster size to be as small as possible when formatting a drive. For example, if you format a 500 MB partition, you get 8 K clusters, for a 170 MB medium 4 K clusters, for a 340 MB medium 8 K clusters.

Examples

Example 1

- An analog channel is recorded with multiple triggered events. Each event is to comprise 2000 samples. A total of 100 events are recorded.
- A "naive" calculation which assumes that the data can be arbitrarily compressed would yield the following results:

Analog channel: 2 byte / sample 100 events x 2000 samples x 2 byte = 100 x 2000 x 2 byte	400 000 byte
1st file header: 1 x 1536 byte*	+ 1 536 byte
Each additional file header: 99 x 512 byte*	+ 50 688 byte
1 cluster for the directory allocation info and 32 bytes there for file allocation info	+ 32 byte
Data size (cannot be stored this compactly!)	**= 452 256 byte***

* File header for trigger events (size is variable)

- In actual fact, the data can not be packed with any arbitrary degree of compression. Instead, the storage medium's cluster structure must be taken into account. Thus, the storage of each individual measurement (trigger release, event) starts at a new cluster.
- Thus in reality the following results would emerge in an example of a 340MB hard drive:

1st event: 1536 bytes (header) + 4000 bytes (samples) = 5536 bytes => 2 x 4 KB	8 KB
Next events: 512 bytes (header) + 4000 bytes (samples) = 4512 bytes => 2 x 4 KB => 99 x 8 KB	792 KB
Directory and file allocation info in FAT (1 cluster)	8 KB
True memory requirement	**= 808 KB***

* File header for trigger events (size is variable)

The results illustrate that the space requirements for this example's files are almost twice the actual data volume.

Example 2

If the measured data are not recorded as multiple triggered events, then a folder is created for each measurement (340MB hard drive). In total, 100 measurements each comprising 2000 samples are performed:

For each measurement: 1536 byte + 4000 byte = 5536 byte = 2 x 4 KB => 100 x 8 KB	800 KB
Directory and file allocation info in FAT (1 cluster) => 100 x 8KB	800 KB
True memory requirement	**= 1 600 KB**

Example 3

In this example, 3 analog channels are recorded with multiple triggered events. Each event contains 2000 samples. A total of 100 events are recorded. Two of the channels are saved directly to the internal storage medium and the other is filtered online (imc Online FAMOS) before being saved (only filtered results are saved). The data is to be stored to an unpartitioned 340MB hard drive (8KB cluster size):

2 analog channels à 2 byte + 1 virtual channel à 4 byte	
1st event:	
1536 byte* + 1 x 2000 x 2 byte = 5,5 KB => 8 KB	8 KB
1536 byte* + 1 x 2000 x 2 byte = 5,5 KB => 8 KB	8 KB
1536 byte* + 1 x 2000 x 4 byte = 9,5 KB => 16 KB	16 KB
Next events:	
512 byte* + 1 x 2000 x 2 byte = 4,5 KB => 8 KB => 99 x 8 KB	792 KB
512 byte* + 1 x 2000 x 2 byte = 4,5 KB => 8 KB => 99 x 8 KB	792 KB
512 byte* + 1 x 2000 x 4 byte = 8,5 KB => 16 KB => 99 x 16 KB	1 584 KB
1 cluster for directory and 3 file allocation infos	8 KB
True memory requirement:	**= 3 208 KB**

* File header for trigger events (size is variable)

12.3.3 Parameter set

What is a parameter set?

A parameter set is a group of parameters which can be used for various purposes. For instance, parameters may be **settings for the device and channel configuration**. They can also be **metadata** and **variables values**.

Example

Example: Channel configuration

Channel name	Save (PC)	Sampling time	Duration	...
Channel_001	True	1 ms	40 s	...
Channel_002	True	100 ms	10 s	...

Example: Metadata

Company name	Test object number	Test part number	Test station photo	Hint
My company	X1	X1_123	.\Picture1.png	speed test 1

Example: Variable values

Name	Category	Value	Typ	...
DisplayVar_01	DisplayVar	1	System.Single	...
DisplayVar_02	DisplayVar	10	System.Single	...
DisplayVar_03	DisplayVar	777	System.Single	...

The parameters can be exported to a file. This file is in most cases read- and writable using a text editor or EXCEL (various data types are available: see *Export formats*). In this way, the parameters can be edited and imported back at a later time. When editing , please observe the notes under: "*Editing the parameter set* 1166".

What can a parameter set be used for?

Parameterizing the device- and channel configuration by means of the file's contents

There are various ways to import the file and to configure the measurement device by this means. You can save multiple parameter sets and import a particular one before the start of each measurement, depending on the requirements.

Thus, the configuration of the measurement device and of the channels can always be correctly adapted without needing to edit it manually each time.

Initializing variable values by means of the file contents

Just as you can parameterize the device configuration, you can also initialize variable values. The files contain values for the variables, which you can import at the desired time.

Saving the configuration in a file as documentation on the experiment

For purposes of documentation on the measured data or the experiment, you can save the parameter sets of device/channel configurations, or current variable values. These are available for view without the need to start imc STUDIO or even for it to be installed. An additional documentation possibility is a report 285. Separate reports can be created with various configurations.

Metadata on the experiment or the measurement

You can save metadata on the experiment or measurement. Experiment metadata are displayed in the project dialogs for the selected experiment, and measurement metadata in the Data Browser for the measurement. You can perform filtering and sorting by them.

Note — Trigger-settings and Application module assignments

Trigger-settings and Application module assignments can not be exported as parameter sets in a useful way.

What contents may a parameter set have?

Content	Description
Texts and numbers	E.g. device and channel configuration (device name, channel name, terminal, coupling, sampling rate, ...), metadata (notes, e-mail address, date), texts and numeric variable values
Paths	E.g. metadata: information on the project documentation, pictures. When a parameter set is created which contains pictures or document, these are saved in an extra folder. The folder name corresponds to the filename with the extension: ".data". The path information provides a relational reference to the file. Upon export, the paths are resolved and the file can thus be imported.

File formats

A variety of file formats is available. Each format has certain advantages and should be selected according to the application.

Format	Description
Parameter set files (*.csv)	Tab-separated Text-file. Easily readable in EXCEL.
Horizontal Parameter set files (*.csv)	Tab-separated Text-file. Easily readable in EXCEL.
Parameter set files (*.txt)	Particularly suitable for metadata having only one line of text (e.g. for the Setup-page *Documentation*). Easily readable with any text editing program

Export parameter set via

Command: Export parameters

The command: "*Export parameters*" allows automated export of the parameters at desired times and according to previously adjusted settings.

There are pre-made export-configurations available such as "Export all channels" (the values of the active channels' variables) or "Export all channels and settings" (the values of the active channels' variables, and channel settings of all analog channels). Or you can manually select which variable values and settings to export.

Ribbon: Project > Import/Export > Export > Export parameter set

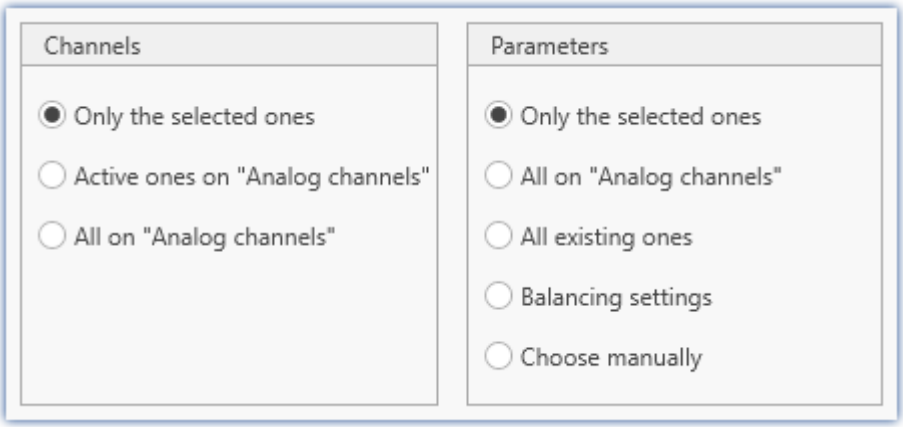
Corresponds to the command: "*Export parameters*", with the limitations that there are no pre-made configurations and that export is performed when you close the dialog.

Ribbon: Setup-Configuration > Export configuration

This function provides a variety of exporting options.

For most of the selection options, only the parameters (the content of the rows and columns) of the table on the currently open Setup page are exported. The table is associated with a so-called table description 274. In the table description is where you set which parameters (e.g. Channel name, Channel status, ...) are present. Not all parameters which would be possible are displayed in the table. For this reason, there are a variety of export options, so that it is possible to export either the visible, or the selected, or even the invisible parameters.

When export of the parameter set is performed via the menu ribbon item "*Setup-Configuration*" > "*Export configuration*", the selection dialog appears as follows:



The **left column** presents the **selection of channels** whose parameters are to be exported. The choices offered are only the selected channels, or only the channels set to "*active*", or all channels.

The **right column** presents the **selection of the parameters** to be exported for the channels.

Selection	Description
Only the selected ones	All parameters which are selected are exported.
All on "Page name"	All parameters are exported which are visible on the page. These included all columns in the table, but also all parameters in the dialogs and on additional tabs (e.g. the three tabs on the page: "*Channel balance*")
All existing ones	All parameters defined in the table description are exported. (Including such which are not present in the table.)
Balancing settings	The balance settings are exported, regardless of the Setup page selected. The record of the experiment includes the respective balance values saved for the various settings. For this reason, the balance values for each measurement range, for example, are exported to the same file. For a detailed explanation, see "*Exporting balance values (taring, scaling, bridge, ...)* 1171". Upon importing, the appropriate balancing settings are imported in accordance with the measurement range set.
Choose manually	A dialog for selecting the desired columns appears. In this dialog, select all parameters which you wish to export. All parameters selected are exported.

Note Export of adjustment values

When parameters which are relevant for adjustment are imported, inconsistent states can occur if not all parameters for the adjustment are appropriate. E.g. if only the compensation values are imported. When these values are exported the system ensures that all adjustment information is available when one of these relevant columns is exported.

All of the available selections also export the "*Balance scaling information*" as soon as a relevant adjustment column is included. Exception: "*Choose columns manually*"; here only items which are actually selected are exported.

The possibility to edit the export file manually still remains, but in consequence, the adjustment information may become inconsistent.

Import parameter set via

Command: Import parameters

The command: "*Import parameters*" allows automated import of the parameters at desired times and according to previously adjusted settings. You can also select from which file to import which variable's values and/or settings.

Ribbon: Project > Import/Export > Import > Import parameter set

Corresponds to the command: "*Import parameters*", with the limitation that import is performed when you close the dialog.

Ribbon: Setup-configuration > Import Configuration

Corresponds to the command: "*Import parameters*", with the limitation that import is performed when you close the dialog. Everything is imported, regardless of the Setup page opened.

Parameter set import-column

With the "*Parameter set import-column*", you can quickly load a variety of settings. When entries in the Setup-table are selected, the associated saved parameter set (*.csv) is loaded. See also "*Creating and configuring additional columns* 251" > "*Parameter set import-column* 267".

Note Importing variables

For importing variables' values, instead of the command: "*Import parameter*", the command "*Load/Refill variable* 1760" should be used. This command is specially designed for variable values.

Using the command: "*Import parameter*" can cause unintended alteration of the device control.

Note How the correct target is found

Parameter sets are saved in the form of tables. When importing previously saved parameter sets, the following rules apply in normal cases:

- **Target table:** As the target table (table description), the table in which the most columns are found is used. Note that the columns are found on the basis of their display names. It doesn't matter whether the short or long display name is used for a column.
- **Language:** The file name will contain a language-indicator abbreviation in most cases. By means of this, a parameter set can also be imported from a different language (see "*Note: Language abbreviation* 1164").
- **Device-/channel configuration or datapool (variable values):** The datapool is recognized as the target as soon as at least the columns "Name" and "value" are present in the parameter set table.
- **Target row**: The target rows of the parameters to be imported are found on the basis of the names of the corresponding elements. (See "*Mapping for import and export* 1164")
- If only one row exists in a target table (e.g. experiment description), then one row is always imported from the parameter set table.

Note Language abbreviation

The csv-files are designed to be language-independent. No in-house designations are used, which makes them easily readable. A language-indicating abbreviation is automatically appended to the file name. Thus, upon import it is possible to recognize for which language the file was generated. Additionally, a parameter set can also be imported from a different language (as long as the source language is installed).

Language recognition: If the abbreviation is not present, the language with the most matching titles is used as the source language.

Special case: There is no language abbreviation, but there are multiple parallel, valid files.

If when importing a file is specified only with the ending ".csv" and without any language-indicator abbreviation, then the system automatically searches for valid files having an abbreviation (Filename.Language.csv). If there is any corresponding file, it will be imported.

If multiple such files exist (varying languages), the file in the current language is used preferentially. If it is not available, the first one found is imported. The command generates an entry in the logbook on which file was imported.

Mapping for import and export

Mapping: Import

Here you can select the parameters according to which to import. At least one parameter must be used for assignment purposes to ensure that the parameters are assigned to the correct channel.

Selection	Description
Import mapping by name	For all parameters (all table descriptions), the settings are imported. The assignment is made on the according to the **name**. This means: All parameters belonging to a source-row (from the parameter set) are imported to a row in the target, if the parameter **Name** is identical in both the source and the target. **Special case:** If the parameter set contains the columns: **Name** and **Channel name** with different names: In this case, the column **Name** is still used for the assignment, but the column **Channel name** for setting the new name! Then it is possible to adapt the name by means of the name assignment. But this means that the parameter set can no longer be used since the previous name then no longer exists. By this method it would be possible, for example, to initialize the channel names. Then assign the new names of the parameters using a second parameter set.
Uniformly per module type	Set up all channels of a module type uniformly. The assignment is based on module type or Device order name, respectively, and channel type (e.g. "ARGFT/UTI-6-SUP", "Analog inputs"). Only one parameter set per module type is required.
Multiple-device operation (Firmware group A)	For all channels/variables, the settings are imported. The assignment is made according to the **connector designation** and **the device serial number**. When using multiple devices, this selection is preferable to "*Single-device operation*", since each connector designation is unique within a device. Disadvantage: it is not possible to assign parameters to other devices this way. If necessary, edit the parameters beforehand, in order to adapt the serial number to the target device. Cannot be used with devices of firmware group B because the connector does not distinguish between the main channel and the monitor channel.
Single-device operation (Firmware group A)	For all channels/variables, the settings are imported. The assignment is made according to the **connector designation**. When using multiple devices, the connector designation is not always unique; the same one could appear in every device. Use this setting only when using only one device. Cannot be used with devices of firmware group B because the connector does not distinguish between the main channel and the monitor channel.
...	Select a user-defined mapping. See "*Mapping - Modifications of the default behavior* 1181".

Assignment: Export

Here you can select what is to be exported.

Selection	Description
Export all channels	All variables having the channel type: "**analog inputs**" are exported.
Export all channels and settings	All content of the table description: "**analog channels**" is exported and all variables having the channel type: "**analog inputs**" are exported.
...	Select a user-defined mapping. See "*Mapping - Modifications of the default behavior* 1181".

Editing the parameter set

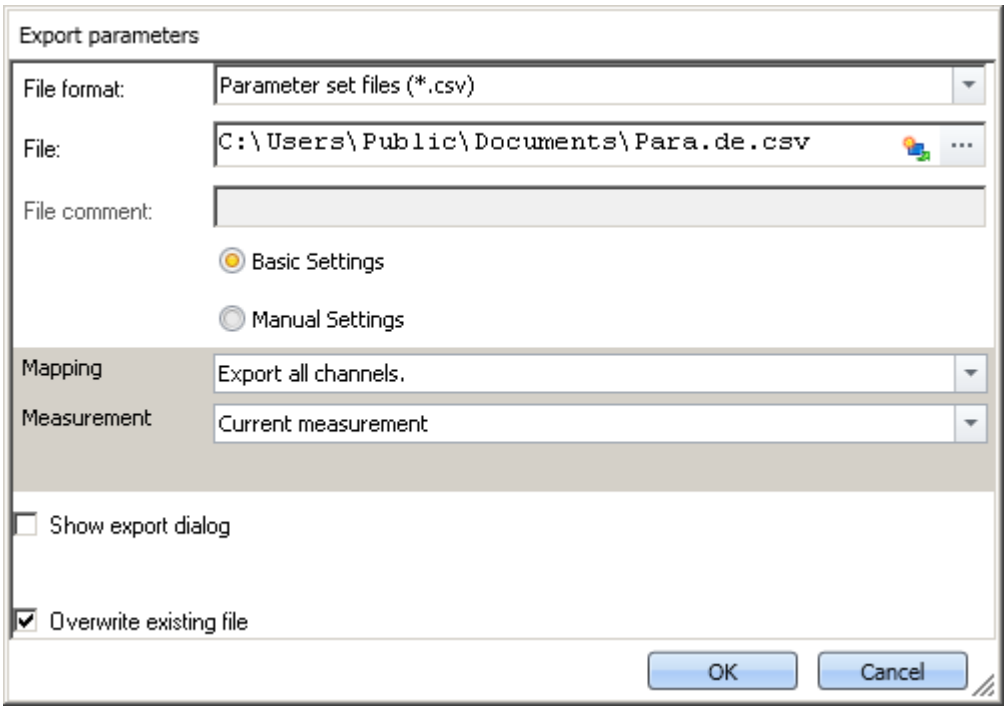
Program	Description
Editing by means of a text editor (Notepad, Notepad++, ...):	Here there are no known limitations.
Editing a csv-file with EXCEL	Load the file into EXCEL by double-clicking or by means of EXCEL's menu. If EXCEL's text conversion wizard appears, select "tab stop" as the separator. When you save the file after having made changes in EXCEL, this message appears; "*The file may contain features which are not compatible with Unicode text. Do you want to keep the workbook in this format?*" • Confirm with "*Yes*" When you close EXCEL, another message appears: "*Do you wish to save changes?*" • Select "*Don't save*" The following formats are not appropriate since the automatically save with the separator character ";": • CSV (separator character) (*.csv) • CSV (Macintosh) (*csv) • CSV (MS-DOS) (*.csv) **Known problems:** Unless different settings have been made and no accompanying unit has been specified, EXCEL automatically interprets numbers with the decimal separator: *Decimal period* as a date and concerts the accordingly. In this case, set the formatting of the cell to "*Text*". Or use the *Decimal comma* as the separator character with making entries.
Editing a csv-file by means of OpenOffice or LibreOffice	Load the file by double-clicking or via the program's menu. If the Text Import Assistant appears, select as the separator character/separator option: "Tabulator". If you save the file after having made changes, a message appears asking for the data saving format. • Click on "*Keep current format*" **Known problems:** • The character ' is used as an internal character in the editing process and it is deleted if it is at the beginning of a cell. This character otherwise has significance, for instance, as the unit for "feet" or "minutes of angle". • Unless different settings have been made and no accompanying unit has been specified, numbers with the decimal separator: *Decimal period* are automatically interpreted as a date and converted accordingly. In this case, set the formatting of the cell to "*Text*". Or use the *Decimal comma* as the separator character with making entries.

12.3.3.1 Export parameters

A parameter set is a group of parameters which can be used for various purposes. For instance, parameters may be **settings for the device and channel configuration**. They can also be **metadata** and **variables values**.

The command: "*Export parameters*" allows automated export of the parameters at desired times and according to previously adjusted settings.

There are pre-made export-configurations available such as "Export all channels" (the values of the active channels' variables) or "Export all channels and settings" (the values of the active channels' variables, and channel settings of all analog channels). Or you can manually select which variable values and settings to export.



Command: Export parameters

Note Import of variables preferably via the command: Export variable

For **exporting variables' values**, instead of the command: "Ex*port parameter*", the command "***Export variable*** 1758" should be used. This command is specially designed for variable values.

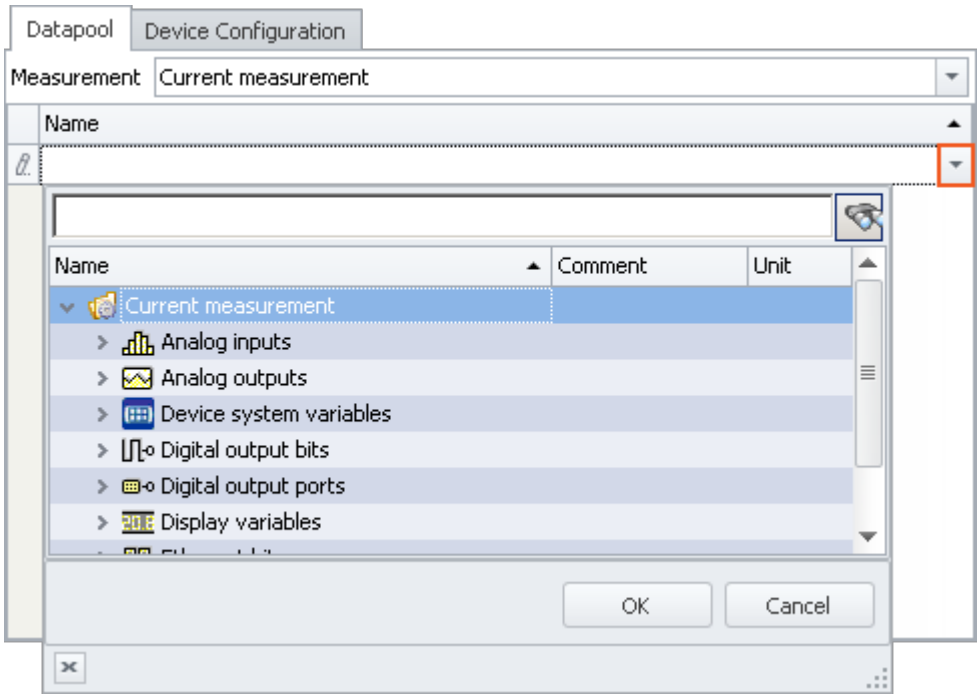
Certain variable types can only be exported by means of the command "*Export variables*"; this included text-variables.

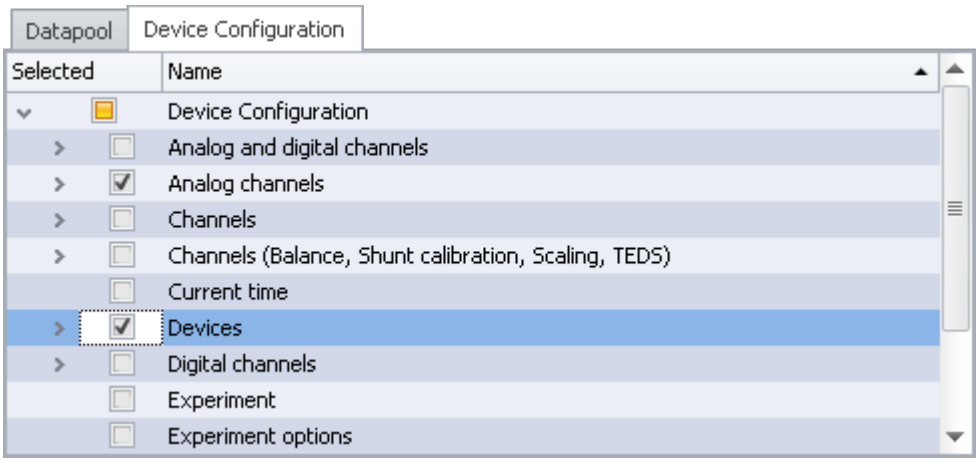
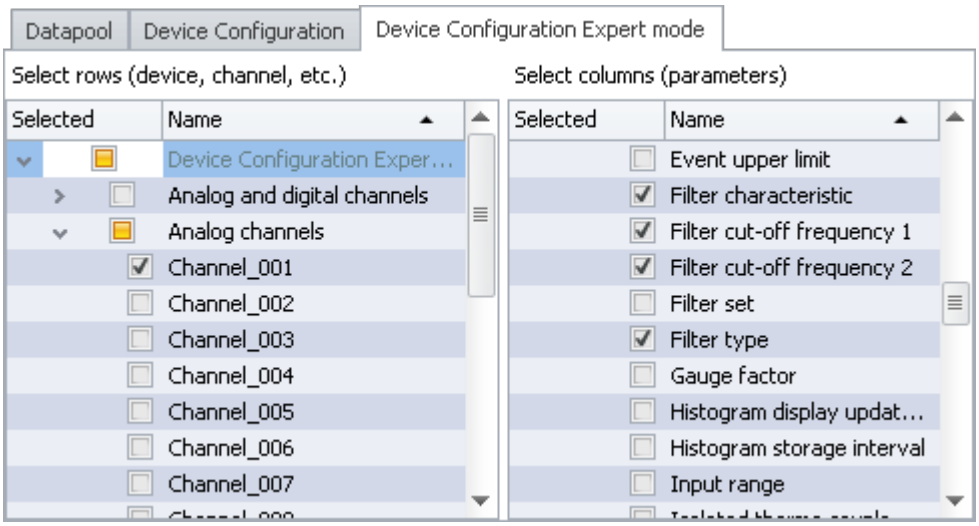
Option	Description
File format	Selection of the export format. For more information, see the description of the parameter set (File formats 1161).
File	Specifies the target file including the pathname.
Basic/Manual Settings	In order to configure the export, you can choose between **basic** and **manual** settings. • **Basic Settings** 1168: Here you find prepared export settings, for instance, one where only all values of the analog channels are exported. • **Manual Settings** 1169: Here you can define exactly what to export.
Measurement	You can export either parameters belonging to the current measurement, or those of a different measurement (identified by the measurement name or the measurement number).
Overwrite existing file	If this option is activated, files having the same name in the destination folder will be overwritten without a confirmation prompt. This option is useful for automated routines.
Show export dialog and Show element list	If one of the two options is activated, then when executing the commands the dialog appears in which to make settings in these two areas: • **Show export dialog:** File options (Save as dialog) • **Show element list:** Datapool (variables) and device configuration. All settings can be inspected and edited. Any changes have no effect on the configuration of the command in the command list, but only on the current export.

Basic Settings

Option	Description
Mapping	Here you can select what to export. • **Export all channels:** All variables having the channel type: "**analog inputs**" are exported. • **Export all channels and settings:** All content of the table description: "**analog channels**" are exported and all variables of the channel type: "**analog inputs**" are exported. • **...:** Select a user-defined assignment. For more information, see the description of the parameter set (Mapping for import and export 1164).

Manual Settings

Tab	Description
Datapool	Selection of the variables to be saved. Here you create a list of the variables to be exported. At runtime, these are exported automatically. The list is not dynamic, it does not adapt when variables are added or deleted. In some cases, the variables do not exist yet when the list needs to be filled. In order to add a variable to the next empty line: • enter the corresponding name, or • click on the button ( ) to the right of the line. When you click on the button, a strip-down Data Browser with a list of all available variables appears. If needed, run a "*Process configuration* 241" ( ) to update the list of variables it contains.  *List of the variables to export* Select all desired variables (multi-selection is also allowed) and click on the button *OK*. All selected variables are added to the list of variables to export.

Tab	Description
Device Configuration	Selection of the device and channel configurations to be saved. Here, select the settings to be exported.  *List of device settings* The individual table descriptions are available for selection. These define which parameters are exported. To limit the export to individual variables, devices or other elements, you can expand the table descriptions and select the desired elements.
Device Configuration Expert mode	This tab is only visible if expert mode for export of parameter sets 134 was activated in the options. Here, you can select individual parameters for export. In the region on the left, select the rows of a table description: the element (e.g. variable or device) whose parameters are to be exported. In the region at right, select the columns of the associated table description: the parameters which are to be exported.  *Left: List of the rows in a table description (devices and variables)* *Right: List of the columns of a table description (parameters)* In the example, the parameters "Filter characteristic", "Filter cut-off frequency 1", "Filter cut-off frequency 2" and "Filter type" belonging to the variable: "Channel_001" are exported.

Note

Keeping variables' values and the device configuration separate

You are able to export variables' values and device configurations together to one file. You are also able to combine the selection in the individual tabs. You can also combine "Device configuration" and the expert mode.

But please note: You will have greater clarity if you keep these things separate. To minimize error sources and to find errors more easily, we recommend exporting them separately.

Important parameters are always exported

Certain important parameters, which are necessary for assignment purposes when importing, are always included in the export process. Even if they are not selected, the following parameters are always included in the export of channel parameter:

Device name, Device identifier, Channel name, Name.

Trigger-settings and Application module assignments

Trigger-settings and Application module assignments can not be exported as parameter sets in a useful way.

Exporting balance values (taring, scaling, bridge, ...)

The balancing process is dependent on various parameters such as the amplifier type, the measurement range, measurement mode and 16 or 24 bit resolution. In order for it to be possible to re-import the balance value, these values need to be compatible. For this reason, when exporting balance values, many additional parameters are exported along with them.

Note

All necessary balance values are recorded in the file in xml-notation.

The balance values are associated with certain channel settings and only apply if these settings are previously restored when importing. E.g., when changing the measurement range, a new balancing procedure must be performed.

The record of the experiment includes the respective balance values saved for the various settings. For this reason, the balance values for each measurement range, for example, are exported to the same file.

Example

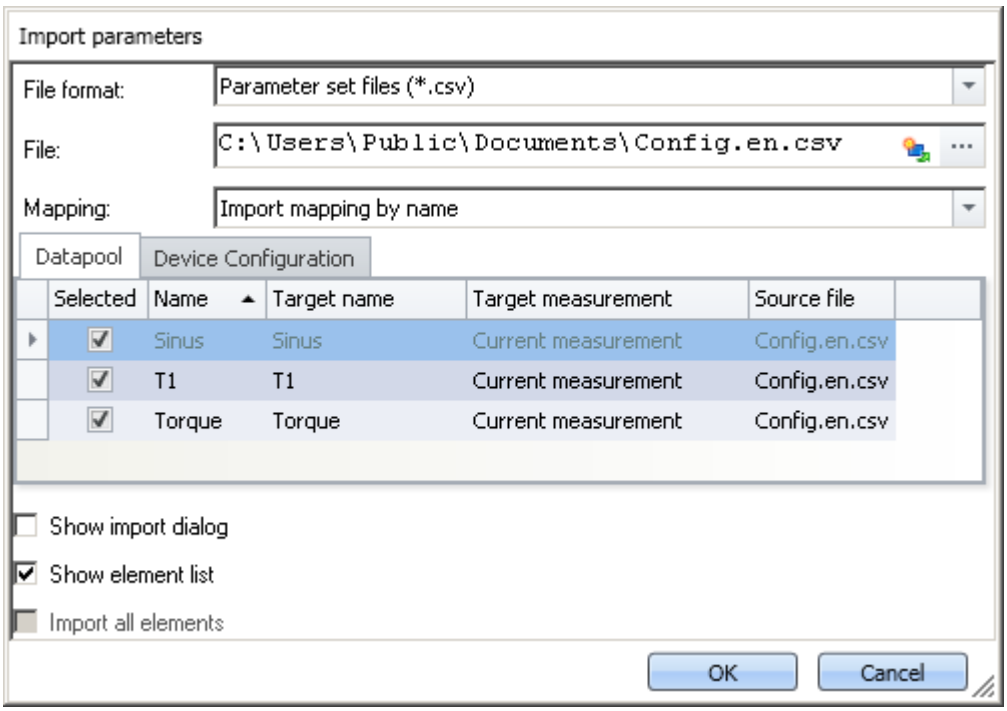
You perform balancing for Channel_001 with the measurement range 10 V. Next, you perform another balancing for the 5 V measurement range and export the result also to the file Balance.en.csv; the file contains both balancing values.

Upon importing, the appropriate balancing settings are imported in accordance with the measurement range set.

12.3.3.2 Import parameters

A parameter set is a group of parameters which can be used for various purposes. For instance, parameters may be **settings for the device and channel configuration**. They can also be **metadata** and **variables values**.

The command: "*Import parameters*" allows automated import of the parameters at desired times and according to previously adjusted settings. You can also select from which file to import which variable's values and/or settings.



Command: Import parameters

Note — Import of variables preferably via the command: Load/Refill variable

For **importing variables' values**, instead of the command: "*Import parameter*", the command "***Load/Refill variable*** 1760" should be used. This command is specially designed for variable values.
Using the command: "*Import parameter*" can cause unintended alteration of the device control.

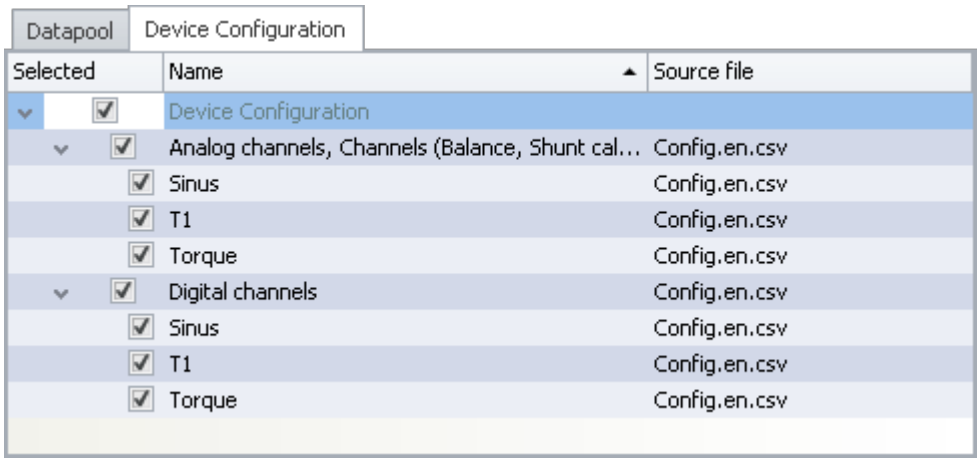
Reference — Description of the parameter set

Basic information on working with a parameter set, as well as about its content and structure, is presented in the section: "*Parameter set* 1159".
Practical examples appear in the section: "*Parameter set in the application* 1177"

Option	Description
File format	Selection of the file format. For more information, see the description of the parameter set (File formats 1161).
File	Specification of the file to import, including the pathname.

Option	Description
Mapping	Here you can select according to which parameters to perform the import. At least one parameter must be used for assignment purposes in order to ensure that the parameters are also assigned to the desired channel. **An unsuitable assignment can cause either no parameters or incorrect ones to be imported!** For more information, see further below under: "*Mapping for import* 1176".
Show export dialog and Show element list	If one of the two options is activated, then when executing the commands the dialog appears in which to make settings in these two areas: • **Show export dialog:** File options (open-dialog) • **Show element list:** Datapool (variables) and device configuration. All settings can be inspected and edited. Any changes have no effect on the configuration of the command in the command list, but only on the current import.
Import all elements	This option is only available if Show element list is not selected. When it is activated, then the complete parameter set, meaning the datapool and the device configuration is always imported according to the assignment instruction. The advantage is that at the moment you create the command, you don't need to know what elements the parameter set contains. **Note:** If the box is checked, then the system no longer distinguishes between the device configuration and the datapool. If only the datapool is exported (e.g. for Channel_001), then when importing, the channel configuration is also imported. E.g. in this case, the unit is also set.

Tab	Description
Datapool	Selection of the variables to be saved. Here you create a list of the variables to be imported. At runtime, these are imported automatically. The list is not dynamic, it does not adapt when variables are added or deleted. Datapool | Device Configuration Selected | Name | Target name | Target measurement | Source file ☑ Sinus | Sinus | Current measurement | Config.en.csv ☑ T1 | T1 | Current measurement | Config.en.csv ☑ Torque | Torque | Current measurement | Config.en.csv *List of variables to import*

Tab	Description
Device Configuration	Selection of the device- and channel configurations to be imported. Here you select which configurations are to be imported from the file.  *List of device settings* Some table descriptions are already available for selection. The displayed table descriptions do not have to match the original table description. The file does not state from which table description the export came. **Example:** In the picture above, you see the table descriptions: "*Analog channels, Channels (Balance, Shunt calibration, Scaling, TEDS)*", and "*Digital channels*". These were determined from the parameters which the file contained. However, the parameter set was generated from the table description "*Analog channels*". Since the table descriptions mostly contain the same parameters, these two are the first ones found which cover the import. If you ensure that all parameters are present on the target computer, then a correct table description will always be selected for import. To limit import to individual variables, devices or other elements, you can expand the table descriptions and select the desired elements.

Note — Importing a configuration

Make certain that the configuration can be imported.

Often certain settings depend on other parameters. For example, the measurement range depends on the offset/factor. When the measurement range is imported, then either the appropriate offset/factor-values must also be imported with it, or be set already.

Many other such interdependencies exist.

If a parameter can not be imported, a corresponding message appears in the logbook.

Note — Importing to the Datapool / Variables

It is not possible to import to the **device channels** belonging to the current measurement (Data Browser: *Current measurement*). This applies to analog channels, incremental counter channels, ...
This does not apply to single value variables, such as Display-variables, DACs, ... It is possible to import to these.

Imprecisions in importing

Import of the measurement range

Due to imprecision from rounding, the measurement is not adopted correctly in some cases.

When there is no match, the next larger measurement range is selected which completely covers the range required. To handle rounding problems, a tolerance of 1% is provided. If a smaller measurement range is below the desired range by only 1%, it is selected.

Example

Possible measurement ranges	0.1 and 0.25	0.75 .. 1.25 and 0.5 .. 1.5
Measurement range in the import file	0.12	0.75 .. 1.263
Import generated	0.25	0.5 .. 1.5
	because it is the next larger range	because it is the next larger range

In case of rounding problems

Possible measurement ranges	0.1 and 0.25	0.75 .. 1.25 and 0.5 .. 1.5
Measurement range in the import file	0.10000001 to 0.101	0.75 .. 1.2500001 bis 0.75 .. 1.262
Import generated	0.1	0.75 .. 1.25
	because this deviates from a measurement range by 1% or less	because this deviates from a measurement range by 1% or less

Importing balance values (taring, scaling, bridge, ...)

The balancing process is dependent on various parameters such as the measurement range and measurement mode. In order for it to be possible to re-import the balance value, these values need to be compatible. For this reason, when exporting balance values, many additional parameters are exported along with them.

Warning Bridge balance

When you import bridge balancing values for a channel whose amplifier type or device model is different (e.g. CRONOScompact with 16-bit Integer resolution to CRONOSflex with 24-bit Float resolution), the resulting measurement values are implausible! By contrast, balancing values from taring only correct the measured value and for this reason they can be imported without any problems.

Mapping for import

Here you can select according to which parameters to perform the import. At least one parameter must be used for assignment purposes, in order to ensure that the parameters are actually assigned to the desired channel.

Selection	Description
Import mapping by name	For all parameters (all table descriptions), the settings are imported. The assignment is made on the according to the **name**. This means: All parameters belonging to a source-row (from the parameter set) are imported to a row in the target, if the parameter **Name** is identical in both the source and the target. **Special case:** If the parameter set contains the columns: **Name** and **Channel name** with different names: In this case, the column **Name** is still used for the assignment, but the column **Channel name** for setting the new name! Then it is possible to adapt the name by means of the name assignment. But this means that the parameter set can no longer be used since the previous name then no longer exists. By this method it would be possible, for example, to initialize the channel names. Then assign the new names of the parameters using a second parameter set.
Uniformly per module type	Set up all channels of a module type uniformly. The assignment is based on module type or Device order name, respectively, and channel type (e.g. "ARGFT/UTI-6-SUP", "Analog inputs"). Only one parameter set per module type is required.
Multiple-device operation (Firmware group A)	For all channels/variables, the settings are imported. The assignment is made according to the **connector designation** and **the device serial number**. When using multiple devices, this selection is preferable to "*Single-device operation*", since each connector designation is unique within a device. Disadvantage: it is not possible to assign parameters to other devices this way. If necessary, edit the parameters beforehand, in order to adapt the serial number to the target device. Cannot be used with devices of firmware group B because the connector does not distinguish between the main channel and the monitor channel.
Single-device operation (Firmware group A)	For all channels/variables, the settings are imported. The assignment is made according to the **connector designation**. When using multiple devices, the connector designation is not always unique; the same one could appear in every device. Use this setting only when using only one device. Cannot be used with devices of firmware group B because the connector does not distinguish between the main channel and the monitor channel.
...	Select a user-defined mapping. See "*Mapping - Modifications of the default behavior* 1181".

FAQ

Question: Why do my changes to the parameter set have no effect following the import?

Answer: There are multiple possible reasons:

- An unsuitable assignment is used.
  For example, the name assignment is used, but the names in the parameter set file do not match the names in the channel list. This can be the case if the name is to be modified by via the parameter set.
  Solution: Use instead, for instance, import according to the connector designation, or observe the notes on the mapping by name (see: "*Mapping for import* 1176").
- The settings are reset by other parameters.
  For instance, if you export the parameter set of the page: *Analog channels*, you obtain among others the parameters for the analog channels and analog Monitor-channels. The parameters of these two channel types are interrelated. For example, if you change the coupling of the analog channel in the parameter set file from "Full bridge" to "DC", but leave the Monitor channel unchanged, the importing the Monitor channel will overwrite the analog channel.
  Solution: Always minimize the parameter set to the most important parameters. Delete everything which is not needed.

12.3.3.3 Parameter set in the application

Loading the factor, offset, unit and channel name from a CSV

This example illustrates how to import various settings in a targeted manner. The example can be modified for the parameters required.

Begin by creating a csv-template from an export of all parameters.

Next, go to the Setup page: "*Analog channels* " and run the export procedure:

Ribbon	View
Setup-Configuration > Export Configuration ( )	Complete

Select "*All on "Analog channels"*" (under "*Channels*" and "*Parameters*") for the export. All parameters defined in the table description are exported. Conclude the export and thus create a file of the file type "*Parameter set file (*.csv)*".

Editing in EXCEL

Open the csv file with EXCEL

- Delete all columns except for the following: "*Name*", "*Channel name*", "*Unit*", "*Scaling offset*" and "*Scaling factor*"
- Also delete all lines except the analog channels desired.
- Be aware of EXCEL's automatic formatting (see "*Editing the parameter set* 1166"): Set the "*Formatting*" of the columns: Scaling offset and Scaling factor to "*Text*"
- Next, change the values. Her is an example:

Name	Channel name	Unit	Scaling offset	Scaling factor
Channel_001	Messpunkt_1_Spannung	V	5	1
Channel_002	Messpunkt_2_Geophon	m/s	0	0,05076
Channel_003	Messpunkt_3_Geophon	m/s	0 m/s	0,05076 m/s/V

Notes on the values in the example:

- The column "*Name*" is used for assignment when importing.
- The column "*Channel name*" is used for the new name.
- The column "*Scaling offset*" depends on the unit and the column "*Scaling factor*" is additionally dependent on the measurement mode. In this example, import of Channel_003 would fail if the measurement mode was set to Bridge. In this case, the unit for the factor would need to be set to "*m/s/"mV/V'*"" for example. As you see with Channel_002, you can also omit the unit.

Save the csv file. Here, too, observe the notes on EXCEL (see "*Editing the parameter set* 1166"):

Upon saving, this message will appear:
"*The file may contain features which are not compatible with Unicode text. Do you want to keep the workbook in this format?*"

- Confirm with "*Yes*"

When you close EXCEL, another message appears:
"*Do you wish to save changes?*"

- Select "*Don't save*"

Import in imc STUDIO

Perform an import:

Ribbon	View
Setup-Configuration > Import Configuration ( )	Complete

For the mapping, use "*Import mapping by name*"

Restricted import of parameters, e.g. without channel name

Requirement: A parameter set contains many parameters, and they are all always included with an import. However, this is not always desirable.

Possible solution: One could edit the parameter set file by deleting all undesired parameters.

This is the easiest approach if it only needs to be done once. However, if the exporting process routinely generates a new file, it can save time to **modify the import mechanism** instead.

Below is an example illustrating the import of balancing values.

Example

When importing balancing values, not all of the parameters need to be imported

The scope of the balancing process comprises certain parameters such as the measurement range and measurement mode. But other parameters such as the channel name, comments, data saving settings, ..., are not to be included in the import.

In one particular application, the balancing values are routinely exported. This means that they can be used in all experiments for the same device. When required, it should be possible to import the file without needing to edit it each time. For this purpose, the "*mapping* 1173"-mechanism is to be modified one single time.

Mapping:

In the mapping instruction, you can select the parameters according to which to perform the import. You can additionally define other restrictions.

Specifically: The **balancing-parameters** associated with the **connection terminal designation "[01] IN01"** (appearing on a line in the parameter set) are mapped to the **device having the corresponding serial number** in the **line displaying the connection terminal designation "[01] IN01"**. Now this is performed for each parameter. Exception: Channel Name; this parameter is skipped.

Information on the mapping:

Each assignment defined will be saved in an XML file.

Pre-made mapping files for the parameter set commands are located following installation (depending on the installation and possibly the version) under the this path: (hot key <win> +r)

shell:common appdata\imc\imc STUDIO\Applications_1\Extensions\Parameterset

The assignment files used are located in the respective import or export folder. **Please do not change the original files!** If you save your own files there, they will be available for import/export.

A detailed description is provided in the chapter: "*Mapping - Modifications of the default behavior* 1181".

Additional configurations for import, examples and ideas are also presented in that chapter. For instance, you can limit import to certain channel types, or exclude other parameters (such as the data storage configuration) from the import.

Proceed as follows:

The best approach is to copy one of the existing files and open it in a text editor (e.g. Notepad++).

The name of the new file must have this extension: ".mapping.xml"; e.g. Balancing_ConnectorAssignment.mapping.xml

Copy the following text to the file (be aware that any extra line break, caused by copying long lines, will need to be deleted):

```
<import>
    <languageblock language="de">
        <caption>
            Abgleichdaten nach Anschluss und Geräteseriennummer importieren
        </caption>
        <description>
            Für alle Kanäle werden die Abgleich Einstellungen importiert. Ignoriert
werden Parameter, wie der Kanalname. Dabei wird die Zuordnung anhand der
Anschlussbezeichnung und der Geräteseriennummer vorgenommen.
        </description>

        <mapping>
            <source>
                SELECT *
            </source>
            <target>
                SELECT *\Name,Kanalname WHERE Anschluss = source.Anschluss AND
Geräteseriennummer = source.Geräteseriennummer
            </target>
        </mapping>
    </languageblock>
</import>
<import>
    <languageblock language="en">
        <caption>
            Import balance settings in accordance with the connection and device serial
number
        </caption>
        <description>
            The balance settings are imported for all channels. The assignments are made
according to the connector designation and the device serial number. Parameters such as
the channel name are ignored.
        </description>

        <mapping>
            <source>
                SELECT *
            </source>
            <target>
                SELECT *\Name,Channel name WHERE Connector = source.Connector AND Device
SN = source.Device SN
            </target>
        </mapping>
    </languageblock>
</import>
```

Subsequently, there is a new item among the available mapping selections for import of the parameter set: "*Import balance settings in accordance with the connection and device serial number*".

In the following line of code, after the string "*\" you specify which parameter are not imported.

```
<target>
    SELECT *\Name,Channel name WHERE ...
</target>
```

For more examples of syntax, see the chapter "*Mapping - Modifications of the default behavior* 1181".

Please also observe the notes about languages 1182.

12.3.3.4 Mapping - Modifications of the default behavior

Note Advanced notes

This chapter provides advanced information on the import- and export mechanism. In most cases this chapter can be skipped.

Warning

The following description is of a modification of the default import and export behavior. For most applications, the existing mapping files are adequate.
Please don't change the existing mapping files, but rather make new assignments, at most.

The **mapping** determines how import and export are governed, for instance, according to what pattern data are imported. In this chapter, you will find some mapping examples. According to this pattern, you can define your own mapping files to design the import and export to corresponds to your requirements.

Each assignment defined will be saved in an XML file.

Pre-made mapping files for the parameter set commands are located following installation (depending on the installation and possibly the version) under the this path: (hot key <win> +r)

shell:common appdata\imc\imc STUDIO\Applications_1\Extensions\Parameterset

The assignment files used are located in the respective import or export folder. **Please do not change the original files!** If you save your own files there, they will be available for import/export.

Scenarios for import

If only certain rows of a table are to be imported, or the parameters are to be imported according to criteria other than the name.

Scenarios for export

For instance, if all channels are to be exported, they must be selected and their names known. However, as soon as the configuration is changed and new channels are added, the export list must be modified. With the help of the export mapping rules, it is possible to export all channels no matter how many and what names are used, or conversely to to export only variables belonging to one type.

Defining a mapping rule

This mapping rule is saved in an XML file with the extension ".mapping.xml". **Be aware that the file must be UTF-8 encoded.** In every case, the language for which the mapping rule is used must also be specified.

In such a file it is possible to specify any desired mapping rules, and a name and a description for the mapping rule.

Example Introductory example of import, with explanations

```
<import>
    <languageblock language="en">
        <caption>
            Here, a designation can be specified
        </caption>
        <description>
            Here, an explanation of this mapping rule can be supplied.
        </description>

        <mapping>
            <source>
                Here is where you define which rows from which table of the parameter set
are to be imported.
            </source>
            <target>
                Here is where you define on which rows and columns the source row data
are to be mapped.
            </target>
        </mapping>
    </languageblock>
</import>
```

Languages in the mapping rule

In order for assignment to be possible, the language of the file to import and of the software in the mapping file must be available. English should be used as the fallback language in case some parameters in the currently used software language are not available.

For each language, create a separate **`languageblock`**-block.

Example **`languageblock`**-block

```
<import>
    <languageblock language="de">
        <caption>
            ...
        </caption>
        ...
    </languageblock>
    <languageblock language="en">
        ...
    </languageblock>
    <languageblock language="fr">
        ...
    </languageblock>
    <languageblock language="ja">
        ...
    </languageblock>
    <languageblock language="zh-CN">
        ...
    </languageblock>
    <languageblock language="zh-TW">
        ...
    </languageblock>
    ...
</import>
```

Syntax-examples for the import:

Syntax <source>	Description
`SELECT * FROM Channels`	All rows are used for import from the source file which can be assigned to the table description **Channels**. Device parameters and parameters of other table descriptions are not imported.
`SELECT *`	All rows are used for the import from the source file.
`SELECT * FROM Channels WHERE Name = 'My_TemplateChannel_Bridge'`	From the source file, all rows are used for the import 1. which were able to be assigned to the table description **Channels**, 2. and have **My_TemplateChannel_Bridge** for the parameter: **Name**.

Syntax <target>	Description
`SELECT * WHERE Name = source.Name`	All parameters of a source row are imported to a row in the target, if the parameter **Name** is identical in the source and in the target.
`SELECT * WHERE Connector = source.Connector`	All parameters of a source row are imported to a row in the target, if the parameter **Connector** is identical in the source and in the target.
`SELECT * WHERE Connector = source.Connector AND Device SN = source.Device SN`	All parameters of a source row are imported to a row in the target, 1. if the parameter **Connector** is identical in the source and in the target and 2. if the parameter **Device SN** is identical in the source and in the target.
`SELECT *\Name,Channel name WHERE Coupling = 'Full bridge'`	All parameters of a source row are imported to a row in the target if the parameter **Coupling** is **Full bridge** in the target. The parameters **Name** and **Channel name** are ignored and not set in the target.

Example Import 1: Channels - Import mapping by name

All rows are used for import from the source file which can be assigned to the table description **Channels**.
Each row in the parameter set is mapped to a row in the target table having the same **Name** as the source row.

```
<import>
    <languageblock language="en">
        <caption>
            Channels - Import mapping by name
        </caption>
        <description>
            The channels of the channel table are ordered by their names
        </description>

        <mapping>
            <source>
                SELECT * FROM Channels
            </source>
            <target>
                SELECT * WHERE Name = source.Name
            </target>
        </mapping>
    </languageblock>
</import>
```

Example Import 2: Import mapping by name

Similar to Example 1.
From the source file, all rows are used for the import, not only for the table description: **Channels**.
Each row in the parameter set is mapped to a row in the target table having the same **Name** as the target row.

```
<import>
    <languageblock language="en">
        <caption>
            Import mapping by name
        </caption>
        <description>
            All parameters are assigned according to their name.
        </description>

        <mapping>
            <source>
                SELECT *
            </source>
            <target>
                SELECT * WHERE Name = source.Name
            </target>
        </mapping>
    </languageblock>
</import>
```

Example Import 3: Single-device operation (Firmware group A)

File: ConnectorAssignment_one_Device.mapping.xml

From the source file, all rows are used for the import.

The settings for all channels are imported. The assignment is performed on the basis of the connector designation. Each row in the parameter set is mapped to a row in the target table having the same **connector designation** as the source row.

```
<import>
    <languageblock language="en">
        <caption>
            Single-device operation (Firmware group A)
        </caption>
        <description>
            Import settings for the device selected for measurement, if the Connector
designations do not include serial numbers.
The assignment is based on the connector (e.g. "[02] IN01").
        </description>

        <mapping>
            <source>
                SELECT *
            </source>
            <target>
                SELECT * WHERE Connector = source.Connector
            </target>
        </mapping>
    </languageblock>
</import>
```

Cannot be used with devices of firmware group B because the connector does not distinguish between the main channel and the monitor channel.

Example Import 4: Multiple-device operation (Firmware group A)

File: ConnectorAssignment.mapping.xml

All rows from the source file are used for the import.

The settings for all channels are imported. The assignment is performed on the basis of the connector designation and the device serial number.Each row in the parameter set is mapped to a row in the target table having the same **connector designation** and **serial number** as the source row.

```
<import>
    <languageblock language="en">
        <caption>
            Multiple-device operation (Firmware group A)
        </caption>
        <description>
            Import settings for particular devices, if the Connector designations do not
include serial numbers.
The assignment is based on connector and device serial number (e.g. "[02] IN01",
"144163").
        </description>

        <mapping>
            <source>
                SELECT *
            </source>
            <target>
                SELECT * WHERE Connector = source.Connector AND Device SN = source.Device
SN
            </target>
        </mapping>
    </languageblock>
</import>
```

Cannot be used with devices of firmware group B because the connector does not distinguish between the main channel and the monitor channel.

Example Import 5

This mapping rule consists of two parts. First, the row of the parameter set which was imported for the channel table and which has the name "My_TemplateChannel_Bridge" is loaded. Next, this row is mapped to all rows in the target table for which "Full bridge" was set as the coupling.

In the second step, the row of the parameter set which was imported for the channel table and which has the name "My_TemplateChannel_DC" is loaded. Next, this row is mapped to all rows in the target table for which "DC" was set as the coupling.

In both cases, the name and the channel name are not set in the target.

```
<import>
    <languageblock language="en">
        <caption>
            Initializing analog channels
        </caption>
        <description>
            Creating a base configuration for all analog channels.
        </description>

        <mapping>
            <source>
                SELECT * FROM Channels WHERE Name = 'My_TemplateChannel_Bridge'
            </source>
            <target>
                SELECT *\Name,Channel name WHERE Coupling = 'Full bridge'
            </target>
        </mapping>

        <mapping>
            <source>
                SELECT * FROM Channels WHERE Name = 'My_TemplateChannel_DC'
            </source>
            <target>
                SELECT *\Name,Channel name WHERE Coupling = 'DC'
            </target>
        </mapping>
    </languageblock>
</import>
```

Example Sample files for example Import 5

Example: csv-file

Name	Channel status	Measurement mode	**Coupling**	Input range
My_TemplateChannel_Bridge	Active	Voltage	**Full bridge**	±100 "mV/V"
My_TemplateChannel_DC	Active	Voltage	**DC**	±5 V
My_Channel_without_effect	Active	Voltage	Quarter bridge	±10 "mV/V"

Example: Channel-configuration prior to import

Name	Channel status	Measurement mode	**Coupling**	Input range
Channel_001	Passive	Voltage	**DC**	±1 V
Channel_002	Passive	Voltage	**DC**	±1 V
Channel_003	Passive	Voltage	**DC**	±1 V
Channel_004	Passive	Voltage	Quarter bridge	±1000 "mV/V"
Channel_005	Passive	Voltage	Quarter bridge	±1000 "mV/V"
Channel_006	Passive	Voltage	**Full bridge**	±1000 "mV/V"
Channel_007	Passive	Voltage	**Full bridge**	±1000 "mV/V"
Channel_008	Passive	Voltage	**Full bridge**	±1000 "mV/V"

Example: Channel-configuration after import

Name	Channel status	Measurement mode	**Coupling**	Input range
Channel_001	Active	Voltage	**DC**	±5 V
Channel_002	Active	Voltage	**DC**	±5 V
Channel_003	Active	Voltage	**DC**	±5 V
Channel_004	*Passive*	*Voltage*	*Quarter bridge*	*±1000 "mV/V"*
Channel_005	*Passive*	*Voltage*	*Quarter bridge*	*±1000 "mV/V"*
Channel_006	Active	Voltage	**Full bridge**	±100 "mV/V"
Channel_007	Active	Voltage	**Full bridge**	±100 "mV/V"
Channel_008	Active	Voltage	**Full bridge**	±100 "mV/V"

The names are ignored. The channels: "*Channel_004*" and "*Channel_005*" remain as they are, since the coupling "*Quarter bridge*" is not in the assignment rule for the target.

The source row "*My_Channel_without_effect*" is also not in the assignment rule and is not used as a source.

Syntax-examples for the export:

Syntax <source>	Description
`SELECT * WHERE Caption = 'analog inputs'`	All contents of the table description and/or all variables of the channel type: "**analog inputs**" are exported. In this case, there is no table description of the name "**analog inputs**". However, there are variables of the channel type "**analog inputs**"
`SELECT * WHERE Caption = 'analog inputs' OR Caption = 'analog channels'`	All contents of the table description and/or all variables of the channel type: "**analog inputs**" or "**analog channels**" are exported. In this case, there is a table description of the name "**analog channels**". And there are also variables of the channel type "**analog inputs**"
`SELECT * WHERE Caption = 'analog channels'`	All contents of the table description and/or all variables of the channel type: "**analog channels**" are exported. In this case, there is a table description of the name "**analog channels**". However, there is no variable of the channel type "**analog channels**".

Syntax <target>	Description
`SELECT *`	obligatory specification

Example — Export 1: Export all channels

All variables of the channel type: "**Analog inputs**" are exported.

```
<import>
    <languageblock language="en">
        <caption>
            Export all channels
        </caption>
        <description>
            Export data from all active channels.
        </description>

        <mapping>
            <source>
                SELECT * WHERE Caption = 'analog inputs'
            </source>
            <target>
                SELECT *
            </target>
        </mapping>
    </languageblock>
</import>
```

Example Export 2: Export all channels and settings

All contents of the table description: "**analog channels**" are exported and all variables of the channel type: "**analog inputs**" are exported.

The only distinction made is between the names: "analog inputs" vs. "analog channels". The names must be unique if they are used for the export.

```
<import>
    <languageblock language="en">
        <caption>
            Export all channels and settings
        </caption>
        <description>
            Export data and settings from all active channels.
        </description>

        <mapping>
            <source>
                SELECT * WHERE Caption = 'analog channels' OR Caption = 'analog inputs'
            </source>
            <target>
                SELECT *
            </target>
        </mapping>
    </languageblock>
</import>
```

12.4 imc Data format

The imc data format (*imc3*) is applied and generated.

One major advantage of this format is the display of large data volumes in the curve window. It is not necessary to load large data volumes completely before they can be viewed. Instead, only so much data are loaded for the curve window as can also be displayed. In this way, you are able to quickly scroll and zoom, no matter how large the volume of data is.

Storage of the data is resilient to interruptions; even incomplete files can be used without the need for repairs. Furthermore, it is possible to save the monitor channels of imc EOS and imc ARGUS*fit* in the envelope type format 1132.

Selecting the data format: imc2 or imc3

In imc FAMOS, it is possible to set whether the files are to be saved in the old imc2 or in the new imc3 format. imc STUDIO applies this setting accordingly. We recommend saving the measured data in imc3 format.

Reference The imc data formats

Information on the imc data formats is presented in the imc FAMOS manual.

12.5 imc Format Converter

The imc Format Converter provides the ability to convert measured data to a different format, e.g. to EXCEL data format. The imc Format Converter is integrated in **imc STUDIO** within the functionality of **Sequencer**'s ***Data Saving Assistent*** 1738 and ***Format Converter*** 1713.

Furthermore, the conversion capabilities of the Format Converter can be used either within a **stand-alone program** by means of the **Windows-Explorer's context menu**, if imc STUDIO or imc FAMOS has been installed.

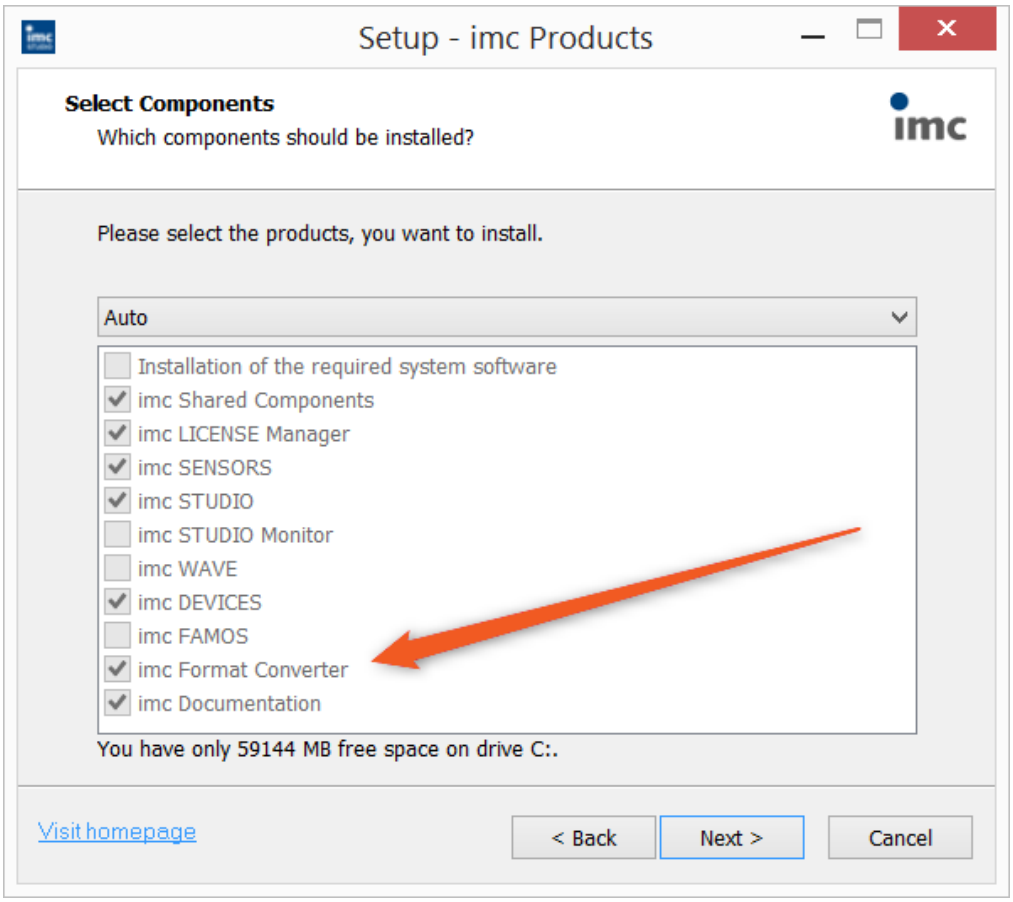
Export formats

The same export formats are supported which are also offered by imc FAMOS. These include:

- user-defined ASCII export-formats
- user-defined EXCEL export-formats
- MDF2.0, MDF3.0, MDF4, ASAM ATFX, ASAM ATFX NVH, Catman 5.0
- Google Earth Export, HEAD acoustics (also 4.5 compatible)
- Matlab 4 and 5, nSoft-DAC, RPC-3, Somat SIF (nCode output)
- TEAC TAFFmat and TEDAM, DIAdem TDM and TDMS
- Binary Universal File Format
- Universal File Format (UFF)

12.5.1 Installation

The installation file is located on the installation medium and is included in the installation by default.



Separate installation

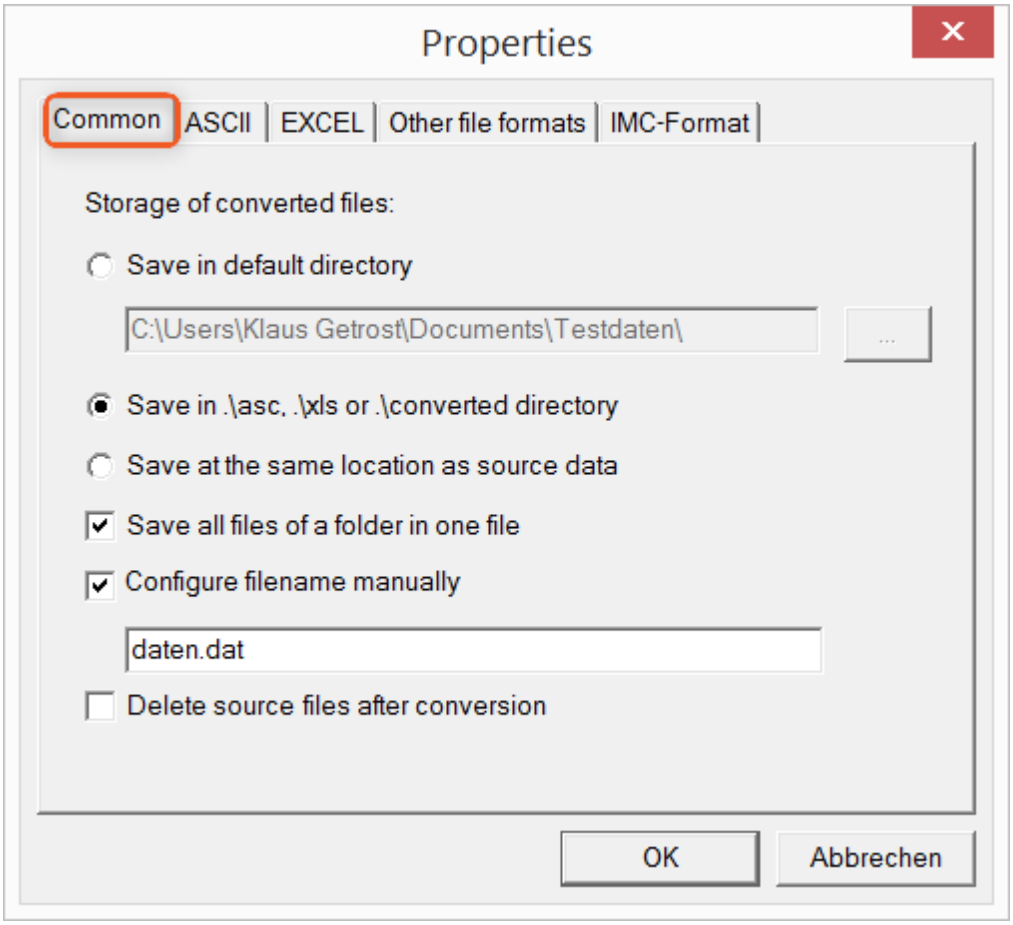
Select the installationvariant: "*User-defined*" and select there only the imc Format Converter.

12.5.2 Settings

In the settings you specify whether the data are saved either individually or together, which templates are used for ASCII and EXCEL, or what other formats are used.

The settings remain intact, so that under normal circumstances you can immediately select data and convert them.

Data saving



Storage location for converted files:

Save in default directory: Free choice of a **target folder**. Even network drives are possible.

Save in .\asc, .\xls oder .\converted directory: The data are saved in their source folder but additionally in a **subfolder**. For ASCII in the *asc* folder, *xls* for EXCEL and *converted* for any other export formats.

Save at same location as source data: The data re saved on the same **level of the tree diagram as the source data, without subfolders**.

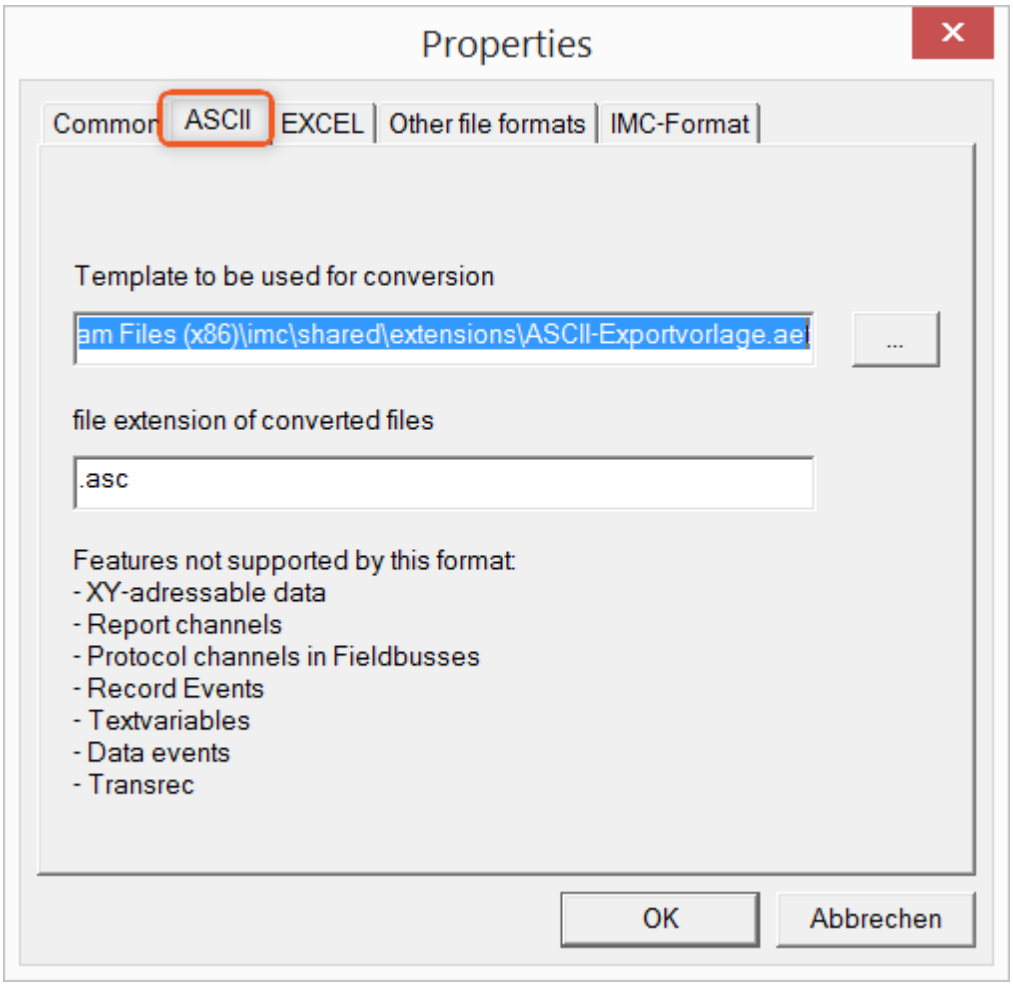
Manner of data saving:

Save all files of a folder in one file: All of a channel's files are saved **together in one file**.

Configure file name manually: **Name of the target file** in which all channel files are saved together.

Delete source files after conversion: Once the data have been converted, the **source files can be deleted automatically**, in order to make memory space free. This setting can make sense when all data are saved together in the imc format.

ASCII



Format template to be used for conversion:

By default, ASCII export templates are located in the folder *"C:\Program Files (x86)\imc\Shared\Extensions"* and use the file extension ***.aet***. Owners of imc FAMOS are also able to personally create or modify these templates. In this case, the aet-files are saved in the folder *"C:\ProgramData\imc\Common\Def"*.

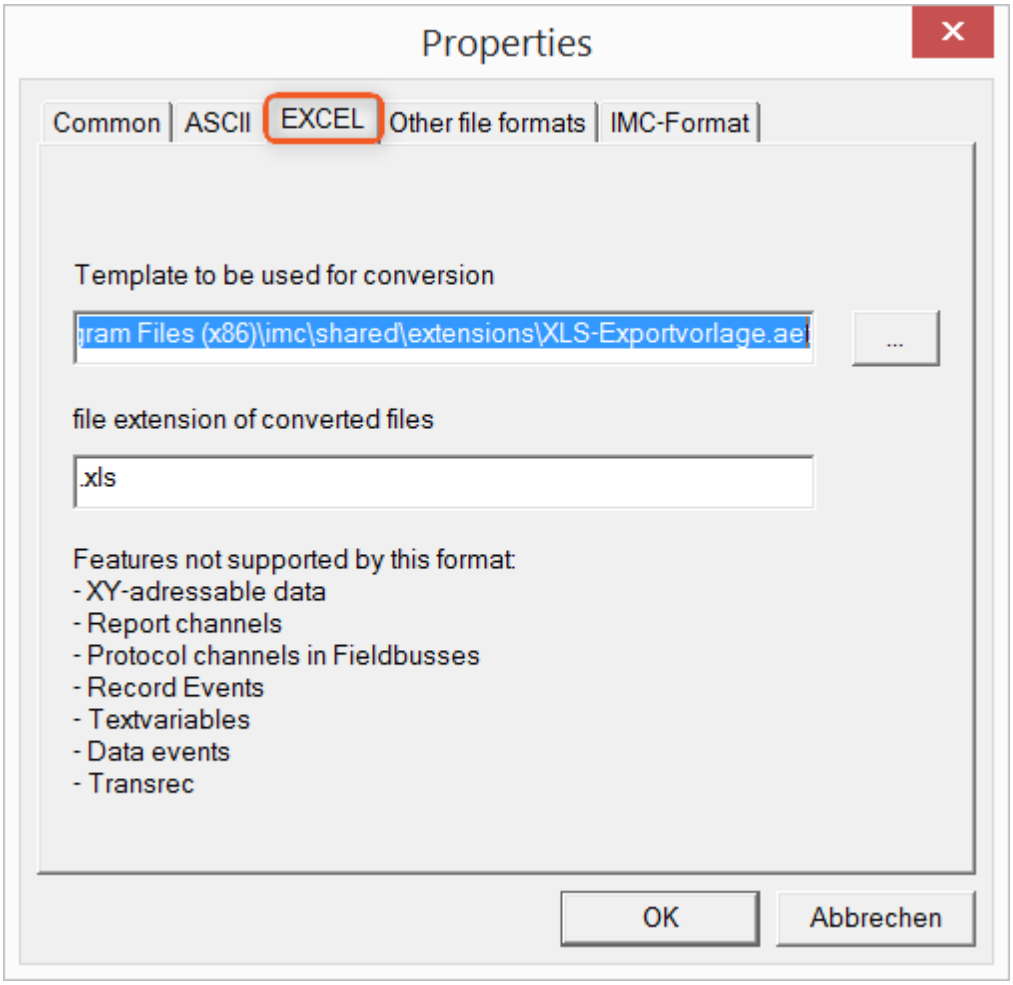
By default, the file extension is *"asc"*, but any arbitrary one can be specified here.

Special variable types which can **not** be exported as ASCII data are listed on this tab page.

Notes — File extensions

In imc FAMOS the file extension is specified with the ASCII template. Within imc FAMOS this file extension is added to the ASCII file. With imc Format Converter the file extension specified here is used instead.

EXCEL



Format template to be used for conversion:

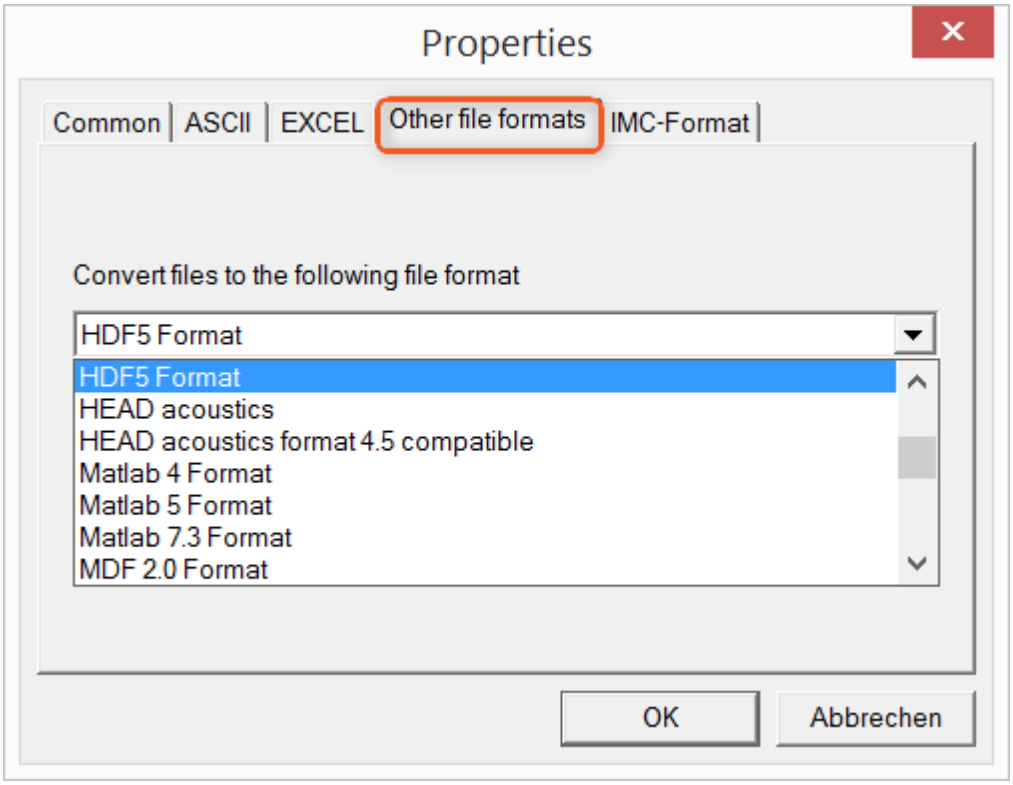
For EXCEL files, a similar technique is used as for ASCII 1194. By default, the file extension is *"XLS"*, but can be arbitrarily specified here.

Variable types which are **not** exported correspond to those for ASCII-export 1194.

Notes File extensions

If the XLS**x** format was selected as the export template, it must be here as the file extension. The imc Format Converter does not import the specified file extension from the AET file.

Other file formats



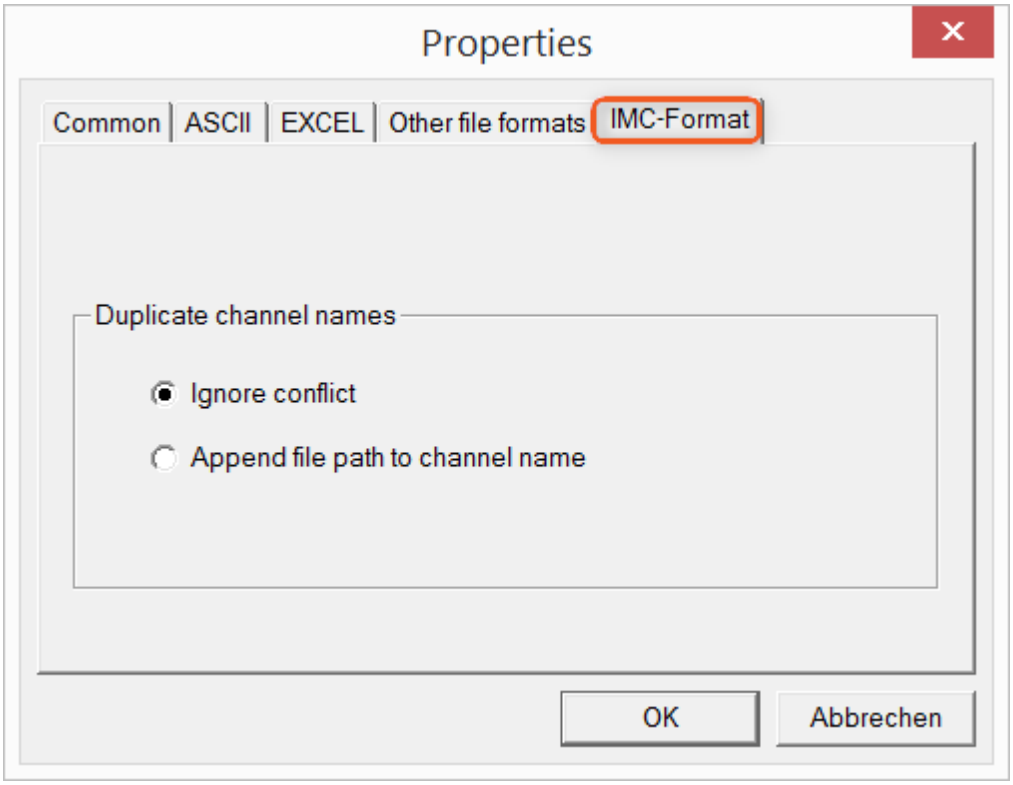
Convert files to the following file format:

Selection of the format in which the data are to be saved. Installation of the imc Format Converter makes certain export formats available.

Options

If the format selected comes with additional export options, these can be set by means of the dialog opened by clicking on the button "*Options*". Descriptions of the options appear in the document ***ImportExportFilter.pdf***, of which a copy is imported along with the imc FAMOS installation. Alternatively, the document is available from the imc homepage.

imc-Format



Handling duplicate channel names

Ignore conflict: If files are overwritten due to duplicate channel names, the conflict is not reported.

Append file path to channel name: A unique channel name is generated from the file path.

Notes Whay save in the imc-format?

The imc Format Converter is designed for imc data, so why is there an imc-Format option? imc devices allow complex data structures in which channels can be recorded at different moments in time (using triggers). For this reason, the channels are saved as individual files. Using the imc Format Converter, you are able to join the data sets accumulated into one single file after the measurement. One appropriate group of settings on the page "*Common* 1193" to accomplish this would be:

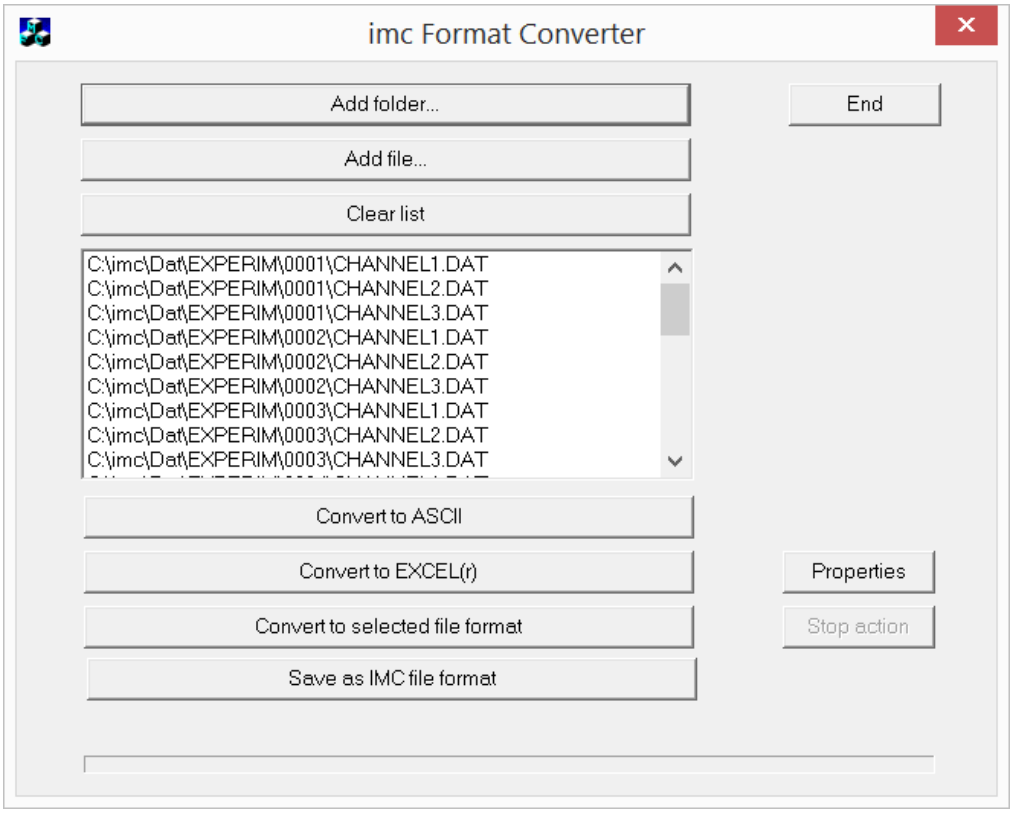
Save at same location as source data = active

Save all files of a folder in one file = active

Delete source files after conversion = active

12.5.3 Format converter as standalone-program

There is no shortcut for starting the standalone version. Start "*imcFrmtCvrt.exe*" directly from the installation folder, normally "*C:\Program Files (x86)\imc\imc Format Converter*"



Add folder...

Select the folder of measured data desired. Subfolders will also be imported and appear in the Files list.

Add file...

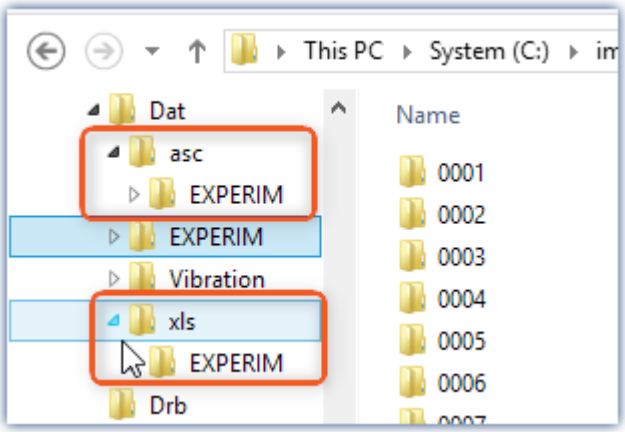
Selection of individual files.

Clear list

All entries are deleted from the Files list.

Convert to ASCII/EXCEL(r)

The standard variant for converting data to ASCII 1194- or EXCEL 1195-files. Toward this end, a "asc" or "xls" folder is created on the same level as the folder of measured data, in which the data are saved in the same folder structure. Depending on the settings, either individual files or tables are generated, in which all data are saved in a file.



Convert to selected file format

The data are saved in the format which was selected under "*Settings* 1196". The target folder is a subfolder of the same folder, with the name "*converted*". This option allows all signals to be saved jointly in a single file. Furthermore, a filename with file extension must be entered in the settings 1193.

Save as IMC file format

For saving all files in a folder to a single file in imc format. Observe the necessary settings on the Common tab page 1197.

12.5.4 Command line parameters

The imc Format Converter can be run via command lines. In the simplest case, the settings last applicable in the imc Format Converter are used, e.g. including the target file. If multiple files are to be converted, these are defined in an XML-file 1200.

Command line options

	Description	Example
-a	conversion to "*ASCII*"	`imcFrmtCvrt.exe -a <files or paths>`
-e	conversion to "*EXCEL*"	`imcFrmtCvrt.exe -e <files or paths>`
-k	conversion to "*Other data formats*"; The format selected in the settings is used, e.g. MDF	`imcFrmtCvrt.exe -k <files or paths>`
-i	import and use configuration from a setup XML file (imc Devices mode) enables complete remote control. Under some circumstance, a variety of settings can be required for the export. Beyond the default settings, it is possible to provide a Setup.xml by means of -s. A description of the XML is then no longer needed.	`imcfrmtcvrt -i -s d:\setup.xml` `imcfrmtcvrt -i d:\setup.xml <files or paths>`
-s	(Setup) without any additional parameters opens the familiar settings dialog	
-h	(hidden) only in conjunction with -i: Suppression of all windows and error messages	`imcfrmtcvrt -i -h d:\setup.xml <files or paths>`
-l	Write all information from windows and error messages to a LOG-file	
-h -l	Write all information from windows to a LOG-file in spite of suppression	`imcfrmtcvrt -i -h -l d:\setup.xml d:\Logfile.log <files or paths>`

For the purpose of determining **target files**, the imc Format Converter **settings** are used.

Specifying the files to convert in an XML file

You can specify in an XML file which files are to be converted. This can be necessary for instance when the command line is not sufficiently long. Depending on the Windows version, the line length is between 2000 and 8000 characters. The files to be converted are then are specified in an XML file.

Example Sample XML-file

This XML file must have the following content:

```
<Converter_V1>
<DestinationPath>
    <SourceFile>d:\Test\data\e01 (08).RAW</SourceFile>
    <SourceFile>d:\Test\data\e02 (08).RAW</SourceFile>
</DestinationPath>
</Converter_V1>
```

Calling the file

Example 1:

`imcfrmtcvrt -a d:\data.xml`

All files specified in the file: "*d:\daten.xml*" are converted to "ASCII" (-a)

Example 2:

`imcfrmtcvrt -i -h -l d:\setup.xml d:\Logfile.log d:\Data.xml`

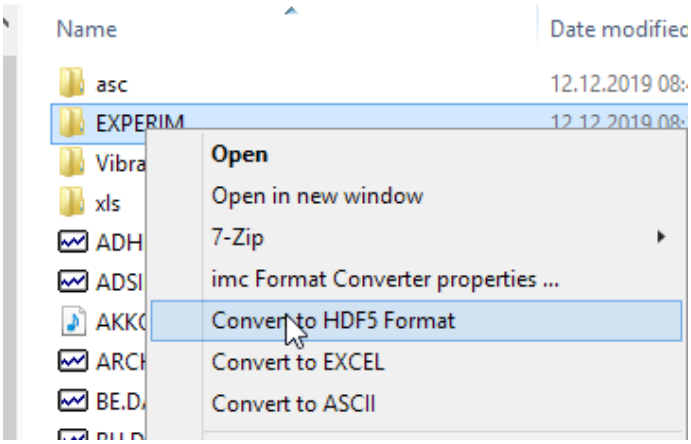
All files specified in the file: "*d:\daten.xml*" are converted.

The parameters for the conversion are defined in the file "*d:\setup.xml*" (-i).

All notifications are suppressed (-h) and saved in the log-file (-l): "*d:\Logfile.log*".

12.5.5 Conversion via the Windows Explorer

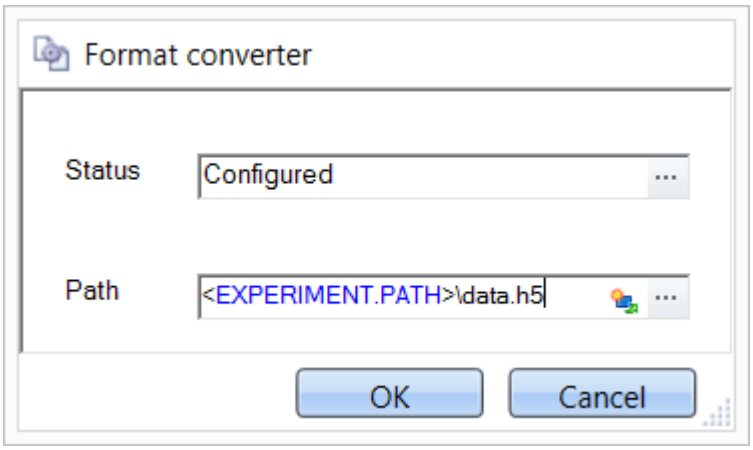
Installation of the imc Format Converter expands the Windows Explorer's context menu. The current templates for export of ASCII 1194 and EXCEL 1195, as well as the export format last selected on the page "*Other file formats* 1196" can then be applied directly to a folder or files:



12.5.6 Format Converter as imc STUDIO command

Calling

Via the Sequencer: "*External calls*" > "*Format converter*":



Configuration of the command: Format Converter

Status

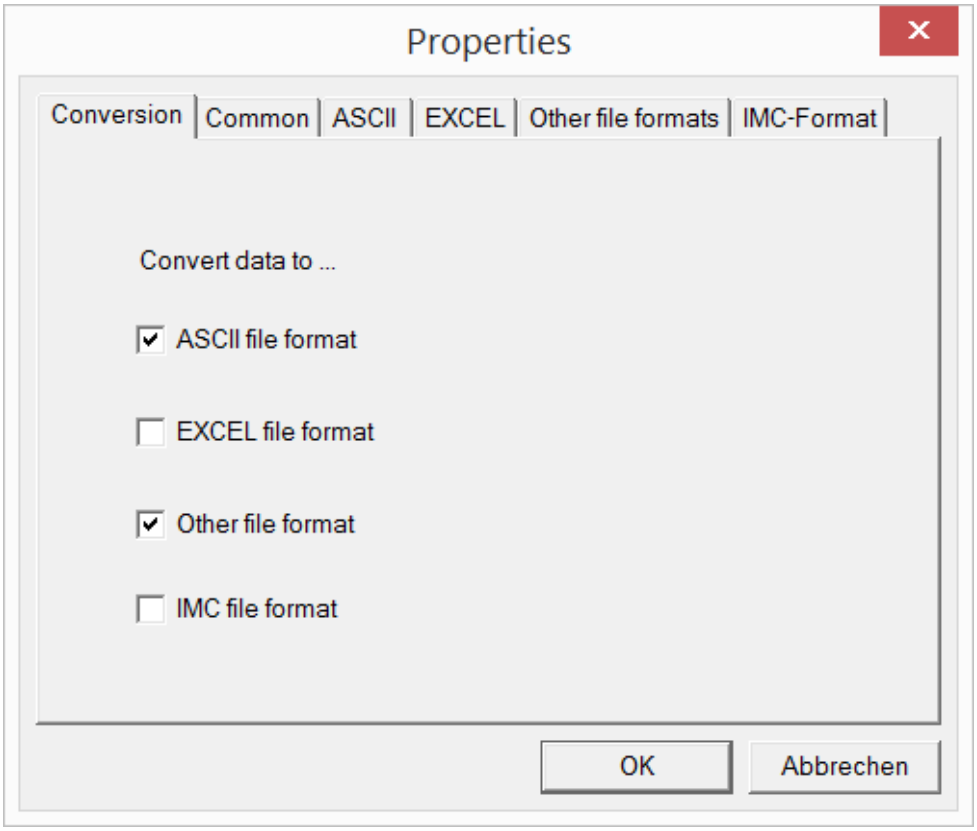
Immediately upon opening the command's configuration, the status is set to "*not configured*".

Path

- If just **one** file (*.dat, *.raw) of measured data is specified here, you can select whether to **convert either only the single file, or all files of measured data** 1193 which the **folder** contains.
- If a **folder** is specified here, then **all files of measured data in the folder** are converted:
  - If the option "*Save all files of a folder in one file*" among the settings on the tab page "*Common* 1193" (see the segment on "*Status*") is selected, then **all files of measured data** present in the specified folder are converted to **a single target file**.
  - If the option is **not selected**, then each file of measured data which the folder contains, the data are converted to **a separate file**.

Using the button ··· in the box "*Status*" you can open a dialog via which you can make all the settings for the conversion:

- Conversion: Target format. It is also possible to select multiple formats.
- Common 1193: Storage location and whether individual files or a multi-file is to be created.
- ASCII 1194: Selection of the export template and specifying of the file extension.
- EXCEL 1195: Selection of the export template and specifying of the file extension xls or xlsx.
- Other file format 1196: Selection of the format and if necessary its formatting options.
- IMC-file format 1197: Save the data to a file in the imc format, or as a copy to a different memory location.



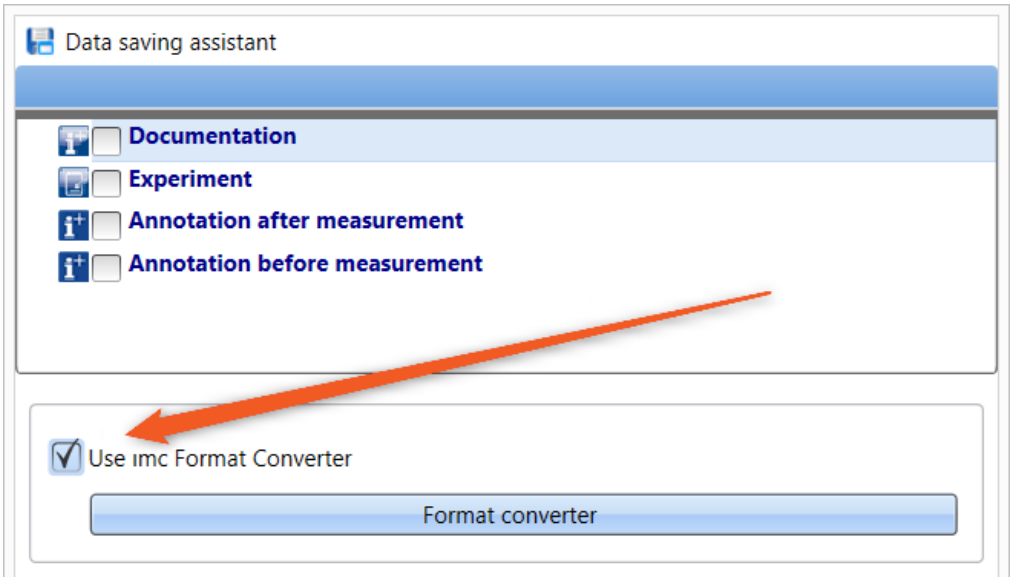
Setting the Format Converter

12.5.7 Converting via the Data saving assistant

Calling

Via the Sequencer: "*Read/write data*" > "*Data saving assistant*"

Along with supplemental information on the measurement, the "*Data saving assistant*" also allows the use of the imc Format Converter. Clicking on the button "*Format Converter*" opens its settings dialog, with which you can configure the imc Format Converter in accordance with the description of the imc STUDIO-command 1202.

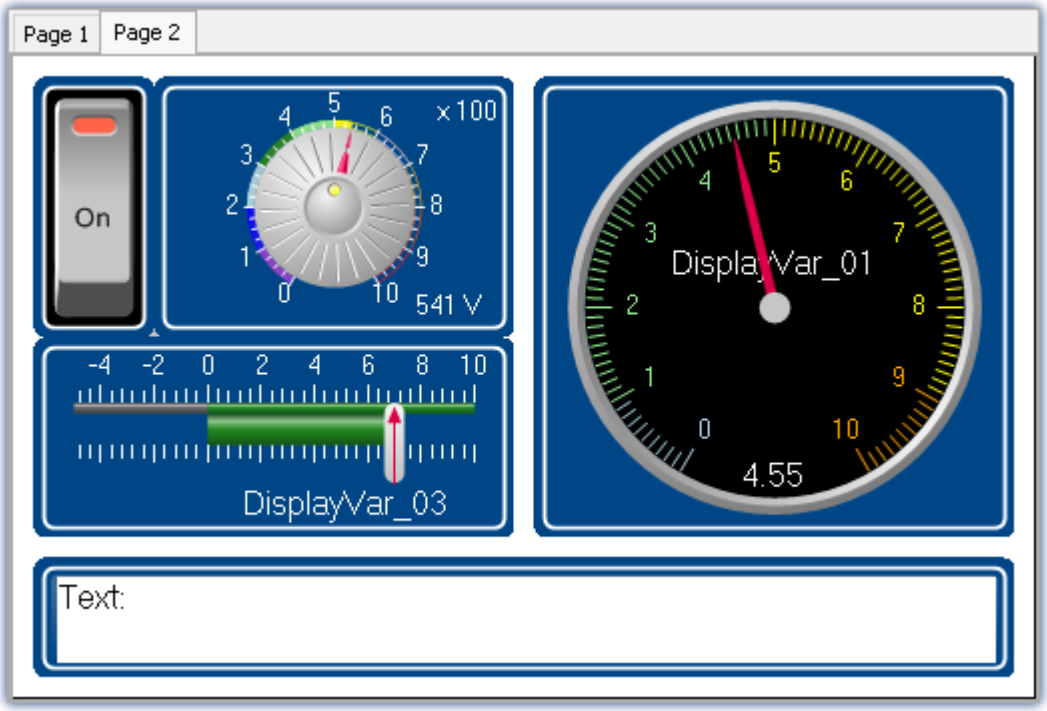


Format Converter as component of the Data Saving Assistant

13 Panel - Data visualization and controls

Panel is the imc STUDIO component with which **measured data and events** can both be **displayed** in instruments, and influenced using other controls (both controls and instruments are called Widgets).

With the Panel you can achieve **personalized user interfaces** (called Panel pages 1535), which are optimized for your specific needs.



imc STUDIO Panel with two pages (example)

Panel page

The Panel consists of one or more Panel pages. On a page, it is possible to layout various Widgets.

There are two kinds of Panel pages available:

- Dialog pages 1535 optimized for on-screen display
- Report pages 1535 optimized for printouts

Widgets

Various Widgets 1213 are available as instruments and controls. The amount of Widgets varies by the product configuration. Even the curve window 1265, familiar from other programs, can be placed on a page.

By means of the Widgets, you can affect the measurement interactively. It is also possible to assign actions (commands) 1263 to Widgets such as switches and buttons.

Operation

For the positioning of Widgets 1235 and the assignment of channels and variables 1550, a variety of operation techniques can be used, such as Drag&Drop or context menus.

The Widgets can be operated by keyboard or mouse.

13.1 Ribbon

13.1.1 Control

Device control

Here you find the menu actions for device control.

For a detailed description, see the manual:
"*Setup - Manual*" > "*Ribbon*" > "*Control* 189".

View

Menu item	Description
Panel Fullscreen	The Panel is displayed in fullscreen mode. The imc STUDIO **user interface is hidden** and can thus no longer be operated or accessed (excepting the Panel pages). See chapter "*Fullscreen* 1544".

Export

Menu item	Description
As PDF (Export)	Saves the current Panel page in PDF format.
As Graphics (Export)	Saves the current Panel page as a graphics file. You can select among the following graphics formats: • Portable Network Graphics (*.png) • Windows Bitmap (*.bmp) • JPEG (*.jpg)

Print

Menu item	Description
Print Preview	Opens a print preview. Here you can see the possible appearance of printing results. This can be helpful when Widgets such as the curve window are used, which have their own color settings for the printout.
Print	Opens the dialog for printing the Panel pages.

13.1.2 Navigation

Control

Menu item	Description
Post-Processing	This option currently has no function.
Find in Variables	This option currently has no function.
To Beginning	This option currently has no function.
Start	This option currently has no function.
Stop	This option currently has no function.
To End	This option currently has no function.
Repeat	This option currently has no function.
Faster	This option currently has no function.
Slower	This option currently has no function.

Curve window

Menu item	Description
Zoom	This option currently has no function.
Absolute Time	This option currently has no function.
Relative Time	This option currently has no function.
Scroll	This option currently has no function.
Stretch	This option currently has no function.
Pause	This option currently has no function.

Design

Menu item	Description
Design Mode	• Activated: the Panel page can be edited, the Widgets can not be operated. • Deactivated: The Panel page can not be edited. The Widgets can be operated.
Navigation Bar	The navigation bar currently has no function.
Selection	The navigation bar currently has no function.
Variables	This option currently has no function.

Export

Menu item	Description
As PDF (Export)	Saves the current Panel page in PDF format.
As Graphics (Export)	Saves the current Panel page as a graphics file. You can select among the following graphics formats: • Portable Network Graphics (*.png) • Windows Bitmap (*.bmp) • JPEG (*.jpg)

Print

Menu item	Description
Print Preview	Opens a print preview. Here you can see the possible appearance of printing results. This can be helpful when Widgets such as the curve window are used, which have their own color settings for the printout.
Print	Opens the dialog for printing the Panel pages.

13.1.3 Design

New

Menu item	Description
Insert Page	Selection window for creating a new Dialog- or Report-page is opened. (see: Insert - Dialog / Report 1535)
Standard Dialog	Creates a new Standard dialog page.

Edit and Clipboard

Menu item	Description
Design Mode	• Activated: the Panel page can be edited, the Widgets can not be operated. • Deactivated: The Panel page can not be edited. The Widgets can be operated.
Lock/Unlock Page	Lock/Unlock the current page. Individual pages can be protected against changes (see Lock and unlock page 1543).
Undo	Reverses the last change.
Redo	Restores a previously reversed change.
Delete	Deletes the selection (the page or the selected Widgets).
Select all	Selects all of the page's Widgets.
Cut	Clips out the selection or the selected Widgets and moves them to the Clipboard.
Copy	Copies the selection or the selected Widgets to the Clipboard.
Paste	Inserts the content of the Clipboard at the selected position.

Navigation

Menu item	Description
Navigation Bar	The navigation bar currently has no function.
Selection	The navigation bar currently has no function.
Variables	This option currently has no function.
Zoom	This option currently has no function.
Absolute Time	This option currently has no function.
Relative Time	This option currently has no function.
Scroll	This option currently has no function.
Stretch	This option currently has no function.
Pause	This option currently has no function.

View

Menu item	Description
Panel Fullscreen	The Panel is displayed in fullscreen mode. The imc STUDIO **user interface is hidden** and can thus no longer be operated or accessed (excepting the Panel pages). See chapter "*Fullscreen* 1544".

Im-/Export

Menu item	Description
Import Page	This lets you import a saved page (file type: dbv). If the page's designation is identical to that of an already existing one, you are prompted whether to overwrite the existing page. If you answer "No", the imported page is appended along with a new page title (incremented number).
As Panel Page (Export)	Saves the current Panel page as a file (file type "Panel pages (*.dbv)").
As PDF (Export)	Saves the current Panel page in PDF format.
As Graphics (Export)	Saves the current Panel page as a graphics file. You can select among the following graphics formats: • Portable Network Graphics (*.png) • Windows Bitmap (*.bmp) • JPEG (*.jpg)

13.2 Tool windows

The following tool windows are available for the purpose of configuring the Panel pages:

Tool Windows	Description
Widgets	Depending on the product configuration, a certain amount of display and control elements (**Widgets**) in various groups.
Data Browser	The tool window **Data Browser** shows the available data such as measurement channels, Display variables or pv-variables in a hierarchical tree diagram. With this tool window you can also set the variable linkage 1550 for a Widget.
Properties 1212	The Properties window changes its contents in response to the object selected. If no Widget is selected, the page properties 1538 are displayed.
Page templates 1229	Ti create new pages there are templates for **dialogs** (optimized for screen display) and **reports** (optimized for printout).

13.2.1 Properties (general)

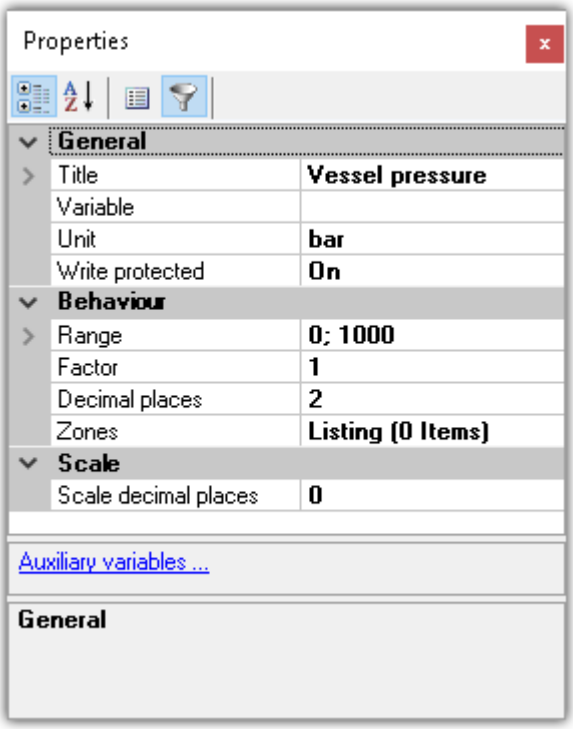
In the tool window "**Properties**" (or via the context menu of the selected Widget, (see context menus of the Widgets 1237), the individual Widgets or the Panel pages can be adapted. The content of the Properties window depends on the respective element.

There are two display modes:

Icon	Description
	Display of the important properties. Properties which normally are adapted to the specific requirements are displayed (e.g. measurement range and variable).
	Display of all properties.

Example

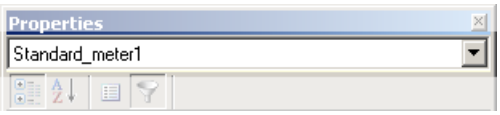
Below, one possible configuration of a *Standard meter (Automotive)* is described, with the important properties indicated.



In the window *Properties*, these properties can be adapted to your wishes. Here, for instance, the title *Vessel pressure* has been added, the unit set to *bar*, and the display range set to 0 through 1000.

Structure and operation

Selection list



Shows the name of the selected element. By means of the selection list, it is possible to select a different element. The selection on the Panel page assimilates itself to the selection.

Only shown in the tool window. Not in the "free-floating" "Properties" window.

Toolbar

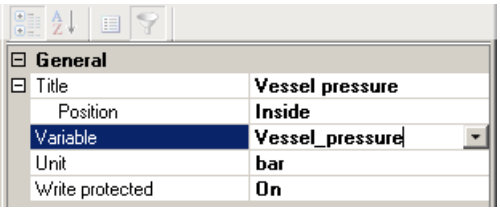
Alphabetical sorting of the properties list.

Grouping of the properties by property type.

Display of all properties.

Display of the important properties. Properties which normally are adapted to the specific requirements are displayed (e.g. measurement range and variable).

Properties



List of all properties.

- Left column: Name of the property
- Right column: Current setting

To edit a setting, click in the associated text box. The desired values can be entered on the keyboard. If there are selection from which to choose, a corresponding button appears at the right margin:

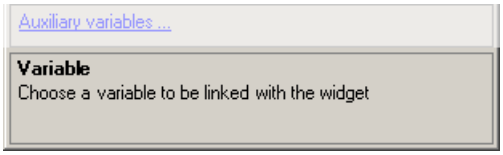
Opens a selection dialog (a list or the Data Browser)

Opens a dialog (for instance Zones-dialog or event)

More dialogs

This area lists Options-dialogs which can be opened at the click of the mouse. Here you will find dialogs in addition to that for the Properties.

Description



Here you have a description of the selected property.

13.2.2 Widgets

The tool window "Widgets" provides the building blocks from which to construct a Panel page. The window contains multiple groups. In general, the groups **Curve window**, **Standard**, **Shapes** and **Devices handling** are available. Furthermore, there are specially designed Widgets in groups such as **Automotive**, **Industrial**, **Designer**, **Aviation**.

Note Which groups are visible

Which groups are available depends on the product configuration.
(see Product Configuration / Licensing 29 or Technical Data Sheet)

Widgets are placed (inserted) 1235 on the Panel page.

- Widgets - Operation and Properties 1235: Select this link to go to further information on operation of the Widgets.
- Special Widgets 1521: The descriptions of certain special Widgets are presented here.

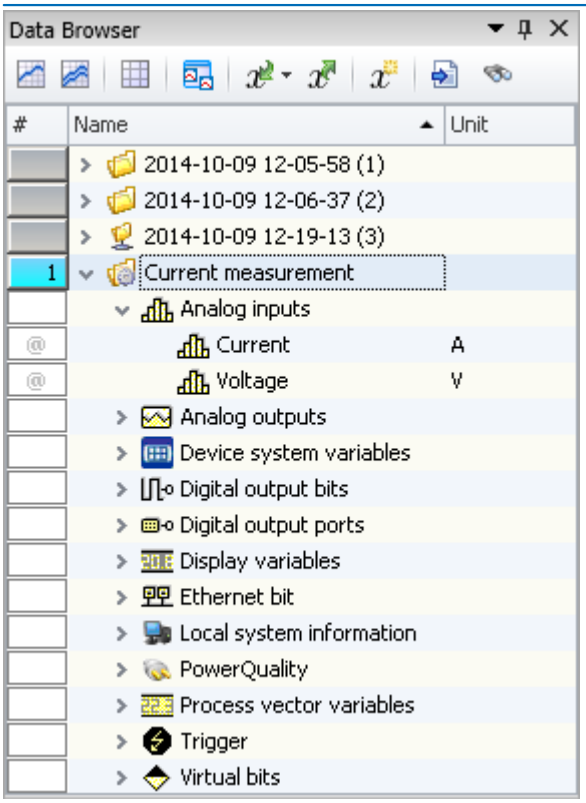
13.2.3 Data Browser

The tool window Data Browser **shows the available data (called variables)** such as measurement channels, Display variables or pv-variables in a hierarchical tree diagram. From this tool window, you can move the variables by means of drag&drop 1551 to a Panel page and link them with Widgets. You can also navigate through measurements 1221 and assign measurement numbers 1220.

FAQ

Why are the current channels not displayed?

Answer: The tool window Data Browser **only shows changed device configurations (e.g. channels) once** the menu action "***Process configuration***" has been selected. On this topic, see the chapter "*Process configuration* 241" in the documentation on the component Setup.



Data Browser in the Panel

Structure

The Data Browser has **a toolbar** 1222 and a table with multiple columns. Columns can be hidden or shown, according to the user's choice.

The **Table** presents the measurements and measured data in a tree diagram. The table features the following columns:

Column	Description
Numbering column (#)	Here, the assigned measurement numbers are displayed. Additionally, it is possible to lock in the measurement number (see "*Assigning / Locking measurement numbers* 1220").
Name	In this column, the measurement name and the variable name are displayed. The measurement name is determined by the data storage settings, but can be edited subsequently.
Event time	If a measurement has multiple channel-start-times (event-times), one entry appears in the column "*Event time*" for each different event time. Click on the column entry in order to select the desired event from the dropdown-list. The associated Widgets then show the selected event along with the associated variables/channels (see "*Event time* 1218"). Preselection: "*All available*"
Comment	Displays the content of the variable's property: *Comment*. With device variables, this can be defined by means of the Setup pages.
Category	Shows the content of the variable's property: *Category*.
Unit	Displays the content of the variable's property: *Unit*. With device variables, this can be defined by means of the Setup pages.

Note — Showing columns

By default, not all columns are shown. You are able to add columns by means of the contextmenu item "Column selection". After any changes, use **save** to preserve the **View**.

Variable groups

The following is a list of the most important variable groups:

Groups	Description
Device variables such as: analog inputs, Display variables, Fieldbus variables, ...	A list of the device variables is provided here: "*Setup - Device Configuration*" > "*Configuring Channels and Variables*" > "*Channel table* 340".
User-defined variables	You can create your own local (PC-) variables. You can also create them in their own separate category. For more info, see the section: "*User-Defined Variables* 1149".
System variables, such as: Trigger, measurement status, system-information and user name	In an expanded tree diagram, a variety of information is presented. For more info see the section: "*Trigger, devices and system variables* 1138".
Virtual channels	Results channel of imc Online FAMOS and imc Inline FAMOS.
Bus Decoder	Results channel of Bus Decoder.
imc FAMOS	Results channel of imc FAMOS.
...	...

The groups can be expanded. All the variables they contain are then listed.

Using personally customized groups

You can deactivate grouping according to category and create your own grouping system instead.

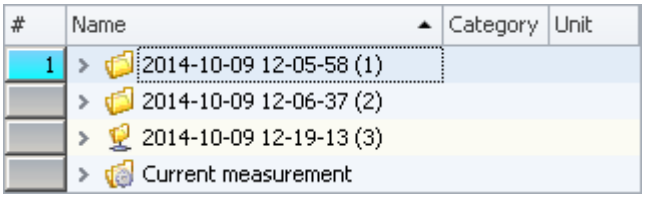
The setting "*Grouping by categories*" can be deactivated in the Options (under "*Data Browser*" > "*Categories*"):

Ribbon	View
Extra > Options ( )	All

Options - Categories	Description
Grouping by categories	The variables are grouped in the Data Browser according to their respective categories (e.g. "Analog inputs", "User-defined variables"). This can be deactivated in order to sort the variables by their names. You can create your own groups by naming the variables according to the following syntax: "Group.Name". For instance, if there are two channels: MeasPoint1.Temperature and MeasPoint1.Voltage, these variables are automatically grouped in the group: "MeasPoint1".

Measurements and measurement names

The Data Browser, along with the entry for the **Current measurement** ( ), displays all **measurements saved** with the experiment.



Measurement	Description
Current measurement	The variables under "*Current measurement*" always reflect the current status. While a measurement is being performed, you can view the current measured data as these variables under "*Current measurement*". **Notes on circular buffer memory** By default, the circular buffer memory in the plug-in Setup is activated for display of the channels. **Under "*Current measurement*", only the content of the circular buffer memory is displayed.**
Saved measurements	The measurements are arranged as list entries on the same level as Current Measurement. Among other factors, the measurement's name depends on the data storage settings (Setup). It may, for example, consist of a time stamp and take the form "*yyyy-MM-dd HH-mm-ss*". (year-month-day hour-minute-second) **Notes regarding measurement** • You are able to expand the entries of all measurements without needing to load 1219 the measurements first. The information on which channels belong to the measurement is also available before the measurement has been loaded. • The measurement name can be arranged to be defined by the user. On this topic, see the option "*Storage location measurement data* 129". Here, you can save measured data in subfolders in order to improve the sorting of data in the Data Browser. • Display of saved measurements in the Data Browser can be deactivated. On this topic, see the option: "*Variables*" > "*Measurement Administation*" > "*Access to stored measurements* 137". This can be helpful, for instance when very many measurements are performed. In such a case, very many entries would appear in the Data Browser. If these measurements do not need to be loaded, then it can provide better clarity of overview to deactivate the display of the saved measurements.

Measurement name, and renaming measurements

The names of the saved measurements correspond to the folder names as they appear on the hard drive. It is possible to rename the measurements subsequently via the Data Browser.

- To do this, open the measurement's context menu and click on "*Rename*".
- Or select the measurement and press the key <F2>.

When you perform renaming, the corresponding **folder on the hard drive is also renamed**.

Note — Renaming

Note that you may only use symbols which are allowed.
If a sub-folder structure was created using the option "*Storage location measurement data*", it can not be subsequently modified in this way. Only the measurement folder itself can be renamed.

Event time

If a measurement has multiple channel-start-times (event-times), one entry appears in the column "*Event time*" for each different event time. Click on the column entry in order to select the desired event from the dropdown-list. The associated Widgets then show the selected event along with the associated variables/channels.

The selection is available if the individual channels' **start times** are **different**. For instance, this is the case when in a triggered measurement the individual channels are triggered in succession by different events.

Display All

The selection list provides the entry "*All available*". This entry is selected by default. It causes all channels and their associated events to be displayed, even if these were started at different points in time.

Exception: "*Save trigger events in individual files*". With this setting, each trigger-event for the channels is saved in a separate folder. In consequence, the same channel-name can exist in multiple trigger folders. In this case, only the channel of the first applicable event is displayed.

Show relationships

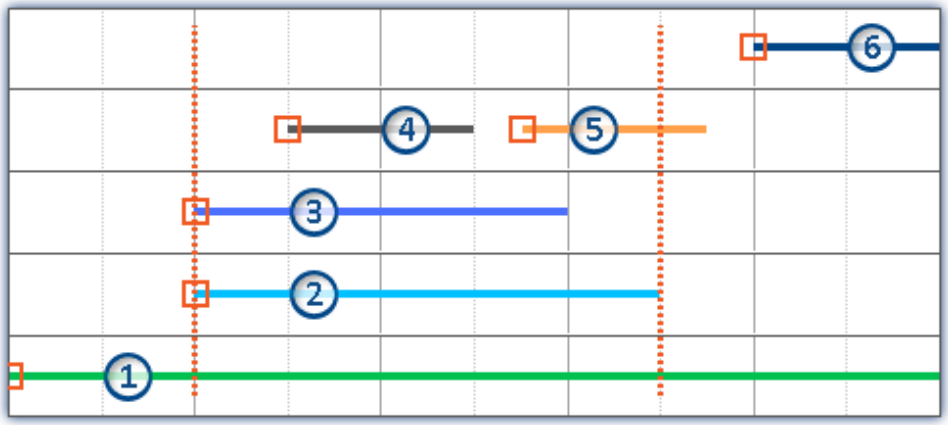
When a defined event is selected, the system will make every attempt to display as many channels as possible. The channels displayed have some time overlap.

Example: A channel starts at 14:00 and ends at 15:00.

If this channel's event time is selected, all channels are displayed which comprise measured data in the time between 14:00 and 15:00.

Exception: "*Save trigger events in individual files*". See the explanation above.

Example



Dotted lines: Start and stop-times of channel 2; e.g. 14:00 and 15:00 from the above example

5 channels are recorded. Counting from the bottom to the top:

- Event 1: The first channel responds to "*BaseTrigger*" (start button).
- Event 2 and 3: The second and third channels are linked to the same trigger (e.g. "*Trigger_01*"). This produces one single event-time.
- Event 4 and 5: The fourth channel has two events. This produces 2 event-times if "*Save trigger events in individual files*" is activated. Otherwise, only one event time.
- Event 6: The fifth channel was triggered once all other channels governed by the triggers have concluded (except the first channel).

Thus, the Events-list shows 4 or 5 event times, depending on the setting "*Save trigger events in individual files*".

Depending on the selection, the relevant individual channels among these are loaded and displayed.

Example: The second event time is selected (Channels 2 and 3). The space of time corresponding to the longest channel belonging to this event is used for selecting the other channels (2). All channels which have values within this space of time are displayed (1, 2, 3, 4 and possibly 5).

Channel 5 is displayed if "*Save trigger events in individual files*" is not activated. When this option is activated, it is only possible to display one event time at once. In that case only the first channel (4) is displayed.

Operation - Loading and unloading measurements/variables

Saved measurements are located on the hard drive. Operating from the Data Browser, it is possible load and display these measurements. Since the saved measurements can be arbitrarily large, by default they are not loaded. The software knows the channel name and the path to their storage location, but the data located there are not in the software's own memory.

However, only variables which are already loaded can be displayed or processed.

Note Resources

It is best to only load measurements/variables which are currently needed. Loaded elements make demands on the memory. If any measurement is no longer needed, it is best to unload it.

Automatic loading on demand: If a Widget is linked with a variable belonging to a saved measurement, this variable is loaded automatically. All of the measurement's other variables remain unloaded. It is exactly the same way with all other linkages to variables (e.g. a imc FAMOS-sequence). A soon as the value of a variable is required, that variable is loaded immediately.

To **load** the **complete measurement**, open the measurement's context menu and click on the corresponding item.

To **load** a **single variable**, display the variable in a Widget for example.

To **unload** the **complete measurement**, open the measurement's context menu and click on the corresponding item.

Folder symbol - Measurement status

The folder symbol in front of the measurement's entry reflects the measurement's status:

Status	Description
Current measurement	The is the current measurement, not a stored measurement.
Saved measurement status: unloaded	This stored measurement is not loaded. It is not used and not displayed.
Saved measurement status: loading	This stored measurement is in the process of being loaded. This is a temporary condition.
Saved measurement status: loaded	This is a saved measurement to which at least one loaded variable belongs. Loaded measurements can be viewed and processed.

Note — Loaded variable - Apply changes

When a variable is loaded, its associated file is loaded from the hard drive in a single action. To apply changes made to these files from outside/by other programs, the file must be re-loaded.
To do this, open the measurement's context menu and select the item: "*Reload measurement(s)*".

Operation - Assigning/Locking measurement numbers

When a saved measurement or "*Current measurement*" is **selected** in the Data Browser, a **measurement number is automatically assigned** to this entry. The measurement number is displayed in the left-hand numbering column (#). E.g., for the measurement number 1: 1. You may assign numbers from 1 through 99.

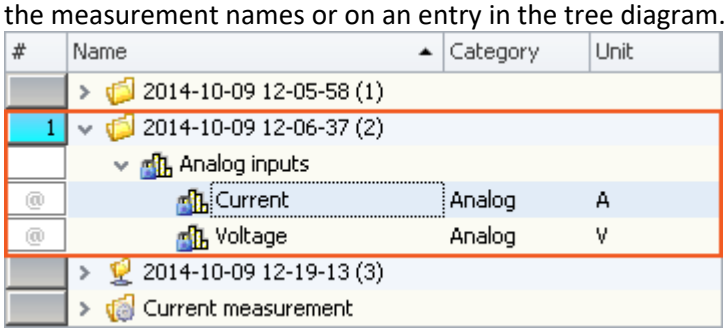
By means of the measurement number, it is possible to access measurements without using or even knowing their respective assigned names.

Example

- You intend to **display all saved measurements in succession** on a Report page. You have a curve window which always displays the channel: "*Channel_001*". However, instead of showing the channel belonging to the current measurement here, you want to display the saved measurement with measurement number: 1. When any measurement is selected, the curve window shows Channel_001 of the selected measurement, since it has the number 1.
- You wish to perform an imc FAMOS analysis on a saved measurement. The imc FAMOS sequence is configured so that the measurement results are obtained from the measurement with the measurement number 1. The imc FAMOS results are then saved back with the measurement having the number 1.
- Each measurement in a curve window is to be compared with a comparison measurement. The comparison measurement is associated with the measurement number 1. The number is locked and remains with this measurement. Any other selection will now be assigned the number 2. The curve window is linked with two channels: one from the measurement having the number 1, and one from the measurement having the number 2. If a different measurement is selected, the reference channel remains in the curve window and the second measurement corresponds to the current selection.

Assigning a measurement number

- To assign a measurement the measurement number 1, click on the cell in the numbering column (#) next to the measurement names or on an entry in the tree diagram.



- To assign multiple measurement numbers, hold down the SHIFT key while clicking on the cells.

Locking/unlocking a measurement number

- If you click again on a previously assigned measurement number, the assignment is locked (1). As a result, the respective number can no longer be assigned.
- To undo the locking, click on the same cell again.

Operation - Control and Navigation

To open or close a branch (e.g. "*Analog inputs*"),

- use the arrow keys: left/right (cursor),
- use the "+" and "-" keys on the numeric keypad
- click on the corresponding symbol (/)

To navigate between the measurements, use

- the mouse or
- the arrow up/arrow down (cursor) keys

13.2.3.1 Toolbar

Toolbar: Variable

Menu item		Description
	Show individually	All selected channel variables (e.g. analog inputs) are opened **in separate** "free-floating" curve windows 1517. All other selected variables (e.g. Display variables) are added to the window "*Current values* 1228".
	Show together	All selected channel variables (e.g. analog inputs) are opened **in a shared** "free-floating" curve window 1517. All other selected variables (e.g. Display variables) are added to the window "*Current values* 1228".
	Show current values	Opens the dialog Current values 1228, which displays the values of selected variables.
	Always on top	If this button is pressed, new display windows ("*Curve windows*" and "*Current values*") are always displayed in the foreground
	Load/Refill variable	Load: Creates or replaces variables with values from a file (user-defined). Refill: Imports values from a file to existing variables (device or user-defined). (See "*Load/Refill variable* 1222")
	Export variable	Exports variables' values to a file. (See "*Export variable* 1758")
	Create user defined variable	Creates user-defined variables, including single value- and text-variables. (More info is provided in the section "*User-defined variables* 1149")
	Find	Displays a search box for the purpose of locating elements in the Data Browser. The Data Browser displays only the entries found. All others are hidden as long as the search box is open.

13.2.3.1.1 Load/Refill variable

Load Variable: The action enables **creation or replace** of user-defined **variables with pre-defined values** from saved files. The user-defined variable inherits all of the properties of the variable from the saved file. (Possible targets: User-defined variables)

- If the variable doesn't exist yet, it is created.
- If the variable exists already, the existing variable is replaced with the new one.

Refill variable: The action enables **setting the values of an existing variable** from saved files. The variable must already exist before the command is run. Only the values are adopted. (Possible targets: among others device or user defined variables)

The values must be saved in a suitable file, for example one which was created using the commando: "*Export Variable*". It is possible to select one or more files. If you wish to use multiple files, they must all be located in one folder.

Note Background information

A function "***Load variable***" is now implemented, which resembles "Load Data" in imc FAMOS. Overwriting existing variables is allowed. A confirmation prompt appears for each conflict of variables. If any of the confirmation prompts is canceled, the entire import automatically cancels. In this way, the system prevents the loading of an incomplete set of controller parameters, for example.

When an existing variable is overwritten in the process of loading, then it is effectively completely replaced, including all of its properties. In consequence, the old variable is neither deleted nor is any new one created, so that no "events" are triggered by the deletion or creation. This is important for scripts, curve windows, etc.

In the process, the user-defined variables' scope is also overwritten. But when a variable for which no scope is defined is imported, the variable becomes "*temporary*".

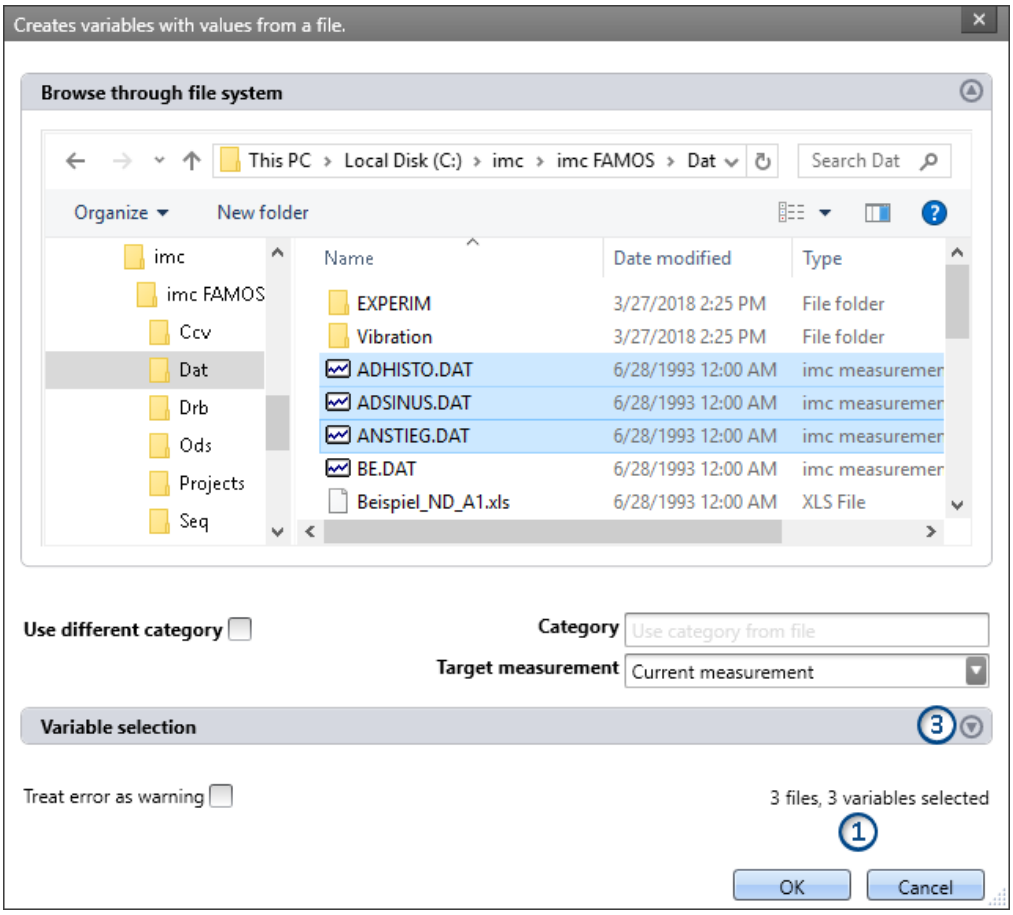
In contrast to imc FAMOS, there are variables which cannot be overwritten, e.g. channels or pv-variables. The reason is that, for instance, these variables belong to the specific device, or possess specific data formats and other properties which must be retained. If the user attempts to overwrite them, an error message is posted indicating that they can't be overwritten.

With the "*Load variable*" command, there is an option for determining whether existing variables can be overwritten with or without a confirmation prompt. By default, this option is activated, meaning that there is no confirmation prompt for overwriting.

By using "***Refill variable***", it is possible to change the content of existing variables. In this way it is possible, for instance, to change the content of pv-variables or user-defined variables. Since this only changes the variables' content and not their type, the target variable and the variable to be loaded must have the same properties, including, for example, the unit. Thus, only values and the sample count may be different. Therefore, please always use "*Load variable*" in cases where changes are possible.

Brief description:

In the upper region, there is an Explorer for selecting files:



Here you can use the "Explorer"-functions to **navigate to the files and select them**. After making a selection, you can close the dialog by clicking "OK". **All data selected are loaded** (or, when using the command, selected for loading).

Exception: If you don't select any file in the Explorer, all importable files belonging to the current folder are automatically selected.

File count (1): At the bottom right there is a readout of how many files are selected and how many variables belonging to these files are selected.

Variables-box (Variable selection) (3): Using the buttons ( ) near the right edge, you can, for example, expand the Variables area. You can also make additional settings such as

- **de-/selecting** specific files or **variables** using the checkbox (☐) , for cases where not all variables belonging to a file are to be loaded. This means that the are present in the list but will not be loaded. They can be activated again at any time.

Variables list	Description
Variable name	Name of the file/variable as found in the file.
Target variable name	Name of the variable as it is to be found in the Data Browser following load.

Other options

Category	Description
Only for: Load Variable	
Use different category	Put a checkmark in the box if you wish to create the variables in a deviating category. An input box for the new category appears. This category is applicable for all variables loaded. • **deactivated:** The variable is displayed **in the variable's category, or without any category** under "no category". • **activated:** The variable is displayed **in the specified category or without any specification under "User-defined variable"**. If you wish to create a category within the category "*User-defined variables*", do the following: write a "\" before the category name. Category: "MeasurementPoint_1". A category called "MeasurementPoint_1" is created in **parallel with** the other categories, such as "*Analog Inputs*". It contains the variable. Category "\MeasurementPoint_1". This creates a category "MeasurementPoint_1" **within the category** "*User-defined variables*". It contains the variable. This applies, for example, to variables created by means of the command: "*Load variable*" or to generating a user-defined variable 1149 by means of the Data Browser.

Target measurement	Description
Current measurement	**Load:** The variable appears in the Data Browser under "*Current measurement*". **Refill:** The variable overwrites the value of any variable existing under "*Current measurement*".
Last completed measurement	**Load:** The variable appears in the measurement which is in the Data Browser and is saved in that measurement's folder. The variable remains always available once the measurement has been loaded.
Measurement number Measurement#<No>	
Fixed measurement name	

Other options	Description
Treat error as warning	If the command is executed even though not all selected variables exist, error messages are posted. In normal cases, the Sequencer is exited when an error message appears. This also causes no variable to be imported. In order that the command always imports all variables, it is possible to activate this option. In this case, no error messages appear in the logbook, but warnings instead. Warnings do not cause the Sequencer to close.

13.2.3.2 Context menu

Context menu in the table - General

By right-clicking on the mouse over the table, the following context menu appears. Depending on the particular position, some of the items are hidden.

Menu item		Description
	Filter	Opens the Editor for creating a filter. The Data Browser then shows only the appropriate content according to the filtering.
	Transfer to imc FAMOS	The selected variables/measurements are transferred to imc FAMOS. imc FAMOS will start if it hasn't already.
	Measurement number #x	Only in the Numbering column. Opens an input box for entering the desired lowest measurement number for the subsequent selection. I.e. if the number "5" is selected, at the next selection the "1" is not assigned to any measurement, but instead "5", if it is still available.
	Export the selected variables	Exports values of variables to a file.
	Create variable	Creates user-defined variables, including single value- and text-variables. (More info is provided in the section "*User-defined variables* 1149")
	Delete selected variables	Deletes the selected variables from the Data Browser. Note that device variables and system variables can not be deleted.
	Edit selected variables	Opens the Editor for changing the properties of user-defined variables. The dialog matches that of "*Create variable*". With some changes, the variable is newly created. In these cases, the current value is discarded. A corresponding warning is issued first. When a device or system-variable is selected, the dialog opens. Here you can view the variable's properties, but not make any changes. (More info is presented in the section "*User-defined Variables* 1149")
	Show individually	All selected channel variables (e.g. analog inputs) are opened **in separate** "free-floating" curve windows 1517. All other selected variables (e.g. Display variables) are added to the window "*Current values* 1228".
	Show together	All selected channel variables (e.g. analog inputs) are opened **in a shared** "free-floating" curve window 1517. All other selected variables (e.g. Display variables) are added to the window "*Current values* 1228".
	Find	Displays the Search box. See the matching description for the toolbar 1222.
	Add Metadata column	Metadata columns can be displayed if saved along with the measured data. Here, you can activate existing metadata columns for display.
	Delete Metadata columns	Opens the dialog for deactivating the display of activated metadata column.

Context menu table - Save measurement

By right-clicking the mouse over the table on saved measurements, more context menu items become available:

Menu item		Description
	Load measurement(s)	Loads saved measurements for viewing and/or processing purposes (see "*Operation - Loading/unloading measurements* 1219")
	Unload measurement(s)	Unloads loaded, saved measurement(s) (see "*Operation - Loading/unloading measurements* 1219")
	Reload measurement(s)	Reloads already loaded measurements in order to undo changes or to load changes of the data to the hard drive (see "*Operation - Loading/unloading measurement* 1219")
	Delete measurement(s)	Deletes saved measurement(s). When a measurement is deleted, it is deleted from the hard drive. Undo is not possible.
	Load measurement settings	Loads the appropriate experiment settings for the measurement. When traceability is activated, all experiment settings of the respective measurement are saved separately. Thus, each measurement's own configuration can be loaded subsequently. See the description: • "*imc STUDIO (general)*": "*Options*" > "*Traceability of measurements* 132" • "*imc STUDIO (general)*": "*Information and tips*" > "*Data management*": ms-files 161
	Rename	Opens the input dialog for changing the measurement's name. If the name is changed, the associated folder on the hard drive is changed as well (see: "*Measurements and measurement names* 1216").
	Copy name	Copies the name of the measurement to the Clipboard.

13.2.3.3 Current values - Inspecting/changing variables

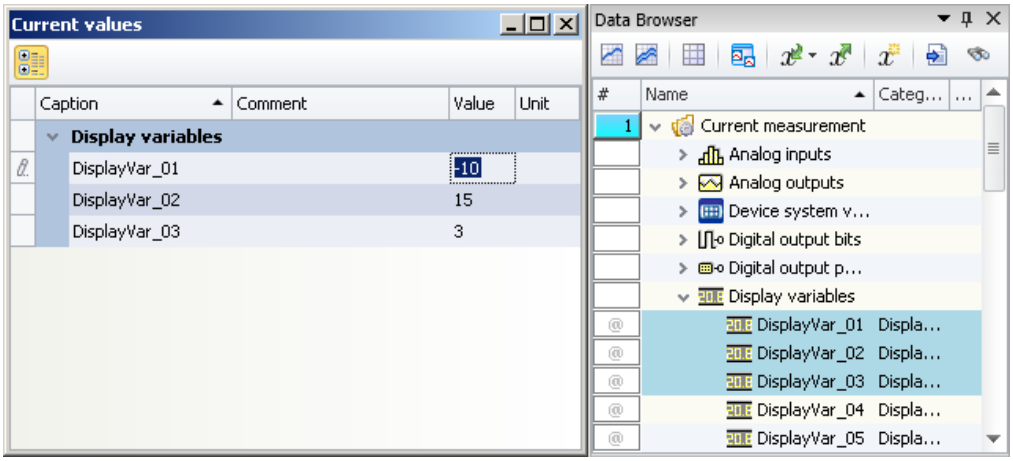
Note

The following description applies to variables which can not be viewed in the curve window, e.g. Display variables.

To directly **display** a variable's **values**, or to interactively alter them independently of a Widget,

- double-click on the entry in the variable in the tool window Data Browser
- or, select the desired variables and click on the "*Show*"-Button (/) in the toolbar of the tool window Data Browser.

Subsequently, a dialog opens, as shown below:



Inspecting or interactively altering variables
Example: three Display variables

When a **variable** is **deleted** from the Data Browser, it is automatically deleted from the Current Values-dialog.

Changing values

To change the value of variables (except bits), proceed as follows:

- click in Edit box
- enter a new value or the new text
- confirm the input with the <ENTER> key

To change a bit's value, proceed as follows:

- click in the checkbox

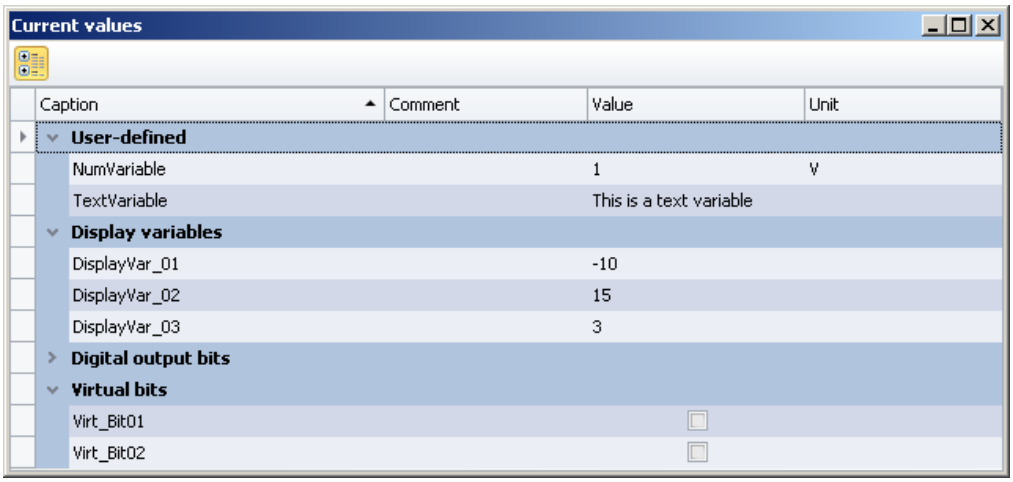
Structure and configuration

The dialog **current values** shows the desired variables in a table with multiple **columns**.

The variables are organized in groups. To activate or terminate the grouping, click on the Grouping button ( ).

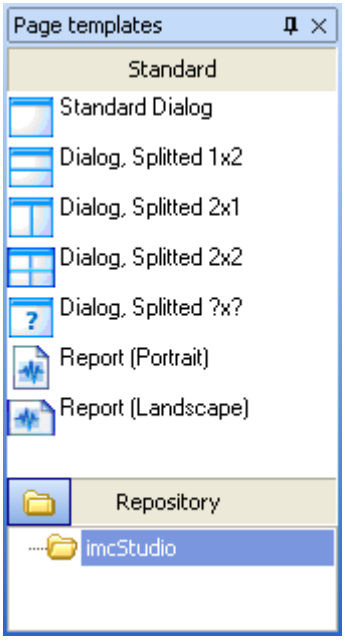
The **table** contains, among other things, the variable names and their current value. The value is displayed in the Edit box as a number or text. For Ethernet bits there is a checkbox.

The dialog settings are saved with the experiment and displayed along with it too, if it was open.



Dialog: Current Values
Sample variables

13.2.4 Page templates



Page templates

In this tool window, you find templates for a new page. The available choices are the same as in the dialogs for adding 1535 a new page.

13.2.5 Repository

The tool window Widgets and page templates each have their own repository. You can export completed pages and Widgets, in order to be able to insert them at other locations later.

Both repositories use the file system for the structuring and naming of the elements in the repository.

By default, the stored files are located here:

- Widget repository: %PUBLIC%\Documents\imc\imc STUDIO\Widgets
- Page repository: %PUBLIC%\Documents\imc\imc STUDIO\PanelPages

Adding elements to the Repository

Elements can be added to the Repository via the context menu or by using the Drag&Drop technique while holding down the <CTRL>-key. The elements in the Repository are assigned a distinctive name, for instance, the current page's name. If the name already exists, a prompt appears regarding whether to overwrite the existing file. If overwriting is denied, a "*Save as*" dialog appears.

It is also possible to add multiple Widgets to the Repository together as a single element. For this purpose, all desired Widgets must be selected when performing the adding procedure.

Importing elements from the Repository

You can use the Drag&Drop technique to move elements from the Repository to the desired location.

Menu items

You can perform the following actions by means of the Repository's menu items:

Menu item	Description
Set the Root Directory	This enables you to change the path to the respective repository. Note that the existing repository is not automatically copied in the process. You can copy it manually.
Delete	Deletes the selected element. This action can not be reversed.
Rename	Renames the element. If the name of a saved page is changed in this way, the page title also adapts itself accordingly when the page is re-imported from the Repository back to the Panel.
Delete	Deletes the selected folder along with all elements belonging to it. This action can not be reversed.
New Folder	Creates a folder which can be used for the purpose of grouping elements.
Rename	Renames the folder.

13.3 Context menu

Context menu: The Panel page's tab

Right-clicking the mouse over the page's tab or on the empty region at right calls the following context menu:

Menu item			Description
	New Page		Selection list for creating a new Dialog or Report-page is opened. (see: Insert - Dialog / Report 1535)
	Import Page		This lets you import a saved page (file type: dbv). If the page's designation is identical to that of an already existing one, you are prompted whether to overwrite the existing page. If you answer "No", the imported page is appended along with a new page title (incremented number).
	Export Page:		
		As Panel Page	Saves the current Panel page as a file (file type "Panel pages (*.dbv)").
		As Graphics	Saves the current Panel page as a graphics file. You can select among the following graphics formats: • Portable Network Graphics (*.png) • Windows Bitmap (*.bmp) • JPEG (*.jpg)
		As PDF	Saves the current Panel page in PDF format.
		Into the Repository	Saves the current Panel page in the repository (Tool Window: Page templates 1229).
	Show Page on Monitor		• embedded: The page is displayed in the plug-in's Panel main window (default selection) • <Monitor name>: The page is displayed in fullscreen mode on the monitor selected. Independent of the opened main window.
	Zoom Zoom (All pages)		Zooms the Panel page for the purpose of better display of, for example, Report-pages (see Zooming Panel pages 1544).
	Page Options:		
		Page Layout for Printing	Opens the dialog: "*Page Setup*" Here, you can change the setup and printout settings for Report-pages (including paper size, source, orientation, margin).

Menu item			Description
			The authorization to edit and display specific pages can be linked to specific user roles.
		Access Rights to the Page	What settings for user rights apply to a page when it is created in "*imc Standard User*" status? Example: All pages are write-protected. You can edit any pages which you yourself have made. With the access rights level "*Panel*" it is possible, for instance, for even the "*imc Standard User*" to create or import custom pages (by default, this is refused). The rights to the individual page depend on the respective user level. Thus, an "*imc Standard User*" who creates a page also has the right to edit and delete it. If a user on a different level creates the page, then these rights remain with an "*imc Advanced User*". The rights to execute these menu actions can be modified at a later time.
		Lock Page	Lock/Unlock the current page. Individual pages can be protected against changes (see Lock and unlock page 1543).
	Page		Deletes the selection (the page or the selected Widgets).

Context menu: Panel page

Right-clicking the mouse over the Panel page when the Design-mode is active calls the following context menu:

Menu item			Description
	Insert element		Creates a Widget (operating or display widget) at the selected location.
	Undo		Reverses the last change.
	Redo		Restores a previously reversed change.
	Paste		Inserts the content of the Clipboard at the selected position.
	Design Mode		• Activated: the Panel page can be edited, the Widgets can not be operated. • Deactivated: The Panel page can not be edited. The Widgets can be operated.
	Skin:		You can choose among multiple pre-defined color schemes (Skin). The color scheme influences the foreground and background colors of the Widgets and pages.
		Blue, Grey, ...	Widget-colors and the background colors of the current page align to the selected skin.
		Report Standard	Skin specially adapted to printout. Widget-colors and the background colors of the current page align to the selected skin.
		Save skin as	A user-defined skin is generated from the current colors of the Widgets. For this purpose, a dialog appears which prompts for confirmation before saving, since this affects all Widgets on the current page. The effects are displayed temporarily on the current page.
		Delete skin	Deletes the user-defined skin used.
	Grid		For the purpose of more easily positioning and moving Widgets, a grid can be imposed over the page (see Aligning to grid 1543).
	Adapt Page Size		See Adapt/automatically adapting the size of pages and Widgets 1541.

Menu item	Description
Properties	Opens the Properties window of the Panel page 1538.

Context menu: Widgets

Right-clicking the mouse on the "crosshairs" ( ) within the selected Widget calls the following context menu:

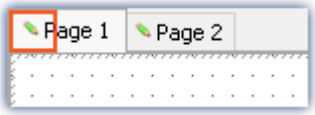
Menu item	Description
Copy	Copies the selection or the selected Widgets to the Clipboard.
Cut	Clips out the selection or the selected Widgets and moves them to the Clipboard.
Delete	Deletes the selection (the page or the selected Widgets).
Arrangement:	
To Foreground	Moves the selected Widget all the way to the foreground, on top of all other Widgets. (see: Widget-levels - Foreground/Background 1241)
To Background	Moves the selected Widget all the way to the background, beneath all other Widgets. (see: Widget-levels - Foreground/Background 1241)
Align	Aligns all selected Widgets to a line. (see: Align and Cascade 1242)
Cascade	Widgets are in stacked at the upper left corner of the Panel page according to their respective Widget-levels. (see: Align and Cascade 1243)
Group	Groups Widgets together (see: Grouping Widgets 1245)
Remove Unused Channels from Curve Windows	If curve windows are associated with channels which no longer exist, these associations can be deleted.
Curve Window Toolbar	• Activated: The curve window's toolbar is always displayed if a curve window is selected. • Deactivated: The toolbar is not displayed. All functions can be accessed via the context menu.
Events	Opens the Editor for linking commands with switches/buttons (see Linking commands with switches 1263)
Into the Repository	Saves the selection or the selected Widgets in the repository (Tool Window: Widgets 1213).
Properties	Opens the Properties window of the Widget 1245.

13.4 Design mode

To edit Widgets on a Panel page, the Panel must be in "*Design Mode*". The Design Mode is recognized by the appearance of the associated symbol.

Ribbon	View
Panel-Navigation > Design Mode ( )	Complete
Panel-Design > Design Mode ( )	Complete
Home > Design Mode ( )	Standard

Or at the associated symbol next to the page name.



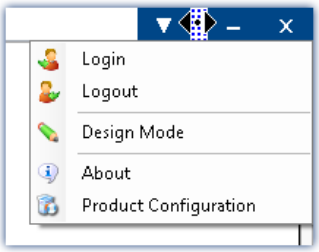
By default, this mode is **deactivated following the program start**. Activation/deactivation of the mode always applies **for all pages**.

Activating Design-mode

- via the menu item, or
- by moving a Widget to a Panel page by means of Drag&Drop.

Activation while in Full-screen mode:

Open the context menu over the bar on the top. In this menu, select the item "*Design Mode*".



Deactivating Design-mode

- by means of the associated menu item, or
- when you start the measurement.

Protecting against changes

You can have a selected page completely locked out of editing 1543 (independently of Design Mode).

Note No protection against deletion of the page

Even when Design Mode is deactivated you can delete or add pages. The Design Mode is not a way to "lock" the Panel component. Only the content of the pages is protected from changes.

13.5 Widgets - operation and properties

Widgets are placed (1235 **inserted** 1235) 1235 on the Panel page.

They offer a variety of **context menus** 1237 and **properties** 1245

In general, there are multiple ways to operate the Widgets:

- Move and change size 1238
- Select 1238
- Editing of Widget-elements 1239
- Move to background/foreground (Widget-levels) 1241
- Align and Cascade 1242
- Docking 1244
- Grouping 1245

Some Widgets have special functions, such as:

- link to variables (see the chapter: Variable linkage 1550)
- Extra variables (multiple scales/pointers and calculations of variables, e.g. Difference)
- Zones 1252
- Linking commands with switches 1263
- Use of placeholders 166

The structure

Each Widget consists of different ***elements*** 1239:

- Boxes displaying the **title** or the current **value**
- **Scales**
- **Numbers**
- **Meters**, **bar meters**, ...

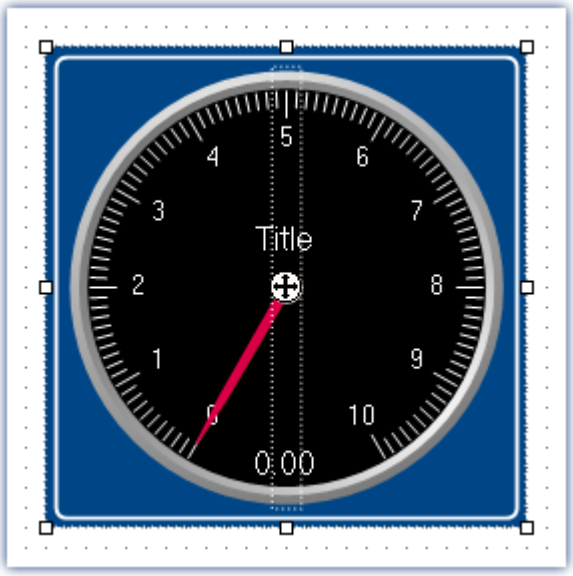
Some of these element have their own context menu 1237 and can be edited 1239. Among other things, they can be moved, or their size and color can be changed. Some of these changes can also be made through the use of the Widget's properties 1245.

13.5.1 Insert Widget

There are various ways to create a Widget on the page:

- Use Drag&Drop to move the Widget from the tool window "*Widgets*" to the Panel page
- Select the desired Widget in the tool window "*Widgets*". Use the mouse to drag a frame (of the desired size) to the Panel page
- Open the context menu 1232 belonging to the Panel page and select "*Insert Element*". Select the desired Widget.
- Use Drag&Drop to move a variable from the tool window **Data Browser** 1214 to Panel page and select the desired Widget (see: Variable linkage by Drag&Drop 1551)

Once you are finished positioning, the Widget appears as selected, as shown in the image below:



Widget in selected state (example)

Note Aligning to the grid can help

An easy way to position and move Widgets is to use a grid on the page (see Aligning to grid 1543).

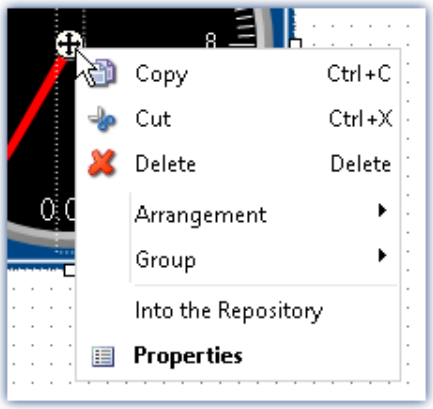
Commonly used Widgets

When you drag a variable to the Panel, a choice of Widgets is offered. The list is automatically modified according to your previous selection. Widgets which you had recently inserted frequently have priority among the choices available.

In the subgroups, all Widgets are available for selection, as accustomed.

13.5.2 Context menus of the Widgets

Widget Opening the context menu



Context menu of a Widget example)

To open this context menu, select the Widget and right-click on the "crosshairs" (⊕).

The context menu for the Widget appears.

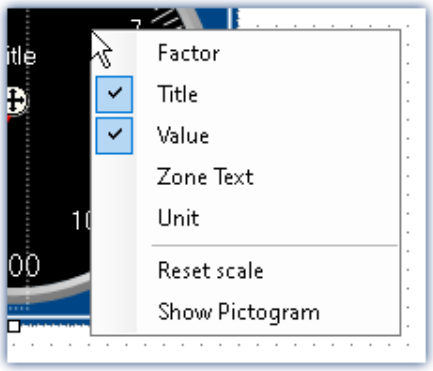
By this means, you can also open the Properties 1245 without using the tool window.

Other context menus

Some Widgets also have additional context menus:

- to show or hide Widget-elements 1239, such as boxes, and for resetting some properties
- to edit selected Widget-elements 1239

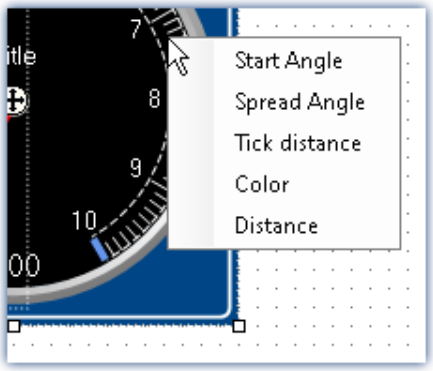
Here, using the example of a *standard meter*.



Context menu for the background

To open this context menu, select the Widget and right-click over the background.

A context menu appears, by which it is possible to show/hide Widget-elements 1239, for example.



Context menu for the Widget element

To open this context menu, select the Widget-elements 1239 and right-click over the background or the control.

A context menu appears, using which the Widget-elements 1239 can be edited.

13.5.3 Operation

Moving and changing size

Moving a Widget

You can move the Widget by clicking on it and dragging it by the "crosshairs" (⊕).

Changing the size of a Widget

Changing size

To change the size/proportions, use the mouse to drag it by the square grips on its edge.

Note Aligning to the grid can help

An easy way to position and move Widgets is to use a grid on the page (see Aligning to grid 1543).

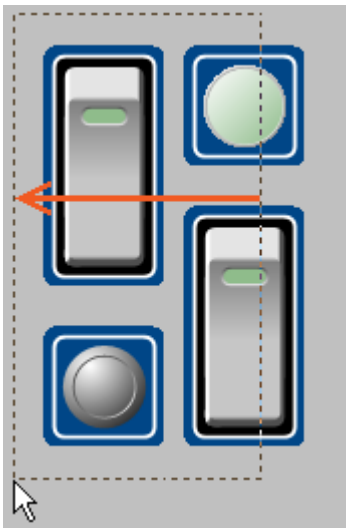
Selection

On a single page, you can select either one or multiple Widgets. The operations below can be performed as desired in any order.

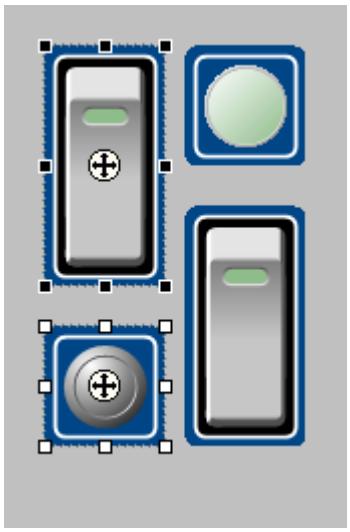
- To **select a Widget**, click on the desired Widget.
- To **de-select an individual Widget**, press the <Ctrl>-key while clicking on the desired Widget.
- To **de-select all Widgets**, click on an empty region of the Panel page.

Selecting multiple Widgets

- Method 1, dragging the frame:
  To select multiple Widgets, use the mouse to drag a rectangular frame around the desired Widgets. The starting point for dragging must be in an empty area of the page:

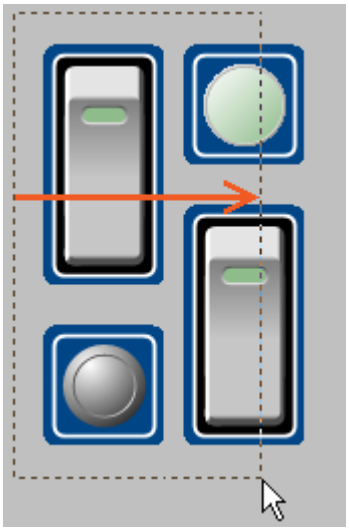


Dragging the frame from right to left

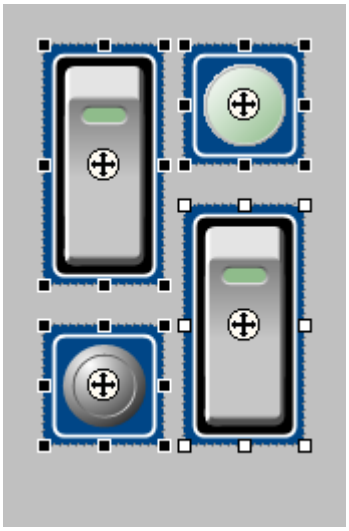


Completely framed Widgets are selected

If you drag the frame from right to left, only Widgets are selected which are completely encompassed within the frame.



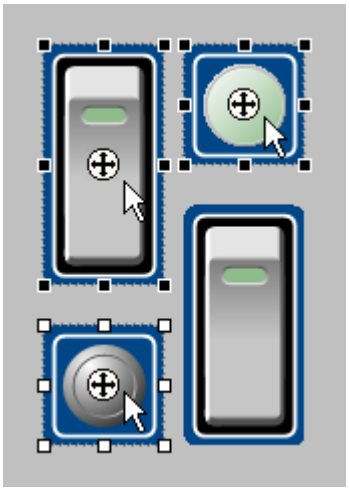
Dragging the frame from left to right



Any Widgets touched are selected

If you drag the frame from left to right, even such Widgets are selected which are only touched by the frame.

- Method 2, <Ctrl>+click
  To select multiple Widgets, you can also hold down the <Ctrl>-key and click on the desired Widgets.



Multiple selected Widgets

Editing Widget-elements

Each Widget consists of various **elements**:

- Boxes indicating the **title** or the current **value**
- **Scales**
- **Numbers**
- **Meters, bar meters**, ...

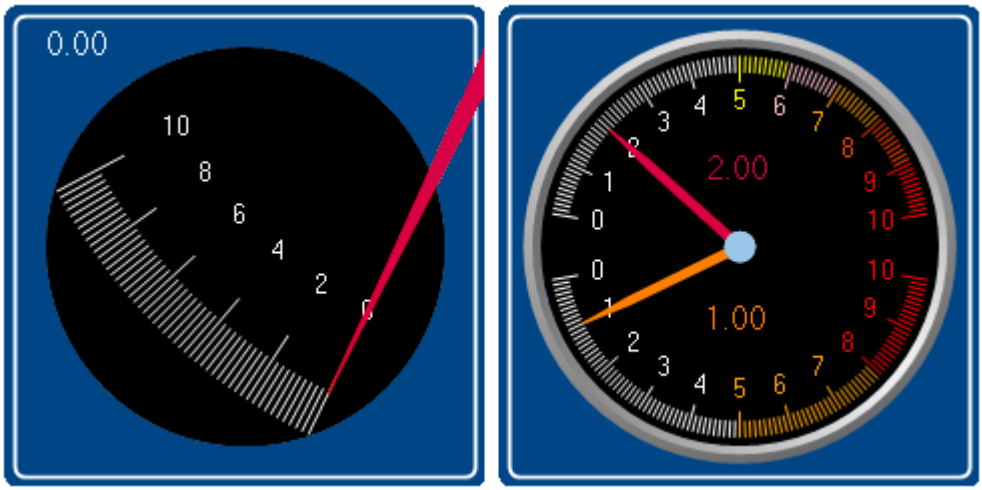
Editing elements

The various Widgets offer many ways to vary the appearance without having to set the Properties 1245. You can:

- move and enlarge boxes
- edit various Widgets elements

Example

Example of three edited Widgets
The one at right consists of two *Standard meters*



Example of three edited Widgets

Each Widget also has other options. To edit the element, you must first select it. Normally a frame around the element and a blue bar, for gripping it, appear.

To illustrate, the images below use the Widget *Standard meter* as an example:

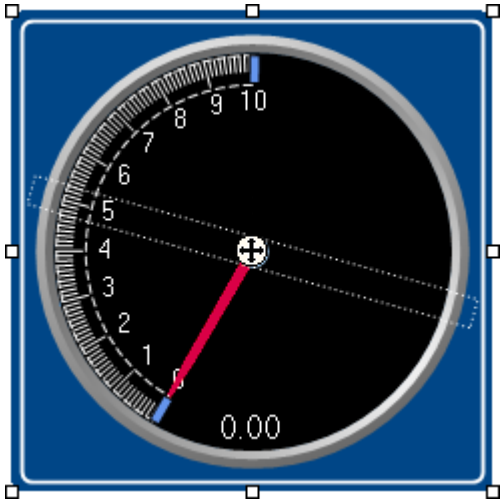
Example 1

Move and change the size of boxes

Move and change the size of boxes

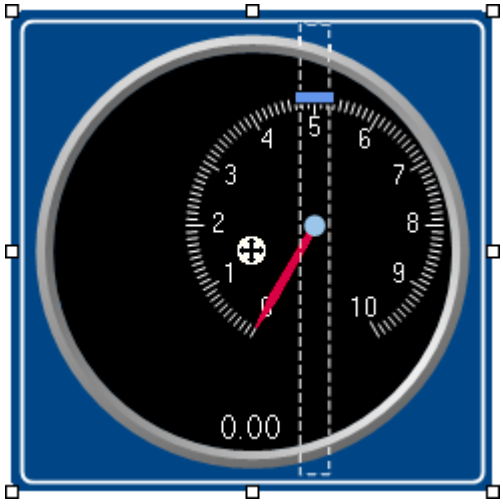
Example 2 Changing the scale angel and the sweep angle

(see also Properties 1245)



Example 3 Changing scale position and radius

(see also Properties 1245)



Widget-levels - Foreground/Background

The Widget-level is redefined upon inserting the Widgets. Each newly inserted Widget is positioned in front of all other Widgets. The levels are important if Widgets overlap or multiple Widgets are **docked** 1244.

By means of the level, you determine whether a Widget appears above or behind another one.

To do this, open the context menu of the"crosshairs" (⊕) within the selected Widget.

- Select *Arrangement*:

Action	Description
To Foreground	Moves the selected Widget all the way to the foreground, in front of all other Widgets
To Background	Moves the selected Widget all the way to the background, behind all other Widgets

Example

Thus, for instance, you can place a *Graphical switch* in the empty space belonging to a *Standard meter*:

Example of Widget-levels

Align and Cascade

Aligning Widgets

Widgets can be aligned along a line.

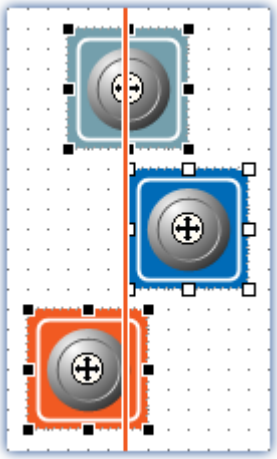
To do this, open the context menu of the "crosshairs" ( ) within the selected Widget.

- Select *Arrangement*:

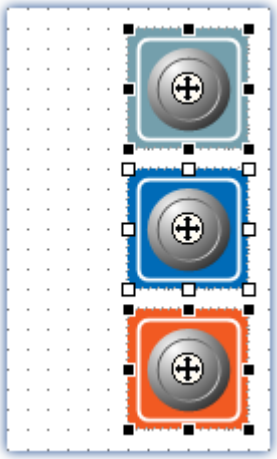
Action	Description
Align top	Alignment to the upper edge of the last Widget selected
Align left	Alignment to the left edge of the last Widget selected
Align Right	Alignment to the right edge of the last Widget selected
Align Bottom	Alignment to the bottom edge of the last Widget selected

Example

Three Widgets are selected. The middle one was selected last. The Widgets are to be aligned to the left edge:



Widgets are aligned to the left edge of the last Widget selected



Alignment results

Cascade Widgets

Widgets can be stacked on top of each other.

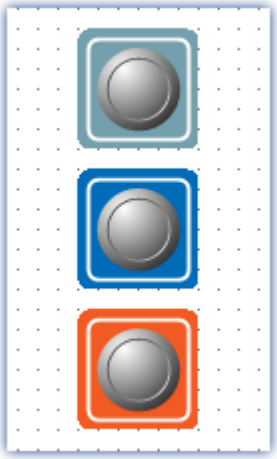
To do this, open the context menu of the "crosshairs" ( ) within the selected Widget.

- Select *Arrangement*:

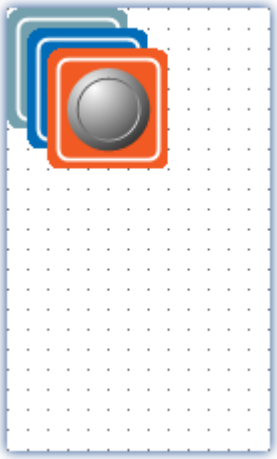
Action	Description
Cascade	Selected Widgets are stacked in the upper left corner of the Panel page according to their respective Widget-levels.

Example

Three Widgets are selected. The Widgets are to be stacked:



Widgets to be stacked



Results of stacking

Docking

With these commands, you can dock a Widget at your choice of edges. The dimensions of the Widget are adapted to the available width. The size can only be reduced from there and depends on the other Widgets and the page size.

To do this, open the properties of the Widget.

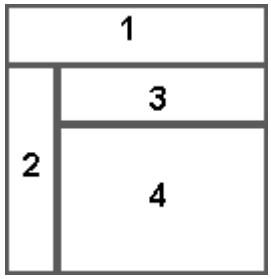
- Select under "*Dock*":

Action	Description
Free	The Widget is not permanently docked to any edge, but can be freely moved. The size can be varied toward any side.
Top	The Widget is permanently docked to the upper edge. The size can only be varied toward downward.
Left	The Widget is permanently docked to the left edge. The size can only be varied toward rightward.
Right	The Widget is permanently docked to the right edge. The size can only be varied toward leftward.
Below	The Widget is permanently docked to the top edge. The size can only be varied toward upward.
Fill	The Widget is extended in size to fill all available room. The size cannot be varied.

If multiple Widgets are docked, their respective positions depend on the **Widget-level**. Widgets positioned further in the background are closer to the edge.

Example — Example for docking in a cell of the split Panel page

Widget-levels:



The docking of the Widgets is as follows:

1 - Top
2 - Left
3 - Top
4 - Fill

Grouping Widgets

Widgets can be grouped together. Grouped Widgets can be moved and magnified jointly. Shared properties can be set in concert.

To do this, select the desired Widgets and open the context menu of the "crosshairs" (⊕) within a selected Widget.

- Select *Group*:

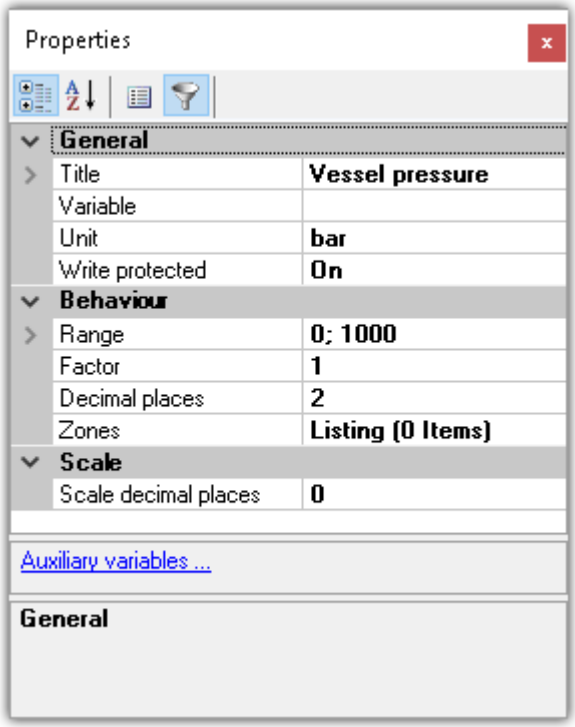
Action	Description
Create Group	Selected Widgets are joined to a group.
Dissolve Group	Dissolves the grouping. The Widgets can then each be edited separately.
Enter Group	Opens the existing group. The properties of the individual Widgets can be edited separately.
Leave Group	Closes an open group.

In order to edit the respective Widgets individually, select *Enter Group*. Then it is possible to edit the Widgets belonging to the groups. In order to exit the group again, select *Leave Group* or reverse the selection.

13.5.4 Properties - Widget

If you have selected a Widget, its properties can be adjusted in the tool window 1211 **Properties** (or via the context menu of the selected Widget, see "*context menus of the Widgets* 1237").

The content of the Properties window depends on the respective Widget selected. For info on the window structure, see "*Properties (general)* 1212".



Below, certain commonly occurring properties are listed.

Note Special properties

- The Widgets each possess only some of these properties.
- Special Widgets have additional properties not listed here.

General

Property	Description
Name	Name of the Widget. Each Widget possesses has a unique name on every page. The name may contain the following characters: letters, underlines, numbers.
Refresh rate	Updating of data. Placement of many Widgets burdens the PC's capacity. You may need to adjust the refresh rates of some Widgets if, for instance, the curve in the curve window shakes. The times are defined as follows: • very fast — 50ms — 20Hz • fast — 200ms — 5Hz • standard — 1s — 1Hz • slow — 10s — 0.1Hz • never — -- — -- Reading and outputting of the new values may be delayed in proportion to the utilization of the PC's resources. With the option "*Refresh rate of newly created Widgets* 131", you specify the default refresh rate of newly created Widgets.
Status	Switches the Widget active/passive. • Active: Widget works normally. • Passive: Widget can not be operated and its color is altered to indicate that it is passive. The current value remains displayed. • ...: depends on a variable 0: Passive 1: Active
Title	Displayed title of the Widget.
Color	Title font's color
Default Source	Source for the title • User-defined: Any text • Name: Name of the variable • Long name: Name of the variable followed by the measurement name. If the Widget is linked to a variable via a measurement number, the measurement name displayed always assimilates to the measurement selected. • Comment: Comment of the variable
Font	Here the name, size and font style can be changed.
Multilingual Title	The title can be pre-defined for multiple languages 1555.
Position	Position of the title (can change the position of the Widget.)
Unit	Displayed unit. If this property is empty, the variable's unit is displayed in the Widget. Otherwise, this unit is always displayed.

Property	Description
Variable	How to link a Widget with a variable is described in the chapter "*Variable linkage* 1550".
Visible in printout	The Widget is hidden in the printout. this also applies to export as PDF or graphic. • Show: Displayed in the export or printout • Hide: Omitted from the export or printout
Write protected	Protects against changed values. • On: The value cannot be changed by this Widget (display element) • Off: The value can be changed by this Widget (control and display element) • ...: depends on a variable 0: Off 1: On

Layout

Property	Description
Dock	Docks Widget at the specified edge. See "*Docking* 1244".
Operable	Protects against editing (supported by the curve window) • Yes: The curve window can be operated and configured. • No: The curve window can no longer be modified. All options are disabled, e.g. axis scaling, Zoom, ... • ...: depends on a variable 0: operable: no 1: operable: yes
Orientation	Displayed when the Widgets can be oriented either horizontally or vertically.
Position	Position and size of the Widget.
Visible	Shows/hides the Widget (not enabled in Design Mode). • Yes: shown • No: hidden • ...: depends on a variable 0: No 1: Yes

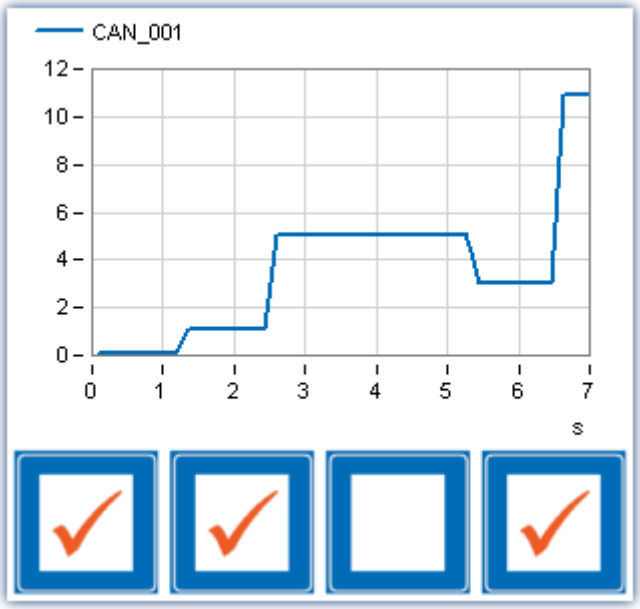
Appearance

Property	Description
Appearance	Different ways to display the Widgets. Usually only affects the appearance (2D/3D). An exception is the clock: here you can toggle between analog and digital clocks.
Background	Some Widgets can change their design. E.g. some Automotive-Widgets can adopt the design of Industrial-Widgets. Note: The Widgets will not look exactly the same and offer all the same settings as the corresponding Widget from the other design. Additionally, not all combinations are possible.
Border	This option lets you activate/deactivate the borders. Additionally, some Widgets can be displayed with sharp and some with round corners. (Only affects Automotive-Widgets)

Property	Description
Coloring	There are different color properties for the individual elements of the Widgets. The colors of most elements can be set separately. In order to apply the colors for other Widgets and for future ones, save the colors in a color scheme.
Font	Here you can adjust the font's name, size and style. This setting overwrites the title's font style.
Pictogram	The standard meter can be provided with a pictogram. This can be an icon for quick visual association. Similar to a speedometer or fuel gauge in a car.
Polygon needle	The design of the standard meter's needle can be modified. You can personally define the needle in terms of various x-y-coordinates. As an example, the coordinates of the standard needle can be loaded (form: "Standard").
Text Off/On	Texts for switch-states. There are advanced settings in the Zones-dialog 1252.
Value representation	If "Value representation" is activated, the value is displayed larger than the caption. (Input, output > Numeric ...)

Behavior

Property	Description
Autosize	Adapts the size of the Widget to the size of the current picture. (e.g. Graphical switch)

Property	Description
Bitmask	Displaying variables' individual bits Selected Widgets offer the option of only displaying individual bits in a variable. Example: A Fieldbus channel returns multiple channel states with: • 0th bit: Sensor connected • 1st bit: Value exceeded • 2nd bit: Error • ... With the property, it is possible to select which bit to display. If the 1st bit is selected, the Widget only shows the value of the 1st bit. Thus, with status indicators on the Panel page, it is easy to present an overview of the status of the various channels.  *A Fieldbus channel returns multiple channel states at one time. In the end, the bits 0, 1, and 3 are set for a total of = 11.*
Decimal places	Number of decimal places for display of the value. (See also "Scale decimal places") • **Automatic formatting:** As many decimal places are displayed as are necessary to make a total of up to 5 significant digits visible, including pre-decimal digits. • **1, 2, or n:** Only takes effect if "*Factor*" is set to a concrete value; otherwise the behavior always matches "*Automatic formatting*". Example of "*Automatic formatting*". For the value 0.001213141 V, the displayed value is as follows: • For factor: "*auto*" -> 1.2131 mV • For factor: "*micro*" -> 1213.1 µV
Edit box color	Graphical display of the zones in text boxes. (Input, output > Numeric ...) The text background can be made to depend on the zones' properties. When you select "Zone color", the corresponding zone color is displayed as the background. When you select "Standard color", the background color set for text is displayed.
Events	Certain Widgets can be linked with commands 1263 which are carried out at state transitions (called events).

Property	Description
Factor	Factor with which the value is to appear. The available selection options depend on the Widget and on the unit. When a unit is specified, you can select "*milli*", "*kilo*", "*mega*", ... for example. If no unit is specified, you can select "*1e-3*", "*1e3*", "*1e6*", ... Additionally, the following selection options are available: • **1:** Output of the variable's unit • **Automatic formatting:** Depending on the value, an appropriate factor is selected. For "*0.01 V*" for example, the factor "*milli*" is used, and "*10 mV*" is displayed. Example: The variable's value is "*0.1*" and the unit "*V*". If you select "*milli*" as the factor, then "*100 mV*" is displayed. If you select "*1*" as the factor, no conversion calculation is performed; the displayed output is "*0.1 V*". **Unit conversion calculations** The following procedure is applied by the Widgets "*Table*" and "*Numeric input*" belonging to the groups: Automotive, Industrial and Designer. 1. Certain units containing a factor, e.g. "*kg*" are converted in calculations. Other units which by chance have the same initial letter as a factor are not processed in this way; for instance "*Gallon*", where the "*G*" is not interpreted as "*giga*". 2. Using the property "*Factor*", you can specify the target factor, even if the variable's unit is "*kg*", for instance. Correct conversion to grams or tons is performed. By means of the factor "*1*", the exact unit entered is outputted. The same applies to many other units. Example: The unit is set to "*kV*", the value to "*0.005*". If the factor is set to "*1*" or to "*kilo*", the Widget displays "*0.005 kV*". If the factor is set to "*milli*", then "*5000 mV*" is displayed. For "*Automatic Formatting*", the displayed output is "*5 V*" (the factor selected by the system then depends on the magnitude of the value).
Increment	Increment size for Widgets with buttons for up and down.

Property	Description
Range	The display range, or the range of valid entries; e.g., for a standard meter, the min- and max-values for the scale. **From Variable** By default, the range is the measurement range if the variable has its own range (e.g. analog channels). The channels can also tranfer other Min- and Max-values. To do this, change the settings on the following page: Setup page "*Analog channels*"; Dialog: "*Curve properties* 372". All other variables automatically use the values defined, even if "*From Variable*" is selected. Special treatment of the scales of the "Potentiometer" and "Standard meter": The Widget adjusts the edge-ticks in such a way as to display "optimum" numbers. Example: For a range of "-97...103", the scales are rounded up to "-100...120". **Defined values**: Maximum and Minimum No values outside of the range specified can be entered. Scale-Widgets modify the scale according to these values. Any channel presets ("*Curve properties*") are ignored. **Infinite** If the range is to be indefinite, you can enter "*Infinite*" or "*-Infinite*" (not possible with scale-Widgets).
Maximum	Maximum value of range
Minimum	Minimum value of range
Scale center	For some bar charts and Widgets with level indicator, the midpoint of the scale can be modified. The display of the bar is based on this value. As the midpoint, any value can be specified or the fixed minimum or maximum value.
Switching behavior	Behavior of the switches/buttons upon clicking the mouse. Some of the Widgets are defined by default as switches, others as pushbuttons. The behavior can be changed by this setting. • Switch: Mouse click -> Widget pressed (on); Mouse click -> Widget comes back (off) • Push-button: mouse button pressed -> Widget pressed (on); Mouse button released -> Widget comes back (off)
Text format	Here you can format the value display. Some Widgets also support placeholders 166. In cases of parallel use with zone-texts, see the notes regarding: "*Zone dialog* 1262"
Zones	Displays existing zones. See "*Zone dialog* 1252" See also the display settings: "Zone representation" and "Edit box color"
Zone representation	Graphical display of the zones. Some Widgets offer various display types for the zones, e.g. standard meter, or potentiometer.

Scale

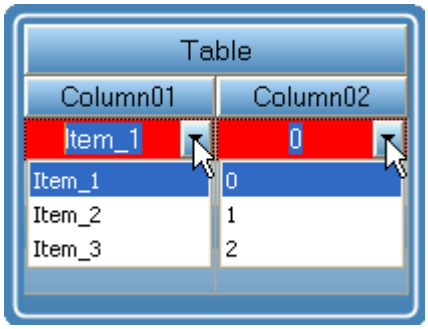
Property	Description
Interval	Smallest step between two values. The values in between are not displayed. E.g. standard meter with an "interval" of 0.5: The pointer always jumps at interval steps: 0; 0.5; 1; 1.5; ...

Property	Description
Scale angle	Opening and starting angle for the scale display. • Start angle: 0° -> Horizontal rightwards • Spread angle: clockwise
Scale decimal places	Number of decimal places for the scale display. The count of decimal places is automatically adjusted when the range is specified by the variable.
Scale position	• X; Y: Positioning of the scale midpoint to the Widget-midpoint • Radius: distance of the scale from the scale-midpoint
Tick distance	Distance between main ticks. The value is automatically aligned to the "range". I.e. when the range is changed, the changed value is discarded.

13.5.4.1 Zone dialog

In this dialog, you can **define which values** are available for selection in a "*selection box*", for example. You can also specify **colored display** for the **value ranges** of various Widgets:

Example of a standard meter

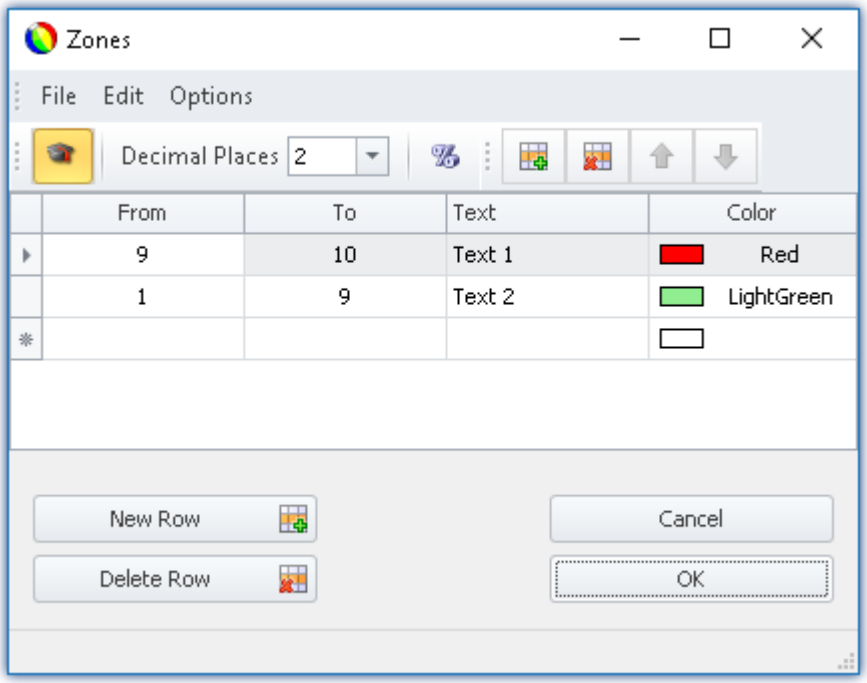


Example of a table with list boxes

To open the dialog, proceed as follows:

- Open the properties of the Widget.
- Click on the button ... in the box "*Zones*"

The Zones-dialog is called, which appears as shown below:



Setting zones for a Widget (example: standard meter)

Note — The dialog sometimes appears different

The dialog's structure and the displayed columns 1255 depend on:

- the Widget (control elements require different settings than display elements)
- the Widget property "*Write protected* 1245" (preset values are not required)
- the option "*Advanced display* 1259" (hidden and automatically calculated columns are displayed)

Here, the value ranges of the Widget can be distinguished by color. In the example, the "*standard meter's*" value range is subdivided as follows:

< 1	default (not defined) e.g. white
1 - 9	green
9 - 10	red
> 10	default (not defined) e.g. white

The image below shows the "*standard meter*" set accordingly:

Example: standard meter

Changing/creating zones

Changing values, texts and colors

To change an entry,

- click in the corresponding box
- perform the change desired

By default, the values are to be entered as absolute quantities. For some Widgets, the values can also be entered as a percentage 1259 of the range.

Adding or deleting a zone

In order to create or delete a zone, use the items: "*New Row*" / "*Delete Row*" (e.g. via the context menu or the menu items).

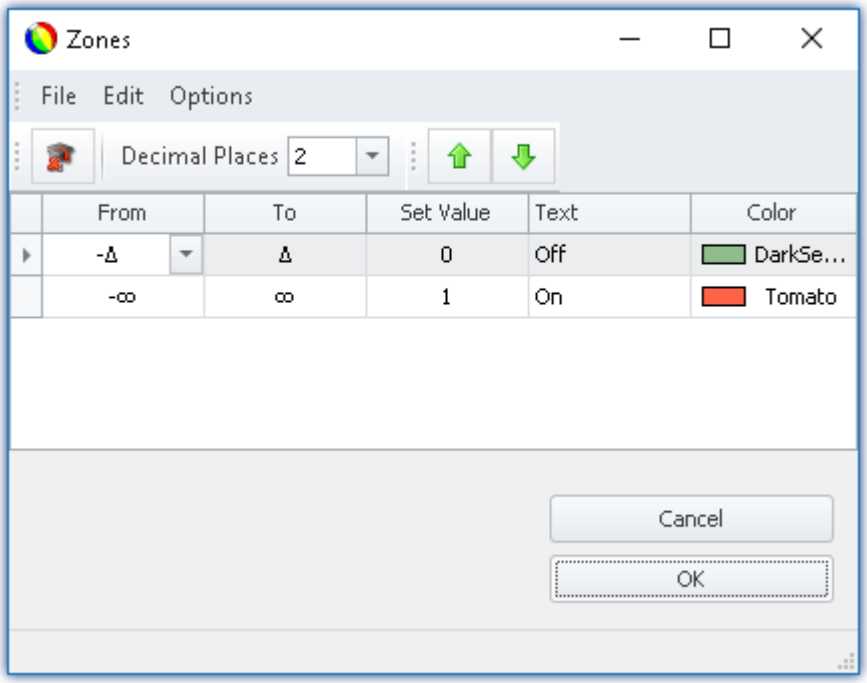
Moving the zones

The top zone has top priority 1257.

To change the order, use the items: "*Move Row Up*" / "*Move Row Down*" (e.g. via the context menu or the menu items).

Infinity and zero

In special cases, symbols are used for *Zero* and *Infinity*. Below, the zones for a **switch** are shown with "*Advanced display* 1259".



Zones of a switch with "Advanced display"

The switch should be "*Off*" when near "*Zero*" and for all other values "*On*".

13.5.4.1.1 The columns and their interdependencies

Some columns are hidden

The structure of the dialogs depends on the Widget:

- If **values are set** with Widget, the column "*Set Value*" is displayed (e.g. "*Listbox*" and "*Switch*")
- If the Widget usually **displays values**, the columns "*From*" and "*To*" are displayed (e.g. "*standard meter*")

The dialog's structure depends on the property "*Write protected*":

- If the Widget is **write-protected**, then **no value can be set** with this Widget. In this case, even for a *Switch* only the columns "*From*" and "*To*" are displayed.

If the "*Advanced display* 1259" is activated, the columns "*Set Value*", "*From*" and "*To*" are displayed if present.

Relationship between set values and value range

The values in the columns are mutually interrelated. The values entered in the column "*Set Value*" should normally lie within the range between "*From*" and "*To*".

If only the column "*Set Value*" is displayed and you change the value, the values in the columns "*From*" and "*To*" change accordingly.

Note Automatic adaptation

- If you have **activated "*Advanced display* 1259", this automatic adaptation is deactivated**. This means you can also set values outside of the range. If you close and then re-open the dialog, the "*Advanced display* 1259" is deactivated again. Each change to the values re-adjusts all the values again!
- Make note that because of the above fact, the Widget may get into an inoperable state.

Example — Example of automated adjustment of the ranges: "*Listbox*"

The Widget has four zones:

Set Value	Text	From (not displayed)	To (not displayed)
0	Eintrag_1	-0,5	0,5
1	Eintrag_2	0,5	1,5
2	Eintrag_3	1,5	2,5
3	Eintrag_4	2,5	3,5

If you change the value sin the column **Set Value** and have not activated the **Advanced display**, the other values adjust themselves.

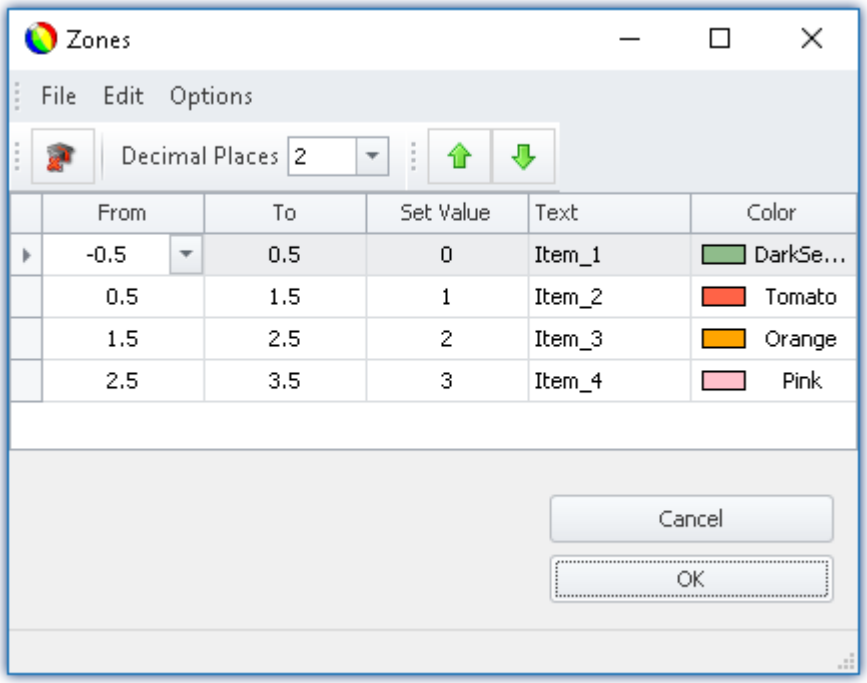
Set Value	Text	From (not displayed)	To (not displayed)
0 > 1	Entry_1	-0.5 > -∞	0.5 > 1.5
1 > 2	Entry_2	0.5 > 1.5	1.5 > 2.5
2 > 3	Entry_3	1.5 > 2.5	2.5 > 3.5
3 > 4	Entry_4	2.5 > 3.5	3.5 > ∞

Additional example of automatically calculated values:

Set Value	Text	From (not displayed)	To (not displayed)
1	Entry_1	-∞	1.5
2	Entry_2	1.5	3.5
5	Entry_3	3.5	12.5
20	Entry_4	12.5	∞

13.5.4.1.2 Order and priority / overlapping zones

In order that no values be omitted, the zones mostly **overlap**. The **top zone has higher priority**. Below an example:



Zone dialog with "Advanced display"

The first zone has the range -0.5 through 0.5. The second has 0.5 through 1.5. The value 0.5 appears in both zones. However the top zone has higher priority.

If the value is exactly 0.5, the zone is displayed with the text "*Item_1*".

Example Example of an LED with overlapping zones

We intend for the LED to shine "*red*" for a value of 5. And "*green*" for all other values:

This amounts to three zones:

1: -infinity to 5
2: 5 to 5
3: 5 to infinity

Ensure that Zone 2 (5 to 5) is not covered over by the others. For that reason, it must be placed on top:

1: 5 to 5
2: -infinity to 5
3: 5 to infinity

It is then also possible to define just two zones:

1: 5 to 5
2: -infinity to infinity

If the value is "5", it lies in the first zone. If it is not "5", it lies in the second zone.

Example — Example of a pointer instrument with overlapping zones

The ideal value lies between 4 and 6 ("*green*").

One value beyond that range in either direction is still OK ("*yellow*").

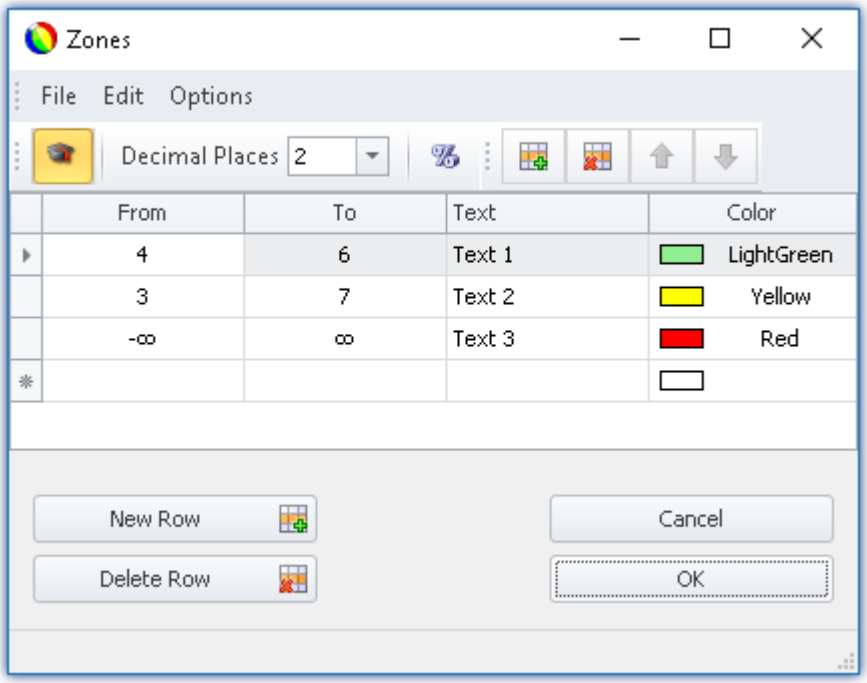
Anything else is either too high or too low ("*red*").

This amounts to five zones:

1: 4 to 6
2: 3 to 4
3: 6 to 7
4: -infinity to 3
5: 7 to infinity

One can make these into three zones:

1: 4 to 5 ("*green*")
2: 3 to 7 ("*yellow*")
2: -infinity to infinity ("*red*")



13.5.4.1.3 Other options and settings

Advanced display

By default the Zones dialog has a **simple display**. The structure is adapted to the normal operating style for the Widget. Accordingly, not all columns are always displayed.

The "*Advanced display*" is required for some complex applications. When the display is activated, all columns are displayed.

Application examples:

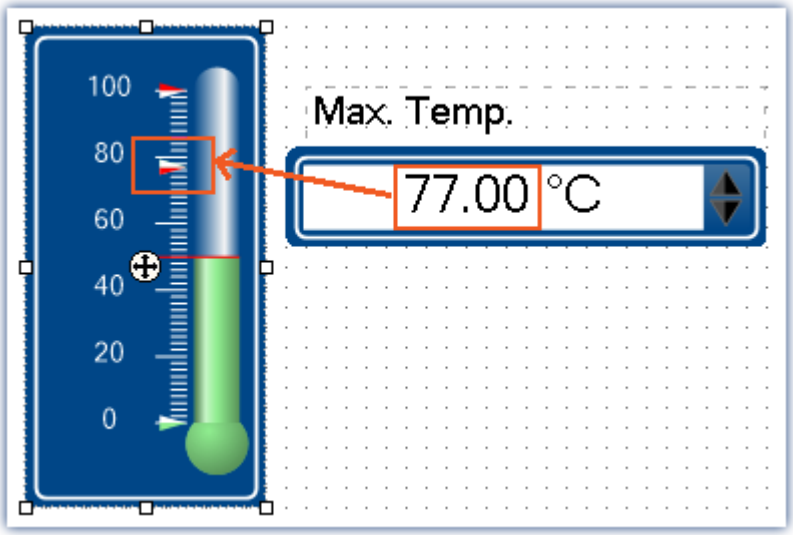
- The "*Listbox*", which is to be used as an input/output element (ranges and set values need to be defined).
- Switches or LEDs having more than two zones (e.g. multi-colored lighted pushbutton)
- Ranges and set values can be defined by variables (e.g. value limits which depend on certain settings)

Activating Advanced Display

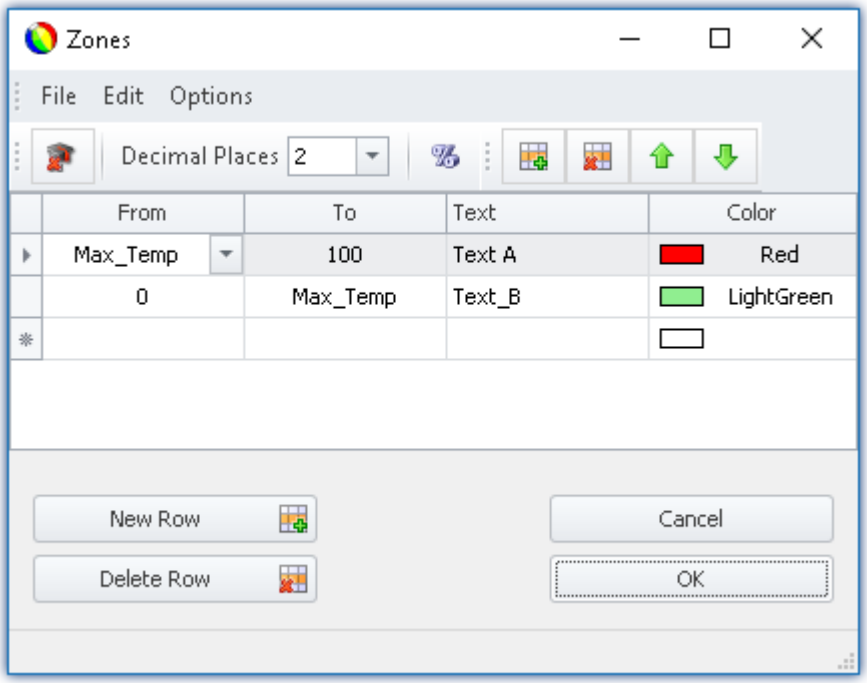
In order to activate "*Advanced display*", use the entry: "Extended display" (e.g. via the menu entry).

Example Variable value limits in a Widget

When a maximum temperature level is exceeded, the thermometer is to turn "red".



Example: Thermometer with user-defined ranges



Zone-settings
The variable: "Max_Temp" is used as the range boundary

Percentage

By default, the values are entered in absolute terms. For some Widgets, you can also enter the values as a percentage of the range.

If at a later time the Widget's display range changes, the zones adjust proportionally. However this means that fixed midpoint values can not be specified (all values adapt to the range).

Activating Percentage Entry

In order to activate "*Percentage entry*", use the entry: "*Percentage*" % (e.g. via the menu entry).

Example

Display range of the Widget: 0-10

Zone: 0-5, 5-7, 7-10 (0%-50%, 50%-70%, 70%-100%)

The range changes to: 0-50

New zones: 0-25, 25-35, 35-50

Numeric Zone / Textual Zone

Some Widgets can **display texts or numerical values** (for instance "*Tables*" belonging to the group Automotive, Industrial and Designer).

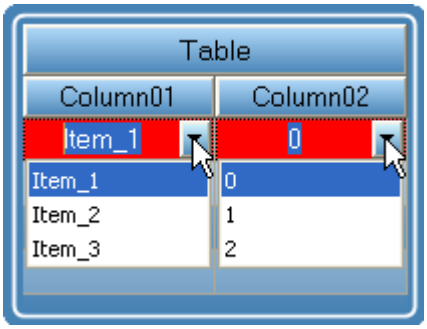


Table with numerical and textual zones

The variable is assigned the same value in both cases. However, it is often necessary to have the user select a text while internally writing a number.

Switching between numeric and textual display

In order to switch between "*Numeric*" and "*Textual Zones*" display modes, use the entry: "*Numerical zones*" 123 / "*Textual zones*" ABC (e.g. via the menu entry).

Note Zone-texts are ignored when ...

- Zone texts are ignored when the **property "*Text*"** is used. The property "*Text*" defines how the value is to be displayed. If you wish to define **a separate display for each zone**, delete the content of the property "Text" and specify the complete representation of the values by means of the zone texts.
- Zone texts are ignored when a **text-variable** is used. Any desired texts can be displayed by means of the text-variable. In order to avoid any duplicates, the zone texts are ignored.

Text as set value

Some Widgets can **set texts** (e.g. "*Tables*" of the groups Automotive, Industrial and Designer). By contrast, most Widgets can only set numbers.

With this setting, all that remains is the set value , and when "*Textual Zone*" is activated, the display value also. It is no longer possible to enter a range. The set value is written to the variable as text, for which purpose a text-variable is needed.

Activating the writing of texts

In order to activate "*Text as set value*", use the entry: "*Text as set value*" AB (e.g. via the menu entry).

Decimal places

Here you set how many decimal places are to be shown in the dialog. The values can also be expressed in exponential notation. For instance, select "*E3*".

The following values can be entered: 0, 1, 2, ..., E0, E1, E2, ...

This setting only refers to the display of numbers in this dialog and has no effect on the Widget.

Setting the amount of decimal places

To change the number of decimal places, select the menu item: "*Decimal places*" (e.g. via the context menu or the menu).

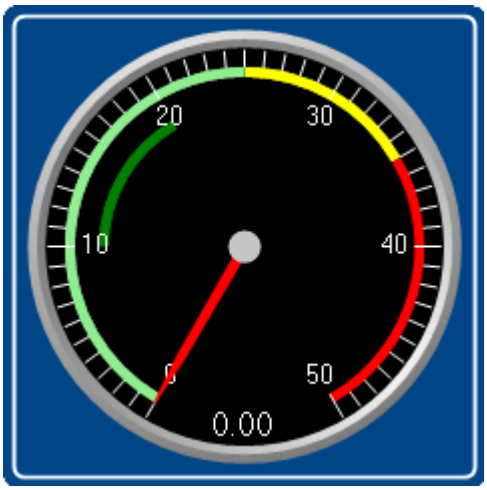
Zone ring

The pointer instrument can have multiple "*Zone rings*".

To **show** the zone ring, the Widget-property: "Zone display" is used. By default, the ring is deactivated.

By means of the menu item "*Edit*" > "Create *Zone Ring*" **you can create an additional zone ring**. If zone display had previously been deactivated, it is activated automatically. Now, two rings are shown (if the second zone ring is supplied with zones).

In the "*Edit*" menu, you can **switch between the zone rings** or **delete a zone ring**.



Two zone rings are displayed. The dark green area is the second ring.

13.5.4.2 Linking commands with switches

Certain Widgets can be linked with ***commands*** which are carried out at state transitions (called ***events***).

As an example, the two events ( ) belonging to a ***Switch*** from the group ***Standard***:

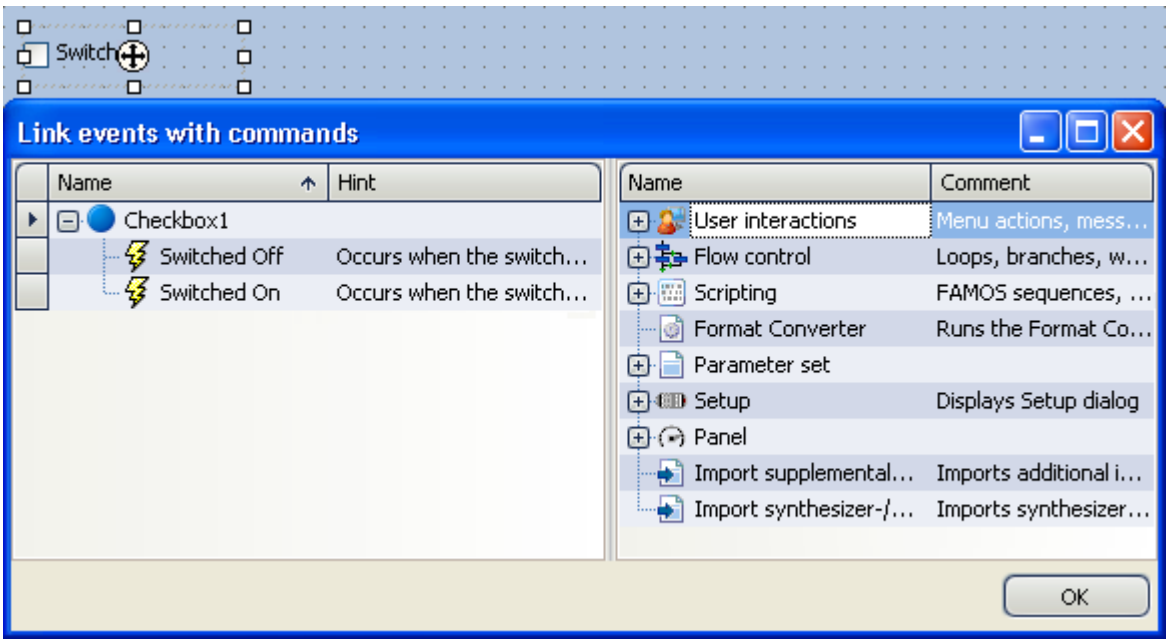
- ***Switched off*** (from On to Off)
- ***Switched on*** (from Off to On)

The Widgets include ***Pushbutton*** and ***Switch*** from the group ***Standard*** and various switches/buttons from other Widgets (***Automotive, Industrial, Designer*** if these are installed).

To assign a **command**:

- Select the Widget
- Open the Properties 1245 of the Widget

- In the Properties window, click on "Events ..." at the lower left. Subsequently, the dialog: **Event Dialog** opens, as seen in the image below:



Event dialog:
Events belonging to a switch from the group Standard

Reference Commandos and Sequences

- The workings of the individual commands are described in the chapter **Command Reference**.
- A **sequence of commands** can be created in various imc STUDIO plug-ins and is described separately (see the chapter "*Sequencer, events and commands* 1681").

Example

Below an example with multiple commands

Name	Hint
Checkbox1	
Switched Off	Occurs when the signal is switched off
#01 Execute menu action: Stop	Execute menu action: acDeviceStop
#02 Execute FAMOS sequence	
#03 Browse in workspace 'Report'	
Switched On	Occurs when the signal is switched on
#01 Execute menu action: Start	Execute menu action: acDeviceStart
#02 Browse in workspace 'Measurement'	

After switch on:

- the measurement starts
- the Panel page *Measurement* is opened

After switch off:

- the measurement stops
- a imc FAMOS analysis is performed (if installed; see *Technical data sheet > Additional imc Software Products*)
- the Panel page *Report* opens

13.6 Curve window

Curve Window shows the graphic representation of measurement signals and computed waveforms, in the simplest case as a labeled coordinate system and a curve. The curve window displays individual and independent objects, which can be positioned, enlarged and edited as desired.

The curve window has an extensive menu offering a multitude of options for display and graphical evaluation of waveforms. All curve windows always display the current values.

The following offers a short overview of the main features of the curve window:

Function	Description
scaling	**Automatic or manual scaling** of the x- and y-axes
Zoom	allows detailed viewing of a curve section
Unzoom	returns to entire curve window display
Measurement values	Two independent measurement cursors are available, which display the x- and y-values of the curves, as well as value differences and slopes. These cursors can be used to select sections of a curve for special processing. Measurement values can be transferred to a report.
Markers	markers can be set to display comments or indicate the coordinates of individual points of a curve in the curve window.
Overview Window	Shows the position within the entire curve of a zoomed curve section.
More Curves	Displays additional curves in the curve window to allow comparisons.
Print	Prints out the curve window's content with the resolution of the designated printer.
Clipboard	Curve window graphs can be copied to the Windows Clipboard. These can be pasted to word processing and DTP applications and combined with text and additional graphics. Several options are available for copying curve graphics to the clipboard or the printer, including font type, line width and dimensions.
Export	The waveform displayed in the curve window can be exported to other imc applications. This is a fast and simple method of data transfer analogous to the Windows clipboard.
Transfer to imc FAMOS	Waveforms can be transferred directly from the curve window to imc FAMOS. A sequence in imc FAMOS can be automatically initiated with a transfer. Since the curve window is used by all imc products, it would be possible, for example, to send data from imc STUDIO to imc FAMOS in this way, and to process them there directly by means of a sequence.
Logarithmic, dB and linear axes	
XY, polar and locus plots	
Colors	The colors for graphics can be set as desired
Grid	The coordinate system can be overlaid with a grid.
Configurations	Curve window configurations can be saved in a file and loaded again.
1/3-octave or octave	The x-axis of a curve window can be set to normal scaling for a **1/3-octave** or **octave** analysis.
overlapping, stacked or waterfall diagram	Several curves can be displayed together for comparison: **overlapping, stacked** or in a **waterfall diagram**. Various options are available for **waterfall display**, e.g. the **angle of view**.

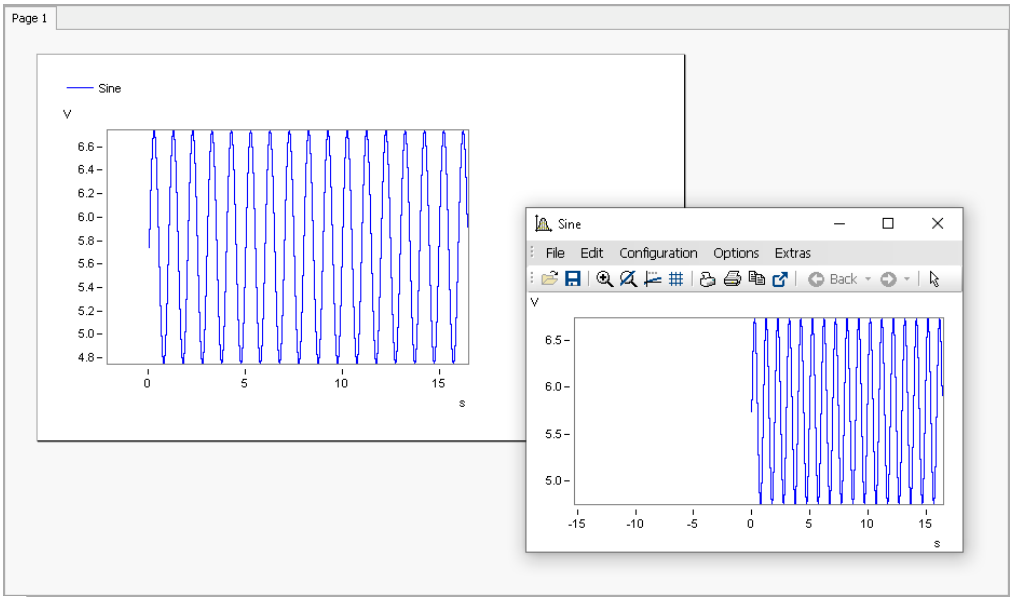
Function	Description
symbols	Curves can be **marked with symbols** to help differentiate between curves in a monochrome display.
periodic waveforms	An option is available for **periodic** waveforms in which all periods are compared with each other. This helps to identify trends and aberrations.
Color maps	A "bird's-eye view" of a colored 3D plot; color-coding indicates height.

Curve Window - Embedded or "Free-floating"

The curve window comes in two varieties:

- "free-floating 1517" and
- embedded on a Panel page

The "free-floating" curve windows are associated with the experiment. They remain open until closed (X) and are displayed even when the Panel is not open.



Example: curve window - embedded and "free-floating"

The embedded curve windows are only visible if the associated Panel page is opened..

13.6.1 Display variants of the curve window

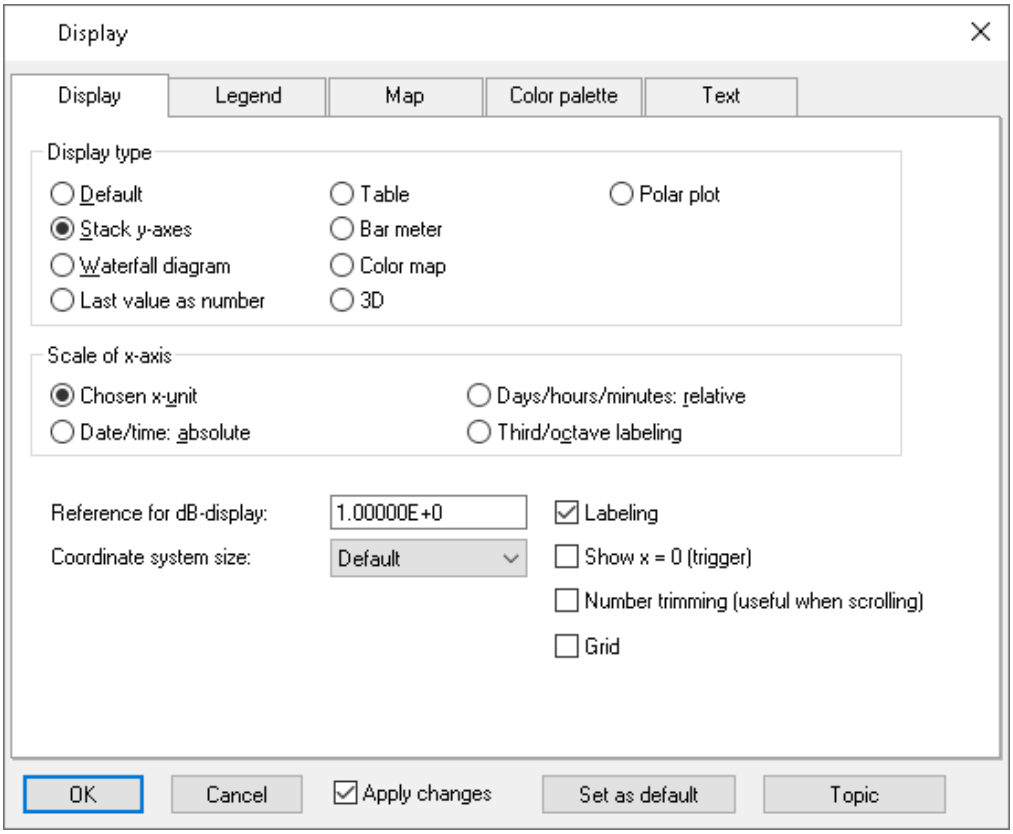
Function

The display mode of a curve window can be defined in a variety of ways. The display options include the appearance of the coordinate system, selection of date/time labeling or 1/3-octave labeling. Several special attributes are also available, including number of secondary ticks, period-comparison, marking of lines with symbols and definition of the reference value for dB displays.

Calling the dialog

Select menu option "*Configuration*" > "*Display*" in the curve window. A dialog appears in which the curve window display can be defined.

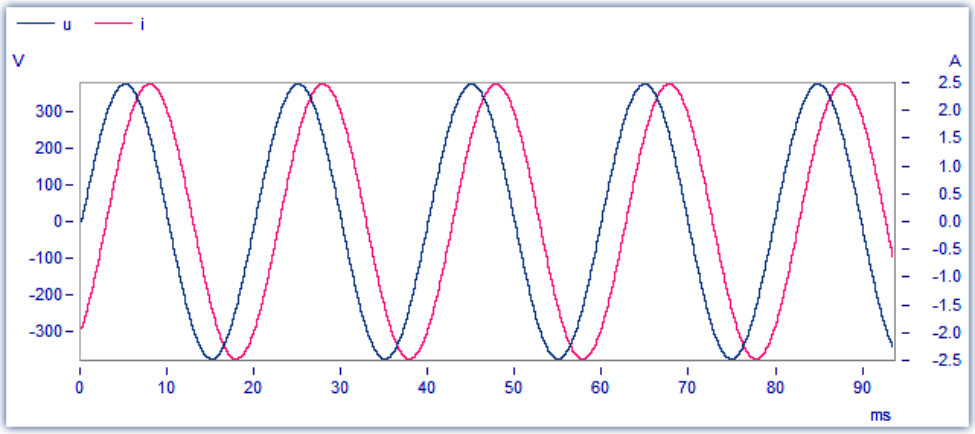
Tap	Description
Display 1267	Display variants of the curve window
Legends 1382	All curve windows can include a legend, with the channel (variable) name, line color etc.
Map 1385	Map lets you place a picture in the background. It is additionally possible to layout the picture in specified coordinates.
Color palette 1398	A channel's color can be governed by a reference signal's amplitude.



Display type

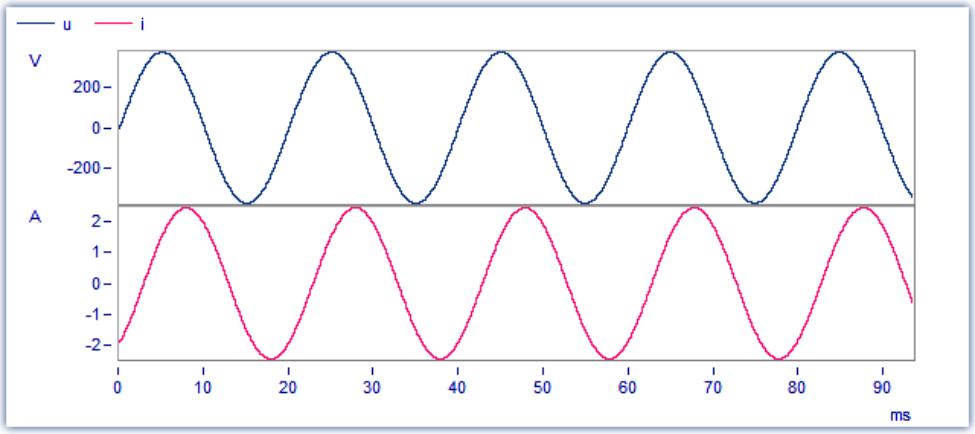
Default

In *Default* display, curve windows with more than one y-axis will show the different axes side-by-side. Several curves are then displayed in the same space.



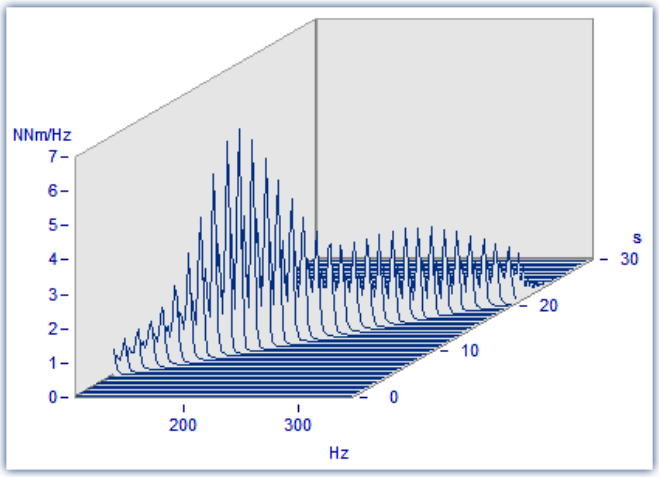
Y-axes stacked

An alternative to the *Default* setting is the *y-axes stacked*. Here, the y-axes are displayed stacked vertically, so that each curve is displayed in its own space, separated from the other curves in the window.



Waterfall Diagram

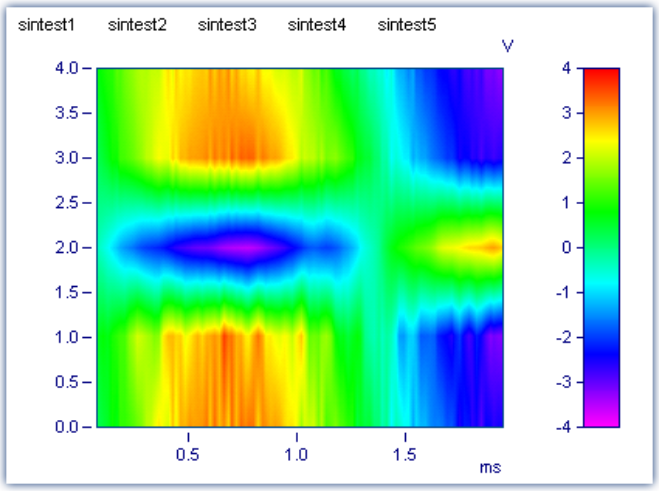
Another alternative is the waterfall diagram, which is especially suited to comparing similar curves. The curves are drawn in a diagonal procession from the front of the screen, with the curves at the back (partly) obscured by those at the front, creating the illusion of a three-dimensional perspective.



See "Waterfall Diagrams 1277"

Color Map

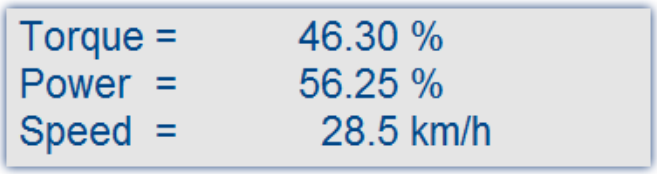
This is a bird's-eye-view of a colored, three-dimensional region. Multiple data sets are each plotted along the horizontal direction, stacked over each other along the y-direction, while their amplitude values are represented by different colors. This style of display is analogous to a geographical map where the landscape contours are indicated by shading or coloring.



See "Color Map Display 1303".

Last value as number

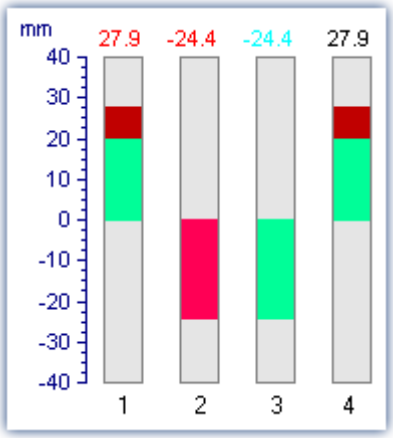
Particularly for online display of measured data, in this mode the last value is displayed as a number. The last value in a waveform is always the most recent measurement value in online display. This mode of display is particularly suited for slowly changing quantities. For example, the amplitude of a spectral line, effective power, or temperature can be monitored as a numerical value. The display of numerical values can be formatted. The corresponding numerical value is displayed for each curve in the curve window.



See "Last Value As Number 1283".

Bar meter

Particularly for online display of measured data, in this mode the last value is displayed in a bar meter.



Section 'Bar gauge' 1294

Table

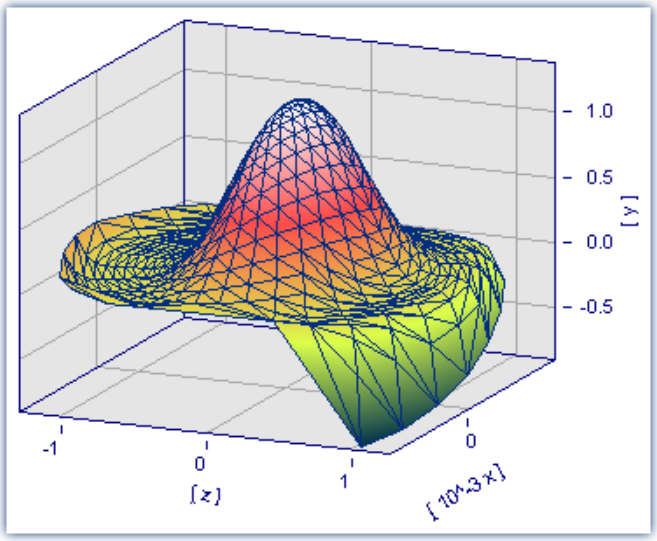
The readings are displayed in tabular form, in chronological order.

t [s]	sintest1 [V]
0.00000	-0.2544
0.00002	0.0349
0.00004	0.0259
0.00006	0.1652
0.00008	0.3037
0.00010	0.4410
0.00012	0.5766
0.00014	0.8592
0.00016	0.6923
0.00018	1.1185
0.00020	1.0945
0.00022	1.2168
0.00024	1.3388
0.00026	1.4670

Section "Tabular display 1288"

3D

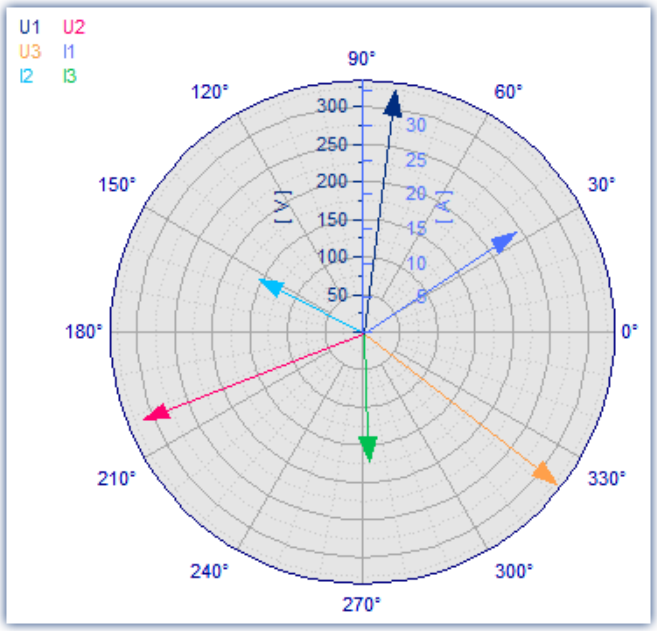
3-D display of waveforms having x,y,and z coordinates with various color map surface patterns, and rotating perspective in all dimensions.



Section on 3D display 1315

Polar plot

Complex data sets can be displayed as a polar diagram.



Complex single values as vector diagram

Section on "Polar plot" 1315

Scale of x-axis

The following settings for scaling the x-axis can also be made using the x-axis' own settings dialog; see the section "Axes 1350".

Selected x-unit

The x-axis is labeled in the x-unit of the waveform, e.g. in "s" (seconds) for measurement over time, or in "A" (amperes) for a characteristic dependent on the current. If several curves are displayed in the window, all waveforms should have the same x-unit.

For example, if a normal waveform with x-offset = 10s, x-delta = 1s and 20 samples is displayed, the x-axis is displayed from 10s to 30s and labeled accordingly.

Date/time: Absolute

For data in the time domain, instead of labeling of the value expressed in seconds, a display of the absolute date and time can be selected. The absolute time is not determined exclusively from the measurement points, but also gives consideration to the generation time of each waveform. This time is recorded in files in the imc FAMOS format and can be inquired and changed with imc FAMOS functions.

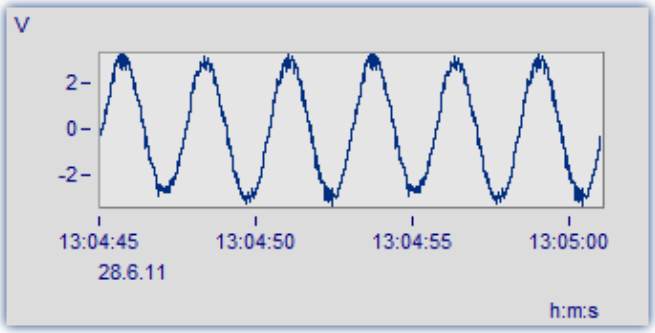
The generation time of a waveform along with the x-coordinate in "s" determines the resulting axis labels.

For example, if the generation time of a waveform is 1.1.92 at 12:00:00, and the waveform has an x-offset of 3600s and a sampling rate of 1800s for 48 measurement values, the range from

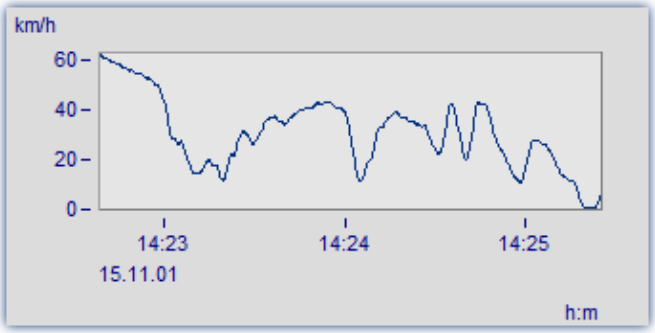
1.1.92, 13:00:00 to 2.1.92, 13:00:00

is displayed (3600s equals one hour). The waveform extends over 48 measurement values, each with a half-hour time difference, resulting in exactly one day of samples.

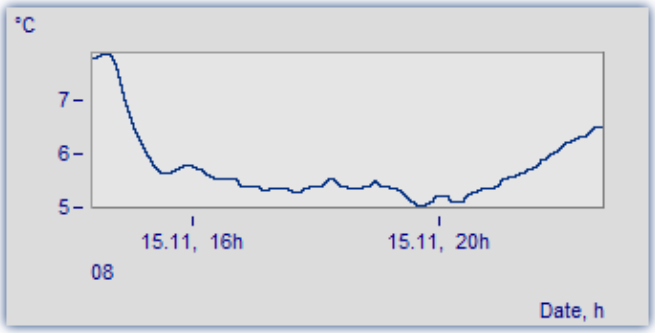
The labeling of the x-axis depends on the length of the displayed time interval, for example:



The segment between 13:04:45 and 13:05:00 encompasses several seconds.



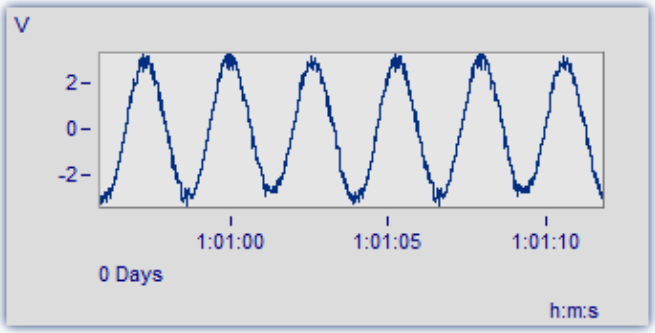
The segment between 14:23 and 14:25 comprises a few minutes.



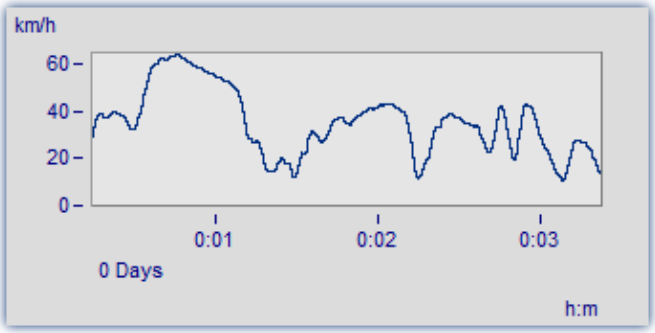
The segment between 4 am and 8 pm encompasses a few hours.

Days/hours/minutes: Relative

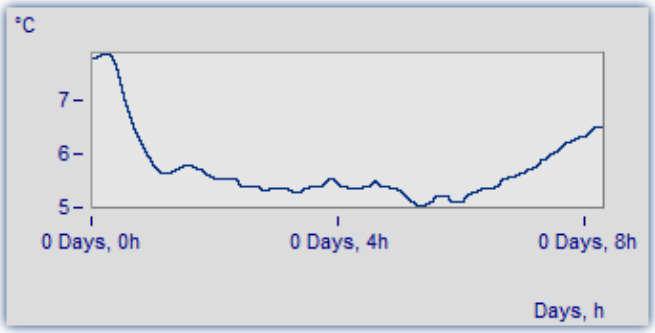
If you wish to view the whole duration of a long-term measurement, then display of relative time in days, hours, minutes and seconds is recommended. The selected unit depends on the interval. Possible variations of this labeling are as follows:



A very narrow section is shown, significantly less than one minute



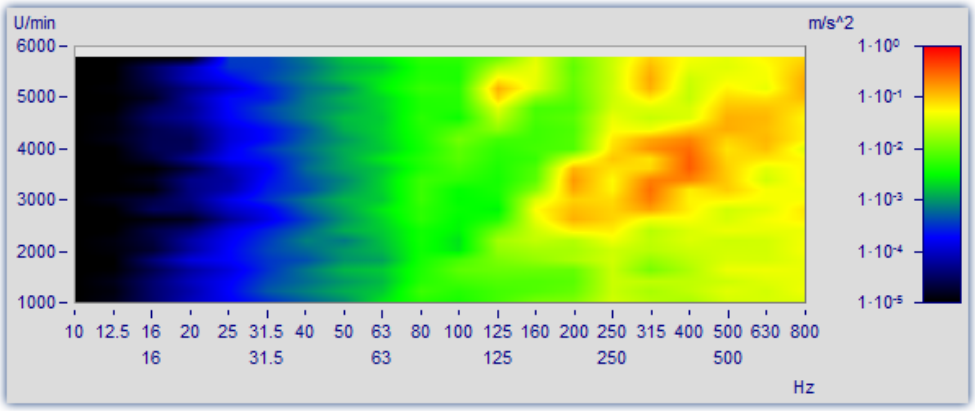
This section is 3 minutes long. The display no longer includes decimal places after the seconds.



The section from 0 to 8 hours is displayed without minutes

1/3-octave, octave labeling

When a waveform contains a 1/3-octave or an octave spectrum, and the x-axis of the waveform is scaled in 1/3-octave, the x-axis can be drawn with the numerical values of the 1/3-octave and octaves according to industry standards.



See "Third-Octave Display 1321".

Further settings

Settings	Description
Axis labeling	Axis labels can be switched off completely, allowing the curve to be displayed in its maximal expanse. When curve windows are reduced significantly in size, the display automatically switches off display of the axis labels.
Show x=0 (trigger)	When this option is selected, a dotted vertical line is displayed in the curve window at x=0. The trigger is very often located at x=0.
Number trimming (useful when scrolling)	When this option is selected, the labeling at the ends of the axes are displayed partially, since the labeling box is too small to display it completely. When scrolling, the numbers "flow" into view. This option may also remain inactive if it is not needed.
Grid	Grid 1404 as default setting.
Reference for dB value	All dB displays in a curve window have the same reference value. This value is "1" as a standard, but can be changed. Each number greater than zero in the valid range can be a reference value. For example, if a value of 10 is specified, all dB values on the axes become 20dB smaller. When decibels are calculated, values are first divided by this reference value. The displayed waveforms are not changed; only the axes labeling is adjusted.
Coordinate system size	Along with the *Default*-view, it is also possible to select a *Maximum*-view in which the axis labels are placed within the coordinate system and the coordinate system fills the entire curve window.

OK, Cancel, Set as default

For details on operation, see the chapter Confirmation bar 1423

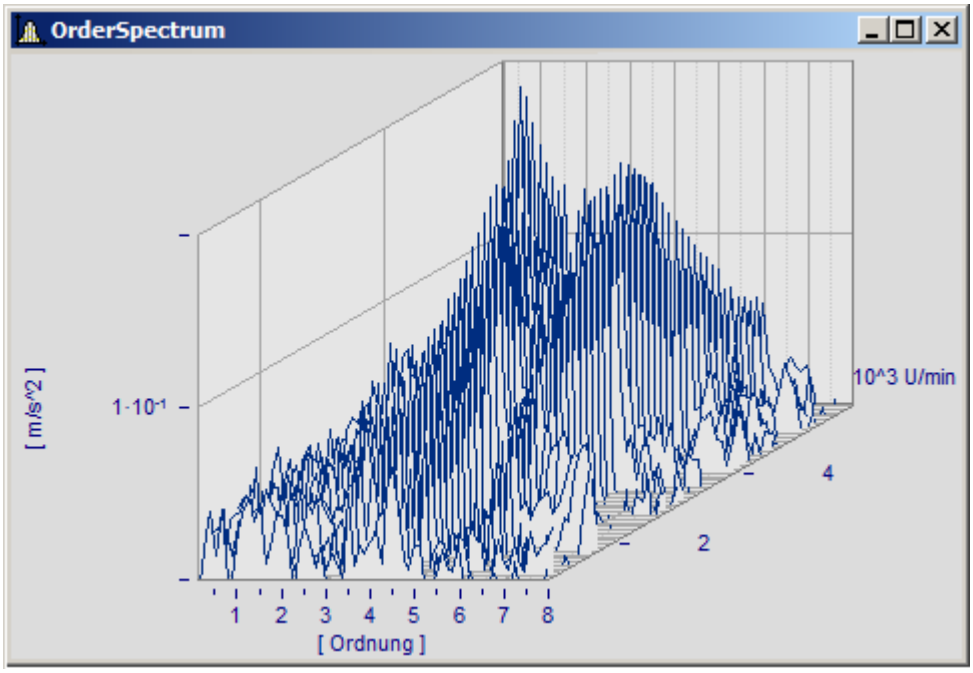
Remarks

Note

- If a waveform is not scaled in seconds, but is to be displayed in absolute or relative time, adjust the sampling time, the x-offset and the x-unit to provide the desired unit "s". For instance, if the waveform has a sampling time of 1h (1 hour), divide the sampling time by 3600 to set the x-unit to "s".
- The modes *Date/time: Absolute* and *Days/hours/minutes: Relative* cannot be implemented if an interval of less than 1ms is set for display of absolute or relative time. The program switches automatically to axis labeling in seconds.

13.6.1.1 Waterfall diagrams

The waterfall diagram serves as a simple tool for displaying a series of curves as arrayed one behind the other, and viewed from a vantage point that makes their line-up appear diagonal. This allows a close comparison of the individual curves, so that trends and deviations are easily recognized. Curves are drawn in a rectangular coordinate system with three axes (x, y, z). The z-axis has a directional component parallel to the angle of view. It is drawn diagonally toward the upper right of the screen at an angle specified by the user.



Not all curves are completely visible in waterfall display. The viewer sees a range of curves from above, with some valleys hidden by peaks. The relative angles and distances between the curves are not always apparent due to overlapping.

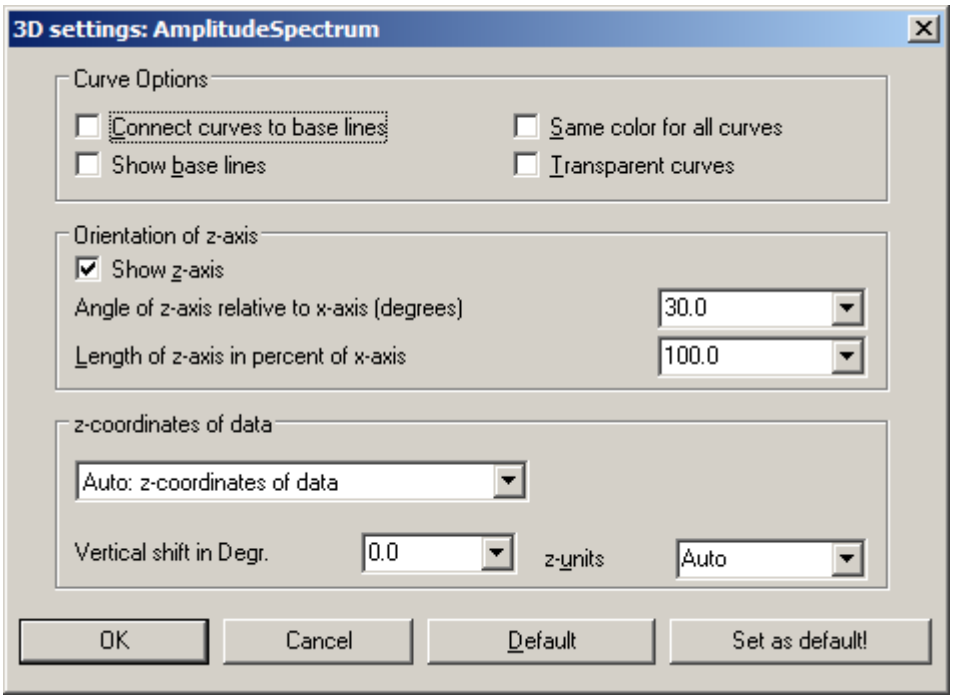
A waterfall display is drawn from back to front. When a new curve is drawn, the area under the curve is filled in with the color of the screen background, covering all lower parts of curves behind it.

The waterfall display is especially useful for comparing:

- multiple spectra; spectra are drawn at regular intervals, so that their development over time can be observed.
- multiple periods in a waveform.
- signals of several sensors on the same object, e.g. multiple temperature recorders arranged on a long pipe, delivering similar signals at different points in time.
- several consecutive measurements of a channel measured with the same trigger.

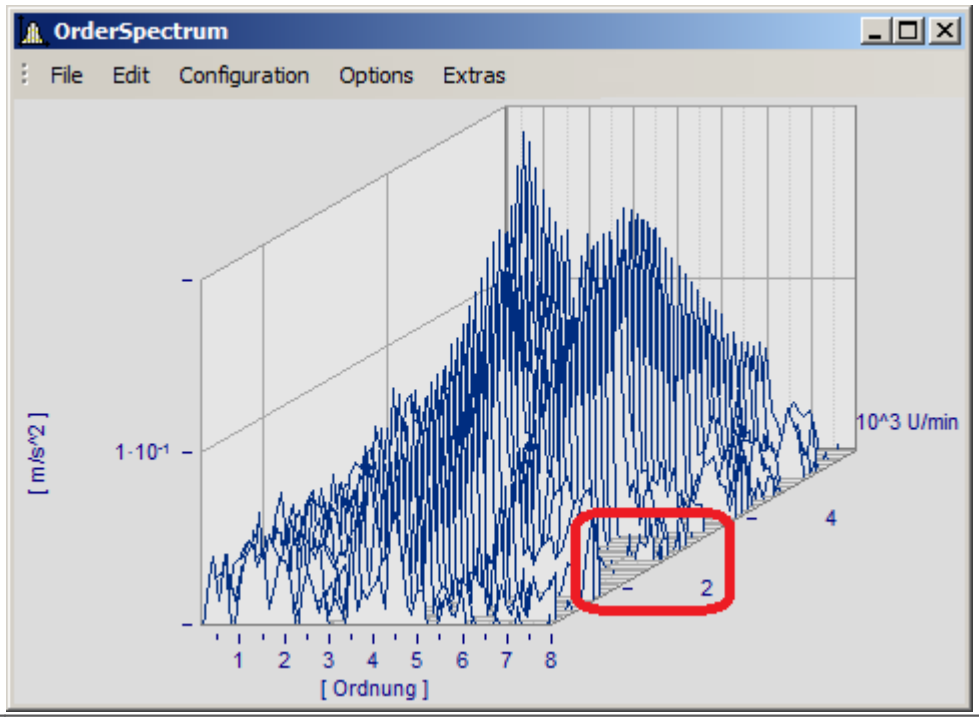
Mouse Operation

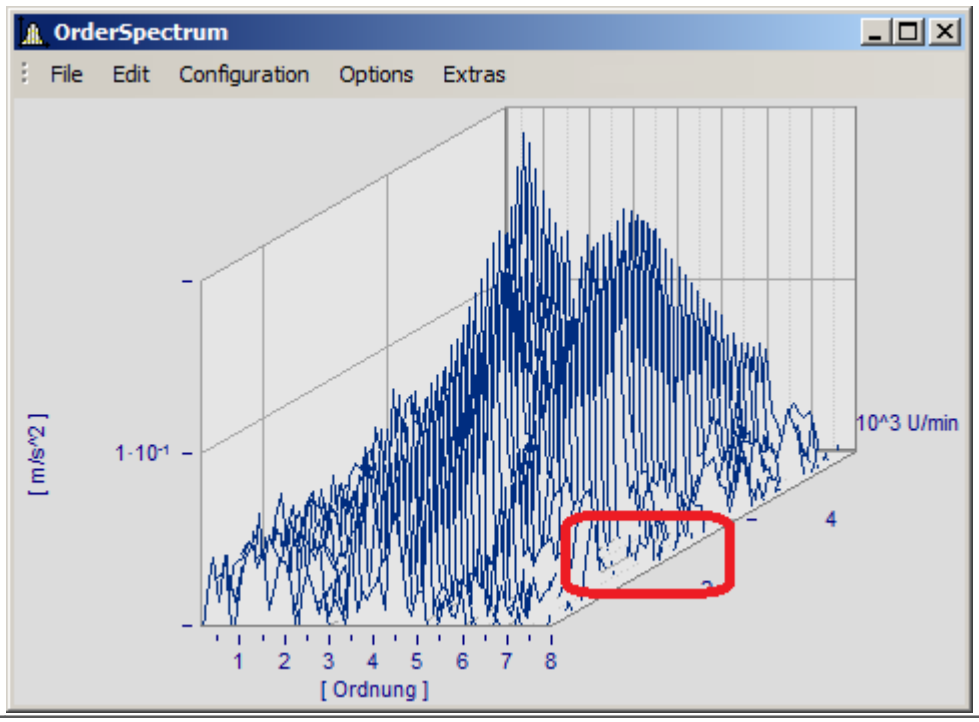
- Call the dialog for setting the type of display using the menu option *Configuration/ Display*.. Select *Waterfall* and end the dialog with *OK*.
- The curves displayed in the window are shown in a waterfall diagram with the effective presettings.
- A dialog is available to change the settings of the waterfall diagram. Select menu option *Configuration/ 3D*.... The following dialog appears:



Curve options

Enter here the settings for the appearance of the curves in the window.

Parameter - Curve options	Description
Connect curves to base lines	This option connects the ends of the curve plot to the base line. The base line runs under the curve, parallel to and in the height of the x-axis. It is the projection of the curve on the plane defined by the labeled x- and z-axes. This option is not active in the following illustration: 

Parameter - Curve options	Description
Show base lines	Base lines are drawn when this option is selected. A base line runs under the curve, parallel to and at the height of the x-axis. It is a projection of the curve on the plane defined by the labeled x- and z-axes. Base lines are also obscured by curves in front of them. Base lines are present in the above illustration; the illustration below shows no base lines: 
Same color for all curves	Select this option to draw all curves in the color of the first (foremost) curve. Otherwise the colors are assigned according to the selected color settings. The color composite is important for a three-dimensional appearance.
Transparent	When this option is selected, there are no hidden lines. However, the three-dimensional effect of the graph is reduced significantly. When this option is not selected, analog data sampled equidistantly are drawn in overwrite mode.

Orientation of z-axis

The options in this group refer to the display of the z-axis.

Parameter - Orientation of z-axis	Description
Show z-axis	Enter here whether the z-axis should be displayed.
Angle in degrees	The angle of the z-axis can be set to between 1 and 89 degrees. An angle of 30 degrees is recommended.
Length of z-axis in percentage of x-axis	The length of the z-axis is specified in percentage of the length of the x-axis. The minimum value is 10%

z-coordinate of curve

Select here between the options *Fixed values: 0,1,2,3,...* and *Fixed range z0, dz,....* In the former case, the first (foremost) curve takes the z-coordinate 0, the next takes 1, etc. The following settings are available when the latter option is selected:

Parameter - z-coordinate of curve	Description
z0	Enter the z-value here; the foremost curve will be displayed at this value.
dz	Enter the z-distance of adjacent curves in this box.
z-unit	A unit for the z-axis can be specified here.

Other options are also provided

Options	Description
Auto: z-coordinates of data	This employs a waveform's z-coordinate. In the imc FAMOS-dialog *Properties* of variables you can assign a z-coordinate to a waveform. The default is "0" The waveforms thus displayed must have unique and increasing z-coordinates. When handling structured waveforms (segmented, events..), this is the best choice. No further settings are necessary; the matrix will automatically be plotted with correct z-scaling.
Ditto with events: relative to first event	Only for multishot waveforms. The z-coordinate is determined from the various trigger times. The first shot is assigned the time 0 and all subsequent ones are designated accordingly. The other, related options listed make reference to other shots (events). E.g., the last, the first displayed, or the last displayed. The z-coordinate of each waveform absolutely must be greater than the last. Otherwise the display will not include all waveforms!
z-axis scaling	To scale the z-axis, double click on this option. Settings for the x- and y-axes, as well as for ticks and marking of the range can be made here.

Reference	Scaling the x-axis

Section 'Scaling the x-axis' 1350

Default

Clicking on this button sets all dialog elements to default values.

OK, Cancel, Set as default

For details on operation, see the chapter Confirmation bar 1423

Remarks

- Select an appropriate viewing angle. An angle of 30 degrees is a good starting point.
- A distorted display results if the angle is very small or very large; often the axes can no longer be labeled.
- The axes are labeled only when the coordinate system is large enough in every direction, i.e. when each axis exceeds a certain length. Enlarge the curve window if no labels appear. If there are still no labels, make any necessary changes to the angle and then the length of the z-axis. If an extremely long z-axis results, the window often has no room left for the labels.
- The area below the curve is **NOT** filled for XY-plots, digital data and reduced data. These displays always use the *Transparent* mode.
- Waterfall diagrams are often more legible when the y-axis is scaled upward very generously. Assume that the curves displayed have a value range from 3 to 12, the y-axis could be scaled from 3 to 30 (or 20 or 40). The axes need not be extended downward, since this <Shift>s the curves upward. Depending on the kind of data, appropriate y-axis scaling can greatly improve legibility.
- Not all data can be represented in a waterfall diagram. For example, noisy data can produce a chaotic image of overlapping lines. Waveforms which are relatively smooth and similar are best for waterfall display.
- When several curves are displayed, the first curve is always drawn in the foreground of the window; all additional waveforms appear behind the first curve.
- Waveforms with different x-axis scaling can be displayed together (i.e. waveforms sampled at different speeds). All curves are displayed correctly with the respect to time on the x-axis, as in all other curve windows.
- Only one y-axis is used for all curves in a waterfall diagram.
- A line display is always implemented for waterfall diagram, even if data were represented with symbols. The symbols are not displayed. The selection of *representation with symbols* is recorded and will be reactivated whenever the waterfall mode is ended.
- Waveforms displayed with a dotted line are interpreted as transparent in the waterfall display.
- Curve display settings such as dots, bar graphs or stair-steps can be set for each waveform individually, or for all waveforms in a window with the option *Valid for all axes* selected in the dialog for scaling the y-axis.
- Only transparent graphics can be outputted to plotters. The algorithm used here, which draws lines and subsequently draws over them with the next plane of display, does not produce the desired effect with plotters.
- Drawing lines which are then covered by the next graphics takes considerably more time than drawing transparent graphics. First try to output transparent graphics before experimenting with various settings and scaling methods.

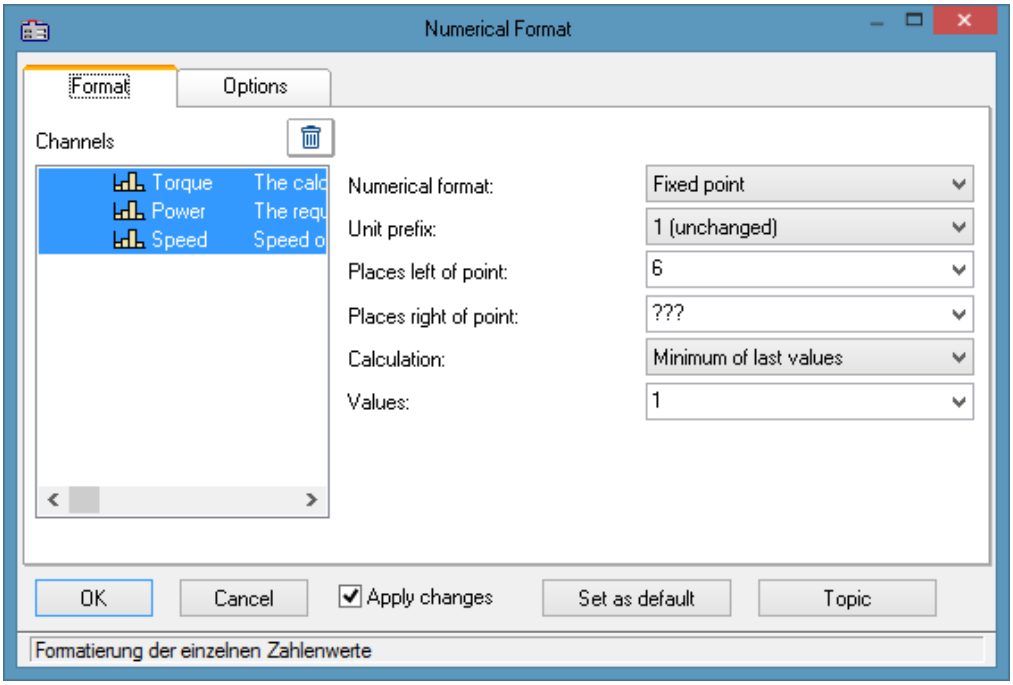
13.6.1.2 Last value as number

This mode, displaying the last value in a waveform as a numerical value, is especially suited for use of the curve manager in data acquisition and on-line display of these data. The last value in a waveform is always the most recent measurement value in on-line display. This mode is especially convenient for display of slowly changing quantities, allowing the user to monitor the amplitude of a spectral line, the effective power or a temperature as a numerical value. The display of the numerical value can be formatted and several channels can be compared with each other.

Mouse Operation

- Select *Configuration/ Display...*
- Select the option *Last value as number* and end the dialog with *OK*
- The curve window now shows the variable's name along with a numerical value. If you display more curves in the window, multiple numerical values appear stacked one above the other:

The format of this display can be changed by double-clicking on the curve window or selecting *Configuration/ Numerical Format...*. The following dialog appears for defining the number format:



Waveforms

A list of all waveforms displayed in the window is found on the left side of the dialog. Select waveforms from this list by moving the mouse to the desired waveform's listing while holding the mouse button. Multiple waveforms can be selected by holding down the <Ctrl> button and clicking on the desired waveforms.

Format (For all selected waveforms)

All settings made here only affect waveforms in the *Waveforms* list.

If the the waveforms selected have differing format settings, question marks appear in the associated boxes. Any settings you enter to replace the question marks will be applied to all waveforms selected.

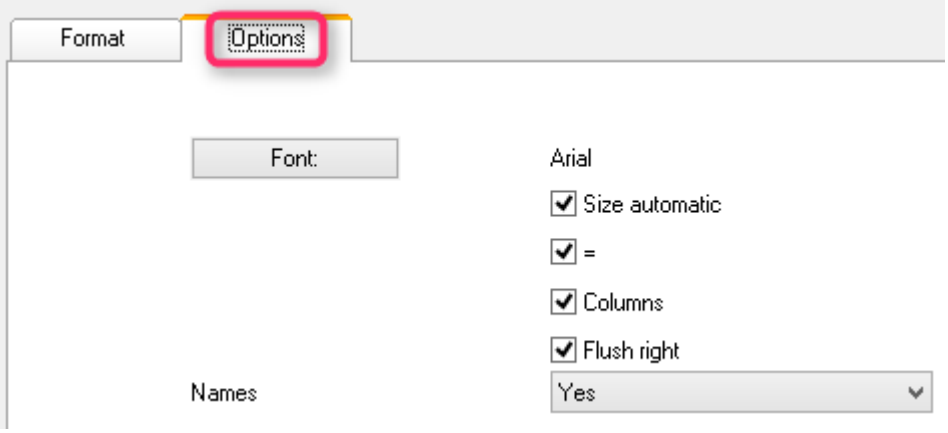
Settings - Format	Description
Numerical format	Representation as *fixed point* is customary. But the numerical values can also be formatted as *Floating point*, in *Hex*-format, or as *Date/Time*. **Numerical format** — **Description** Floating point — *Floating point*-representation uses the E-character to express the exponent, followed by the sign and a two-digit exponent. Hex (1, 2, 4 Byte) — Hex always displays the lowest selected Bytes, thus for example, the 1027 and 1 Byte is represented as 03h. Time relative-, Date/time auto — The point in time when the last sample was recorded is formatted in accordance with the display option applicable to the x-axis 1267. Time relative-, Date/time fixed — The formatting of the time and date is specified by means of placeholders, see the chapter "*Scale* 1351".
Magnitude	Numerical values can be associated with a fixed order of magnitude. Take as an example a current of 0.1A. To display the current in mA, select *milli*. The numerical value is then multiplied by 1000 and the prefix milli (m) is set in front of the unit, if it is not empty. Selectable prefixes range from pico (p) to giga (G). Select 1 if numerical values and unit should be displayed unchanged. Selecting a fixed order of magnitude is recommended for fixed-comma notation; if numbers vary greatly in their order, it is recommended to use floating-point notation with unchanged order.
Fixed-point, floating-point notation	Floating point notation uses the exponential notation, "e" and a two-digit signed power of 10. Fixed-point example: 0.123, -123, 888.987 Floating-point example: 1.45E+03, -1E-01, 1.4444E+00 Select Yes for Fixed-point or no for Floating point. With floating point, one digit always appears before the decimal.
Places left of point	The amount of digits before the decimal point, max. 15. With Fixed-point notation, sufficient digits must always be specified, otherwise a black bar appears. With floating-point format, one digit before the decimal point is sufficient.
Places right of point	The amount of digits after the decimal point, between 0 and 20. If no digits (0) appear after the decimal point, it is omitted.

Settings - Format	Description
Calculation	The numerical value displayed can be calculated from the most recent values:

Calculation	Description
Maximum of the last values	The maximum is calculated from the last *n* values in the waveform. This function can be interpreted in on-line-display as peak hold.
Minimum of the last values	The minimum is calculated from the last *n* values in the waveform.
Average of last values	The arithmetic mean is calculated from the last *n* values in the waveform. The reduces possible noise in display of the measurement data.

Values	Specify here the number of values from which the average, minimum or maximum is calculated. Any number from 1 to 30000 can be specified. If the waveform has fewer values than specified here, all values in the waveform are used for the calculation. Note that with larger numbers, the calculation must be performed for display of the curves. When 1 is specified, the result of calculating the minimum, maximum and mean value is always the last value in the waveform.

Options (Curve Window Defaults)



Settings made in this group of elements are effective for the entire curve window and are not made individually for each waveform. The font and several other basic display types can be set here.

Settings - Options	Description
Font	For selecting the font, font size and style. The size set here is ignored if the option Size: auto is activated.
Size automatic	The font size is automatically selected to be as large as will fit. However, only the size is adjusted.
Columns	The name, equals sign, numerical values, and units are displayed in columns for a clearer presentation. Without the option *Columns*, the entries are separated by spaces.
"=" (equal sign)	An equal sign appears after the variable's name, or else a colon. The colon appears directly with the name, but the equal sign appears in its own column.
Flush right	All numerical values appear flush along the right margin, independently of the format set. Without *Flush right*, all decimal periods are aligned one above the other. The condition for the aligned writing is that the *Columns* option is selected!
Names	Variables names and or comments which appear before the numerical value. If only one variable is in the window, the designator can be switched off since it is already in the title bar.

Set as default

The settings made initially only apply to the current window. If you click on the button *Set as Default*, any new curve windows created apply these settings whenever the display stye *Last value as number* is selected. You should only click on this button if no question marks are showing in the dialog.

The defaults for the format are **NOT** saved for each data set individually.

OK, Cancel

For details on operation, see the chapter Confirmation bar 1423.

Remarks

- Several menu items in the curve window, such as Measure, Scale and Overview Window cannot be selected when the numerical value display option is selected.
- Displaying numbers in fixed-point notation may have two effects in connection with the range of numbers. If the size of a number is too small, only zeros are displayed. On the other hand, when a number is too large to be displayed in the available space, a symbol appears which indicates an overflow. In this case, select more digits left of the decimal point or floating-point notation.
- With numerical display of online measurement data at high sampling rates, it can occur that the numbers change so fast that they are no longer visible. In that case, select a calculation and control the update rate for the values by making the appropriate setting under Values. Then select an appropriately small number of places right of the decimal. Usually only the digits to the right of the decimal point change strongly because the measurement signal changes only slightly.
- When time-stamped data are subjected to data reduction, the amount of values on which the calculation is performed is the same as the amount of measured points before the data reduction. Thus, specifying the amount of values to which the calculation is applied always fixes the time segment on which the calculation is performed, independently of how much the measured data are reduced.
- The color of the texts and the background in numerical display can be changed uniformly for all curve windows in the dialog *Options/ Colors...*. The following elements are available for selection:
  - *Numbers: foreground* or *background*

13.6.1.2.1 Single value display

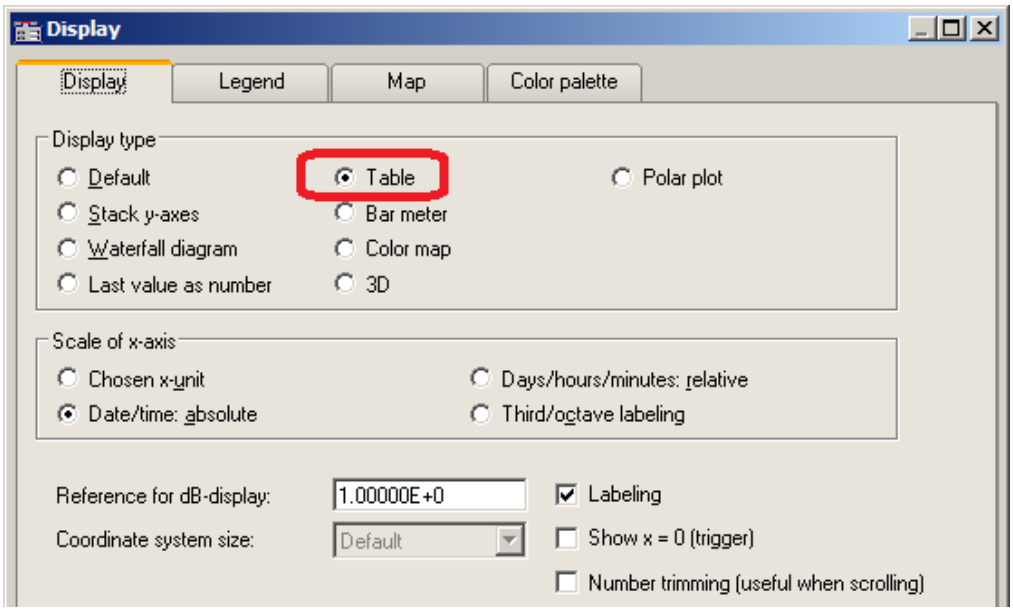
The display mode *Last value as number* is automatically activated if real waveforms with a length of 1 are displayed.

13.6.1.3 Tabular representation

Measurement readings can also be displayed in a table, where the readings are listed in chronological order. This can appear as a protocol readout or as a log book.

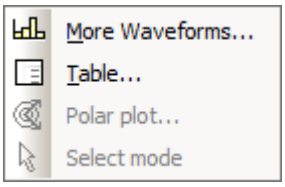
Operation

Select the curve window menu item *Configuration / Display...*. This calls a dialog for defining the curve window's display style. Select the entry, *Table*. Even the time display types *Chosen x-unit* and *Date/time absolute*, are applied in a tabular display.



The curve window then appears as shown below:

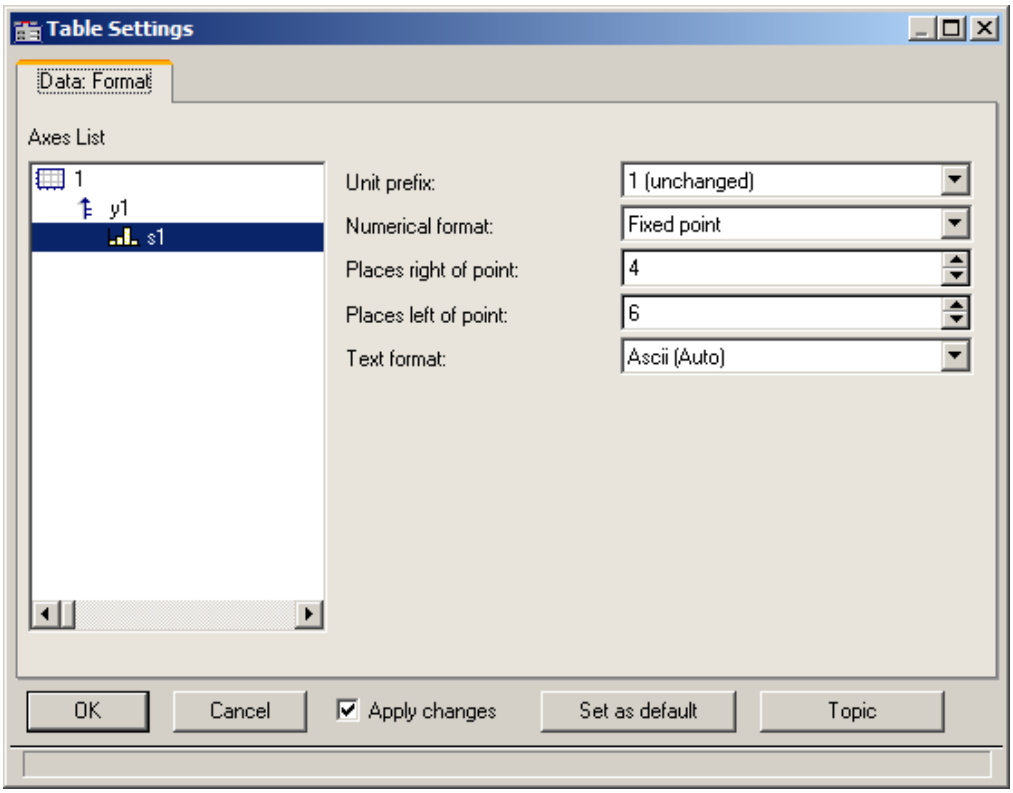
t ["order"]	OrderSpectrum [m/s²]
0.125	0.0092
0.125	0.0157
0.125	0.0161
0.125	0.0101
0.125	0.0083
0.125	0.0150
0.125	0.0194
0.125	0.0107
0.125	0.0142



It's possible to move through the list using the scrollbar at right.

Right-clicking the mouse on the curve window calls the context menu which now looks like this.

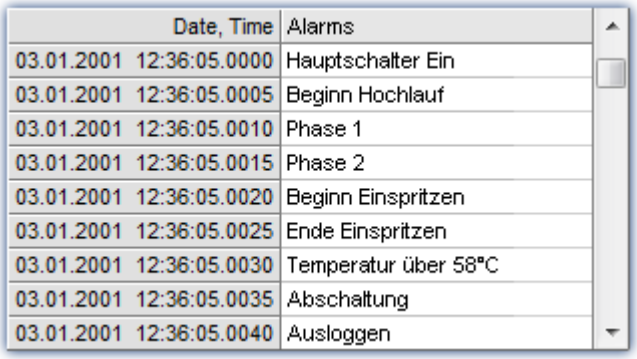
The menu item *Table...* calls the dialog offering controls as shown below:



The left portion of the dialog indicates all the channels displayed in the curve window. How these channel listings are arranged in terms of axes and coordinate systems doesn't matter here. On the right side of the dialog, the controls are located, which are for making settings for the channels selected in the left part.

13.6.1.3.1 Settings

If the curve window displays channels with measurement readings (default), then you need to deal with the settings for the *numerical format*, *namely Unit prefix*, *Numerical format*, *Places*, *right/left of point*. If texts are displayed (time stamp-ASCII data), then the text format becomes relevant.

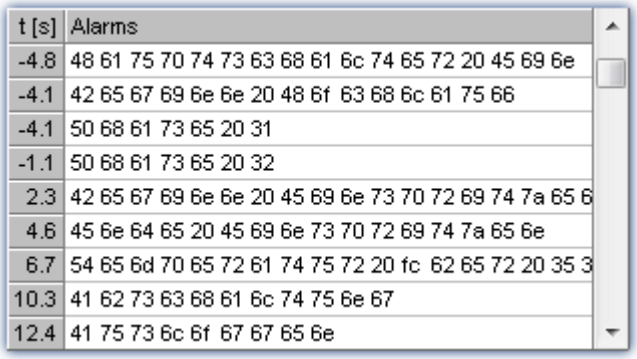
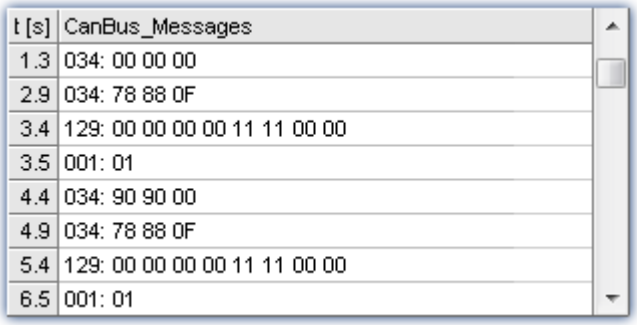
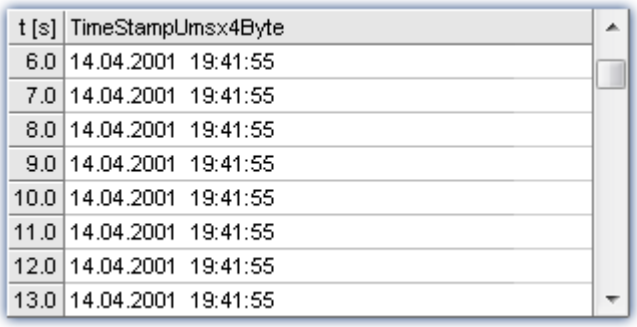
Settings	Description
Unit prefix	This consists of the order of magnitude and, if appropriate, a prefix such as kilo, milli, Mega or even just 1 (default, use of unit without prefix), which appears after the number before the unit. If the reading is 10 and takes the unit V (Volts), then for a selection of "m" (milli), a value of 10000mV is returned.
Numerical format	Fixed point can be selected for readout with a fixed number of decimal places and without an exponent; e.g. 17, 17.35, -0.0017. Floating point can be selected for readout with fixed relative precision and an exponent, e.g. -1.28E-7 or 3.4E+0
Places right of point	This is the number of decimal places to show. 0..15 places are possible.
Places left of point	Only for fixed point format, a specified amount of places before (to the left of) the decimal point. The number should be large enough to ensure that the numerical value is indicated with all its digits.
Text format	This setting only pertains to channels with information presented as strings, in other words on the case of time stamp-Ascii data. Such data are, for instance, generated by imc Online FAMOS , but also by functions belonging to the Time Stamp Ascii Kit (TSA-Kit). This data type contains texts, where each text has a time stamp, e.g. log-book entry representation with absolute time selected: 

Date, Time	Alarms
03.01.2001 12:36:05.0000	Hauptschalter Ein
03.01.2001 12:36:05.0005	Beginn Hochlauf
03.01.2001 12:36:05.0010	Phase 1
03.01.2001 12:36:05.0015	Phase 2
03.01.2001 12:36:05.0020	Beginn Einspritzen
03.01.2001 12:36:05.0025	Ende Einspritzen
03.01.2001 12:36:05.0030	Temperatur über 58°C
03.01.2001 12:36:05.0035	Abschaltung
03.01.2001 12:36:05.0040	Ausloggen

Text format

The following options are available:

Text format	Description
Ascii (Auto)	Display of ASCII characters in legible form (see above.).

Text format	Description
Hex	Display of hexadecimal values 00H .. FFH.  t [s] Alarms -4.8 48 61 75 70 74 73 63 68 61 6c 74 65 72 20 45 69 6e -4.1 42 65 67 69 6e 6e 20 48 6f 63 68 6c 61 75 66 -4.1 50 68 61 73 65 20 31 -1.1 50 68 61 73 65 20 32 2.3 42 65 67 69 6e 6e 20 45 69 6e 73 70 72 69 74 7a 65 6 4.6 45 6e 64 65 20 45 69 6e 73 70 72 69 74 7a 65 6e 6.7 54 65 6d 70 65 72 61 74 75 72 20 fc 62 65 72 20 35 3 10.3 41 62 73 63 68 61 6c 74 75 6e 67 12.4 41 75 73 6c 6f 67 67 65 6e
CAN, LIN message	For all devices with a CAN/LIN-connection: display with identifier and contents of the CAN/LIN-messages. The identifier (appearing before the colon) and the bytes belonging to the message are displayed as hexadecimal values.  t [s] CanBus_Messages 1.3 034: 00 00 00 2.9 034: 78 88 0F 3.4 129: 00 00 00 00 11 11 00 00 3.5 001: 01 4.4 034: 90 90 00 4.9 034: 78 88 0F 5.4 129: 00 00 00 00 11 11 00 00 6.5 001: 01 **Note** — **Messages with extended identifier** In Hex 1291 representation of the content of a message, the extended bit is always visible. In case of CAN, LIN message format, the bit is taken into account to output 4 or 8 hex characters for the identifier. But in the output table the bit is hidden.
4 byte abs. time	For all devices which can read in 4 Byte time stamp data (seconds since 1.1.1980), there is a specialized display style.  t [s] TimeStampUmsx4Byte 6.0 14.04.2001 19:41:55 7.0 14.04.2001 19:41:55 8.0 14.04.2001 19:41:55 9.0 14.04.2001 19:41:55 10.0 14.04.2001 19:41:55 11.0 14.04.2001 19:41:55 12.0 14.04.2001 19:41:55 13.0 14.04.2001 19:41:55

Text format	Description
3-number sets (hex)	Display of hexadecimal values in 3-number sets.

t [s]	Hex_3er_Messages_S1_K2			
2.70843	64 00 00	00 00 00	00 00 00	00 f e f f
2.72640	65 00 00	00 00 00	00 00 f e	f f 00 00
2.79394	66 00 00	00 22 00		
2.80843	64 00 00	00 00 00	00 00 00	00 f e f f
2.82641	65 00 00	00 00 00	00 00 f e	f f 00 00
2.89394	66 00 00	00 23 00		
2.90843	64 00 00	00 00 00	00 00 00	00 f e f f
2.92641	65 00 00	00 00 00	00 00 f e	f f 00 00
2.99396	66 00 00	00 24 00		

Flexray-message	For all devices with a Flexray-connection: display with identifier and contents of the Flexray-messages. The identifier (appearing before the colon) and the bytes belonging to the message are displayed as hexadecimal values.

t [s]	Flexray_Messages_S1_K2
33.81970	65 0000 00: 00 00 00 00 f e f f 00 00
33.88721	66 0000 00: 2d 00
33.90171	64 0000 00: 00 00 00 00 00 00 f e f f
33.91970	65 0000 00: 00 00 00 00 f e f f 00 00
33.98723	66 0000 00: 2e 00
34.00171	64 0000 00: 00 00 00 00 00 00 f e f f
34.01970	65 0000 00: 00 00 00 00 f e f f 00 00
34.08724	66 0000 00: 2f 00
34.10171	64 0000 00: 00 00 00 00 00 00 f e f f

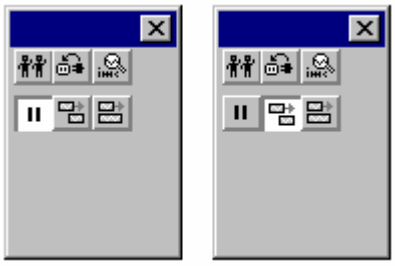
Multi-channel displays

Display of multiple channels at once also uses correctly synchronized ordering. For every measurement reading, the source channel is indicated:

t [s]	Channel	Value
0.0000	s2	0.8399 m/s^2
0.0000	s1	0.1601 m/s^2
0.0005	s2	0.4382 m/s^2
0.0005	s1	0.5618 m/s^2
0.0010	s2	0.7121 m/s^2
0.0010	s1	0.2879 m/s^2
0.0015	s2	0.5478 m/s^2
0.0015	s1	0.4522 m/s^2

Online-Display

If a table is displayed during a running measurement, it may be advantageous to have a readout of the current readings, at the bottom of the table. For this purpose, the roll-mode is provided, just like for time-based displays in coordinate systems.



OK, Cancel, Set as default

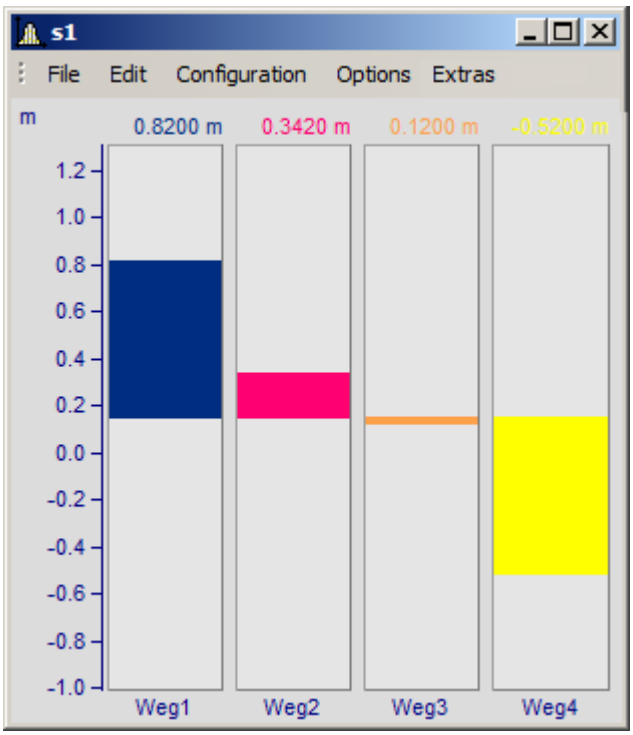
For details on operation, see the chapter Confirmation bar 1423

Remarks

- The entries are always ordered chronologically.
- The condition for using this display type is that the time coordinate in each channel behaves like time, in other words, increases continuously.
- If the time coordinate decreases in a data set, for example, in the case of a XY-waveform, this display type cannot be used.
- For information on the buttons *Ok*, *Cancel* and *Set as default*, refer to the chapter on Confirmation bar 1423.

13.6.1.4 Bar meter

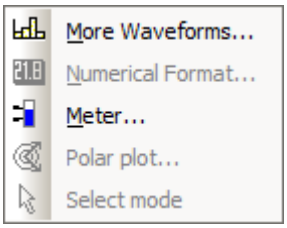
In addition to the numerical data display *Last value as number*, there is also a data display style having the form of a vertical meter bar. The height of the bar is determined from the last value in the waveform. This representation style is particularly useful for online viewing, as it offers a convenient overview of signal developments in multiple channels. The representation style incorporates graphical features such as different colors, slave pointers, margin checking etc.



Settings

Select menu item *Configuration/ Display* in the curve window. Select the *Bar Meter* radio button.

It is possible to display several channels together using a common y-axis. Select the channels to be displayed in the curve window.



To configure the bar meter itself, right-click on the curve window to open the context menu:

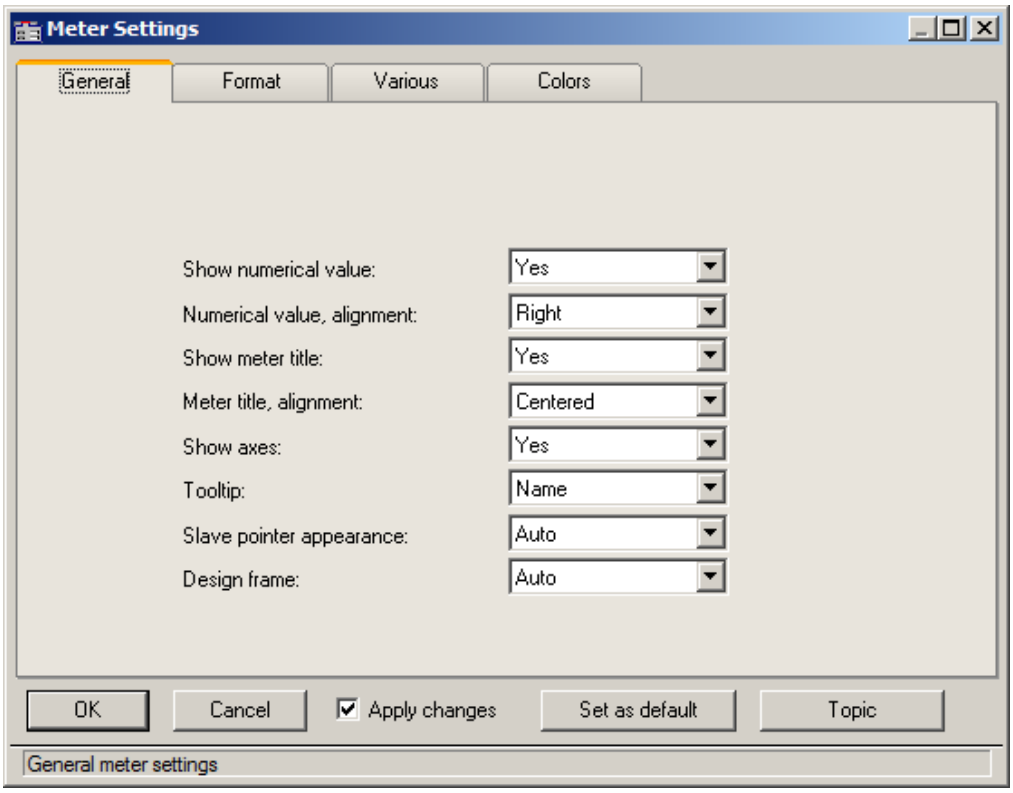
Select *Meter...* to open the *Meter Settings* dialog. This dialog consists of several sub-dialogs, each of which contains controls for different settings.

If you have preferences for certain settings, use the *Set as default* button. These settings will be implemented automatically each time bar meter display mode is used.

The following is a description of each sub-dialog in the *Bar Meter Settings* dialog.

Settings: General

This is where settings are made which affect all bars in the curve window alike.



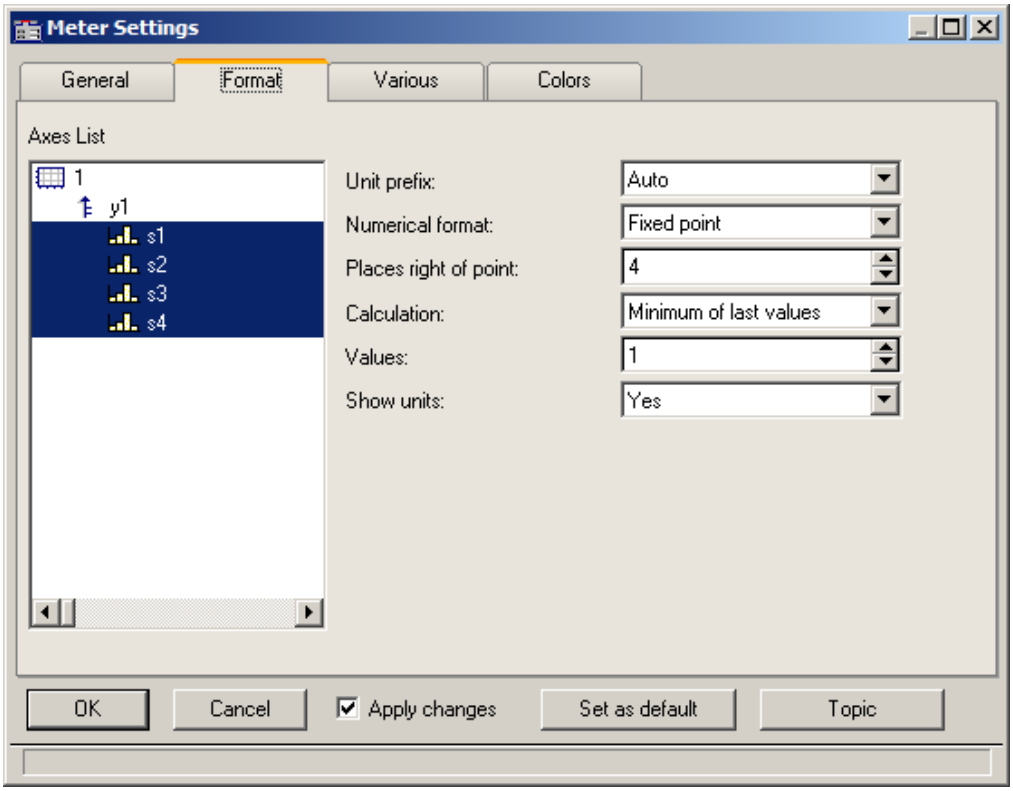
Settings	Description
Show numerical value	This setting governs whether to display the numerical value of a signal together with its bar meter representation. The numerical value is written above the bar.
Numerical value, alignment	This controls the horizontal alignment of the numerical value (in case it is specified to appear). The recommended setting is *right*, but *left* and *centered* are offered.
Show meter title	This determines whether to display the corresponding channel name under the bar meter of a signal. This is particularly useful when multiple channels' signals are being shown.
Meter title, alignment	The horizontal alignment of the title text, which is always written in one line. Choices offered are *left*, *right* and *center*.
Show axes	A y-axis can be specified to appear adjacent to the bar meter at left. The scaling of the axis can be set by the usual method for curve windows. This axis determines the value range of the bar (lower-limit physical value at the bottom edge, upper-limit physical value at the top edge). This is also where to specify whether the scale is indicated or not. Even if the scale is not depicted, it is still applied to the bar's value range. The settings in the curve window's menu item Configuration/ Axes must be made afterwards.
Tooltip	For *Name*, the channel name is displayed; for *no*, it is not.
Slave pointer appearance	*Line*, or a clamp for *auto*.

Settings	Description
Design frame	With *auto*, each channel has its own frame; with *Common frame*, not.

Settings: Format

This filecard mainly concerns the settings for the numerical format and for computing the numerical value. The numerical value is used for the readout above the bar and for determining the height of the bar.

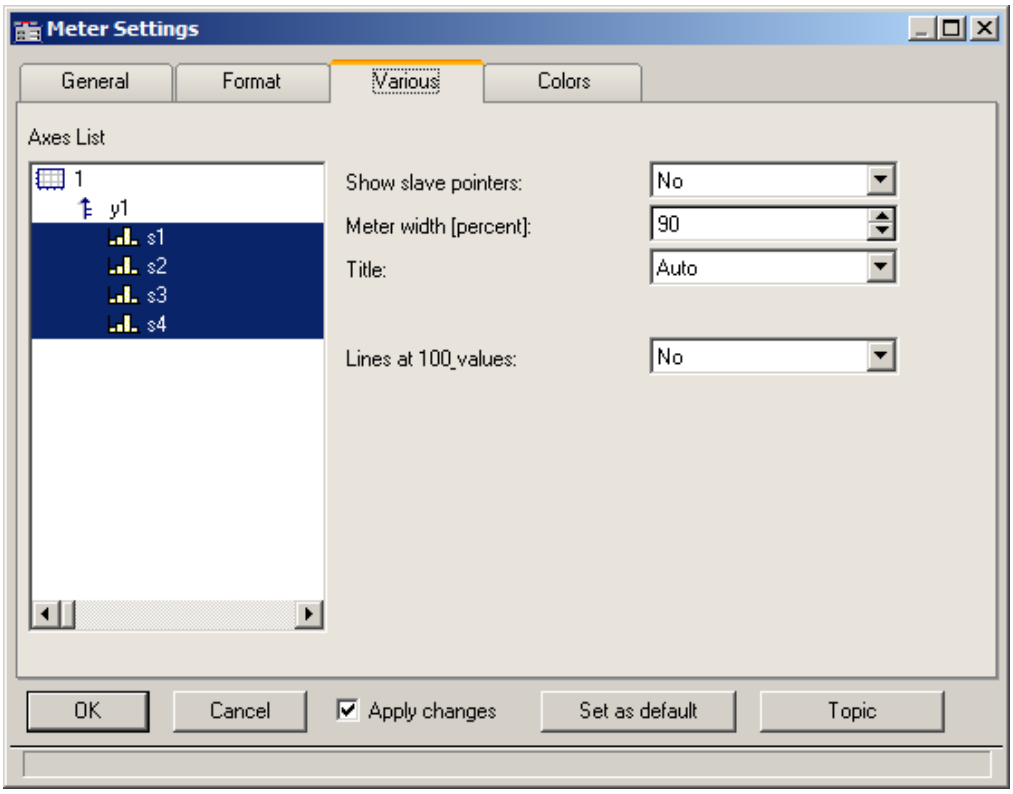
On this and the following sub-dialogs, the first step is to select all the channels to which the settings made should apply in common. Multi-selection can be performed as described in the instructions for the *axes list* 1341. If multiple, selected channels have mutually different settings for a property, the setting indicated for that property will be "???".



Settings	Description
Unit prefix	You can specify a unit prefix, reflecting the 10-based exponent of the signal value, to accompany the displayed numerical value. Prefixes from piko (p) to Giga (G) are supplied. When *auto* is set, the exponent is oriented to the exponent of the y-axis scaling, which is the recommended setting. For fixed point notation, a fixed exponent is recommendable, as well as for the y-axis scaling - see the description of the y-axis. Example: the waveform involved is a power measurement taking units of Watts and extending in range from 1e4 to 1e6 W. As the fixed unit prefix we choose "kilo" (k), resulting in values of 10kW to 1000kW. This setting is unnecessary if the numerical value of the signal isn't set to be displayed.
Numerical format	Fixed point (e.g., 0.01, 100, 365.25) or floating point notation (e.g., 3.5E-3) are available. If the value range of the signal is known, fixed point notation is better. This setting is unnecessary if the numerical value of the signal isn't set to be displayed.
Places right of point	Zero to fifteen places after the decimal point can be specified. This setting is unnecessary if the numerical value of the signal isn't set to be displayed.
Calculation	The displayed numerical value can be the most recently measured value, or can be a function of the N most recent values. The functions of N values offered are mean, minimum and maximum. If the waveform has fewer than N values, all available values are used in the calculation. If N = 1, the particular function used is irrelevant. The height of the meter bars, as well, is derived from the setting made here.
Values	Enter the number N of values, from which the mean, minimum, or maximum is to be calculated. Only integers >= 1 are permitted. A large value for N needs much time for computation, but 100 values is an amount the system can comfortably and quickly handle.
Show units	This determines whether the physical unit, together with the exponential prefix (p, n, µ, m, ...) is to be displayed with the numerical measurement value. It is sometimes necessary to include an indication of the physical unit, in order to identify the value's order of magnitude. The unit should be omitted only if it is already accurately reflected by the y-axis scaling.

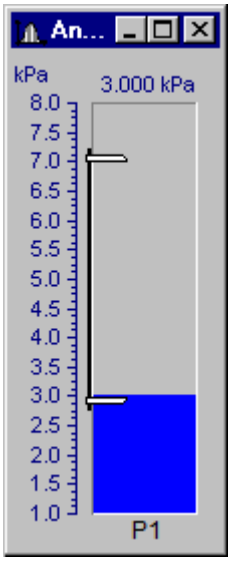
Settings: Various

This sub-dialog contains an assortment of setting controls, such as the width of the meter bars, or settings concerning the slave pointer.



Settings	Description
Show slave pointers	Specify here whether a meter bar should be equipped with slave pointers. These are markings appearing along a bar, which at any given time are situated where the signal's minimum and maximum encountered values were.
Meter width (percent)	Specify here the width of the meter bar relative to the width of the bar's "cell", which is the approximate width of the curve window divided by the number of bars in the window. Any fraction of the cell width, from 0% to 100% can be entered; a setting of 100% causes the bars to touch each other and thus 90% is recommended for greatest clarity.
Title	The title of the bar meter can be the channel to which its signal belongs (setting: *auto*), or any arbitrary name (*Fixed title*). If you select *Fixed title* from the list box, the entry box of the same name will appear below. Fixed title: Enter here an arbitrary designation for a bar meter. Up to 20 characters are allowed.

Slave Pointers



Slave pointers display the largest and smallest levels of a channel. At the beginning of a measurement, they lie together. They generally move away from each other during a measurement - the top one follows the largest values while the bottom one follows the smallest values. The slave pointers may also lie outside of the currently visible region.

The bar's slave pointers are represented by a graphic resembling a bar clamp. The slave pointers may also be located outside of the visible range.

Whenever a curve window is first opened or a new curve configuration is loaded, the slave pointers in that window are reset. The slave pointers are also not displayed when no data is present.

Slave Pointer: Computation

The slave pointers are rapidly computed by simply tracking the minimum and maximum values in a waveform. Therefore, slave pointers can be updated quickly even with large amounts of data. In typical online measurement applications, where the incoming data is simply appended to the data record, the use of slave pointers is appropriate. In off-line data analysis, however, they are not particularly useful because the entire data record is loaded at once.

Starting a new measurement on a channel with slave pointers does not reset them. They retain their former value until new data forces them to a new position.

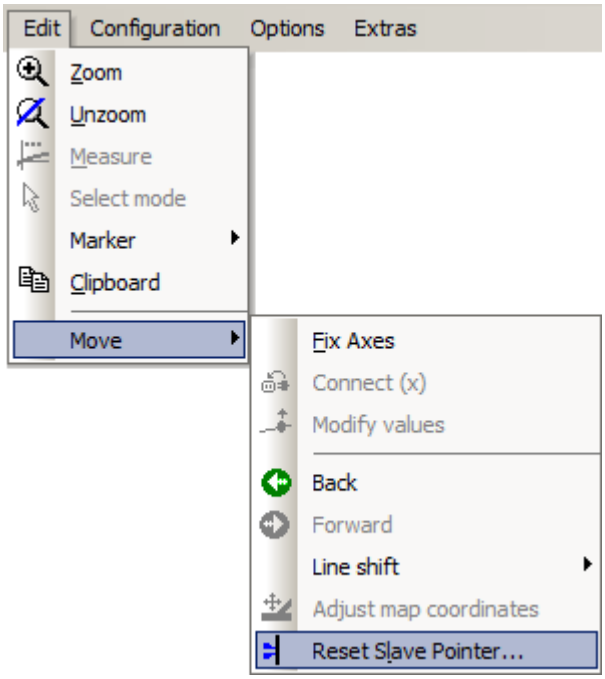
Slave pointers are computed from the actual measurement data and not from a value computed from the channel for the bar meter. For example, consider a bar meter which is configured to display the average of the last 10 values. Although the meter is unlikely to show transient maximum values, these will affect the position of the slave pointers.

To avoid this effect from occurring, compute the average *online* and then display the resulting *virtual channel* in a bar meter. The slave pointers will then agree with the displayed maximum and minimum values.

Slave pointers are accurate to 6 significant figures, i.e. they use 4 Byte representation.

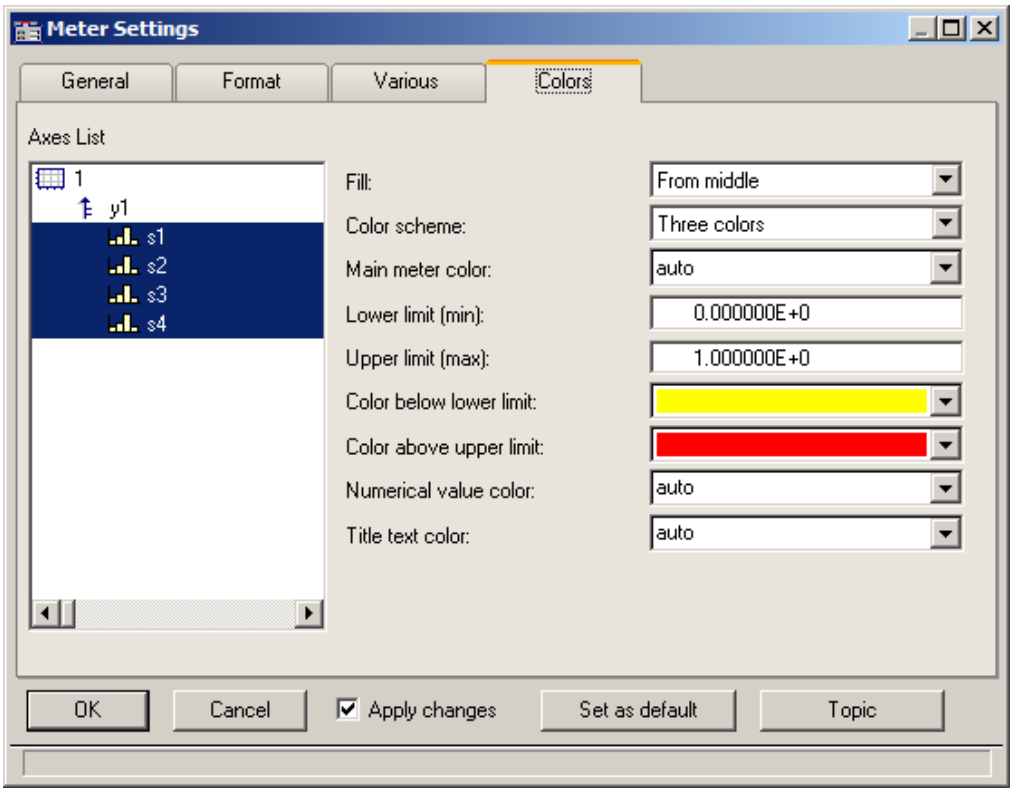
Slave Pointer: Reset

Resetting a slave pointer causes *all* slave pointers in that curve window to be reset. When you reset a slave pointer, it slides back to the current measurement value. Slave pointers have to be reset manually in each window. When a new measurement is started, this does *not* automatically reset the slave pointers!
To reset the slave pointers in a curve window, select *Edit / Move / Reset Slave Pointer...*.



Settings: Colors

This sub-dialog lets you decide on colors for the bars.

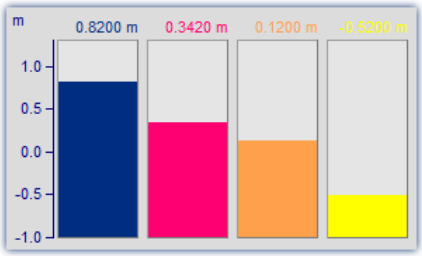


Fill

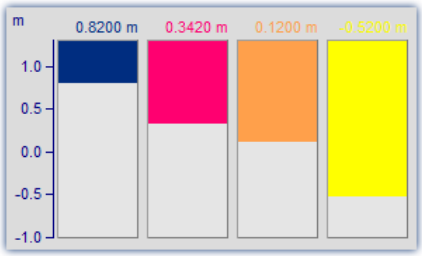
This setting determines from where the meter bars grow with increasing signal measurement values, as represented visually by a column filling with color.

The available options are from the top, from the bottom, from the center, and from y = 0 (the height along the bar, where the y-axis is equal 0.0).

From below



From above



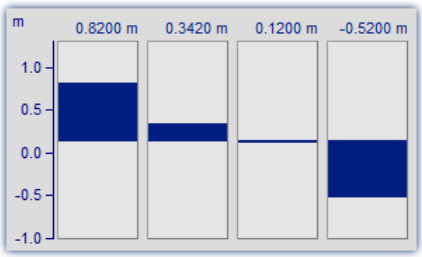
From y = zero

Alternating up and down from the 0 line of the axis

Color scheme

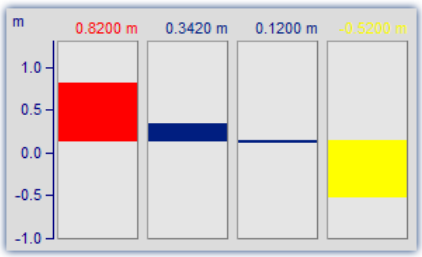
The bar can have only one color, three colors in succession or up to three colors at once. Three colored bars are used to indicate whether the measurement signal is currently within a tolerance range, which can be specified by the user in terms of an upper and a lower threshold value.

One color



A one-colored bar has the same color at all times.

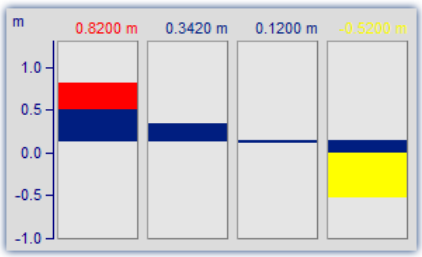
Three colors



The three color bar (*one color at a time*) shows the whole filled area in one color, which depends on whether the signal is below range, within range, or above range.

- The 1st color, if: Value <= Lower limit
- The 2nd color, if: Lower limit < Value < Upper limit
- The 3rd color, if: Value <=Upper limit

Three colors



Where a bar has *Three colors together*, each of its colors is shown only within a certain range of the bar's height, as outlined below:

- The 1st color, in the range: fill height <= Lower limit
- The 2nd color, in the range: Lower limit < fill height < Upper limit
- The 3rd color, in the range: Upper limit <= fill height

Depending on the circumstances, therefore, up to all three of the colors can be visible in a bar.

Settings	Description
Main meter color	This is the only color of the meter bar if only one color style is specified; for three color style it is the middle color (the color of the bar indicating a signal within the limits set by the user). When the valid setting is *auto*, the automatic selection which would have been made for a waveform in the curve window is applied to the bar. The background color of the bar is the background color of the coordinate system, which is set via dialog Options/ Colors 1493 of the curve window. If a waveform doesn't contain any values, the bar meter remains empty as well.
Lower limit (Minimum)	For three-color display modes, a lower limit must be specified in this entry box. If there shouldn't be any lower limit, it is possible to enter -1e30, which is generally lower than any signal which will ever be measured.
Upper limit (Maximum)	For three-color display modes, an upper limit must be specified in this entry box. If there shouldn't be any upper limit, it is possible to enter 1e30, which is generally higher than any signal which will ever be measured. The upper limit must be greater than the lower limit.
Color below lower limit	In three-colored bars, this is the first color.
Color above upper limit	In three-colored bars, this is the last color.
Numerical value color	The numerical value, if it is specified to be displayed above the bar meter, can be assigned a fixed color. Or, if set to *auto*, the color is the same as of the bar itself.

Settings	Description
Title text color	The title below the bar can be assigned a fixed color. Or, if set to *auto*, the color is the same as of the bar itself.

Note

- **Reducing flicker:** If flickering occurs in a bar meter during a measurement, then you should use averaging (Meter Settings dialog: *Format/ Calculation/ Average of last values*). Be sure to set the numerical format to floating point. Restrict the decimal places right of the comma to just a few and use right alignment.
- **Appearance:** In general, tall, slim meters appear better than short, wide ones. If necessary, the curve window should be resized.

Limitations

- With data containing events or segments, the structure is ignored. If results are being computed from the last N values, the data set is simply considered a long waveform. Thus, computations can be performed on more than one event. IMPORTANT: Because this behavior may be changed in upcoming versions, we do not recommend that you use it.
- Only one y-axis is possible. All bar meters in a curve window have to use the same y-scaling. If you wish to use different y-scaling, than use separate curve windows.
- Slave pointers are only appropriate for online measurements. If data is processed off-line (e.g. with imc FAMOS), the slave pointers will not always function properly, i.e. they will not display the maximum / minimum values. However, if values have only been appended to a data record, the slave pointers will function properly. At the beginning of a new measurement, the data record has to be emptied before appending new data. This allows the slave pointers to correctly interpret the new data.
- With XY-data as well as with data reduced using Transitional Recording, computations over the last N values always use a different time interval. This behavior may be changed in upcoming versions.
- The slave pointers can only properly interpret data which has been correctly appended to a data record. For example, data which has not been recorded via the measurement device and therefore has not passed through the Data Manager is not properly interpreted.

13.6.1.5 Color map display

This style of data representation is analogous to a geographical map with color-coding of surface relief.

Multiple waveforms are arranged in a stacked layout (in the y-direction), where each one's values are plotted from left to right (in the x-direction). The amplitude is plotted in the z-direction, in other words, extending vertically out from the screen. The result is a contoured surface extending out from the xy-plane, onto which the observed looks with a bird's-eye view. The hight of the surface is indicated by color coding, where different ranges of height correspond to specific colors.

There is additionally the possibility of superimpose x-y waveforms over the color map, and to have the map bordered by the waveforms. For more on this topic, see the section *Lines / Extras* 1377.

Prerequisites

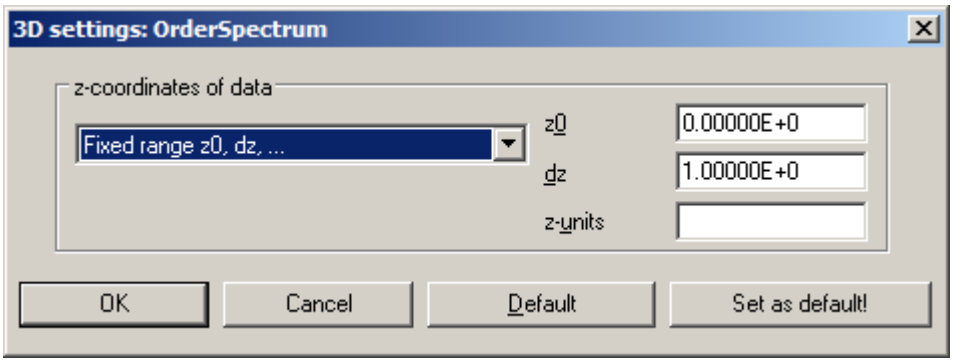
- Color video card with more than 256 colors, in other words 16 bit or 24 bit colors.
- Color printer of DIN A4 format.
- A fast PC (>= 586, 100MHz) with plenty of RAM (>= 32MB)
- Since the graphics display is based on large bitmaps (pixel graphics), large displays in particular require much storage space and computation
- With a DIN A4 color printer having 300dpi, one has about 2000 by 3000 pixels and with 24 bits per pixel one already arrives at about 18Mbytes!

Operation

First display all desired channels in the curve window. Then select the menu option *Color Map* in the dialog *Configuration / Display...*.

Setting the Z-coordinate

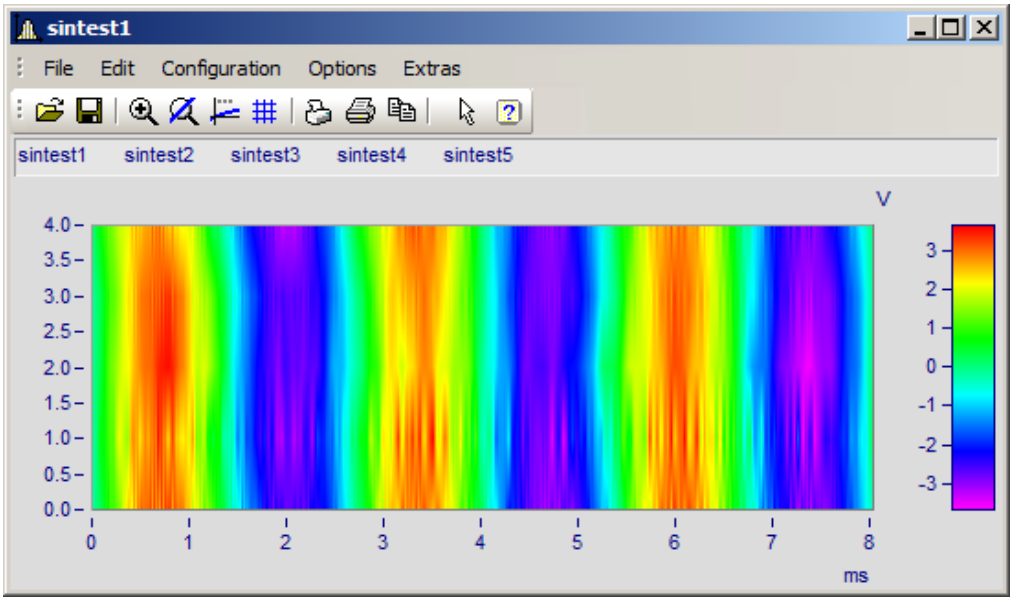
In the dialog: *Configuration / 3D* 1315 of the curve window, select an appropriate z-coordinate (see below):



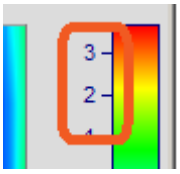
Reference Waterfall diagrams z-coordinate

More on the topic of Waterfall diagrams z-coordinate 1277

Result



Z-Axis Scaling



To adjust the color axis, double click on the scaling of the colored legend. The color axis is the same thing as the z-axis and is scaled as such.

Color Selection

The colors are set by means of the Color Map dialog: *Configuration/Color Map Display* 1306.

Interpreting the Z-coordinate

The z-coordinate of the data (an otherwise undefined coordinate in imc FAMOS) is to be set via the dialog Configuration/ 3D in the curve window (see above). For example, The z-coordinate of the first waveform (channel) in a group could be set to z-min, the next to z-min + dz, the next to z-min + 2dz etc.

Orientation of the Coordinates

The component waveforms in the map are represented as multicolored horizontal lines, whose x-range is plotted from the x-minimum on the left edge of the graph to the x-maximum on the right edge. The height on the screen of a waveform's line or strip is determined by that waveform's specified z-coordinate; a waveform's z-coordinate is thus plotted along the y-axis, in a manner of speaking. Conversely, a waveform's y-coordinate, being its measurement value, is represented by a color; thus one can say a waveform's y-coordinate is plotted along the color axis, which for these purposes is also referred to as the z-axis. The scaling of the y-axis, then, serves to landmark regions of the pictorial representation, without pinpointing the location of any physical value in that representation.

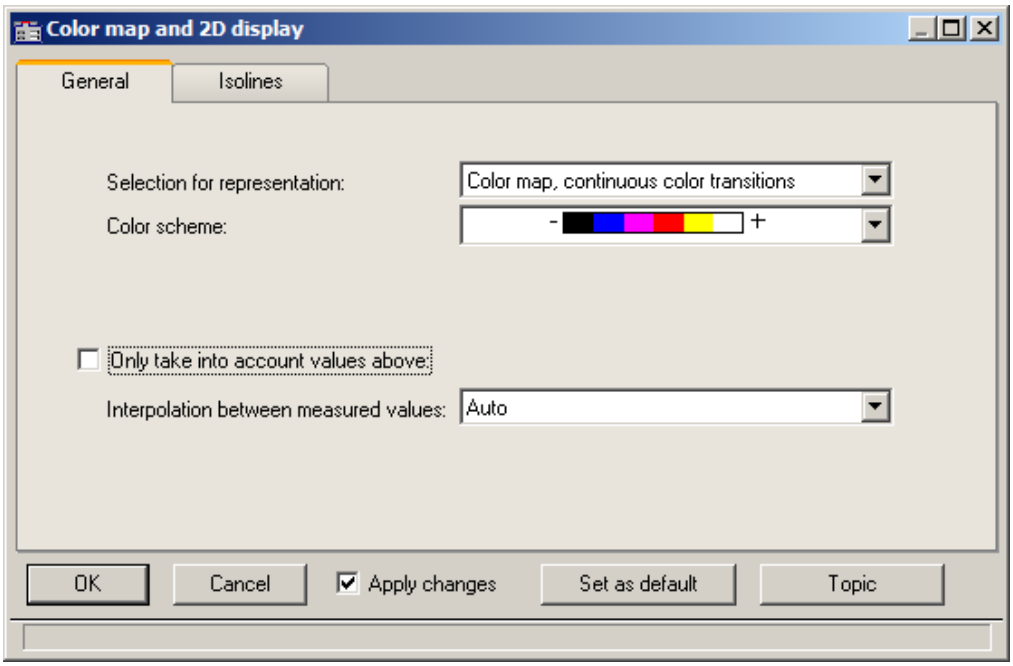
We thus achieve the illusion of a colorful landscape depicted on a map with green signifying valleys, the mountains brown and the snowy peaks white (in our case the colors follow the order of the physical spectrum).

Limits

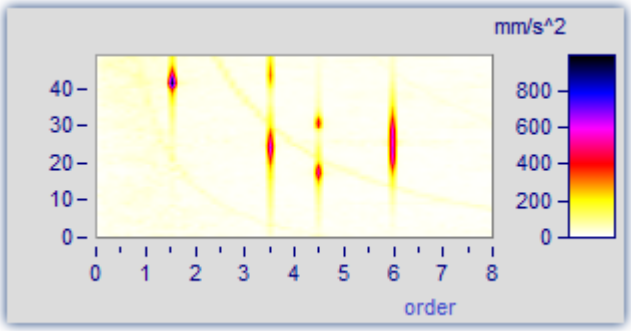
- With locus and XY plots a measurement cursor is not available for the z-coordinate.
- The colored map representation is somewhat slow by comparison with other representation styles. Printing in particular requires large amounts of storage capacity. This may also require patience, therefore.
- The interpolation between two neighboring waveforms is fixed.
- The data's z-coordinate must increase strictly monotonically.

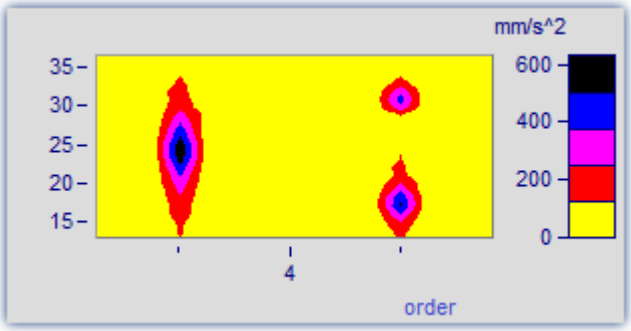
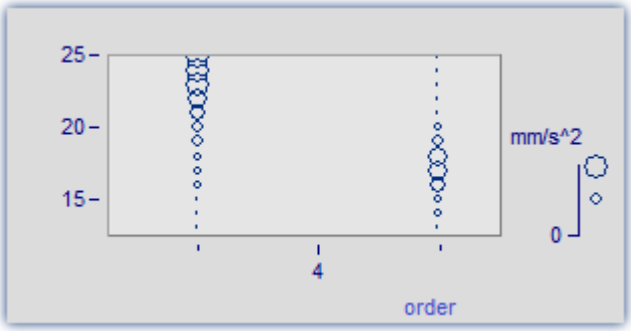
13.6.1.5.1 Color map dialog, "General"

The options for color map display are set in the dialog called by means of the menu item *Configuration/Color Map*:



Selection for representation

Representation	Description
auto	Default colors from magenta to red, or from white to black for B/W display. No other options.
Color map, continuous color transitions	Interpolated color transitions. For each amplitude to represent, an appropriate color is calculated. The color scheme can be selected. 

Representation	Description
Color map, graduated colors	Specified fixed number of colors without interpolation. The color scheme can be selected. **Number of different colors:** The amount of different colors must be between 2 and 1000. The suggested values in the list can be overwritten. 
Symbols, size corresponds to filling (Campbell)	Each measured value is represented by a symbol, e.g. a circle. The **size** of the symbol represents the respective amplitude. The minimum symbol size represents a value of zero, while the maximum represents the maximum value. The intermediate transitions are linear. The symbol is centered around the measured value.  The *Color of the symbol* corresponds to that of the first curve and can not be changed here. Under *Symbol selections*, ellipses, rectangles and diamonds, either drawn as outlines or filled, are available.
Remarks on display as symbols	• With symbol display, the appearance of the symbol's size in the legend is as true to reality as is possible, however, in consideration of how much space in the curve window is taken up by the legend, there are limits to the size. • With symbol display, the readings must increase in the x-direction strictly monotonously. Otherwise, no display is possible. • If XY-data having varying dx increments, or data with varying distance in the z-direction are graphed, then there is no determined size for symbols with (actually constant) size but variable filling. In the legend, an automatically determined value is used for the size.

Representation	Description
Symbols, filling corresponds to amplitude	Each reading is represented by a symbol, e.g. a box. The degree of the symbol's **filling** depends on the signal's amplitude, where an empty symbol (box) corresponds to the minimum and the maximum is represented by the largest size symbol available. The scaling between these extremes is linear. The *Fill color* and *Frame color* can be adjusted. For the frame, the setting *auto* is available, which represents the color of the first curve. For *transparent*, the frame is omitted. The filling increases, depending on the style chosen, either from the outside to the inside, or vice-versa, or from one side or corner to the other. The measurement value is located at the symbol's lower left corner. For the display with symbols of varying size, the maximum symbol size can be set. The size then depends on the font size (of text appearing as labels in the curve window): The value is specified in percent. The size denotes the maximum diameter of the symbols. A size of 50% .. 200% is usually best.

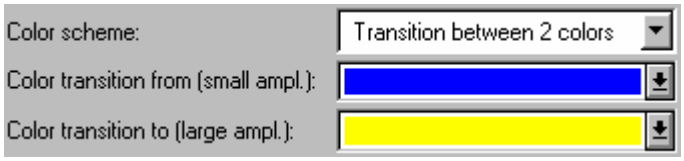
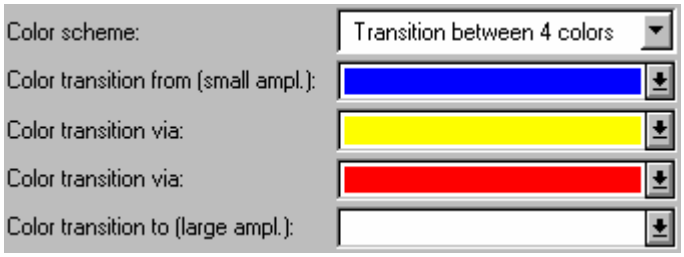
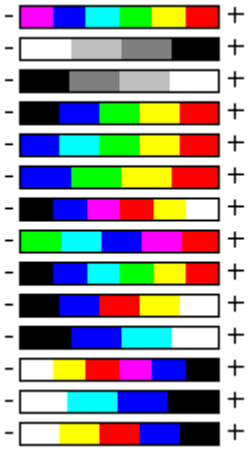
Color scheme

The options depend on what display type is selected.

The color scheme can be set for color map display with graduated or continuous color transition.

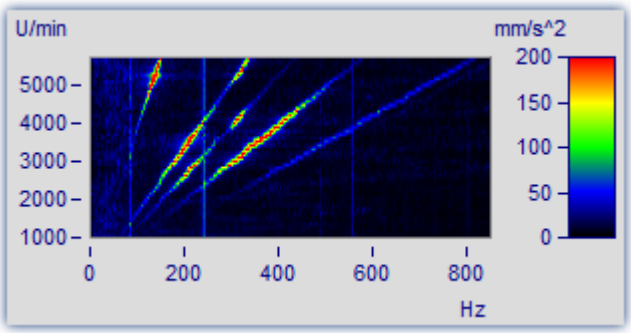
All color schemes (except *auto*) are independent of the capabilities of the output device.

Color scheme	Description
auto	For color output devices, the colors Magenta to Red are used; for b&w printers, the shades from White to Black.

Color scheme	Description
Transition between 2, 3 or 4 colors	Here you have a choice of using 2, 3 or 4 fixed colors. Then additional controls appear to let you select the particular colors.  *Dialog with 2 colors.*  *Dialog with 4 colors.* Between the fixed colors, any intermediate colors are generated by linear interpolation of the fixed colors. For instance, if color map display with 5 graduated colors is chosen, and a transition between two fixed colors is also set, the two colors at the ends of the scale are fixed and the other 3 are determined by interpolation. In the dialog, the color for the smallest amplitude is indicated first (above), and the color for the highest amplitude in the lowest position.
Fixed color schemes	These color schemes are available for direct selection. At left (where "-" appears), the color for the smallest amplitude is shown, at right ("+") the color for the highest. 

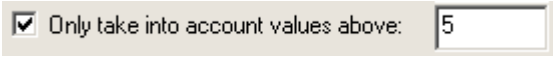
Only take into account values above

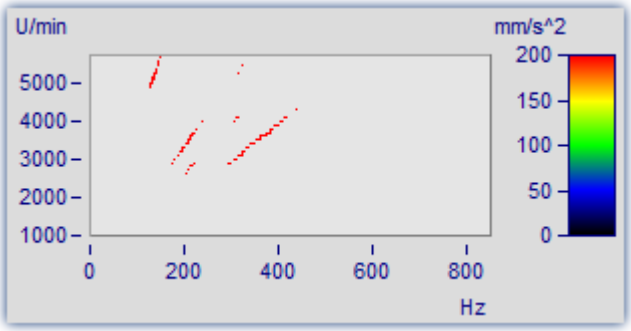
The options depend on what display type is selected.



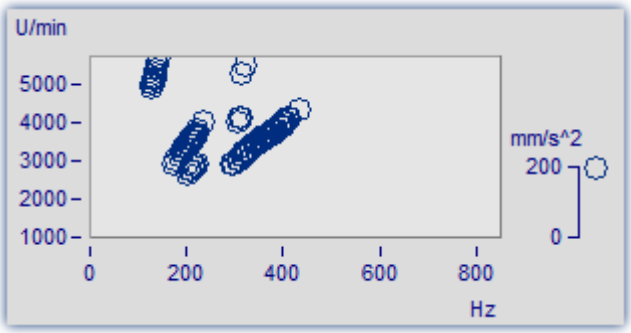
If this option box is not check-marked, all readings are taken into account (default).

If the checkbox is marked, then a lower limit for the amplitudes to be displayed can be specified.





Only relatively large reading values are given consideration. All others merge into the background.

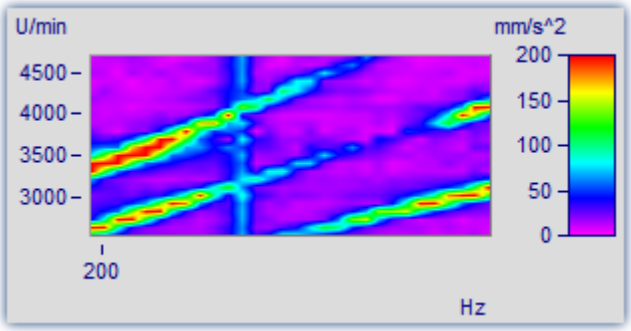
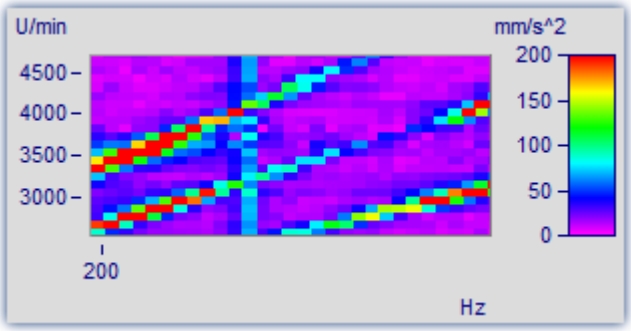
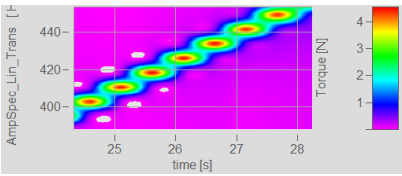


The same graph in the mode *Symbols*, *size corresponding to amplitude (Campbell)*

Interpolation between measured values

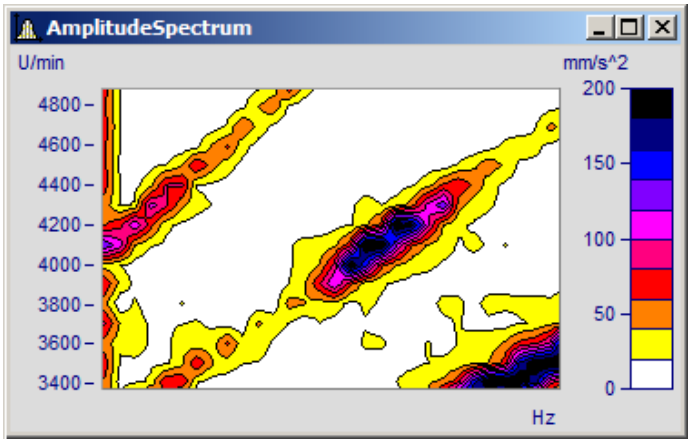
The options depend on what display type is selected.

This refers to interpolation in xy-plane between readings. Not to be confused with interpolation in the z-direction (interpolation of colors).

Interpolation	Description
auto, linear	In the x- and y-directions (horizontal and vertical on-screen, respectively) interpolated values between the specific readings are determined. This corresponds to linear interpolated lines between plotted points. 
Constant extension to upper right	In the x- or respectively, the y-direction, all measured values are extended at constant magnitude until the next measured value. This corresponds to stair-step or bar meter display of measured values. 
Interpolation, cubic polynomials	In x- and y-direction (horizontal and vertical on the screen respectively) interpolation is done between all measured values with cubic splines. 

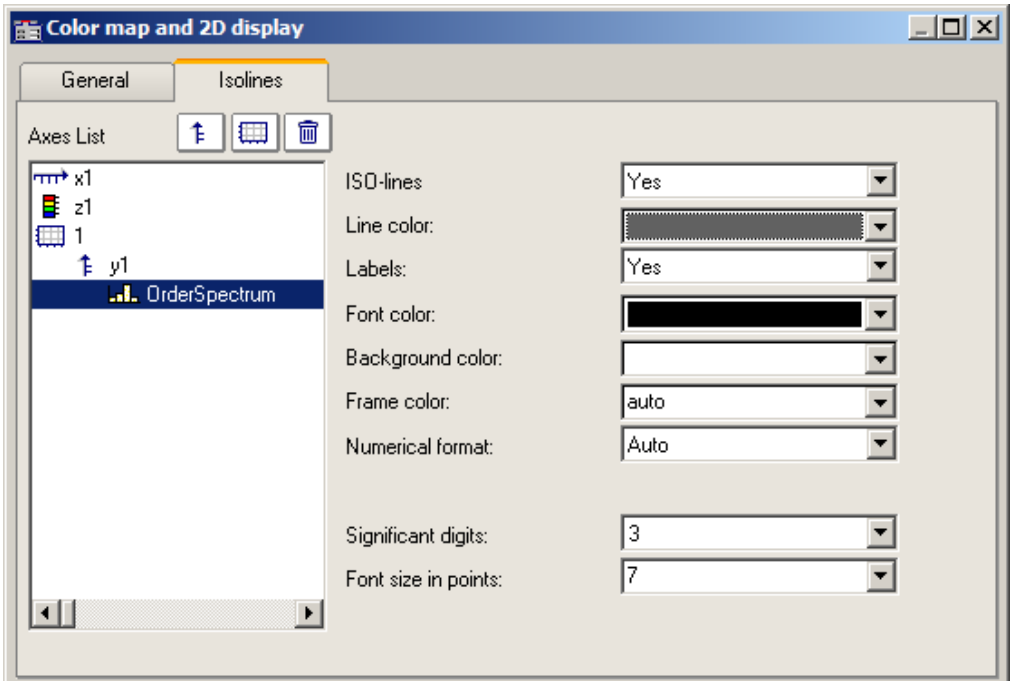
13.6.1.5.2 Color map dialog, ISO-lines

ISO-lines are contours, in other words, lines joining coordinates of equal height. They are used here in the same way as in maps. Since the lines appear in conjunction with graduated colors, they mark the boundaries between the colors.



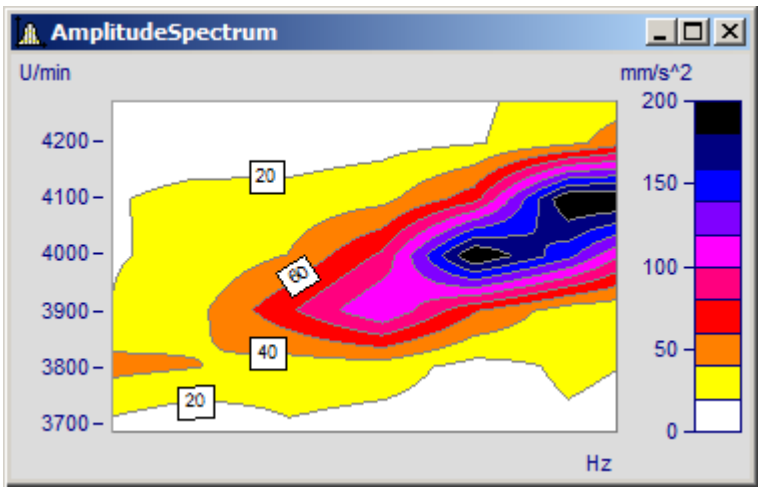
ISO-lines can only be set for *color map display with graduated colors* 1307.

The "*Isolines*" index card of the Color map settings dialog



The following options can be selected:

Options	Description
ISO-Line	Display of the isolines. This option is only available for the display types *Color maps, graduated colors*.
Line color	The color of the isolines is set here.

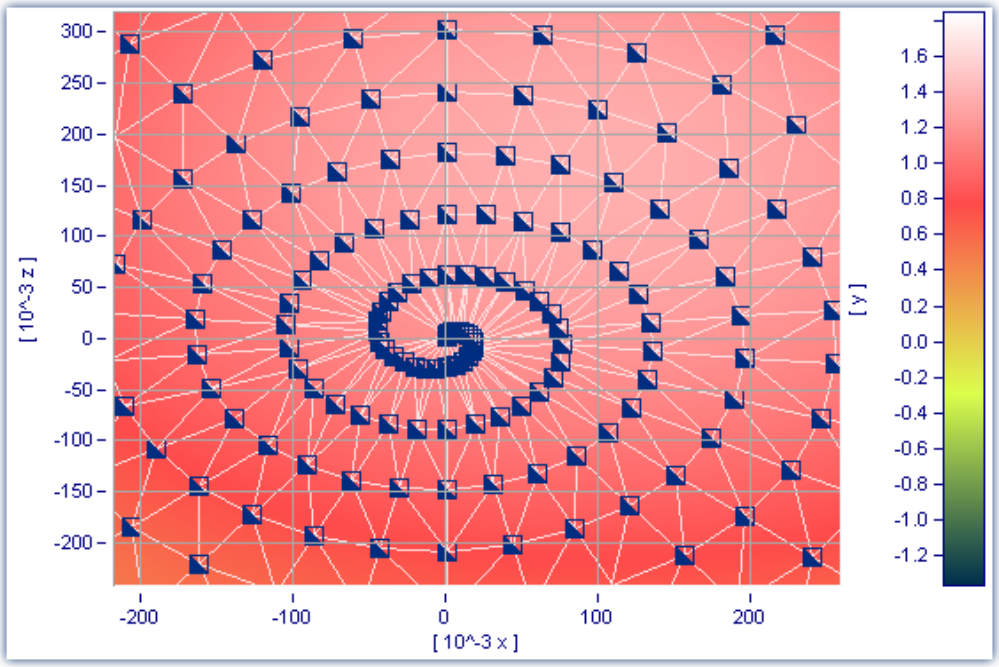
Options	Description
Labels	• *auto*, *no*: The isolines are not labeled. • *yes*: The isolines are labeled; small tags are attached to the lines, as space allows.  All further options only pertain to the labeling of the isolines.
Font color	Determines the color of the label text.
Background color	Determines the color of the label background. • *auto*: Transparent background, but covering the contour line. • *transparent*: The label background is totally transparent. The contour line between the colors is visible as well. • *Fixed color*: The label background is given a fixed color and is thus not transparent.
Numerical format	• *Fixed point*: Fixed point with a fixed number of decimal places to the right of the point e.g. 17, 17.35, -0.0017. • *Floating point*: Floating point with a fixed relative precision and exponent, e.g. -1.28E-7 or 3.4E+0. • *auto*: Automatic selection, partially dependent on the numerical representation of the z-axis.
Unit prefix	This consists of the order of magnitude and, if appropriate, a prefix such as kilo, milli, Mega or even just 1 (default, use of unit without prefix), which appears after the number before the unit. If the reading is 10 and takes the unit V (Volts), then for a selection of "m" (milli), a value of 10000mV is returned. For *automatic numerical format*, the same order of magnitude is set as for the labeling of the z-axis.
Places right of point	Number of decimal places for *Numerical format Fixed point* or *Floating point*: 0 to 15.
Valid digits	For *numerical format auto* the number of decimal places cannot be set, instead, the number of significant digits. Thus, the numbers 3.4, 3.4E-4 and 0.034 each have 2 significant digits.
Font size in points	The font size in points (pt): 4pt to 10pt. Typical values are 6 and 8 points.

Reference Waterfall Diagrams

See chapter Waterfall Diagrams 1277

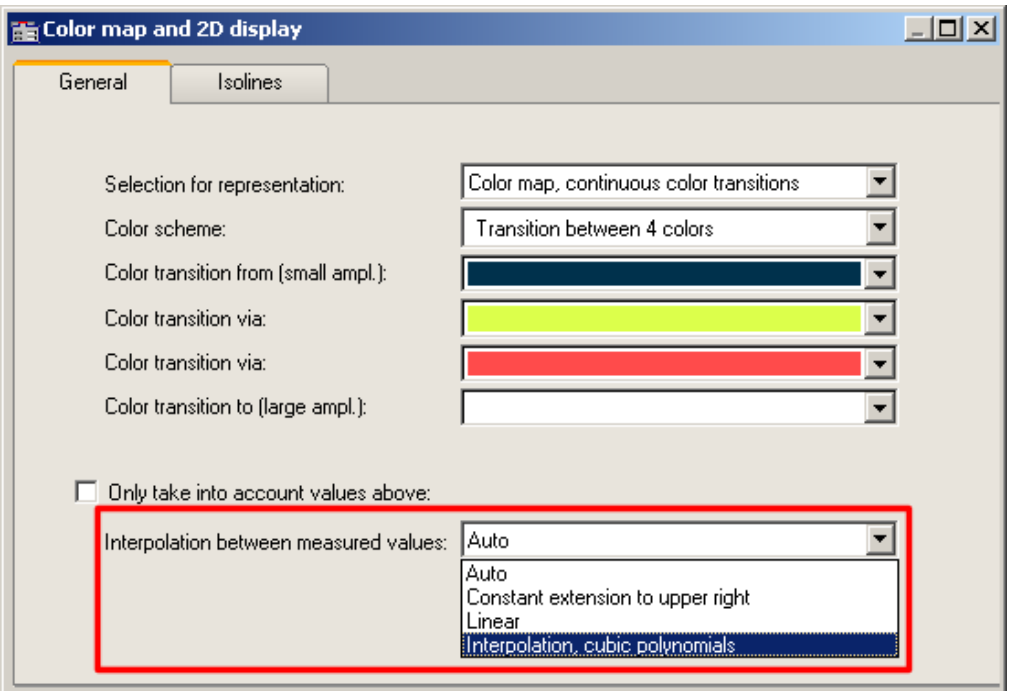
13.6.1.5.3 Waveforms with associated x-,y-,z-variables

A an alternative to 3D-display, waveforms with associated variables can also be displayed in a color map.



Color map display of a data set with multiple dimensions

It is also possible to specify interpolation between measured data points for this display style. To do this, open the color map's Properties dialog and on the *General* page select an interpolation type from the dropdown-list.



Interpolation between measured values in a color map

13.6.1.6 3D display

Another display style for datasets with associations of x, y, and z variables 1346, for segmented waveforms is *3D* display.

In this display style, the perspective can be freely rotated, and the Axis Navigation Bar 1516 can be used to change the view. It is possible to display more than one dataset with associations of x, y and z variables. This makes **comparison** of multiple superimposed 3D data sets possible.

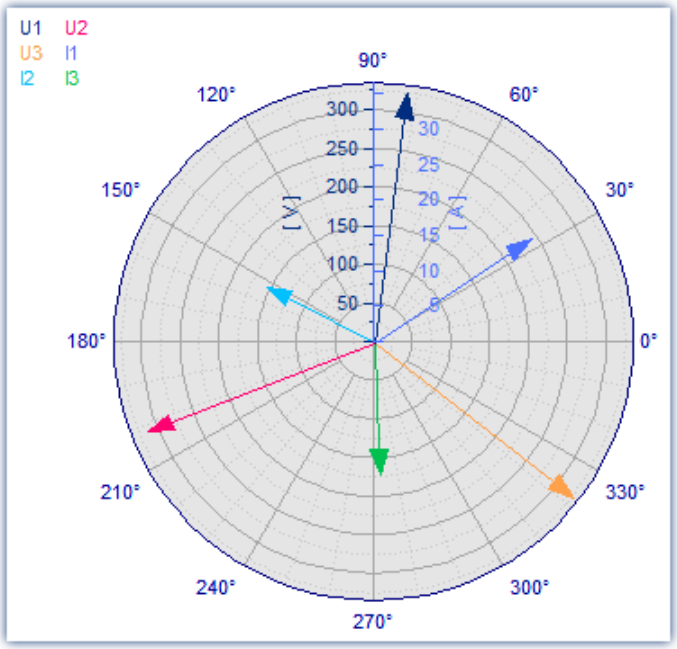
Other curve window functions such as *zooming*, *setting markers* or *line-shifting* are not available here.

For more on using 3D-display, see the section on *Configuration / 3D* 1404.

13.6.1.7 Polar plot

Complex data sets can be displayed as a polar diagram. To do this, select from the menu *Configuration\Display* the the display type *Polar plot*.

Example: Pointer diagram



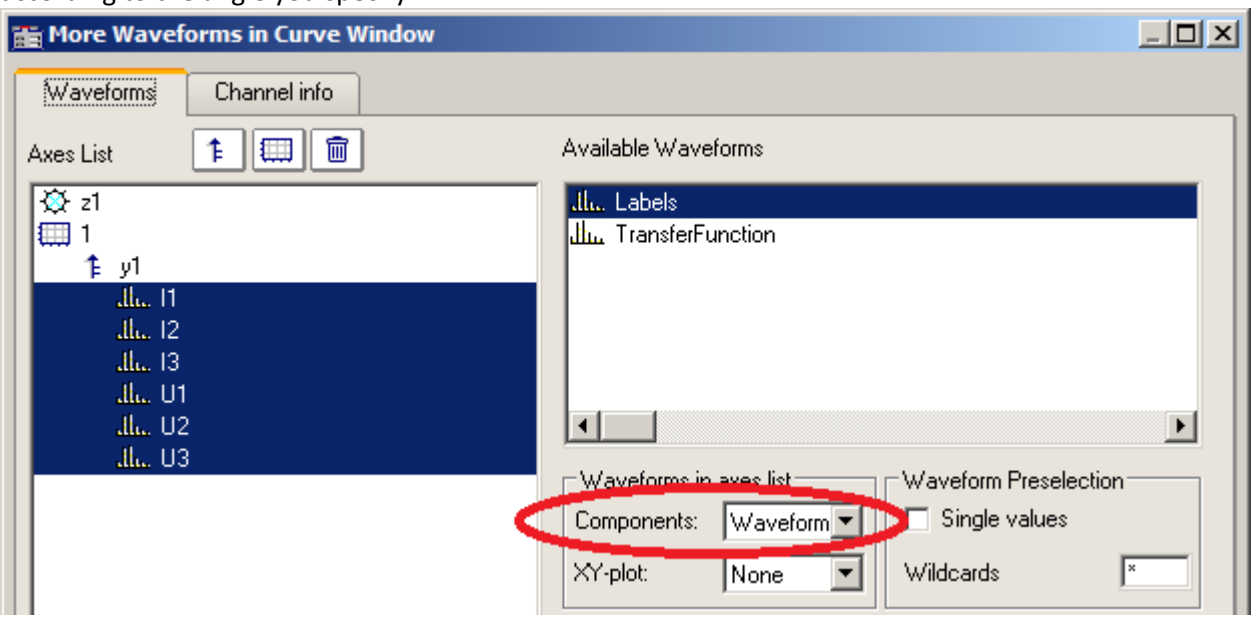
Complex single values as vector diagram

In imc FAMOS you can create a complex single value using the function Cmpl().

The pointer graph above is created as follows:

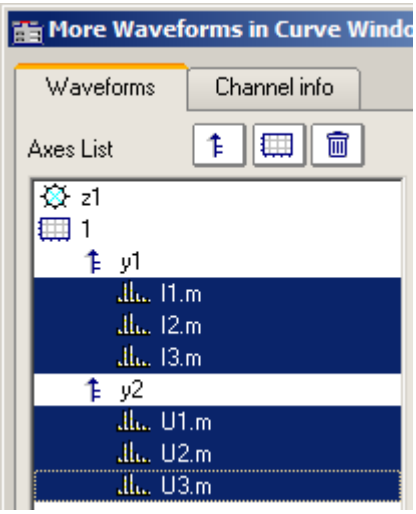
1. Create the desired single values:
   I1= Compl (27'A', 33'Degr')
   I2= Compl (17'A', 153'Degr')
   I3= Compl (19'A', 273'Degr')
   U1= Compl (327'V', 82'Degr')
   U2= Compl (315'V', 202'Degr')
   U3= Compl (331'V', 322'Degr')
2. Display the single values in a curve window and select the display style *Polar diagram* in the menu *Configuration\Display.*

3. By default, the curve window only shows the magnitude component of a variable. For this reason, all arrows point upward. With the right mouse button, open the context menu and there select *More Channels...*. Select the variables and select *Waveforms in axes list\Components: Waveform.* The arrows will then be drawn according to the angle you specify.

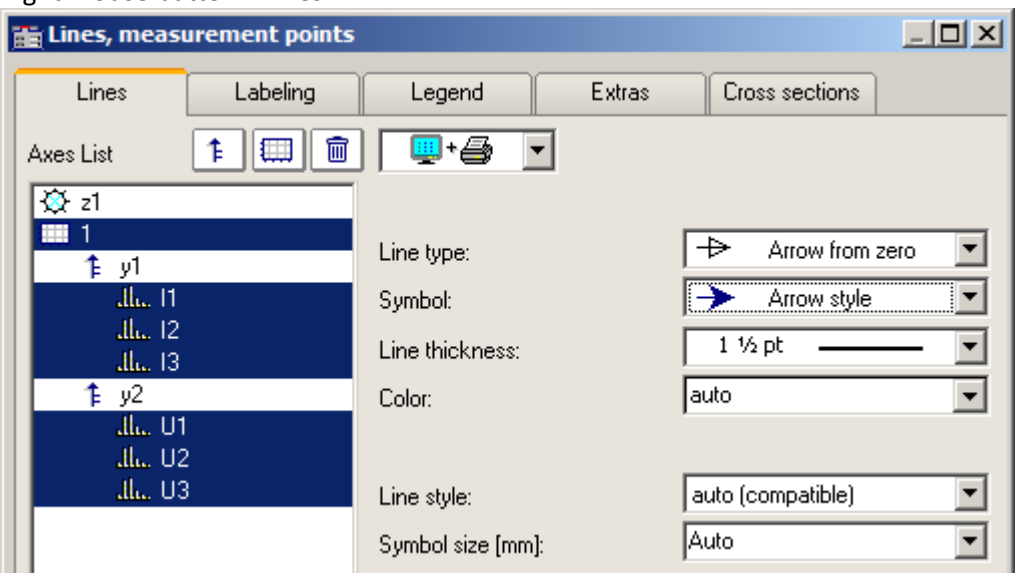


Observing each component of a complex waveform

4. As in a normal curve window, all variables are assigned to an axis. The current which are comparatively low in magnitude therefore appear in the zero point. Create one axis each for current and voltage: right mouse button for context menu -> "*More Channels*"

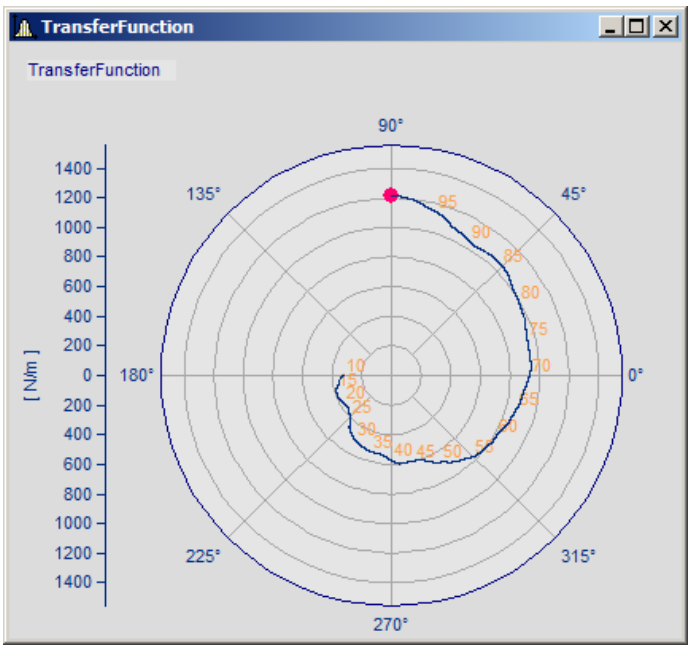


5. Using the Lines dialog, you can now adapt the display of the arrows.
   Right mouse button: "*Lines*"



Dialog lines: Display with arrows

Example: Transfer function



Example transfer function

There are further possibilities for displaying a complex data set:

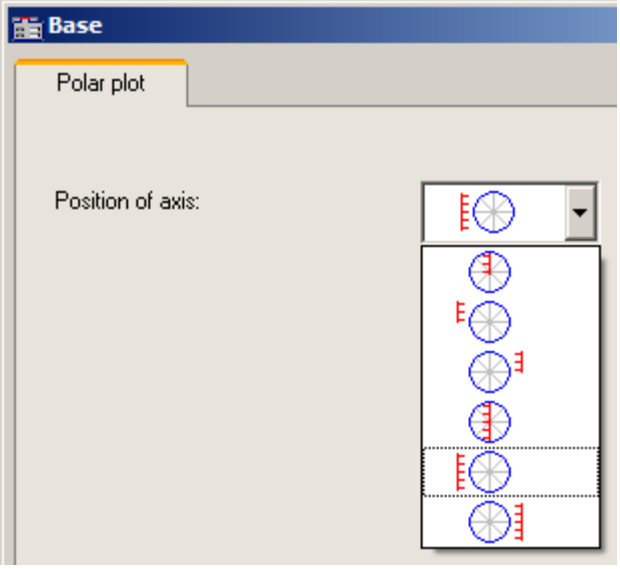
1. Use imc FAMOS to create a complex data set:

```
; transfer function diagram
t = ramp ( 0, 1, 11000 )
m = t * 0.1 + 200 + 1000*smo ( Random(leng?(t), 2, 0, 0, 3), 1000 )
p = t * 0.03 +  150
m = xoff ( xdel ( CutIndex ( m, 1001, 10000 ), 0.01 ), 10 )
p = xoff ( xdel ( CutIndex ( p, 1001, 10000 ), 0.01 ), 10 )
yUnit p Degr
yUnit m N/m
xUnit p Hz
xUnit m Hz
TransferFunction = compl ( m, p )
Labels = red ( TransferFunction, 500 )
```

   Important Note: The angle specification distinguishes between degrees and RAD as the unit. If no unit is specified, then RAD is applied, meaning 2PI instead of 360°. For this reason, it is important to use yUnit to set the unit as Degr, as shown in the example.

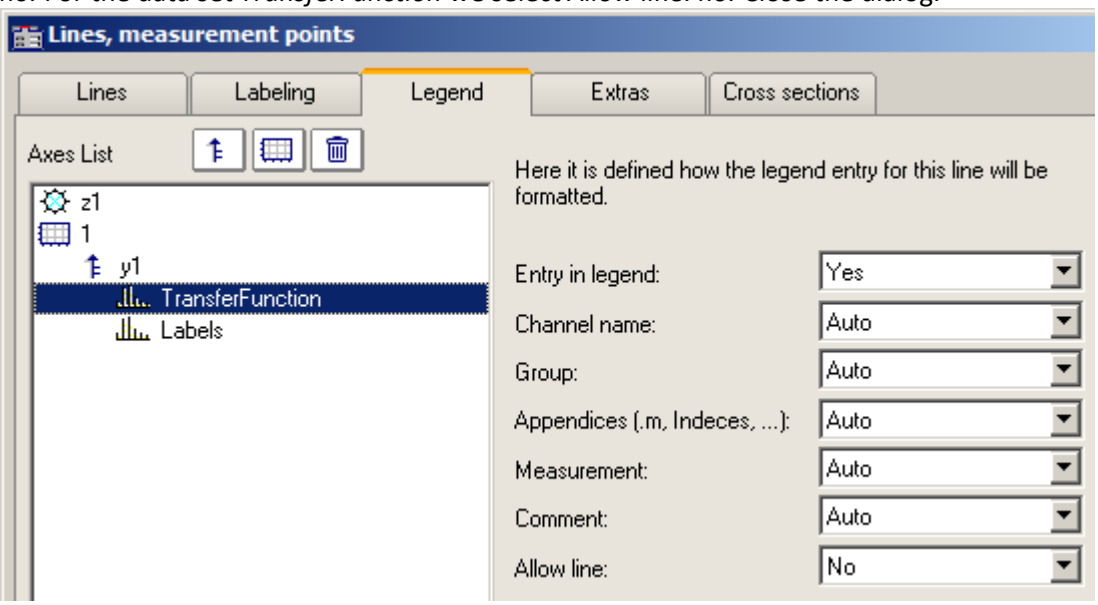
2. Display the data set *TransferFunction* in a curve window and select the display style *Polar plot* in the menu*Configuration\Display.*
3. Right-click the mouse to open the context menu and there select *More Channels...*. Select the variable and under *Waveforms in axes list\Components: Waveform.* The angle data is now reflected in the display.

4. **Axes position:** By default, the Y-axis is displayed only in the positive direction. You can change both the axis' position and its range: Right mouse click: *Polar plot*.



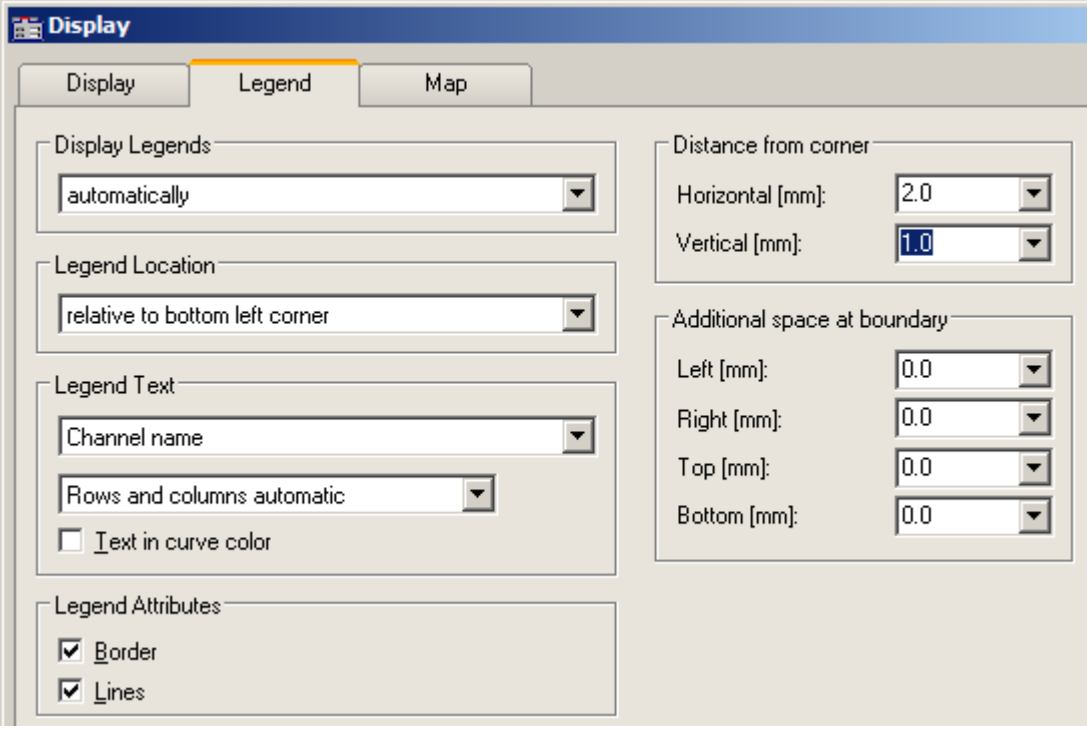
Position of axis

5. **Labeling the values:** The individual values could be labeled in any curve window with their values. However, in this example the amount of data would make that impossible to read. For this reason, the data set *Labels* was created, which is an image of *TransferFunction* reduced by a factor of 500.
   Select the right mouse button: *More Channels...* and assign *Labels* to the same axis.
   Click the button *Topic* and select *Lines*.
   Go to the page *Labeling* and activate it.
   By default, *Labeling* displays the magnitude. In our example we wish to display the frequency. For this reason, select for *Value selection* the entry *Parameter*.
6. **Legend:** The legend's position can be moved. First of all, the data set *Labels* is not to appear in the legend. For this reason, in the *Lines dialog* on the page *Legend* we select the data set *Labels* and select *Entry in legend: no.* For the data set *TransferFunction* we select *Allow line: no.* Close the dialog.



Legend display on/off for each channel separately

7. Open the dialog *Configuration \ Legend.* Position the legend in the upper left corner at some distance from the edge.

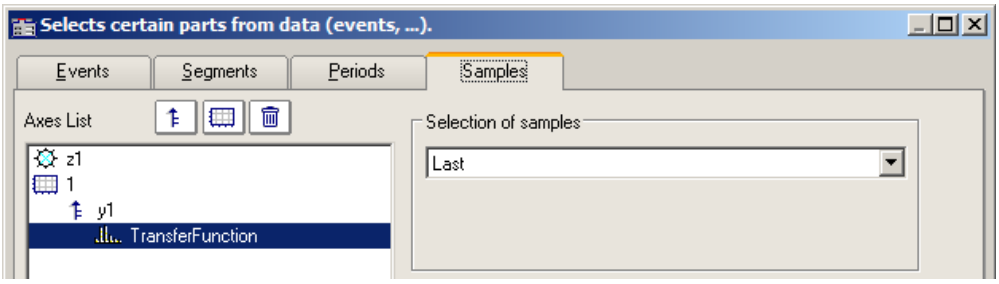


Position of legend

Note — Notes on online display

For measurements with imc CANSAS, the measurement device does not generate complex data sets. For instance, with an online FFT, the magnitude and phase are each transferred as separate channels. These can be combined to form complex data sets in the curve window under *More Channels...* 1346.

If one wishes to display the complex data as a pointer diagram 1315 like above, then besides the procedure outlined above the following step is also necessary: Since only the channel's current value, in other words, the last one, is to be displayed, select *Configuration \ Events, segments periods...*Go to the page *Samples* and select *Last* under *Selection of samples*.



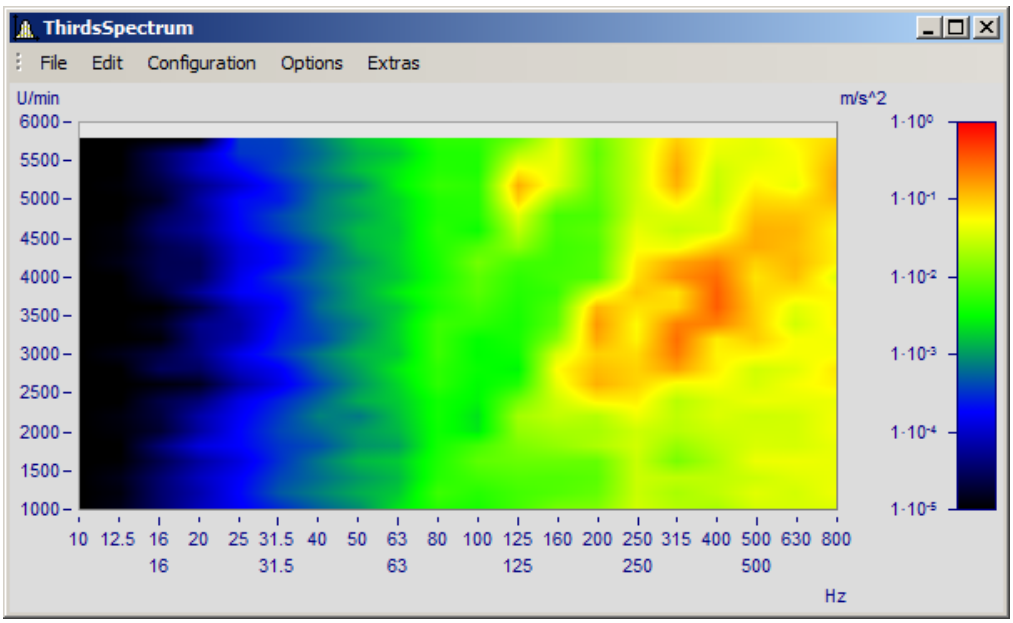
Displaying only the last sample

13.6.1.8 Third-octave display

Certain methods and forms of representation in the analysis of sounds and vibrations have gained widespread acceptance in industry. 1/3-octave, octave and narrow-band analysis require mathematical operations to be performed, in which the spectra in question are computed from the time signals of vibration pick-ups. The x-axis of the coordinate system is labeled according to the selected frequency bands; the frequencies themselves are determined by industry standards.

Example

Using the mathematical functions in imc FAMOS, the 1/3-octave spectrum was calculated over time. The following color map diagram displays the result, with the x-axis labeled in 1/3-octaves and octaves.



Description

The following nominal pass ranges apply to octave filters. Higher and lower values can be computed by taking the first ten values from the list and multiplying them by ten-to-the-power-of any multiple of three.

Octaves

Center frequency (Hz)	Lower limit (Hz)	Upper limit (Hz)
16	11.2	22.4
31.5	22.4	45
63	45	90
125	90	180
250	180	355
500	355	710
1000	710	1400
2000	1400	2800
4000	2800	5600
8000	5600	11200
16000	11200	22400

The following frequencies apply to 1/3-octaves. Higher and lower values can be computed by taking the first ten values from the list and multiplying them by ten raised to an integer exponent.

1/3-octaves

Center frequency (Hz)	Lower limit (Hz)	Upper limit (Hz)
1000	900	1120
1250	1120	1400
1600	1400	1800
2000	1800	2240
2500	2240	2800
3150	2800	3550
4000	3550	4500
5000	4500	5600
6300	5600	7100
8000	7100	9000
10000	9000	11200

The center frequencies of 1/12- and 1/24-octave bands are located at the center frequencies of the 1/3-octaves and at the intermediate values located at logarithmically equal distances. Here the edges of the 1/3-octave range are used as additional frequency check points.

1/12 octaves

Center frequency (Hz)	Lower limit (Hz)	Upper limit (Hz)
1000	974	1029
1058	1029	1089
1120	1089	1151
1183	1151	1216
1250	1216	1286
1323	1286	1361
1400	1361	1448
1497	1448	1547
..	..	..

1/24 octaves

Center frequency (Hz)	Lower limit (Hz)	Upper limit (Hz)
1000	987	1014
1029	1014	1043
1058	1043	1073
1089	1073	1104
1120	1104	1135
1151	1135	1167
1183	1167	1200
1216	1200	1233
1250	1233	1268
1286	1268	1304
1323	1304	1342
..	..	..

© 2023 imc Test & Measurement GmbH

When waveforms in the frequency range are displayed as the result of a frequency-band analysis, the program expects the x-axis to have a certain kind of scaling. Since the frequency bands are spaced logarithmically, the x-axis is labeled with the logarithm of the frequency to correspond to the mathematical functions. The logarithm is then expanded again for display, and the frequencies are marked along the axis, according to the German industry standard. The following table illustrates by a few examples the relation between the x-scaling of the data and the frequency bands, based on the following rule: the $\log_{10}$ of the center frequency is multiplied by ten and then this value is rounded.

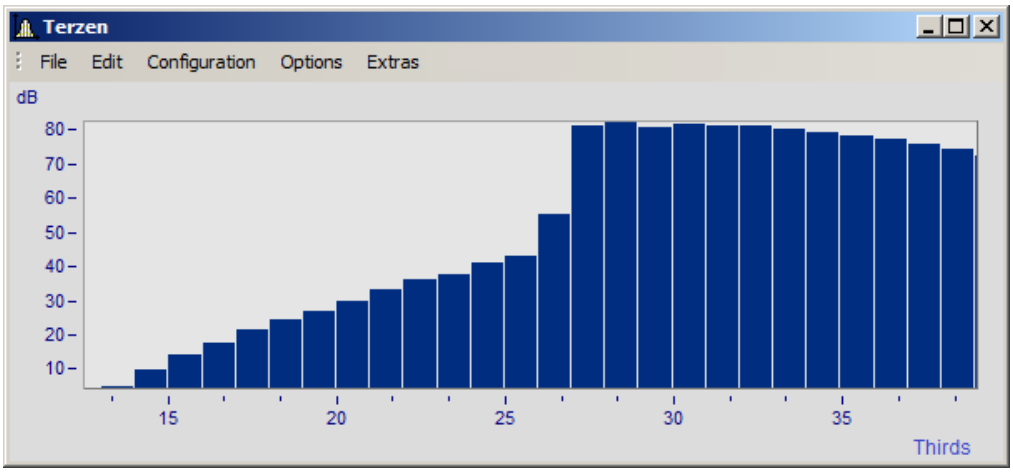
x-scaling	Center frequency (Hz)
..	..
-3	0.5
-2	0.63
-1	0.8
0	1
1	1.25
2	1.6
3	2
4	2.5
5	3.15
6	4
7	5
8	6.3
9	8
10	10
11	12.5
..	..
20	100
30	1000
40	10000
41	12500
43	20000
..	..

The 1/3-octaves are found at the x-positions 0, 1, 2..., the octaves at the positions 0, 3, 6, 9, 12..., 1/12-octaves at the positions 0, 0.25, 0.5, 0.75, 1, 1.25..., and 1/24-octaves are found at all multiples of 1/8.

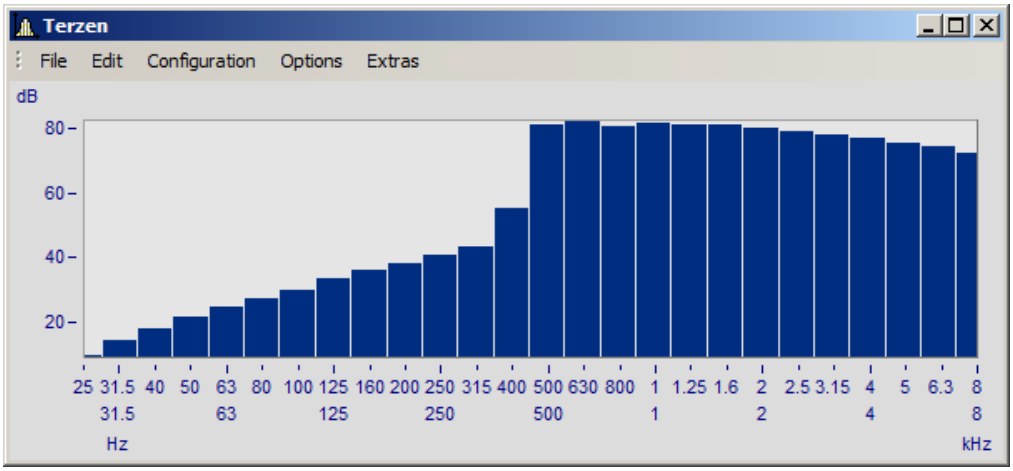
This means that the various bandwidths are expressed as a multiple of a 1/3-octave. Therefore, the delta-x of the x-axis is set to the following values for frequency scaling:

Bandwidth	Delta-X
Octave	3
1/3-octave	1
1/12 octave	0.25
1/24 octave	0.125

Example



A curve window with standard labeling

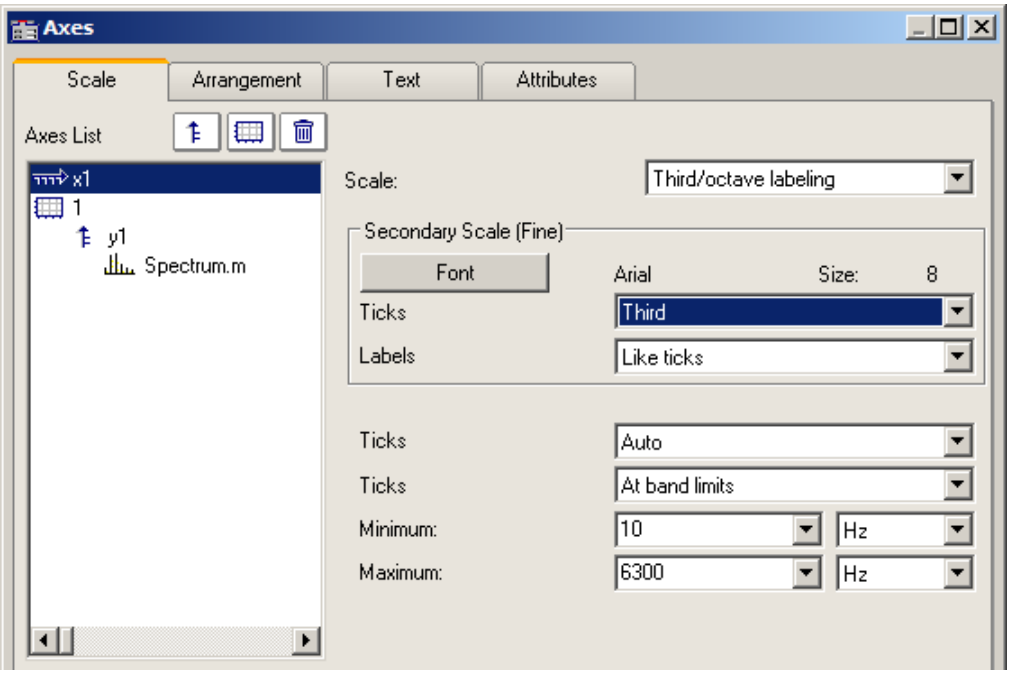


In 1/3-octave / octave labeling

Here, the connection between labeling with the logarithm and the expanded 1/3-octave scaling is apparent.

Mouse Operation

- Set the labeling of the x-axis in 1/3-octave / octaves in the *Axes* dialog called via the menu item *Configuration/ Axes*... menu. Alternatively, you can double-click on the x-axis labeling.

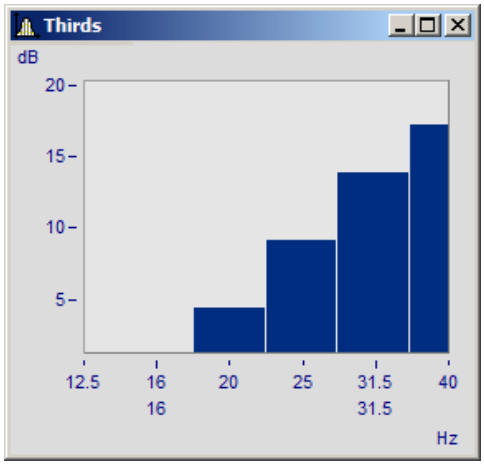
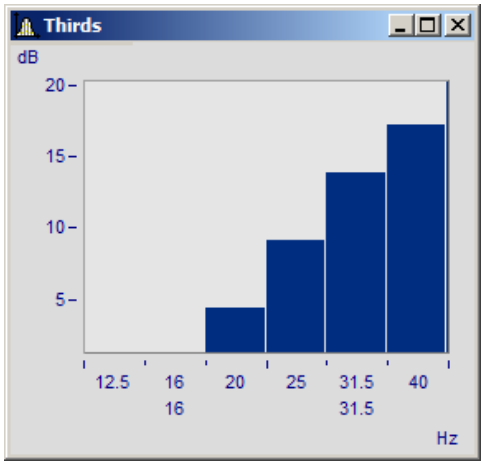


- In the list at left, select the x-axis. Then select under the control *Scale* the option *Third/ octave labeling*

General information

The x-axis scaling is organized into three horizontal layers. The upper layer consists of small ticks and labels (e.g. 1/3-octaves). The middle layer consists of large ticks and the standard font type for the curve window (large labels). The units (Hz, kHz...) are found in the lowest line.

Settings	Description
Secondary Scale (Fine): Font	Establish the font type for the small labels. For true-type fonts, the six-point size is generally quite legible for numbers on the screen.
Ticks	For the small and the large labels, the user can specify the interval between ticks on the axis. Choose from the following options: • Auto: Depending on the domain and the window size, one of the following options can be chosen. • Octaves • 1/3-octaves • 1/12-octaves • 1/24-octaves

Settings	Description
Ticks	Options for centering or aligning ticks. *Centered ticks* are placed exactly at the *center* frequency of the corresponding band, with labeling centered under the tick. The edges of the frequency bands are not visible. In the option *Ticks to edge*, the ticks are located exactly at the edges of the frequency bands, with labels centered between the ticks and at the center frequency.  *Ticks centered*  *At band limits*
Minimum, Maximum	The limits of the frequency range to be displayed are set using the combobox *Minimum* and *Maximum*. The contents of the list with numerical values are based on the previous selection of ticks for small labeling, which the program assumes is the highest resolution. The entries in the list boxes for *Minimum* and *Maximum* are always center frequencies of the band. If numerical values are entered using the keyboard, the precise center frequency will be automatically selected. The range of selectable numbers extends from 10µHz to 400THz, ensuring that the significant physical ranges are covered. Conventional applications will always lie within the range from 10Hz to 20kHz.

Remarks

- **Curves:** When a curve is displayed in a stair-step or bar graph display, the curve is <Shift>ed in the x-direction so that the center of the bars or steps are centered over the labels of the center frequency. The curve is moved to the left by one-half of the sampling time. This applies only to data sampled equidistantly, not to XY-displays.
- Note that in a "steps"-representation of a waveform, the sample-points of the conventional linear display would normally coincide with the corners of the steps. This is not the case for an x-axis labeled with frequency bands, where the sample point coincides with the center frequency, i.e. the midpoint of the step!
- When a small window is scaled over several decades, it is generally not possible to label the axis in 1/24 octaves. Select a coarser resolution in labeling, e.g. octaves.

13.6.1.9 XY-plots

Normally, curve windows display waveforms as time functions. This means the signal's values are indicated as y-coordinates plotted over time represented by the x-axis.

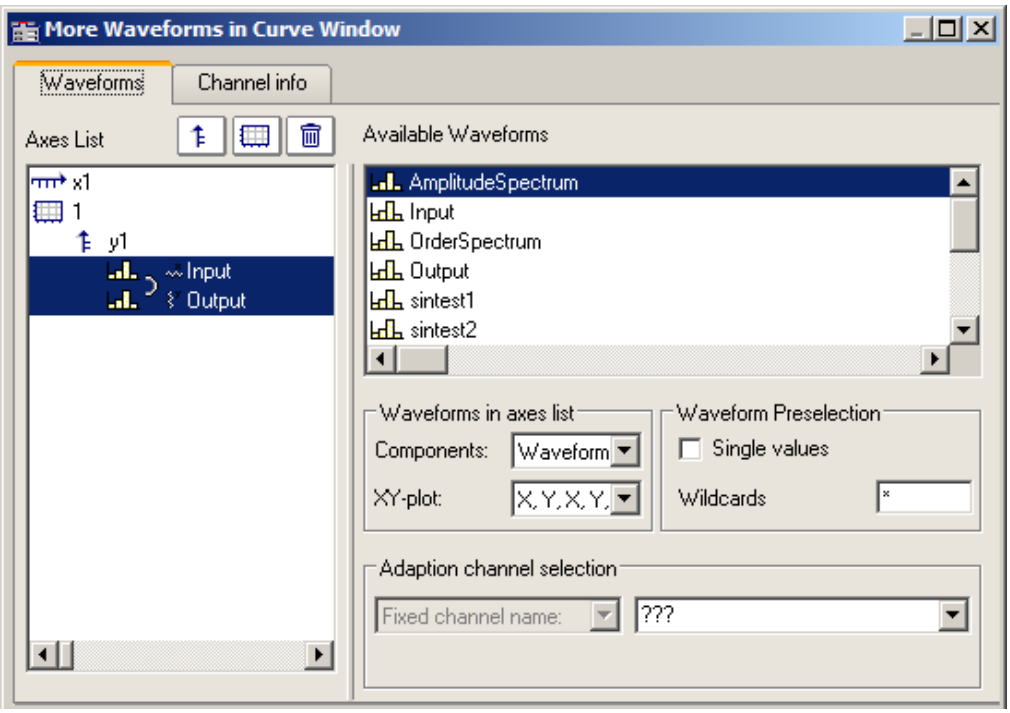
The curve window is also able to display XY-plots, meaning plotting one waveform's values in correlation to another waveform's values.

Examples of XY-plots include **Lissajous figures**, which consist of an XY-plot of sine shaped waveforms of differing frequencies and phases. **Hysteresis curves** for magnetic materials display the correlation between magnetic flux density and magnetic field intensity. **Characteristic curves** enable the assignment of an output variable value to a corresponding input variable value.

Polar plots are another type of XY data. Polar plots are used to display complex waveforms. This type of display can be used for a complex waveform in rectangular coordinates (real and imaginary part) or polar coordinates (magnitude and phase).

Operation

Select "*More Channels in Curve Window* 1339" (by right-clicking the mouse in the curve window). Stack the XY-plots vertically. Select the two waveforms and then in the box *Waveforms in axes list* Select the entry *x,y,x,y* from the drop-down list for *XY-plot:*. If you wish to change the axes, select instead the other order: *y,x,y,x.*



Remarks

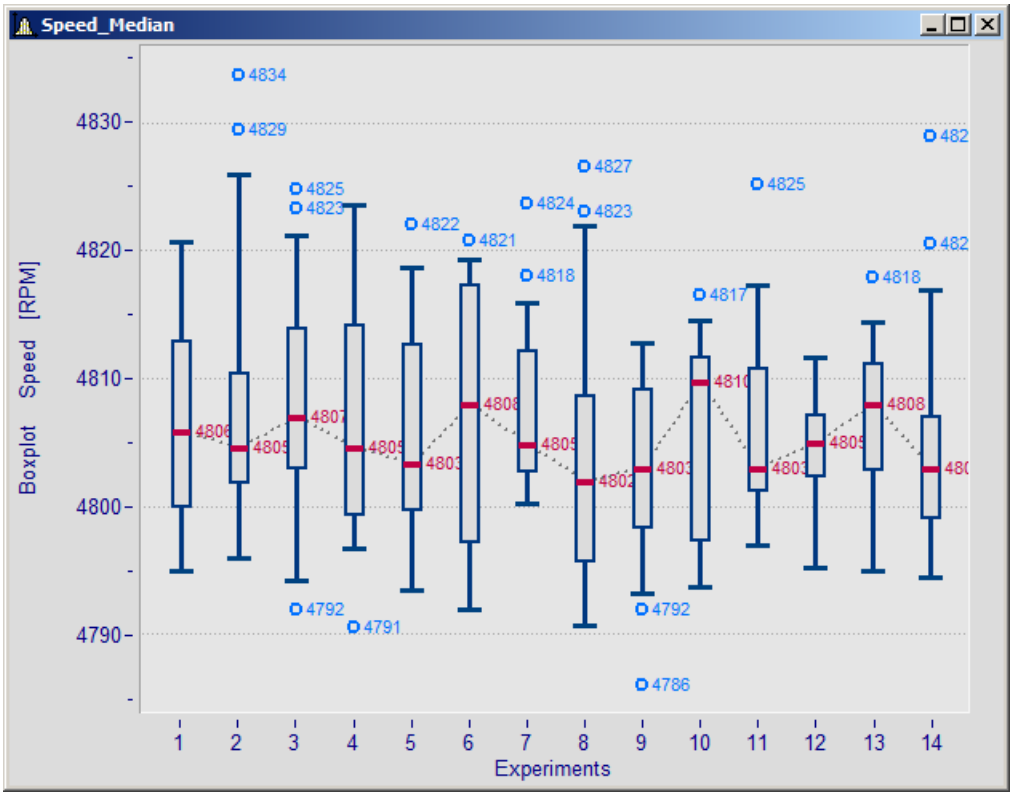
- In all XY-plots, the measurement cursors follow the parameters during operation in "Measure" mode. See chapter section 'Measure 1427'.
- Computing a polar plot requires less computation if the complex waveform is provided in rectangular coordinates. To increase the speed of the graphic display for a waveform in polar coordinates, transfer it into rectangular coordinates.
- The x- and y-values in an XY plot are paired *correctly* with respect to time, *not* point by point! Thus, the two waveforms used for an XY-plot do not have to have the same x-scale! Therefore x-offset and x-delta (sampling rate) are considered, but not the trigger time.
- If a waveform is already defined as complex or of XY-type, simply click on the curve window icon to display it.

13.6.1.10 Boxplot

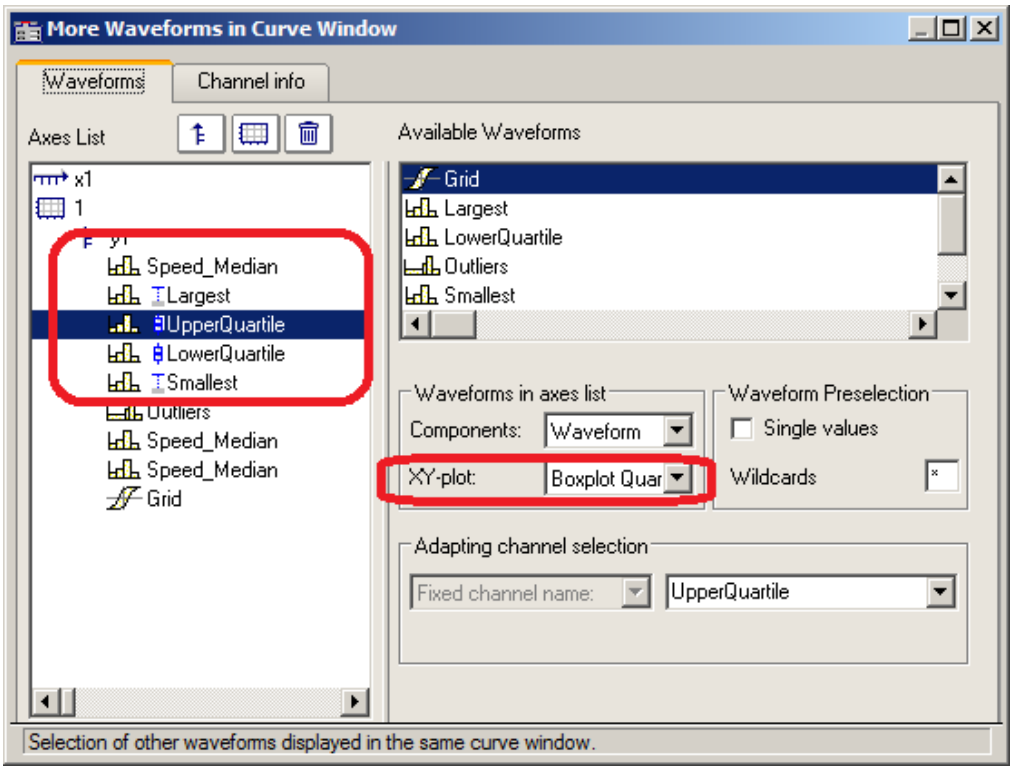
The Boxplot (also called Box Whisker Diagram) depicts a statistical distribution by displaying the minimum value, the maximum value, bottom quartile, top quartile and the median.

In this display, the minimum and maximum values are represented by whiskers, the quartile boundaries as a box, the median as a line.

For a curve window display, each of these statistics is saved in a separate channel, e.g. one channel with the median values, another channel of maximum values...

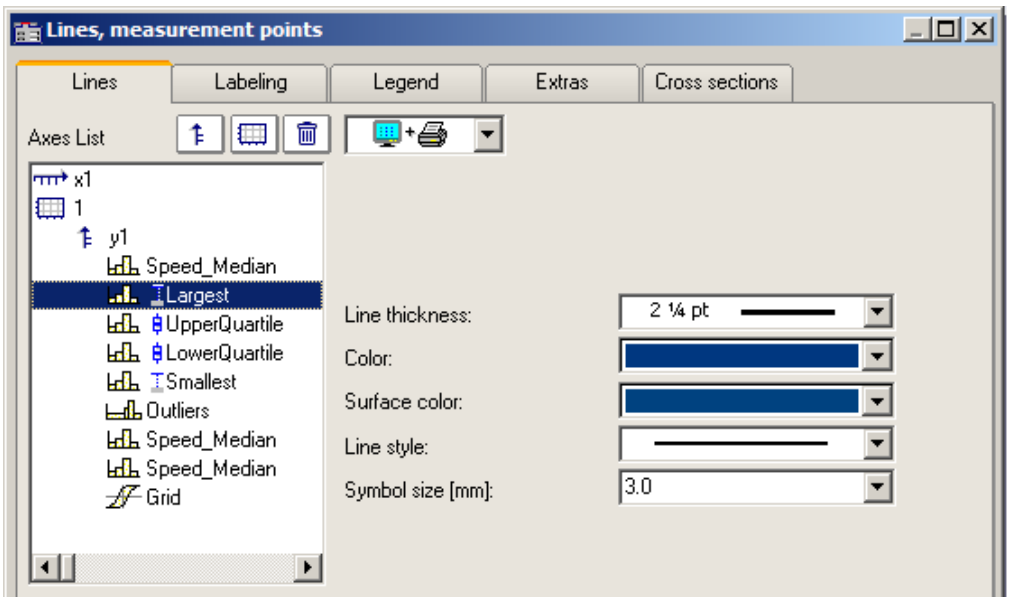


Making the settings:



In the dialog "*More Channels... in Curve Window*", the channel with the median value is initially added. For this channel, no special settings need to be made. Next, the channels for the whiskers and the quartiles are added. For each of these channels, the property *XY-plot* is set to "*Boxplot Whisker*" or "*Boxplot Quartile*", where it does not matter in this case whether a whisker is the top or bottom one; the curve window finds out on the basis of the numerical value. The order of the channels below is arbitrary. They do not even need to be complete, however, the median must always be the first channel.

Subsequently, the line properties are defined:



Depending on the line selected, different properties are available for selection on the right side. The line structure is only applied to the vertical whisker line. The surface color is used for filling the box representing the quartiles and for filling the whiskers' horizontal bar. Automatic color selection for the surface results in transparent fill of the quartiles' box. The color (line color) is used for connecting- and borderlines. Automatic color selection causes this color to be same as the median. Most properties can be set separately, however, for the 2nd quartile defined, only the fill color needs to be selected, since the rest has already been determined by the 1st quartile defined.

The display is performed point by data set point. All data sets must have the same sampling rate. The median itself may be an XY-channel, but all others may only be normal, equidistant data sets (it may be necessary to display the .y-component alone). In the case of an XY-channel, the sampling time of the y-component must match that of the other channels.

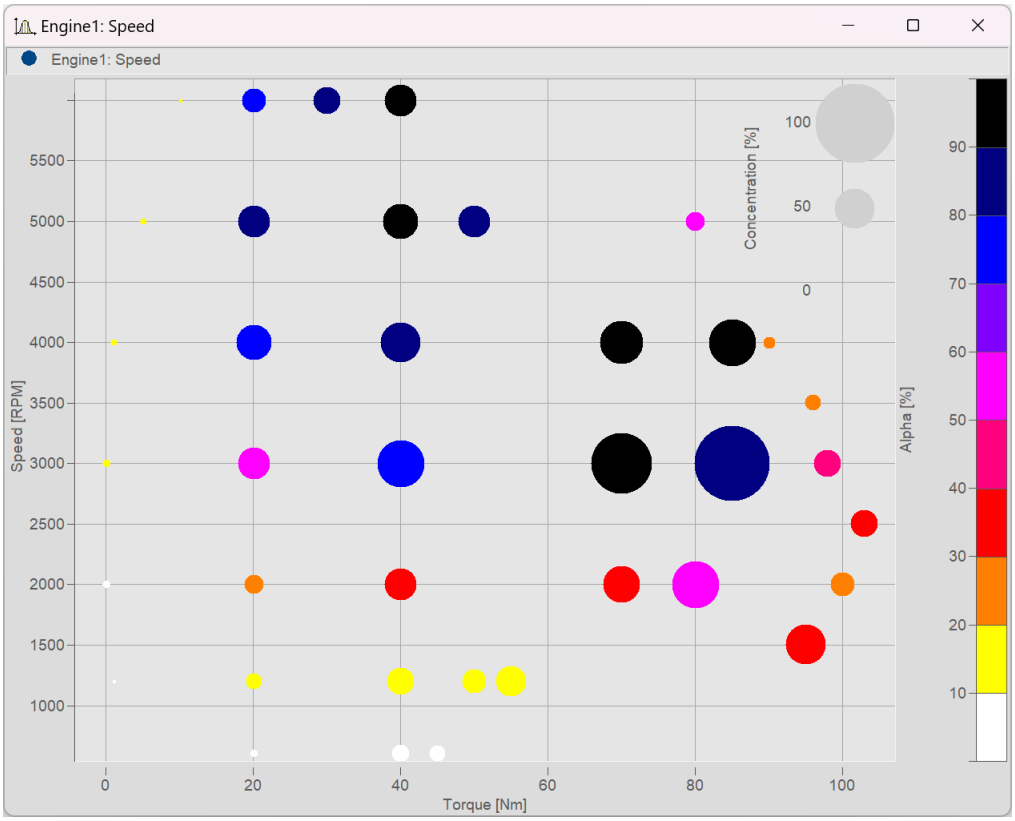
Remarks

- A combination with a color-determining channel is not possible.
- Measurement cursors and other curve window functions have only restricted functionality.
- To display any outliers, supplemental channels are used, for which the legend entry can be suppressed, for instance.

13.6.1.11 Bubble Plot

In a **Bubble Plot** (also called "Bubble Chart"), symbols are displayed which represent three (or optionally four) numerical values. Each point in the diagram is symbolized by a disk shape (bubble) in a typically Cartesian coordinate system. One data set (or in the case of XY, two data sets) determines the **position** of the bubble's center point. One data set determines its **size**, and optionally an additional data set, which could represent a kind of categorization of the data points, determines the **color** of the bubble.

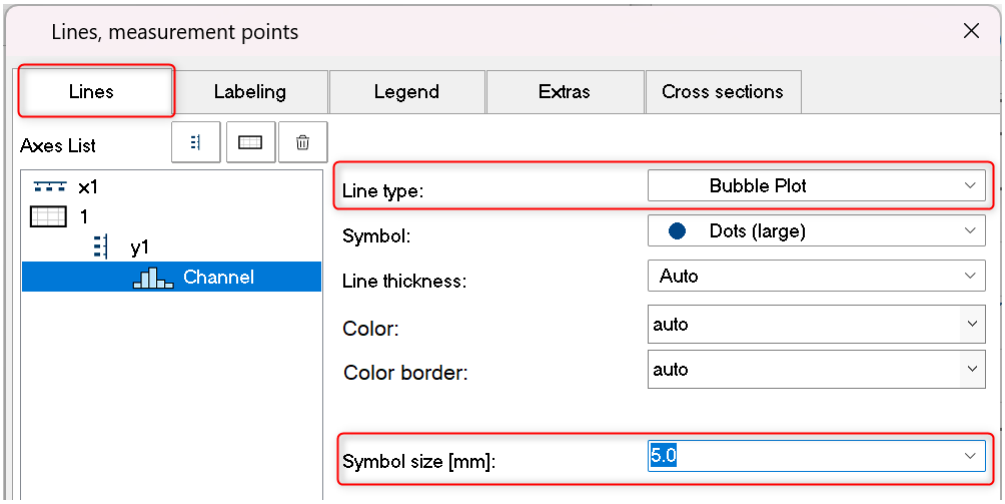
The **necessary condition** is that the **sampling rate** of all the data sets be identical.



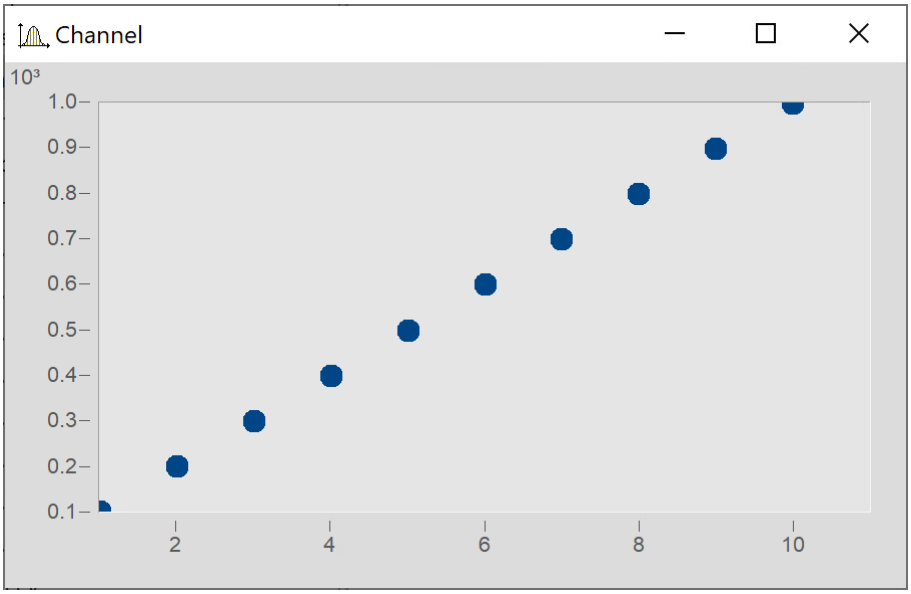
Bubble-plot example

Settings:

Within the **Display type** "*Default*", begin by setting the *Line type* to "*Bubble Plot*" in the settings under **Lines**. The initial size can be modified using *Symbol size [mm]*:

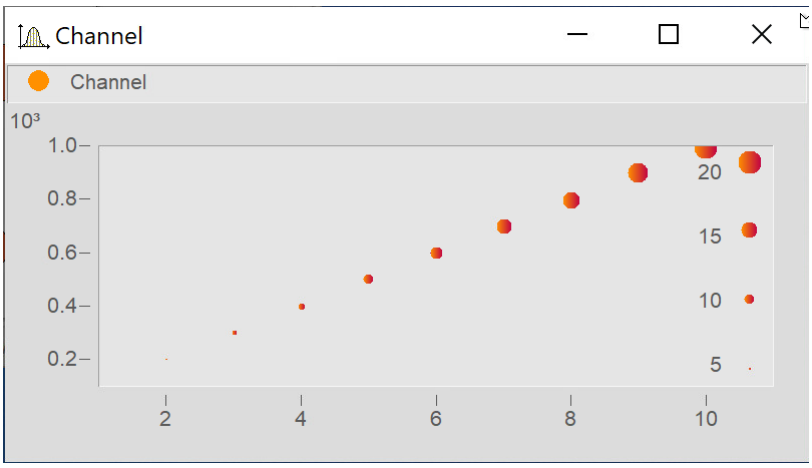


Depending on the *Symbol* selected, all values are displayed as *Dots (large)* etc. in the same size.

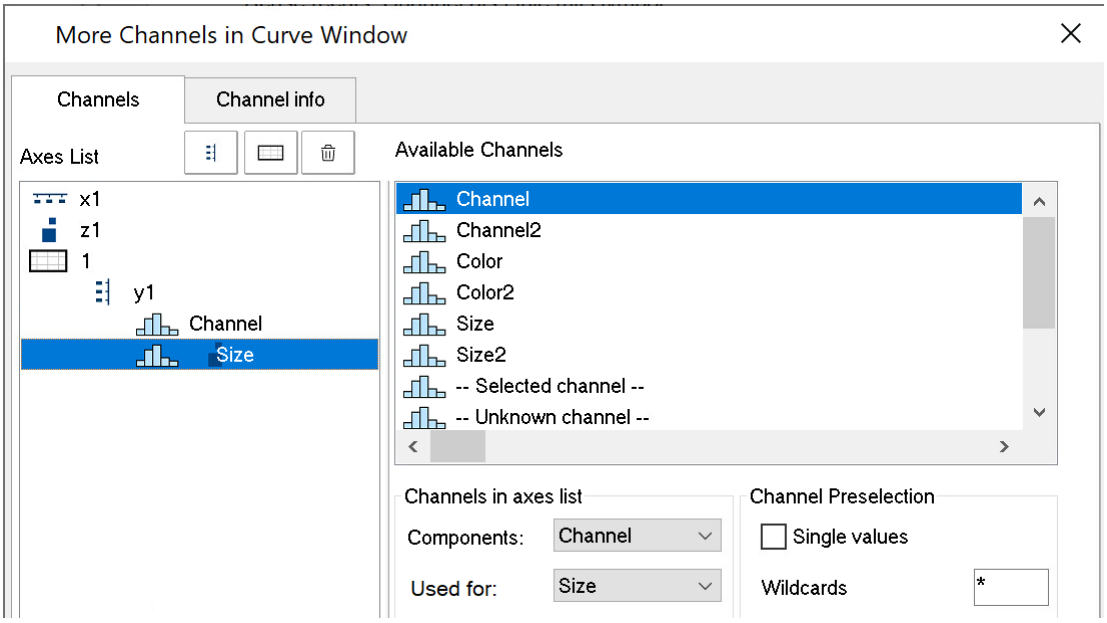


Size of bubbles corresponding to values of another data set

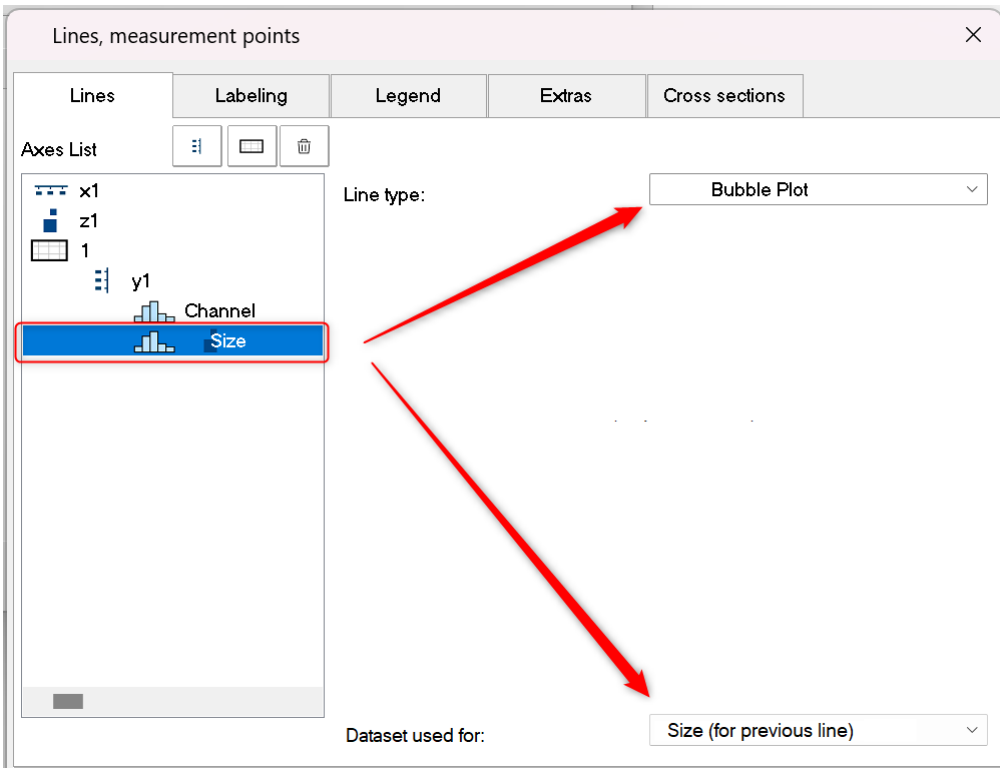
Next, the variable "Size" is intended to determine the size of the bubbles, as illustrated below:



In the dialog **More Channels in Curve Window...** add the variable "Size".



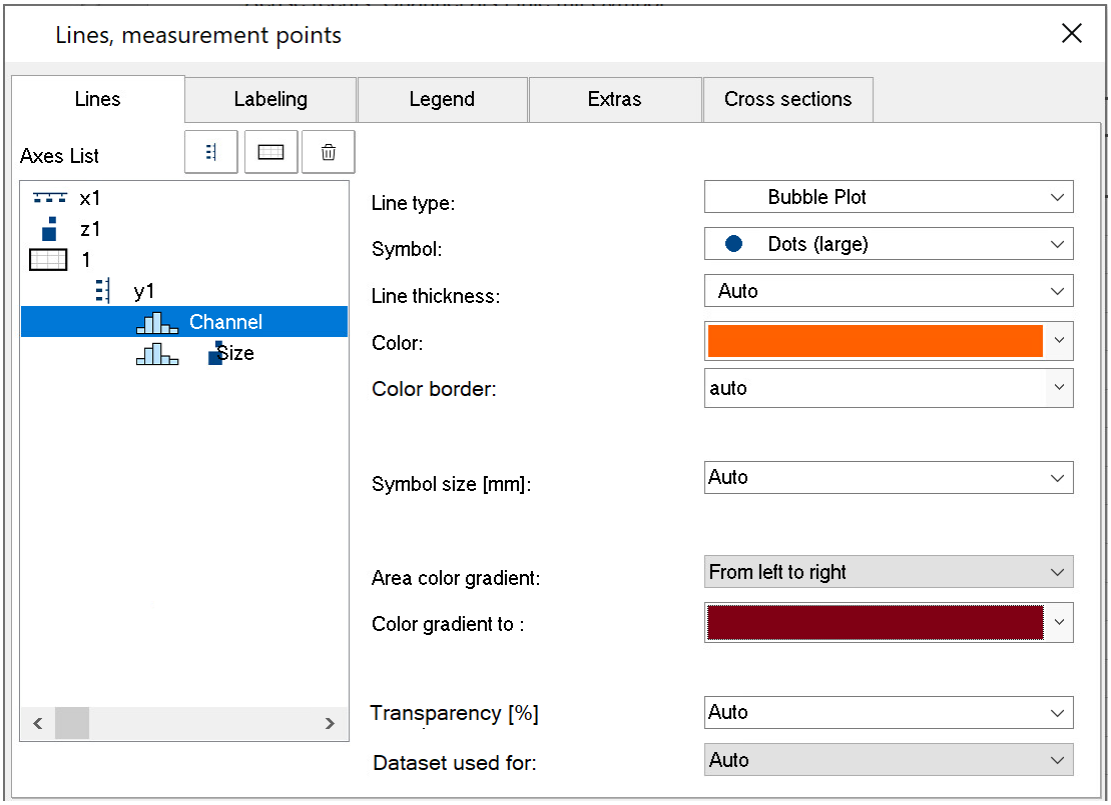
Go to the dialog **Lines** by clicking on the button *Topic* and set the *Line type* for the variable "Size" also to *Bubble Plot.*
Next, under *Dataset used for:* select *Size (for previous line)*:



Next, in the dialog **More Channels...** set *Used for:* to *Size*

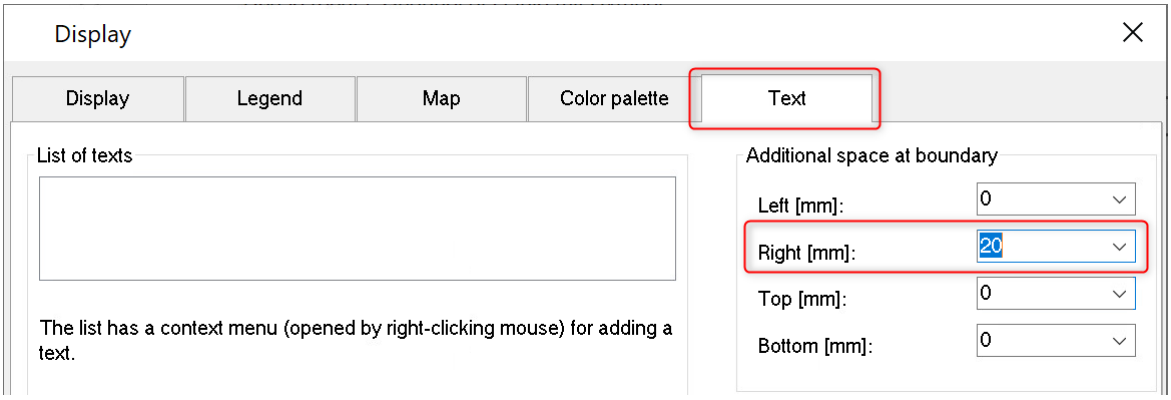
Bubbles with color gradient

Coloring the symbols with a color gradient:

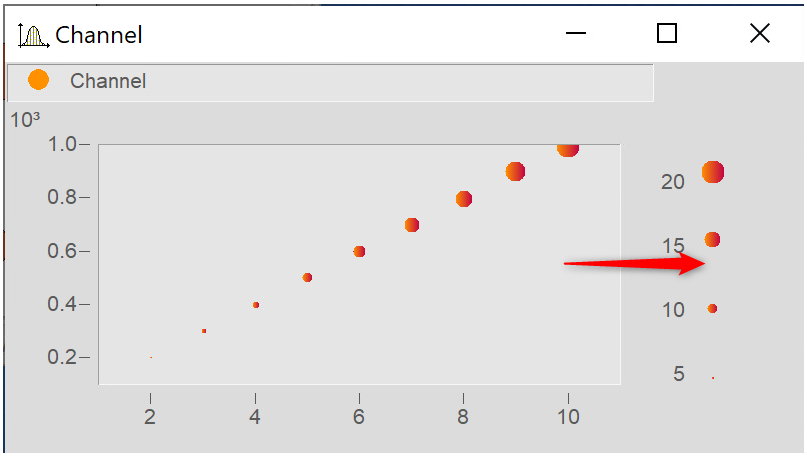


Adjusting the boundary

By default, the scaling axis for the variable is located within the curve window. In order to position it on the outside, it is necessary to first expand the boundary. On the tab *Text* in the dialog for the menu item **Display**, enter a right-side boundary:

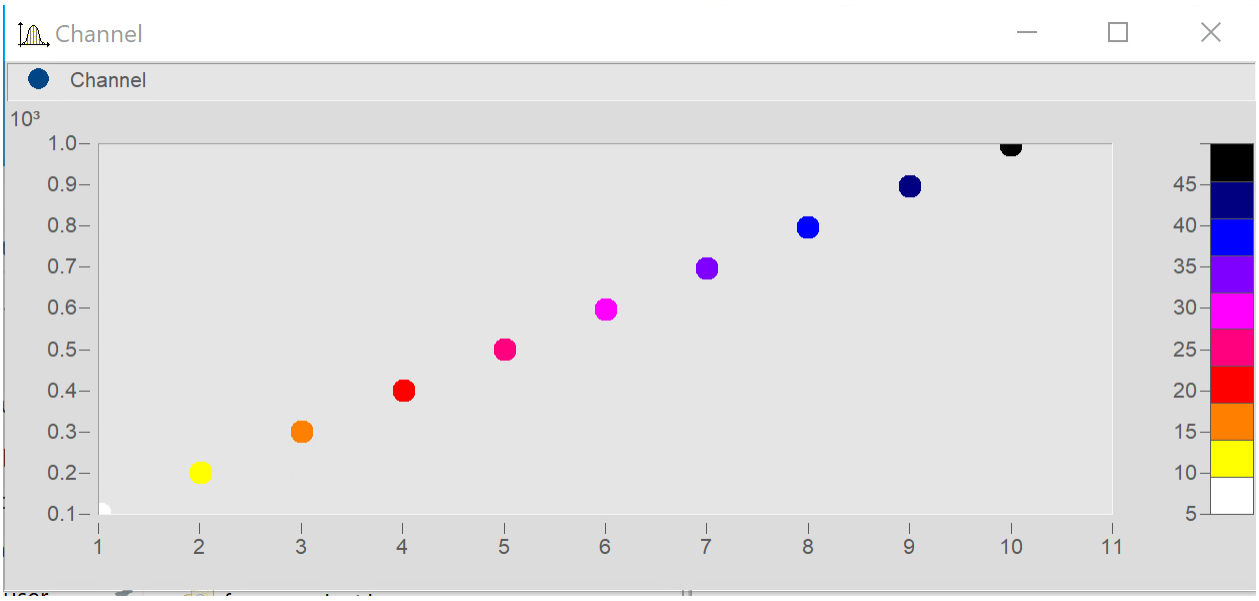


Subsequently, use the mouse to move the axis to the edge:



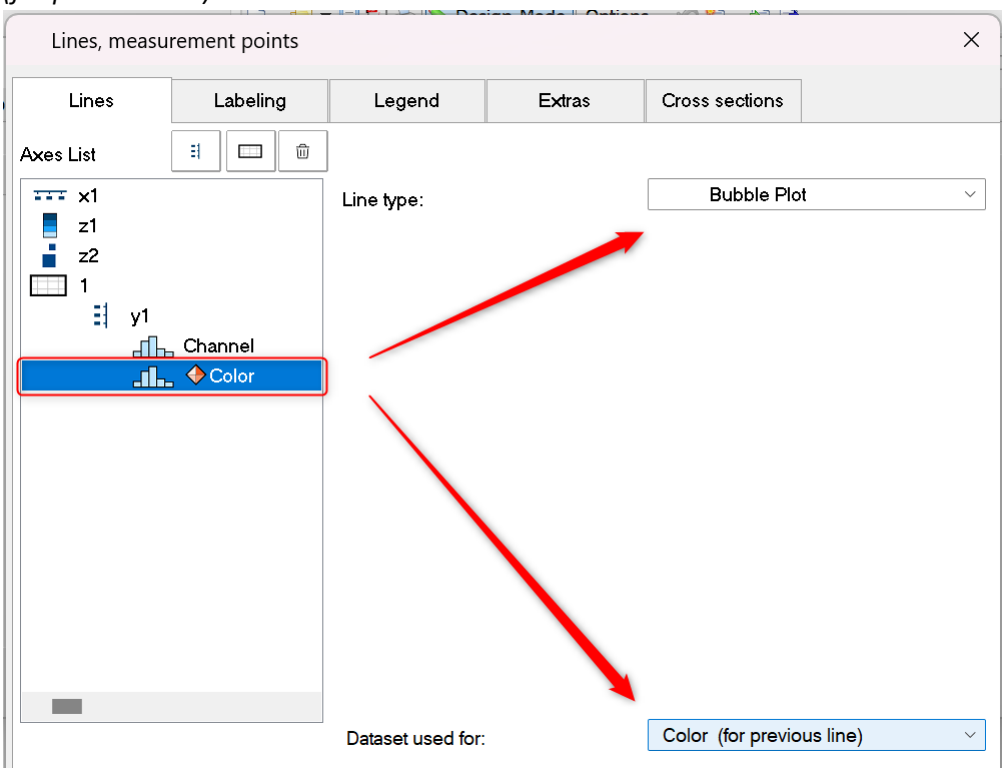
13.6.1.11.1 Additional Bubble Plot displays

Bubble Plot with color palette

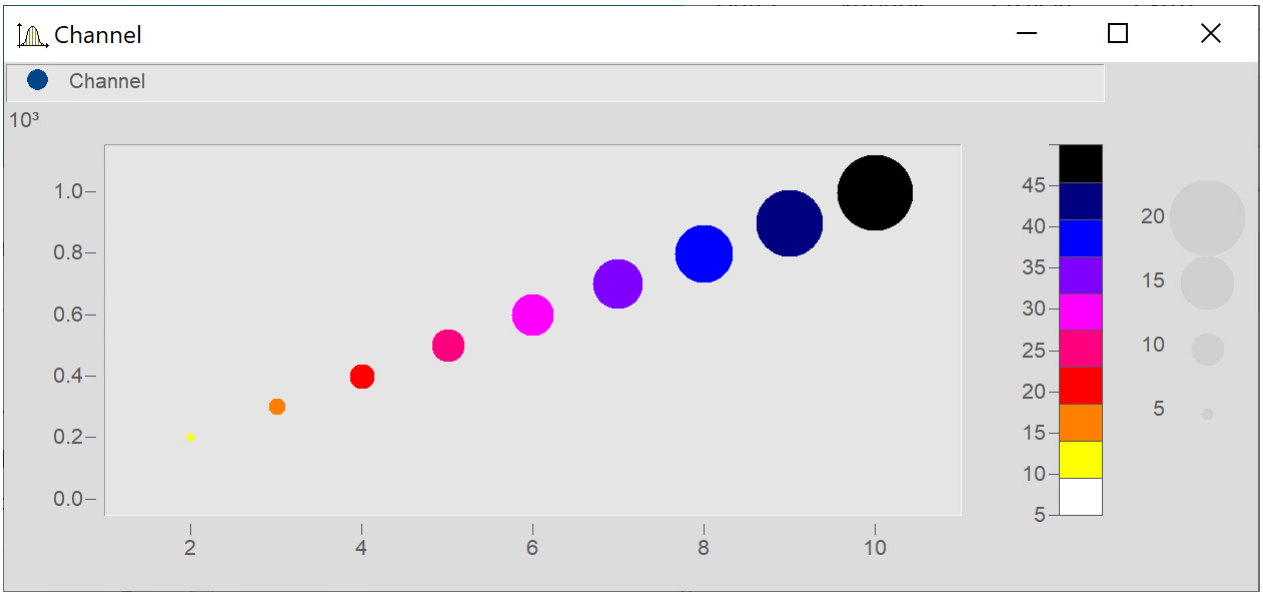


Matches "Size of bubbles corresponding to values of another data set 1333, except that the second data set is used for the color palette.

In the dialog **Lines**, set the color-determining channel to *Line type: Bubble Plot* and *Dataset used for* to *Color (for previous line).*



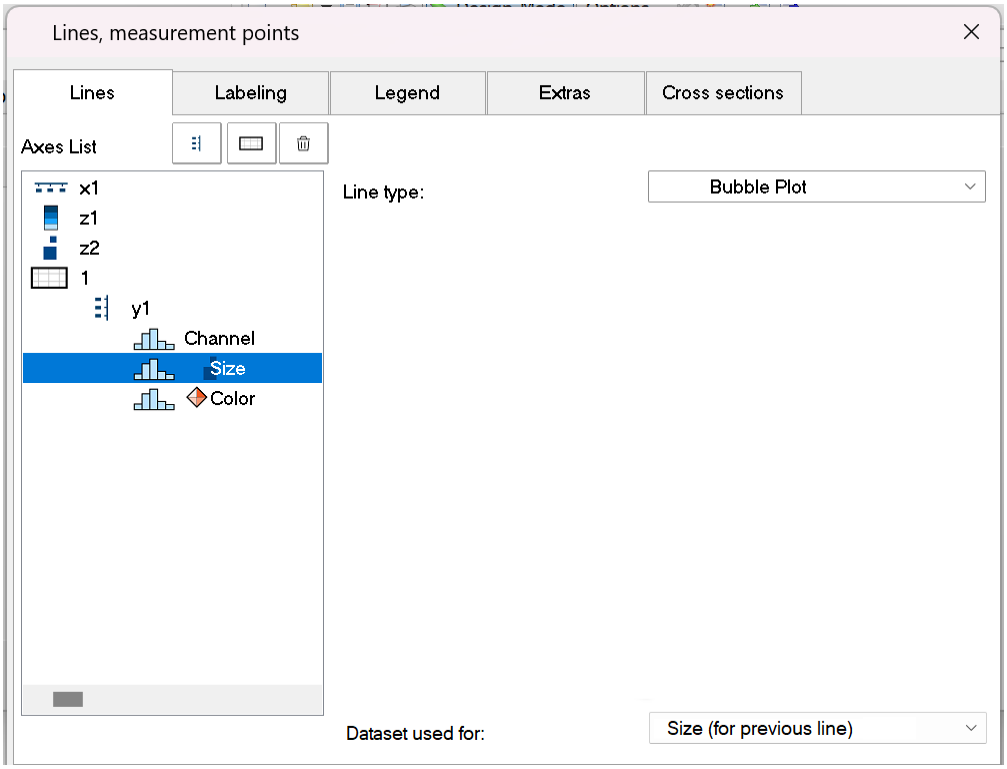
Bubble Plot with size channel and color palette



Matches "Size of bubbles corresponding to values of another data set 1333" except for the following:

A third data set is used for the color palette.

Add the data set for the color and set it in accordance with Bubble Plot with color palette 1336.



In the dialog **Lines**, set all three data sets to *Line type: Bubble Plot*.

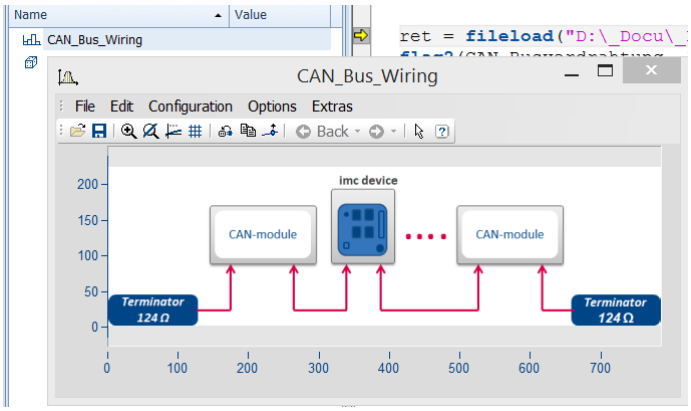
Make the following settings for the respective channels: *Datset used for: Color (for previous line)* and *Size (for previous line)*.

Expand the edge (see here 1335) and use the mouse to arrange the scaling axes for the variables Color and Size.

13.6.1.12 RGB-image

An RGB-image variable consists of a segmented data set in which each sample represents a pixel's RGB-code. When loading an image using **FileLoad**(..."#Picture.dll|Picture Format"...), a color-flag is set in the properties of the variable, which can be queried by means of **Flag?**(Variable, 1). If the RGB-flag is set, when the curve window is opened, the property *Image from RGB-values* 1377 is automatically set under *Effect* in the dialog *Lines\Extras*.

The y-axis and x-axis must both have the same scaling so that the image will be displayed without distortion. This is ensured by means of the property *Resolution* in the dialog *Axes\Arrangement* 1361.



The color stages of the pixels are displayed exactly for the Line type setting 1370 "Steps". For the line type "Lines", the color gradients are interpolated.

If a data set has **multiple events** 1415 each having one image, these can also be displayed when the events have appropriate coordinates. With overlapping coordinates, only the last event may be visible.

The **X-offset** determines the offset in the graph, the event's trigger time is not taken into account.

If any settings are made which the system does not support, the image is not displayed. Settings which are not supported include invalid data formats (XY-data, TSA), XY-overlays over the curve window, selections of individual samples, skipping of segments, and period comparisons.

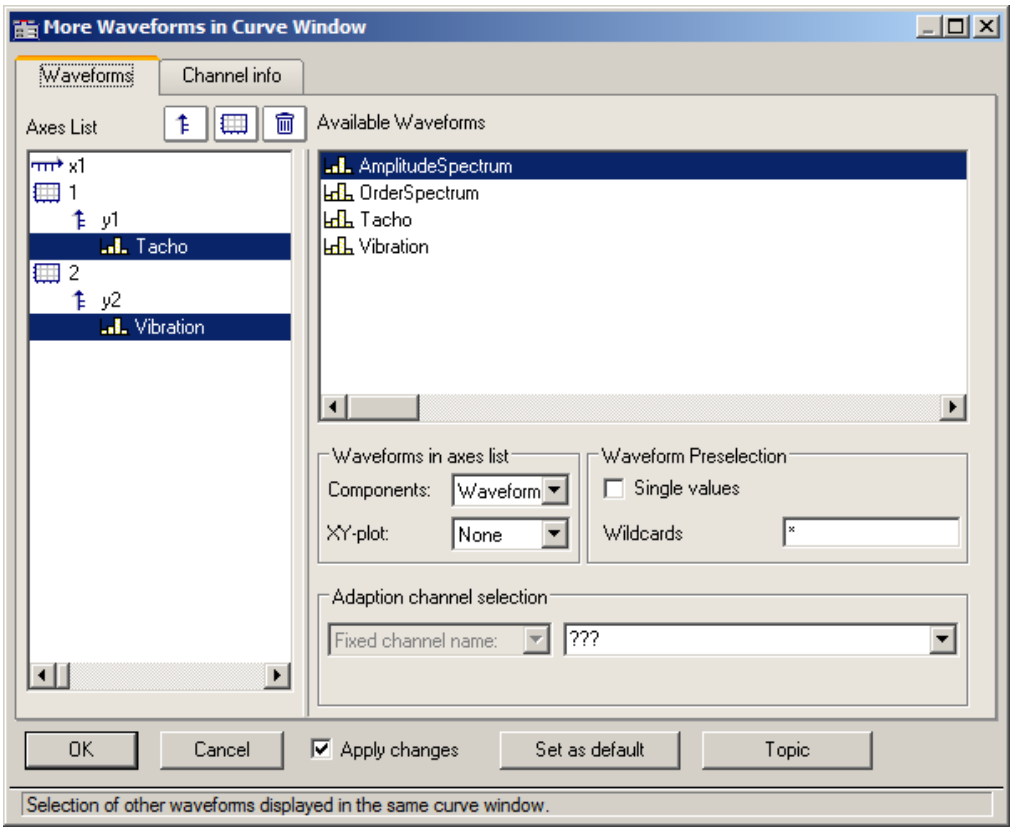
13.6.2 Showing data in curve window

A dialog is available for setting the waveforms to be displayed, the coordinate system and the y-axes in the curve window.

Up to 40 coordinate systems can be represented in a stacked display in the curve window, and each coordinate system can have y- and z-axes with different scaling. A coordinate system and a y-axis (and possibly a z-axis) from this coordinate system must be assigned to each waveform displayed in the window.

Open the dialog by right-clicking the mouse in the curve window and selecting *More Channels.....*

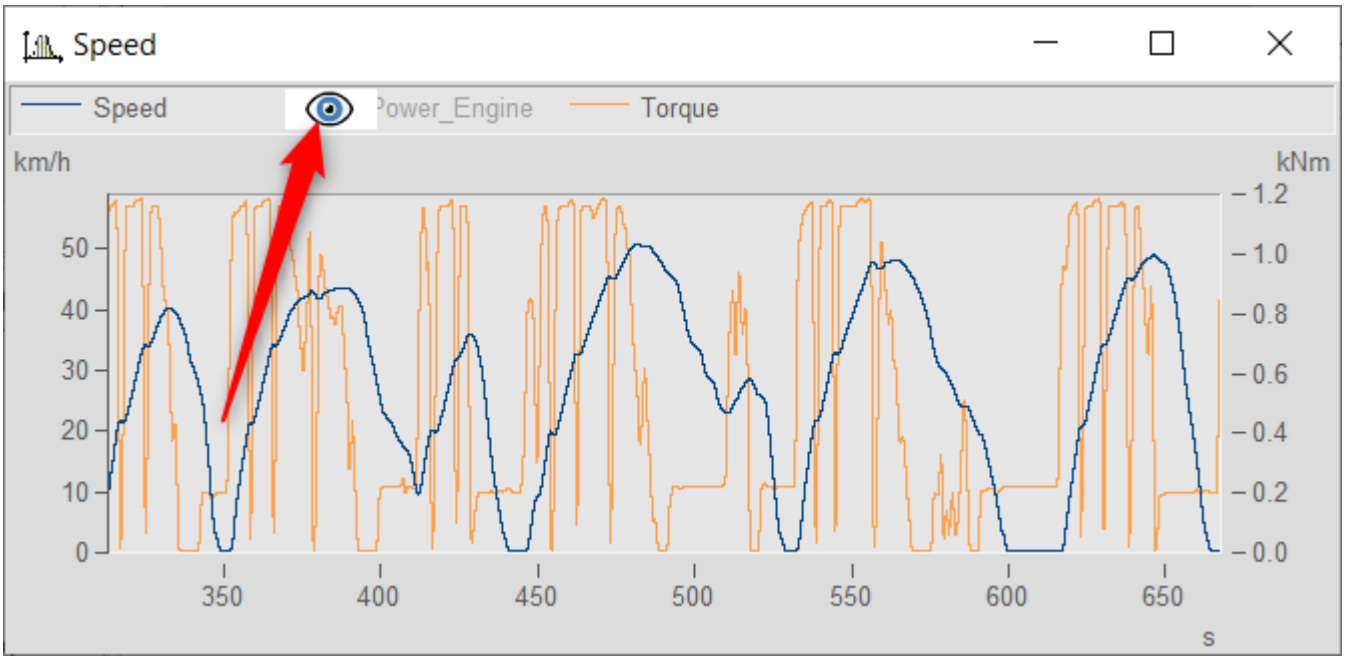
A short help message appears when the mouse pointer is held for a few seconds over a dialog object (provided the curve window help options are enabled.

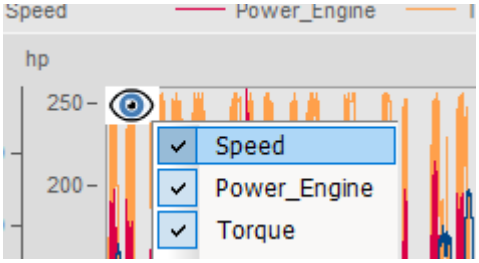


The space allocated in the layout to the Axes list and the Available Channels respectively can be altered by dragging the border between them using the mouse.

The subsections below respectively describe the individual dialog elements.

Showing/hiding channels at the click of the mouse





In the curve window, it is possible to show or hide individual channels at the click of a mouse button. To do this, you move the mouse pointer over the line ample of the a channel in the legende. Click on the eye-icon which then appears there to toggle the channel's display on/off .

Alternatively, use the pop-up menu that appears in the upper left corner of the curve window.

13.6.2.1 Axes list

This *Axes list* reflects the current structure of the curve window. The symbols for the individual coordinate systems are found in the first column. The y-axes displayed in a coordinate system are indicated underneath, somewhat indented. Waveforms assigned to a y-axis are listed after the y-axis. Each waveform is specified by a symbol and its name.

Waveforms can be **added to *Axes list*** from the list of *Available Channels* using drag&drop.

Drag&drop applied to one or more selected entries **moves** these. When you hold down the CTRL key, a copy of the respective item is moved.

Buttons for editing the axes list

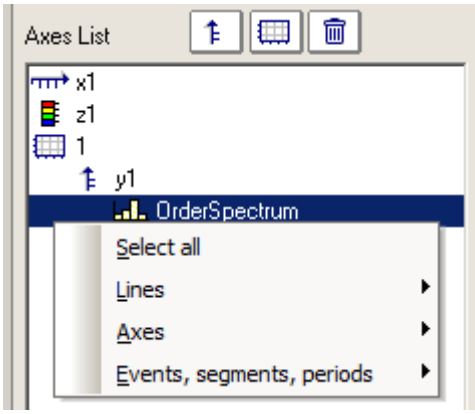
Icon	Description
	A new y-axis is created. To do this, click briefly on this button, which changes the mouse pointer to a suitcase symbol. Click on the target position desired. The new y-axis is inserted above the row over which the mouse pointer is located.
	A new coordinate system is created.
	The lines selected in the axes list (coordinate systems, axes, waveforms) are deleted. Use the <Del> button as an alternative.

Symbols used in the axes list

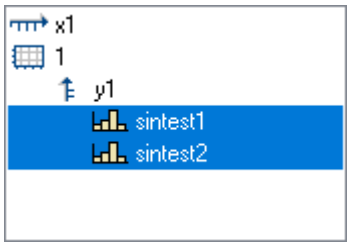
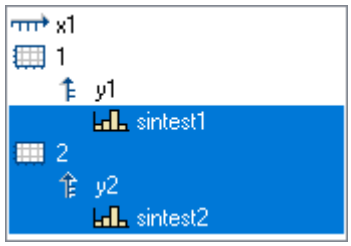
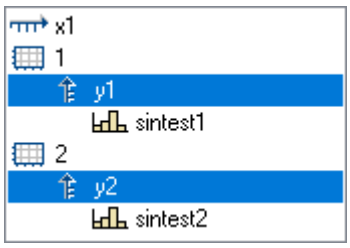
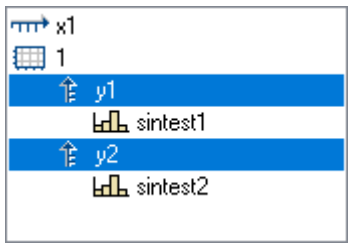
Icon	Description
	x-axis; always at beginning of axes list. All following coordinate systems are scaled on the same x-axis
	z-axis; only in *3D, color map and waterfall diagram*. The z-axis can be scaled only with a fixed z0 and delta-z .
	Indicates an independent **coordinate system**. All further axes and waveforms listed until the next such symbol appears or until the end of the list belong to this coordinate system.
	Indicates a **y-axis**. All further waveforms listed until the next y-axis belong to this Y-axis.
	Two consecutive waveforms in the list are superposed
	XY-Display: Two consecutive waveforms in the list are superposed; the data of the waveform denoted by a "y" are plotted over the values of the waveform denoted by a "x"
	3D-Display: Three consecutive waveforms in the list are joined in a multi-dimensional relationship. The data of a waveform designated y and the data of a waveform designated z are associated with the data of a waveform designated as x. This relationship is for the purpose of 3D display, but can also be used for a color map.
	Variable not present: The original waveform at this position has been deleted. Or: a loaded curve configuration expects a waveform at this position
	Second component missing: The waveform is already designated as one component of an XY-waveform, the other component is missing
	A **normal waveform** with equidistant x-scaling
	Empty waveform: A normal waveform with a length of 0
	Single value, normal waveform with a length of 1
	XY-waveform with monotonous x-track (time waveform)
	XY-waveform with **NON-monotonous** x-track
	A **digital** waveform

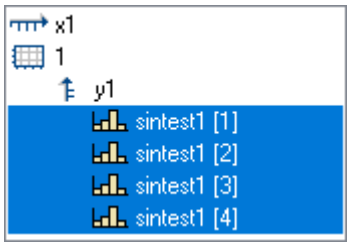
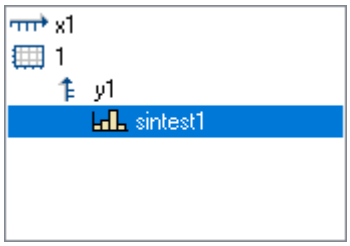
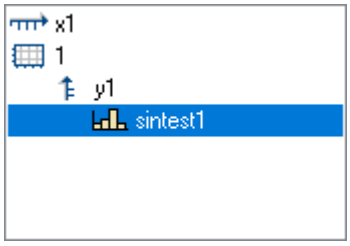
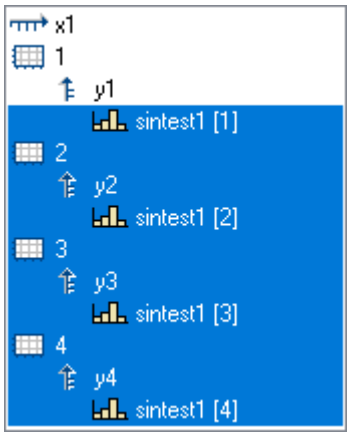
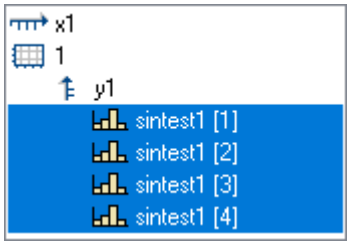
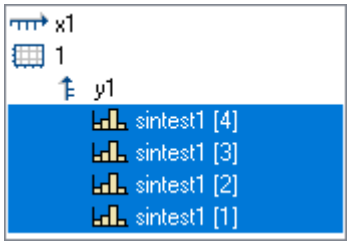
Icon	Description
	A **complex waveform** in magnitude/phase- or real/imaginary display

Context menu in Axes List



The following menu items are available:

Menu item	Description
Select all	Selects all lines in the list.
Lines	A *line* is the graphical representation of one or more data sets, in other words a *waveform*. Normally an entry in the axes list will indicate the simple representation of one data set as a line. The following items affect the selected lines.
Individual coordinate systems	This produces a *stacked* representation of curves.  *before*  *after*
All on 1 axis	All selected lines are displayed with only one axis. Any obsolete axes and coordinate systems are deleted.
Individual axes	A separate axis is displayed for each line.
Axes	The following items affect the selected axes.
All in 1 coordinate system	All axes are incorporated into only one coordinate system. Obsolete coordinate systems are deleted.  *before*  *after*

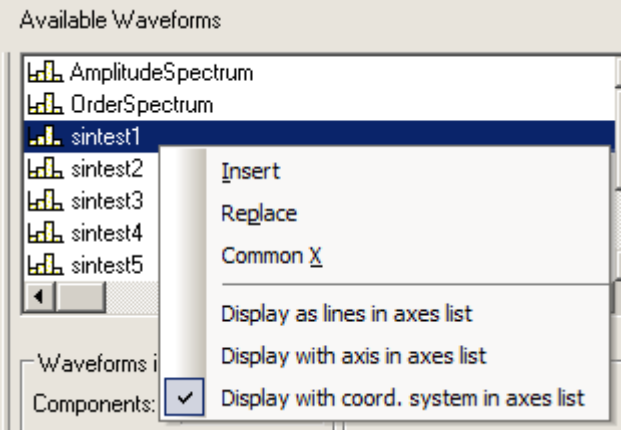
Menu item	Description
Individual coordinate systems	A separate coordinate system is displayed for every (y-) axis.
Events, Segments, Periods	The following items are relevant only to special data types (segmented and multi-shot waveforms)
Combine	Various lines are attached to each other, if the numbering permits. The range of possibilities is dictated by the exigencies of the corresponding dialogs for selecting structural elements.  *before*  *after*
Separate with same layout	One line consisting of several structural elements is dissolved into its components. The order of the elements is retained. If the original line was provided with, say, a coordinate system, then all the new lines will each receive one as well.  *before*  *after*
Separate into individual lines	This is the contrary of *Combine*. Multiple lines, which each amount to a structural element, are created.
Separate into individual coord. systems	Each structural element receives its own coordinate system.
Reverse order	Reverses the order in which the lines appear.  *before*  *after*

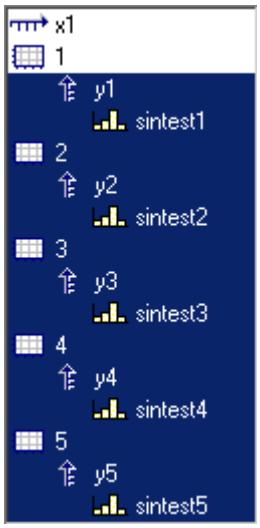
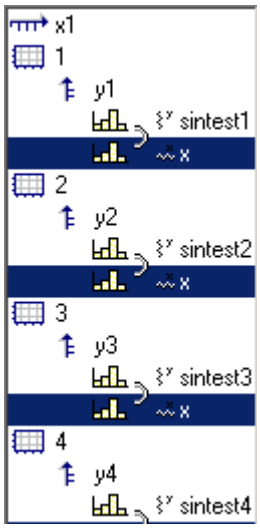
13.6.2.2 Available channels

This list contains the waveforms which can be displayed in the curve window. Waveforms selected from this list can be fetched into the axes list using drag&drop (starting from the left edge of the list, the mouse pointer has the shape of a hand).

Context menu in available channels

Right-clicking mouse in the *Available Channels* list to open the context menu:



Menu item	Description
Insert	The waveform selected is added to the axes list. Depending on how the controls below are set, the waveforms are automatically created along with y-axes or coordinate systems: • Display as lines in axes list, Display as axis in axes list, Display with coord. system in axes list
Replace	The waveforms in the axes list are replaces with the waveform selected under *Available Channels*. If more than one waveform is selected under *Available Channels*, this menu item is not enabled.
Common X	The waveforms selected in the axes list contain an x-component.  *before*  *after*

13.6.2.3 Waveform preselection

Waveform Preselection: Single values

Single values are only indicated in the list of Available Channels if this option is selected. Frequently, single values are generated in FRAME programs or imc FAMOS sequences, for example, as control variables, parameters for functions or indexed variables in loops. The single values are not of interest for display in a curve window.

This name only

The user can specify here a filter for the name of the waveforms to be indicated in the list. Only those waveforms whose names match the specified filters are included. The program does **not differentiate** between **upper** and **lower case letters**. Use the wildcards '*' and '?' when specifying the filter.

The wildcard '*' indicates any number of any given characters, '?' indicates any one given character. The wildcard can also be placed at the beginning or end of the filter.

Examples for wildcards

*	All waveforms
a*	All waveforms whose names begin with 'a'
Channel?	All waveforms whose names consist of 'Channel', followed by any character
Channel	All waveforms in whose names the string 'Channel' appears
a*;t*	different filters are separated by ";"

13.6.2.4 Waveforms in axes list

Waveforms in axes list: Components

What components are selected affects the current selection in the axes list. For two-component waveforms, specify whether the complete waveform should be indicated (XY, polar plot) or only a particular component of the waveform.

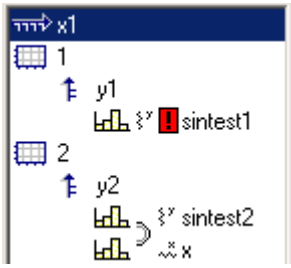
Component	Description
???	The settings for the selected waveforms do not match.
Waveform	Complete display of variables according to type. XY-Waveform: the entire waveform "Y over X" Complex data: polar plot
.x .y	The specified components of XY-waveforms are displayed
.r .i	Real or imaginary part of a complex waveform in real/imaginary display
.m .p	Magnitude or phase of a complex waveform in magnitude/phase or decibel/phase display

Waveforms in axes list: XY-plots

Superposition of normal single-component waveforms.

XY-plots	Description
???	The settings for the selected waveforms do not match
None	No XY-plot has been determined
x of XY	In an XY-plot, this waveform is the x-component
y of XY	In a XY-plot, this waveform is the y-component
x, y, x, y ..	The selected waveforms become x- and y-components in this order
y, x, y, x ..	The selected waveforms become y- and x-components in this order
z	In multi-dimensional displays, this waveform is the z-component.

An XY-plot always involves two or three consecutive waveforms in the axes list, which must both belong to the same y-axis. All must be single-component waveforms (a normal waveform or a specified component of a 2- or 3-component waveform). The two waveforms in an XY-plot are marked by a parenthesis in front of their names.



A small symbol indicates whether a waveform acts as the x-, y- or z-component in an XY-plot.

If a waveform has already been designated as part of an XY-plot, but it's counterpart is missing, this is indicated by a red exclamation mark in front of its name.

Dimensions for 3D displays

If you select the display type *3D*, the available dimensioning options for 3-component waveforms are presented here. Three waveforms as associated spatial dimensions are denoted by a parenthesis in front of the name.

XY-plots	Description
???	this means the waveforms selected have different settings
None	no multi-dimensional relationship defined
x	this waveform is the x-component in a multi-dimensional relationship
y	this waveform is the y-component in a multi-dimensional relationship
z	this waveform is the z-component in a multi-dimensional relationship
y, x, z	The selected waveforms are respectively the x-, y- and z-component of a 3D display

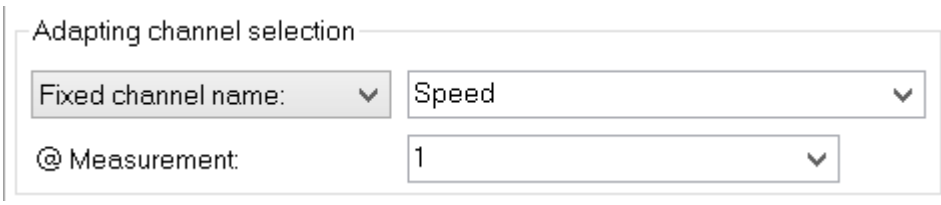
The multi-dimensional relationship always applies to three consecutive waveforms belonging to the same y-axis. All waveforms must be one-dimensional (so either normal waveforms or defined as one component of a 3-component waveform).

Also, there is a small symbol indicating whether a waveform is acting as the x-, y- or z-component. If a waveform is already defined as a dimension in a multi-dimensional relationship, but the complementing dimensions are still missing, a red exclamation mark will appear in front of the waveform's name.

The z component of the display is always the 3rd component. X, y or y, x come first. Please note that the purpose is to display a surface. This is a function y = f (x, z). Compare this with the display of with normal time domain data, where y = f (x). The difference is the same as with waterfall display. Y is the amplitude. X and Z are the independent coordinates. This corresponds to segmented waveforms where you have dx and dz for the two dimensions of the matrix. The values contained are y values. So a segmented waveform is a function as well: y = f (x, z). Thus, a surface will be plotted like with segmented waveforms, but with segmented waveforms all x and z values are equidistant. With xyz display you can chose any pairs of x and z values.

13.6.2.5 Adapting channel selection

The dialog portion *Adapting channel selection* is needed in conjunction with the Data Browser 1214.



Fixed channel name

Lets one specify that only a waveform having the channel name entered in the edit box at the right is displayed. Toward this end, "Unknown channel" from the window *Available Channels*: is added to the axes list and configured accordingly.

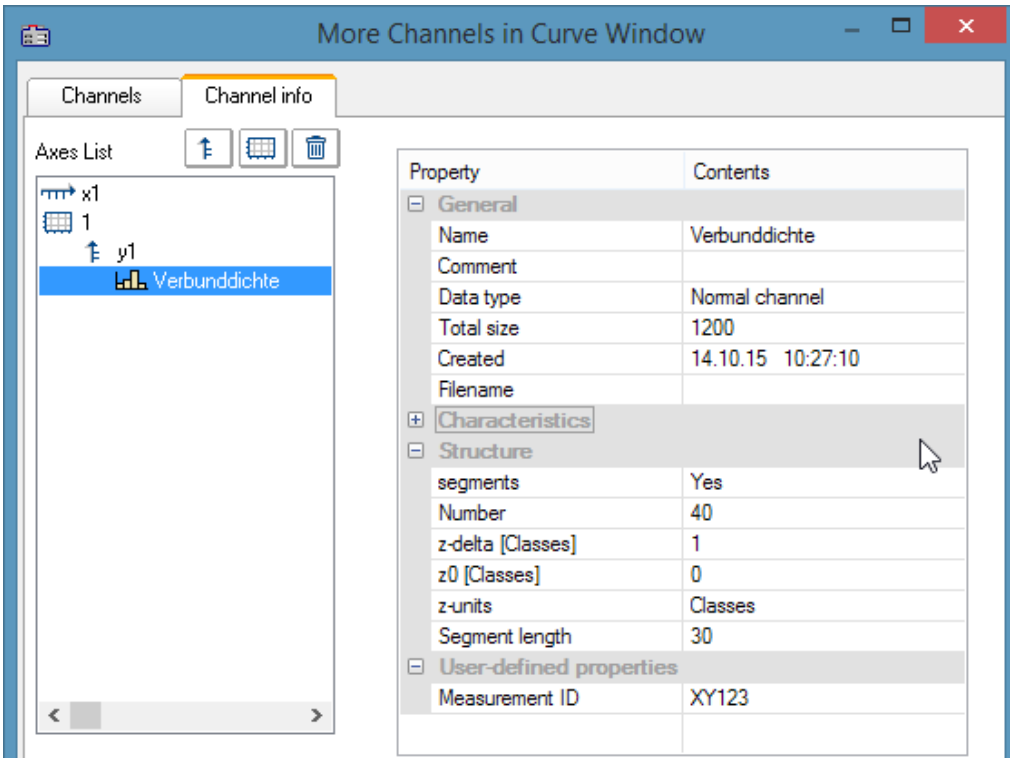
Selected channel

The curve window can be configured so that the waveform having the corresponding number in the Data-Browser is displayed at the this location. The pertinent number for the selected channel can be set in the dropdown list. Toward this end, "Unknown channel" from the window *Available Channels*: is added to the axes list and configured accordingly.

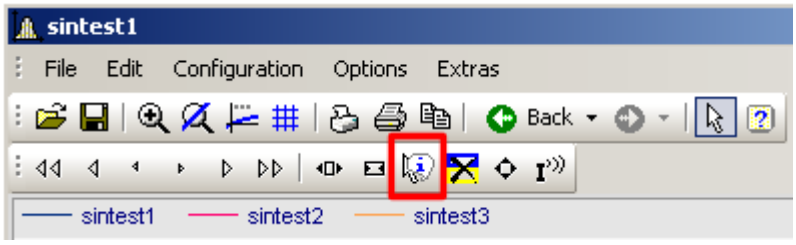
Additionally, the measurement selected in the Data-Browser can be set as a further display criterion in the dropdown-list *@Measurement*.

13.6.2.6 Channel info

The dialog "*More Channels in Curve Window*" contains a second page, *Channel Info*. Here you will find information about the properties or contents of any selected waveforms. The user-defined properties are also listed here.



In Select-mode, the dialog can also be reached via a line's context menu under the heading *Channel properties*, or in the menu ender *Configuration / Arrangement / Channel properties* 1422 or via the corresponding symbol in the *Navigation Toolbar*.

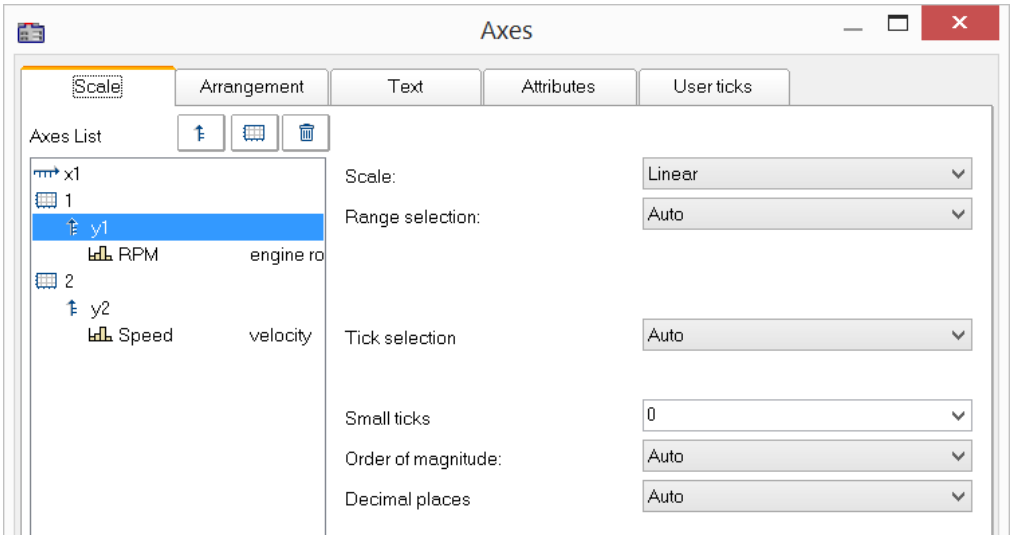


13.6.3 Axis configuration

Each axis (x, y, z) can be scaled manually, with a linear, logarithmic or dB scale. Linear scaling is useful for all time functions, whereas logarithmic scaling is appropriate for spectra. In logarithmic scaling, the waveform is stretched in small coordinate range and compressed at higher x-values.

Mouse Operation

- In the curve window's menu *Configuration*, select the item *Axes*... or to double-click in the region of the axis' scale labels. A dialog for scaling the axes then appears.



The Axes list shows the curve window's structure. Here, select the axis, which you wish to edit on the right side of the dialog. Selecting multiple axes is also possible.

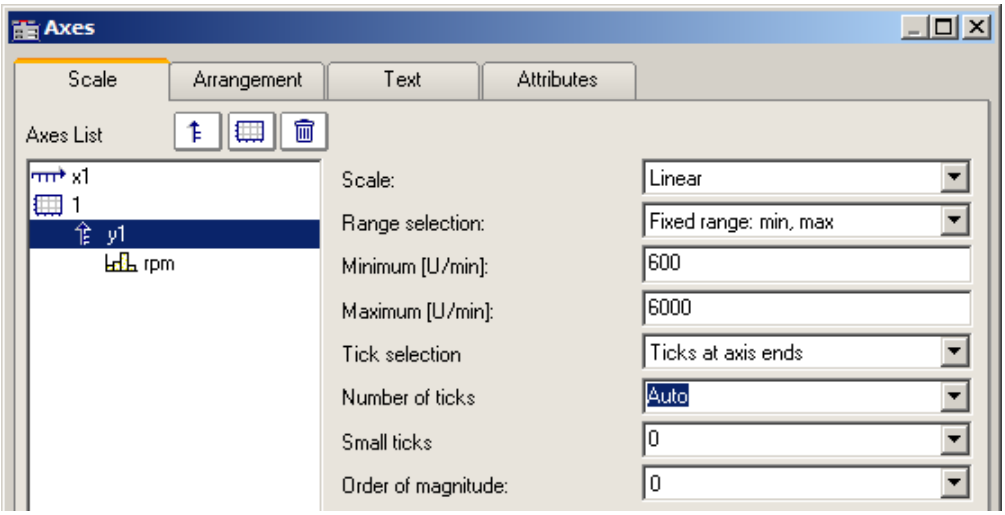
13.6.3.1 Scale

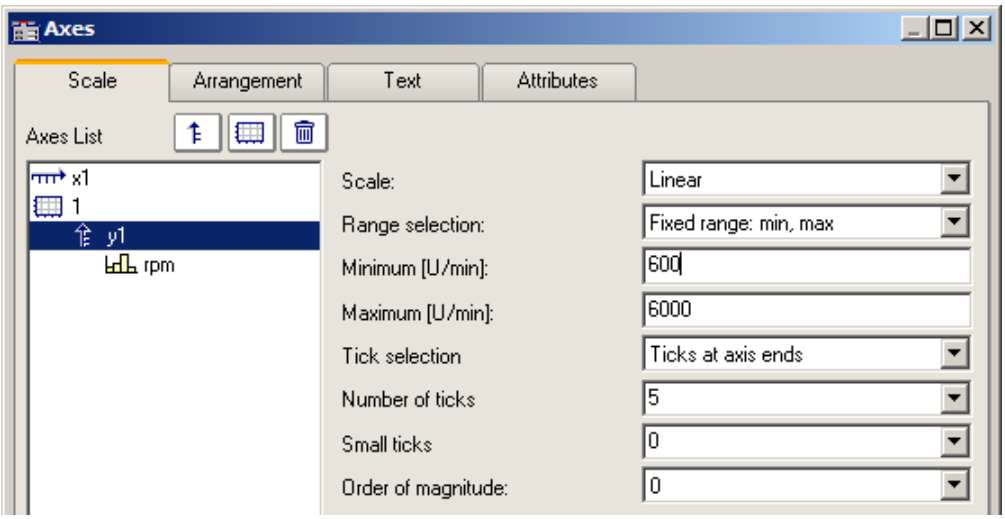
Scaling of the axes. An axis' domain can be defined in a number of ways.

Scale

The axis can be scaled as *linear*, *logarithmic* or in *dB*. In Date/time display, the scaling is always linear. When the mode *Rounding* is selected for logarithmic labeling, powers-of-ten will be written to the axis as long as the range displayed is large enough.

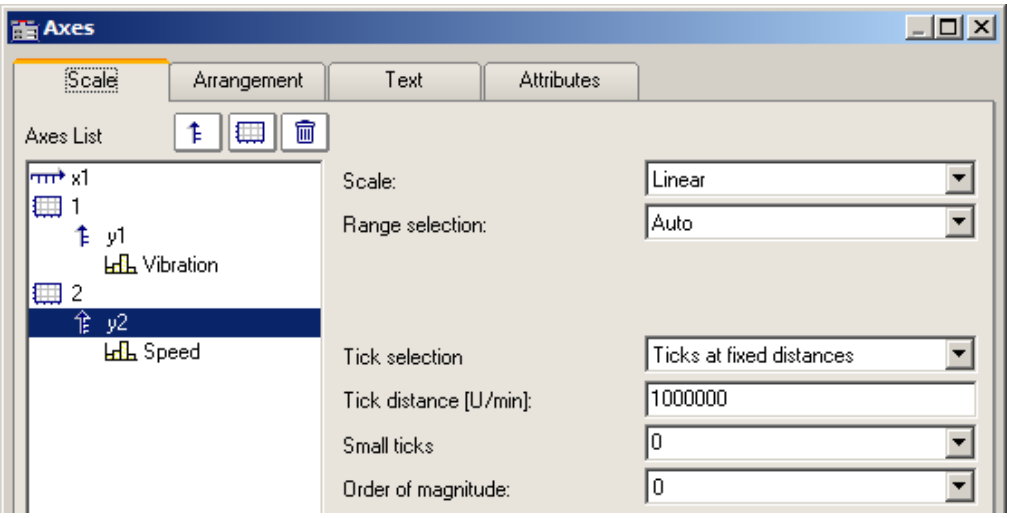
Range selection

Range selection	Description
Rounding: min, max	Specify the axis' domain by specifying the minimum and maximum. The specified values of the waveform are then rounded off in such a way that, taken together with the set number of markings on the x-axis, that axis appears labeled with nice, round numbers. Note that the maximum must always be greater than the minimum.  This menu option is not enabled when the option *Tick selection* is set to *Ticks at fixed distances*.

Range selection	Description
Fixed range: min, max	Here, the range limits and the number of markings are defined. For instance, a minimum specified as 10.0, a maximum of 40.0 and four markings on the axis result in the values 10.0, 20.0, 30 and 40.0 being marked along the axis.  In logarithmic display, the factor with which to multiply the individual tick points must be stated, and it must be more than 1. If you set the first value to 10, the factor to 2 and the number of ticks to 3, then the values 10, 20 and 40 are marked on the axis.
Auto	Determining the range is automatic. The entire curve is displayed in the curve window.
Automatic with zero	With this setting, the zero point is always visible. If the signal values lie within the range 2.0 ... 2.5, the range displayed is 0.0 ... 2.5. This corresponds to an oscilloscope's DC-setting. If the function values are distributed at only a small distance around the mean value, the deviations from the mean value will be interpreted as interference or noise. In that case, a display with visible zero intercept is always set automatically.
Like previous axis	This option is only available if more than one y-axis is displayed in the curve window and an axis other than the first one is selected. If you select this display style, the axis concerned appears in the same style as the preceding axis in the list. In this way you can make multiple curves be displayed with the same axis, and have this one axis state the scaling for all curves correctly.

Tick selection

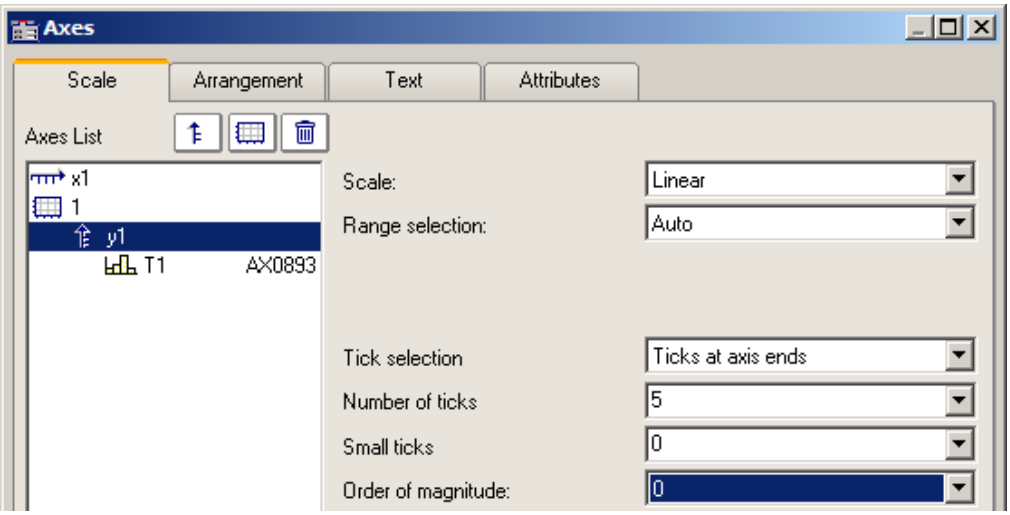
The main ticks, along with which the axes labeling appears in the curve window, can be placed anywhere on the axis. If you select the option *Ticks at fixed distances* in the *Tick selection* control, the additional control *Tick distance* becomes available. Here you have the ability to set the amount of ticks per unit. For instance, when the unit is ms, the ticks appear every 3ms, if 3 was entered in the text box *Ticks distance*.



Select the option *Automatic Labels*, if the ticks should be placed at the ends of the axis. The increment between ticks then depends on the total amount of ticks set.

Automatic tick spacing is recommended; for this, simply select *Auto* under *Tick selection*.

If the option *Automatic Labels* was selected in the *Tick selection* control, the number of ticks can be entered in a text box. The number must be greater than or equal to 2.



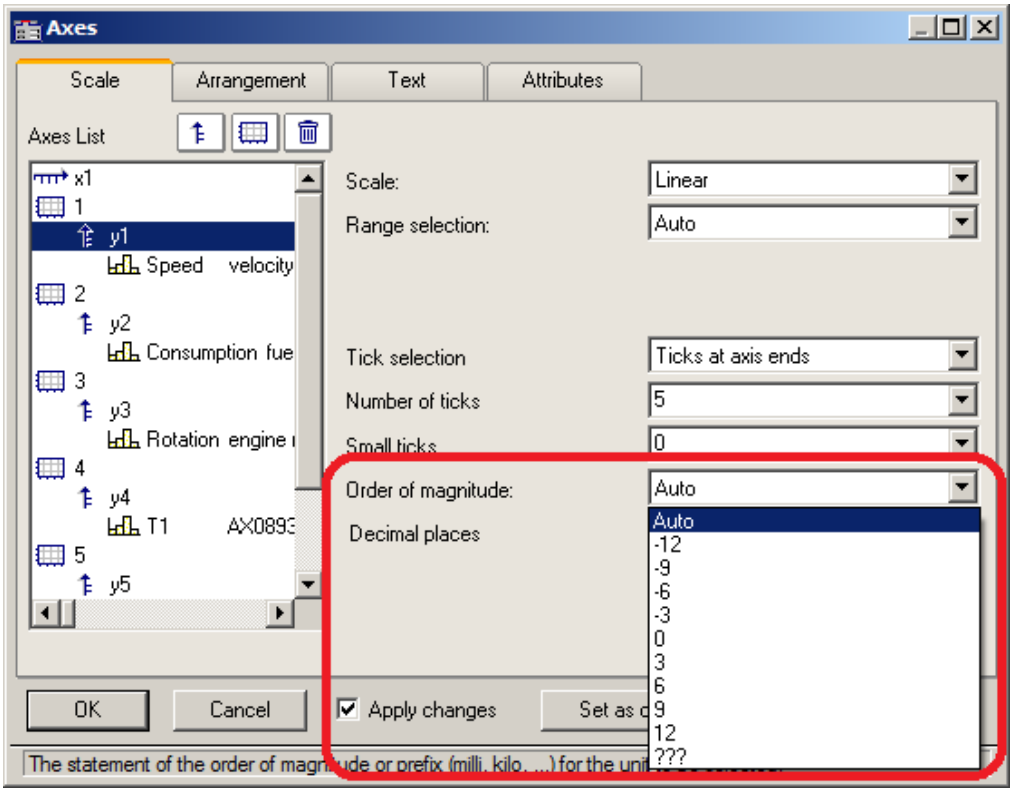
The number of ticks per unit is calculated by subtracting one (1) from the number of markings and dividing the difference by the displayed interval in units. For example, with an interval of 3ms and seven markings specified, two ticks per unit (ms) will be placed on the axis.

Remarks

- An error message is generated when clicking on *OK* if inapplicable values were entered in the text boxes. Error messages occur on account of inapplicable values (too large) or ranges (no positive values in logarithmic display, minimum not smaller than maximum, or inapplicable number of markings). Correct the relevant text boxes and then click on *OK* again.
- The distance (difference) between xmin and xmax may not be too small in relation to the maximum magnitude of xmin and xmax. For example, a range of
  1.0000000000000001 ..... 1.0000000000000002
  CANNOT be displayed. The maximum permissible factor between difference and maximum magnitude is 1E-13.
- When the display mode *1/3-octave/octave* labeling is selected, then the scaling of the x-axis is accomplished as detailed in the appropriate chapter. A different dialog appears for scaling the x-axis.

Order of magnitude

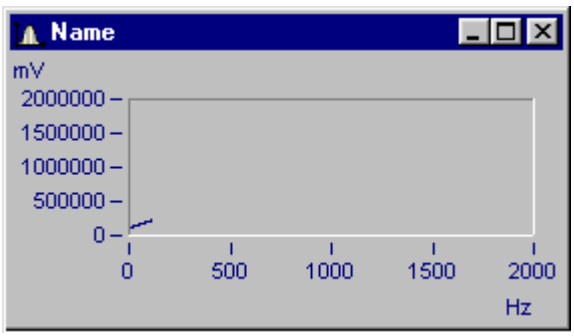
It is possible to arbitrarily determine the power-of-ten of the axis-scaling. If the axis is labeled with a unit, and appropriate prefix will be appended to the unit symbol, if applicable (e.g., mV or MV).



Example x-axis with a fixed

For an x-axis with a fixed scaling range of 0...2000V:

Order of magnitude	Display
Auto:	0..2 kV
+6	0..0.002 MV
0	0..2000 V
-	0..2000000 mV

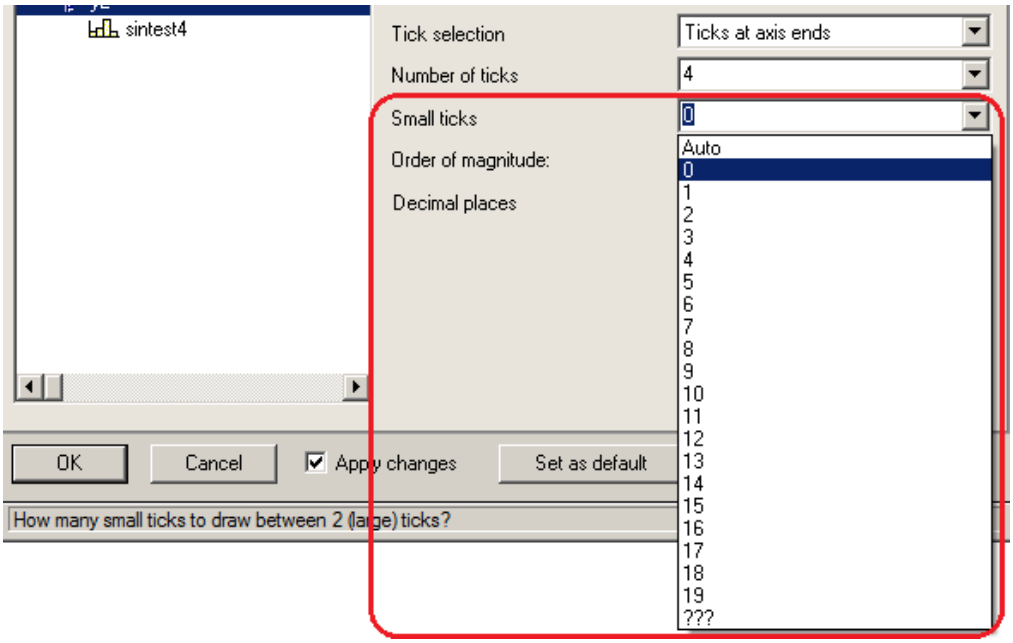


Curve window with inappropriate scaling

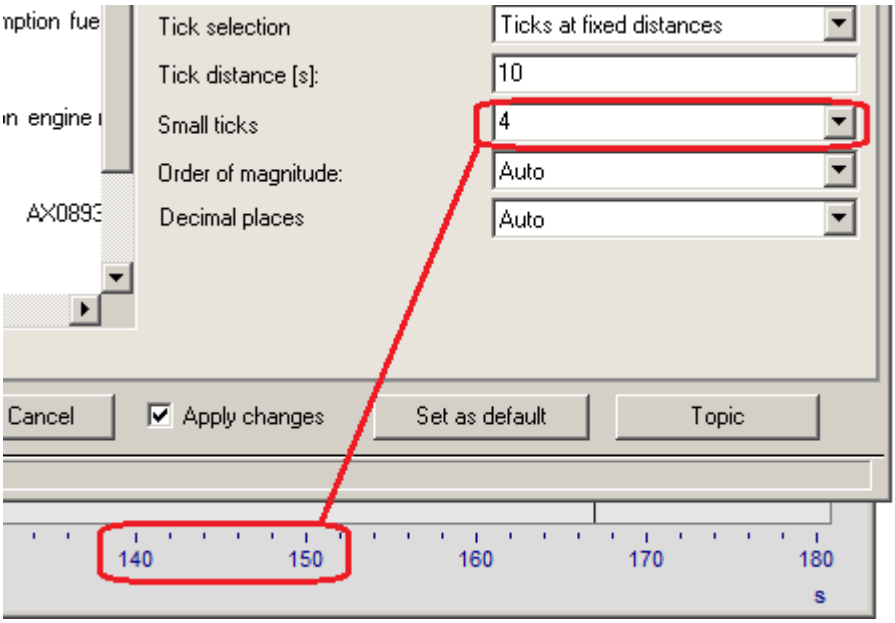
x-axis scaled with exponent 0, y-axis with exponent -3.

Small ticks

Small ticks can be positioned along the axis between the big ticks (main ticks):



To do this, select the amount desired from the pop-down list. If you don't want any, select 0.



The small ticks are not labeled. If the curve window featured a grid, then supplementary grid lines (which may appear thin in a printout) will appear at the small tick coordinates. For details, see the discussion of the menu item *Configuration/ Grid* 1404.

Decimal places

With linear axes, the number of decimal digits can be set here.

Format (for Date/time absolute)

If the scaling of the x-axis is in abs./rel time, the format of the labeling can be specified:

- *Auto ,Auto 1 line, Auto 2 lines*
- *fix 1 line or fix 2 lines.*

The display of time and date makes use of placeholders.

Placeholders in absolute time:

Time: h, hh for hours; m, mm for minutes; s through ss.ssssss for seconds
Date: D, DD for day, M, MM for month; YY, YYYY for year
Names: DDD for abbreviated weekday, DDDD weekday, MMM abbreviated month, MMMM month
A.M., a.m., AM, am for AM/PM format

The placeholders and special characters appear in angle brackets. Furthermore, other characters can be added.

Duplications << or >> for one < or > character in the output

Example

<hh:mm:ss.ss>

<hh:mm a.m.>

<DD.MM.YYYY, hh:mm>

<DDD, DD.MMM.YYYY>

date=<DD>.<MM>.<YY>

Placeholders with relative time:

h, hh for hours; m, mm for minutes; s to ss.ssssss for seconds

D to DDDDDD for days; o to oooooo for hours without days

The placeholders and special characters are expressed in angle brackets. Furthermore, other characters can be added.

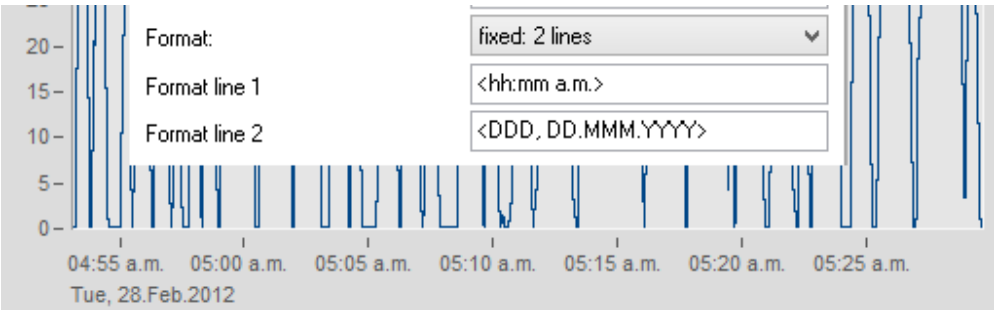
Example

<hh:mm:ss.ss>

<D> Tage

<o:mm:ss>

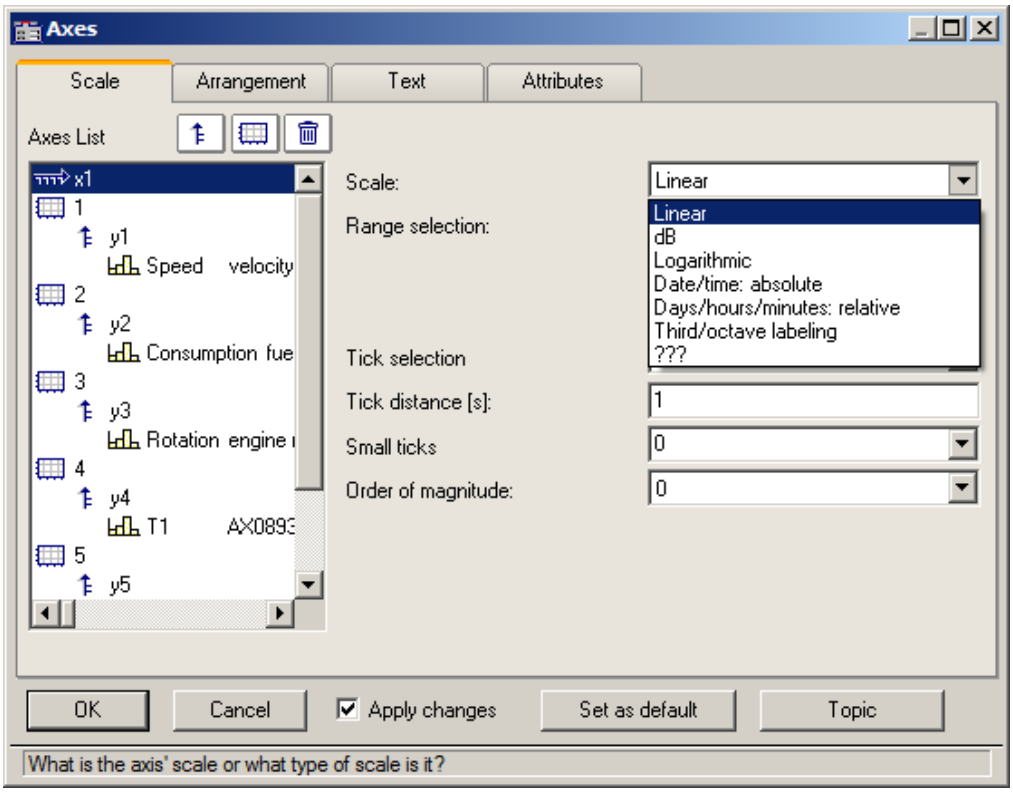
<o> Std, <mm:ss>



Example: line 1: <hh:mm a.m.> line 2: <DDD, DD.MMM.YYYY>

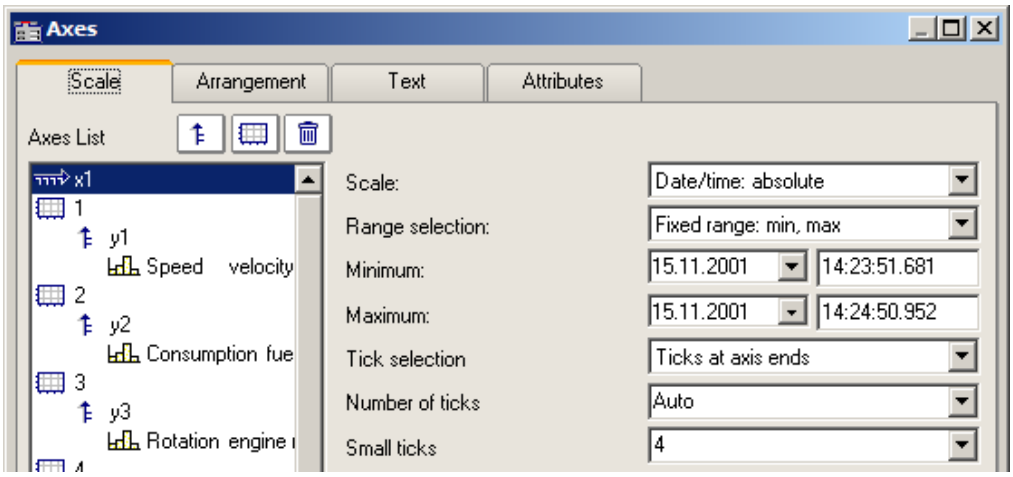
13.6.3.1.1 Particularities of the x-axes

The following concerns special display types only available for x-axes. The x-axis alone must be selected in order for these options to be enabled.



Scale

Absolute date, time: The minimum and maximum are each specified separately in terms of the time and date. Selection of the date is accomplished by means of a calendar. The time is entered into a text box in compact form. The seconds in the time can have decimal places.

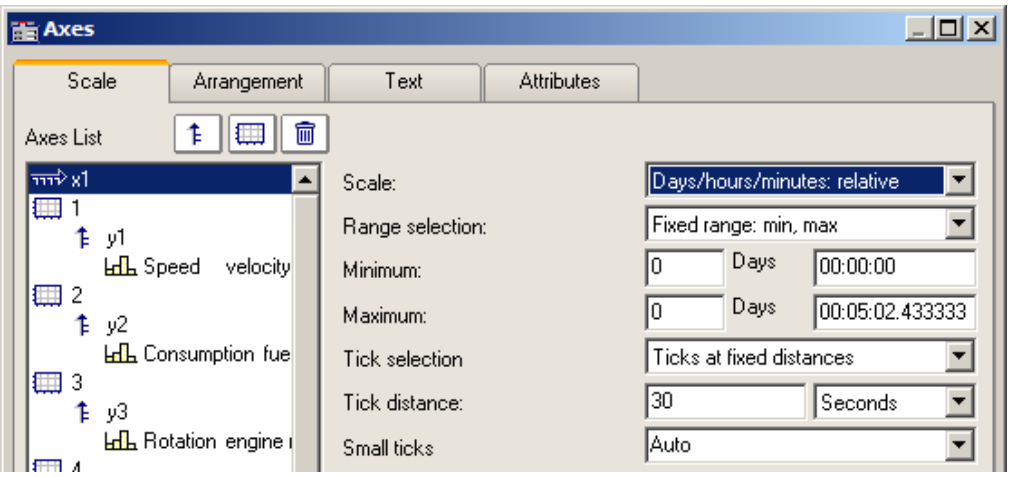


The distance between ticks can be set to be determined automatically or to be fixed. With a fixed setting, the unit can be anything from seconds to days.

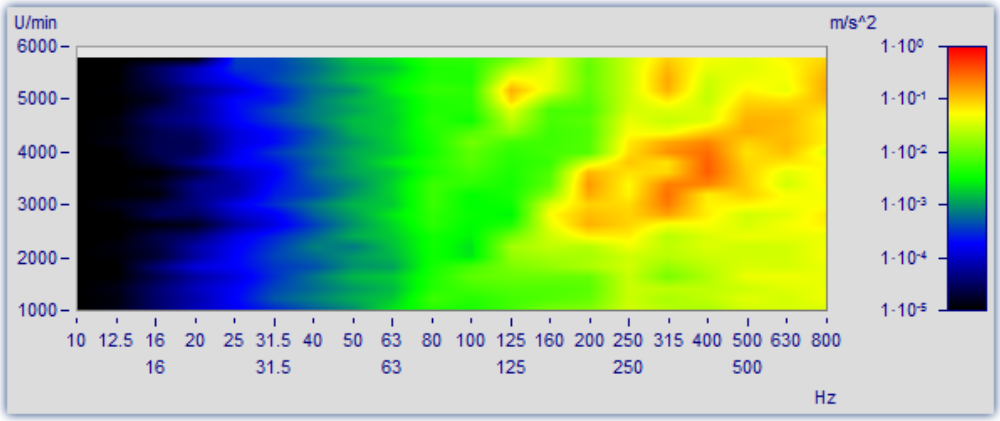
The channels are displayed in reference to absolute time. The absolute time of the measured points is usually derived from the sum of the specified absolute trigger time and the relative time for a point measured after the measurement's start.

Days, hours, minutes: relative: The amount of day and of hours, minutes and seconds (the latter may have decimal places) can be specified. As in linear display mode, the display of the measured data doesn't refer to the absolute triggering time. Thus, only the time intervals between the measured points and the trigger time are indicated.

It is also possible to specify a negative amount of days, in order to indicate points in time preceding the trigger.



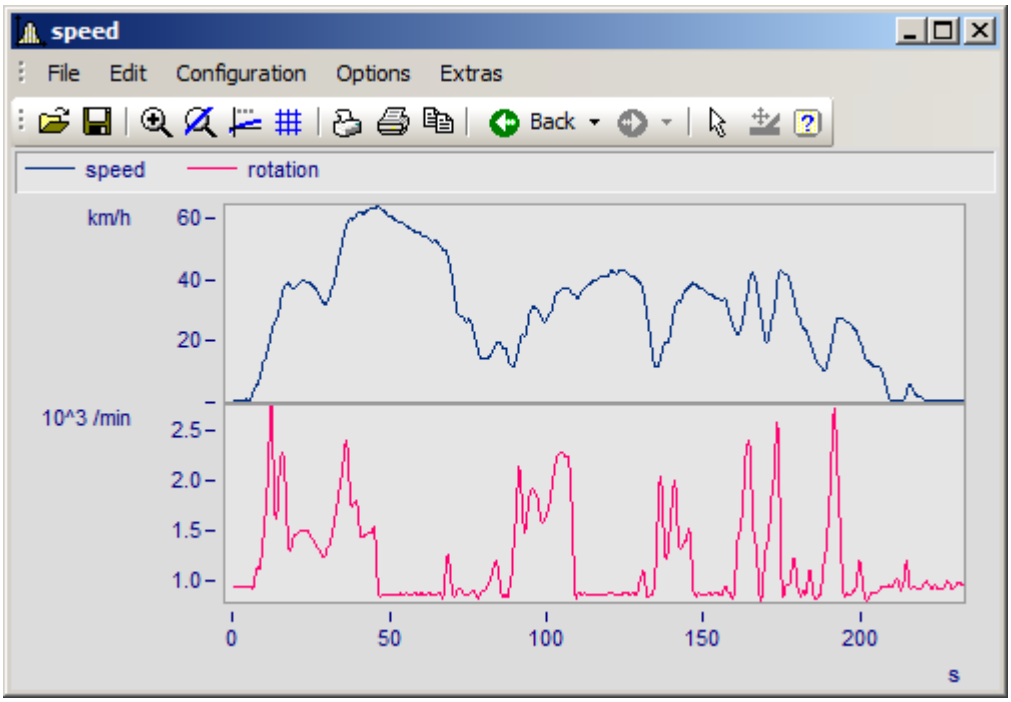
Third/ octave labeling: If a waveform contains a 1/3-octave or octave spectrum and the waveform's x-axis is scaled in 1/3-octaves, then the x-axis can be drawn with numerical values of the 1/3-octaves and octaves, in accordance with DIN.



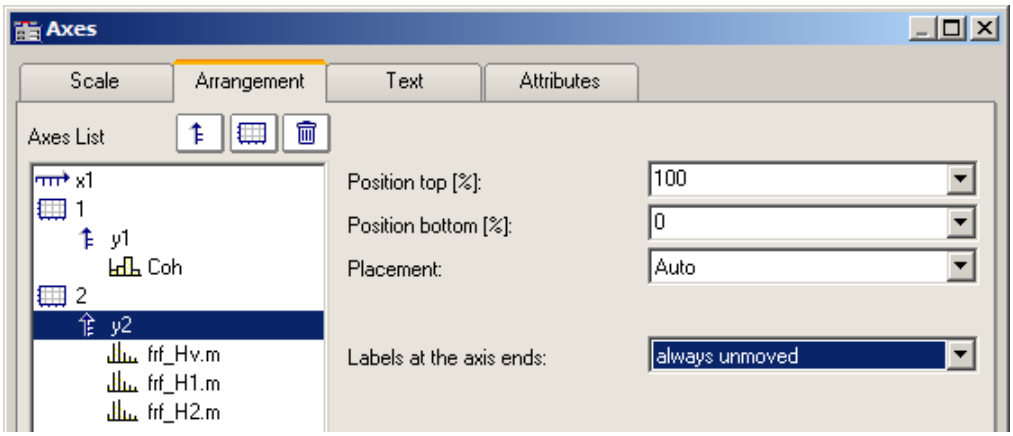
For details, see the dedicated section on "Third / octave labeling".

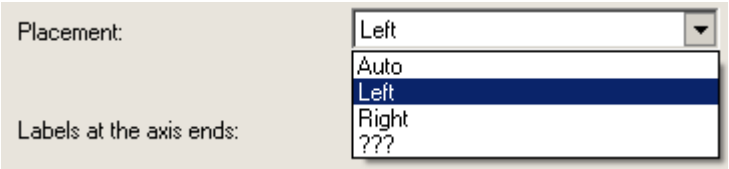
13.6.3.2 Arrangement

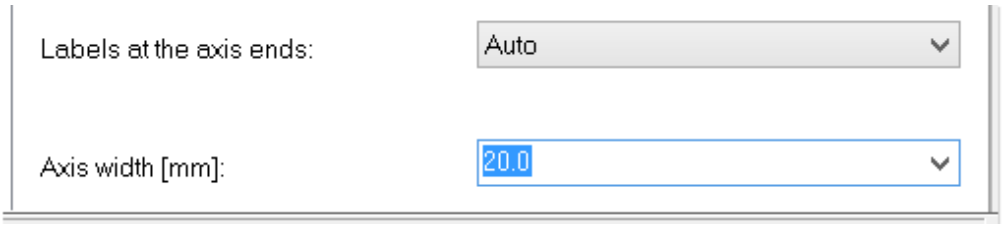
It is possible to stack multiple y-axes in a single coordinate system. It is also possible to set whether to layout the axes a the left or right of the coordinate system.

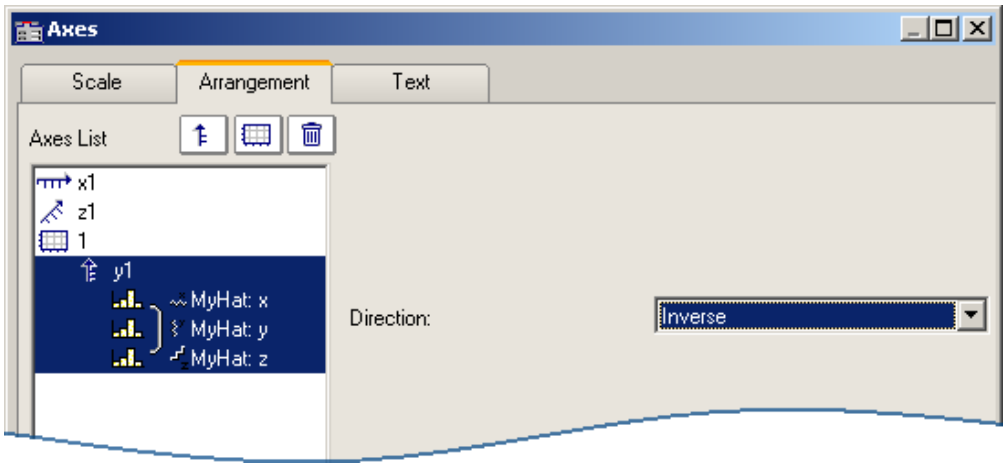
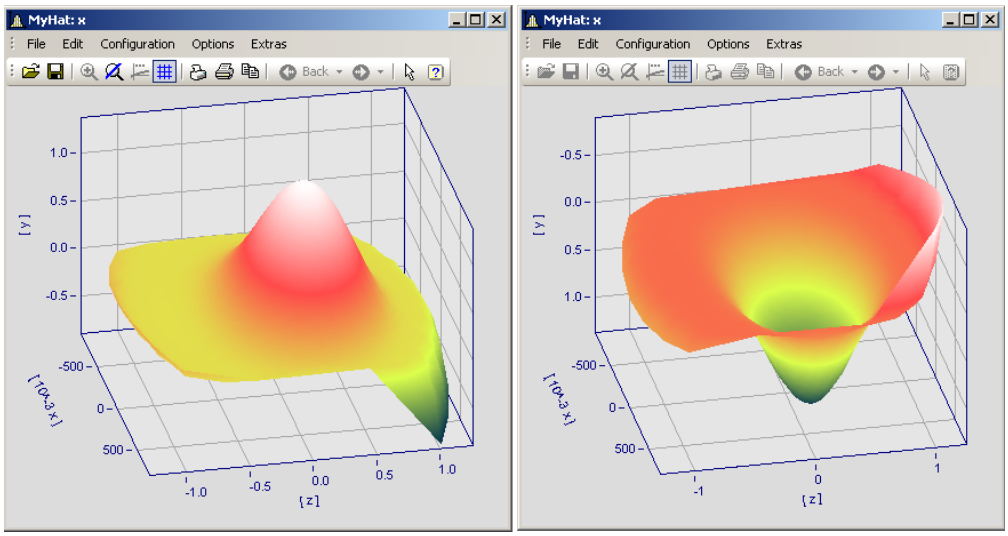


To do this, select the page *Arrangement* in the dialog *Axes*.

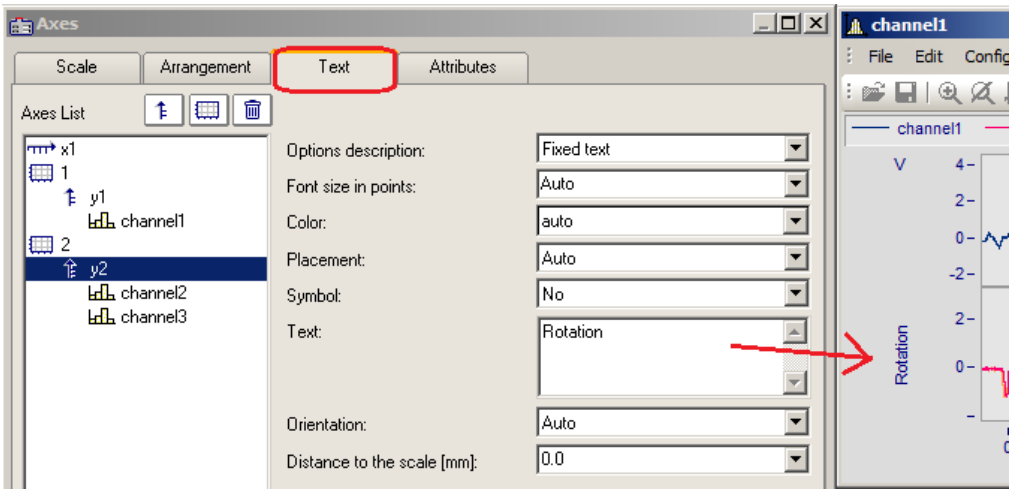


Settings	Description
Position	The relative position along the full height of the coordinate system can be specified for each axis. The setting 100% represents the position all the way at the top, 0% the very bottom. If the axes overlap each other, new "columns" are automatically set up.
Placement	The control *placement* sets whether to position the axis at the left or right of the coordinate system. 

Settings	Description
Labels at the axis ends:	Alongside automatic orientation of the labeling, this control also offers ***always unmoved*** and ***move if necessary***. 
Axis width	The width of the Y- and X-axis can be specified manually. In particular when using "User Ticks 1364", this helps in providing sufficient room for the labeling.
Resolution	Setting the scaling of the y-axis *auto*: individually *same resolution as x-axis*: Generates a constant aspect ratio. In consequence, the ensures among other things that for an RGB-image 1338 the image is displayed without distortion. This setting only works when the same setting (linear or logarithmic) applies to both the x-axis and y-axis, and only for either standard display or Y-axes stacked. Not available for abs./rel time, and 1/3-octaves.

Settings	Description
Inverting the axis direction	With 3D display, you can invert the direction of the Y-axis. Use the menu: *Configuration\Axes* and select the page *Arrangement*. In the dropdown list you will find the entry *Inverse*.   *Example: Invert the axis direction* *At left the normal orientation, at right the inverted Y-axis*

13.6.3.3 Text



Along with the normal axis labeling in the respective units, there is also the possibility to use the page "Text" to personally define the axis labels and to set their parameters. If there is text for a y-axis, it will be printed in vertical alignment in the curve window. Text for the x-axis is aligned horizontally and placed in the center. The following settings are available for the text:

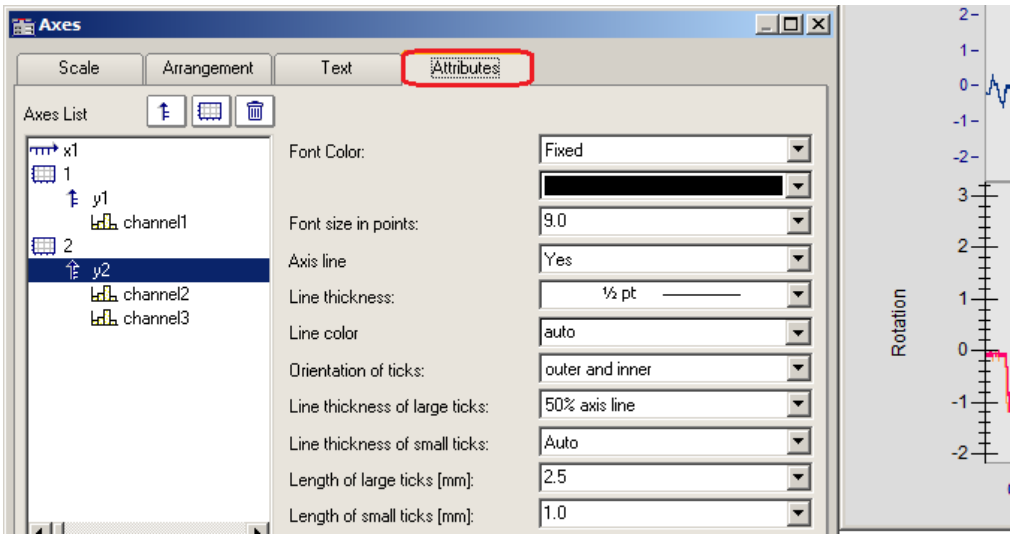
Settings - Options description	Description
No text	The default setting, in which the unit associated with the waveform is displayed horizontally at the upper end of the axis. For "No text", the additional option "*Unit indication*" is shown. This can also be hidden, for instance when using "User Ticks 1364".
Fixed text	Here you can enter any desired text, such as "Length [m]", as a permanent feature. Seed also the notes on Greek letters 1512.
Unit	The unit saved with the waveform is displayed.
[Unit]	The unit saved with the waveform is displayed in angular brackets.
Name, Unit	The channel name is displayed in the first line of the coordinate system and the associated unit saved with the waveform are displayed.
Definable with placeholders and formatting instructions	The text can be given along with fixed components and placeholders. Available placeholders are: • <name> for the channel name • <unit> for the unit • <comment> for a comment on the channel • <e>exponent</e> for an exponent, as an alternative to a^b. • <s>index</s> formats the text as small index. Nesting of the index or exponent is no allowed. • <g*b> for Greek letters. Also in the exponent: A<e>-<g*a>t</e> => $A^{\alpha t.}$ • x<s>i</s><e>e</e> for exponent together with index.

Additionally, it is possible to set the ***font size***, font ***color***, ***placement*** along the axis, and whether the ***symbol*** for the first line appears in front of the signal name, as it does in the legend. The symbol can only be set in the dialog for y-axes. The ***Orientation*** determines whether the text is displayed across the axis or parallel to it. Further, using ***Distance to the scale***, it is possible to specify a minimum distance to the text.

Reference Further texts

Further texts can be added on the tap *Display/Text* 1402 and in chapter *User ticks* 1364.

13.6.3.4 Attributes



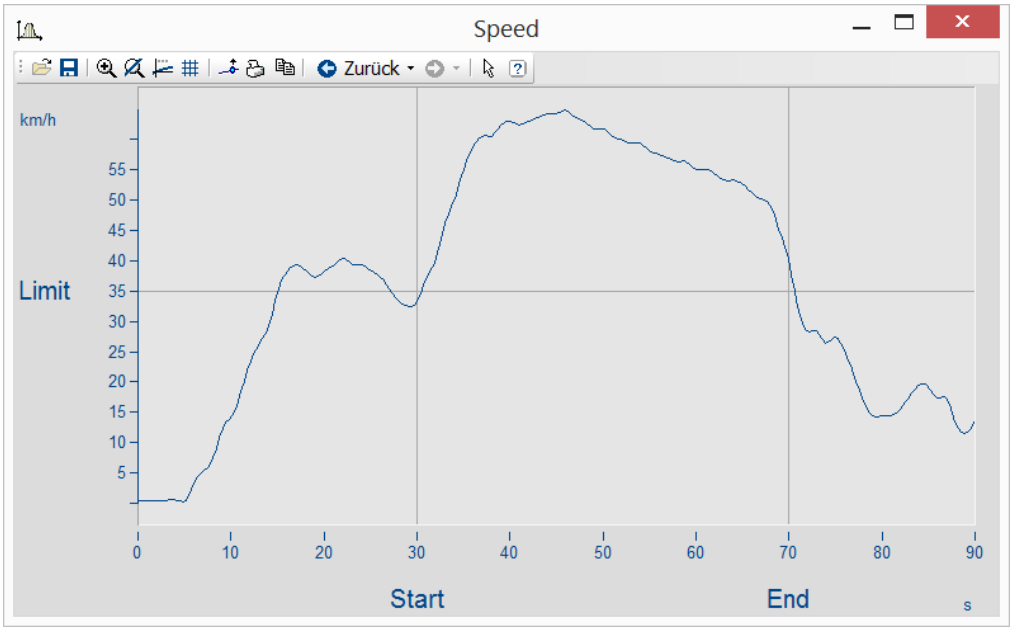
Settings for the axis

The control ***Font Color*** allows the color of the first data set to be applied, or a freely selected color to be set.

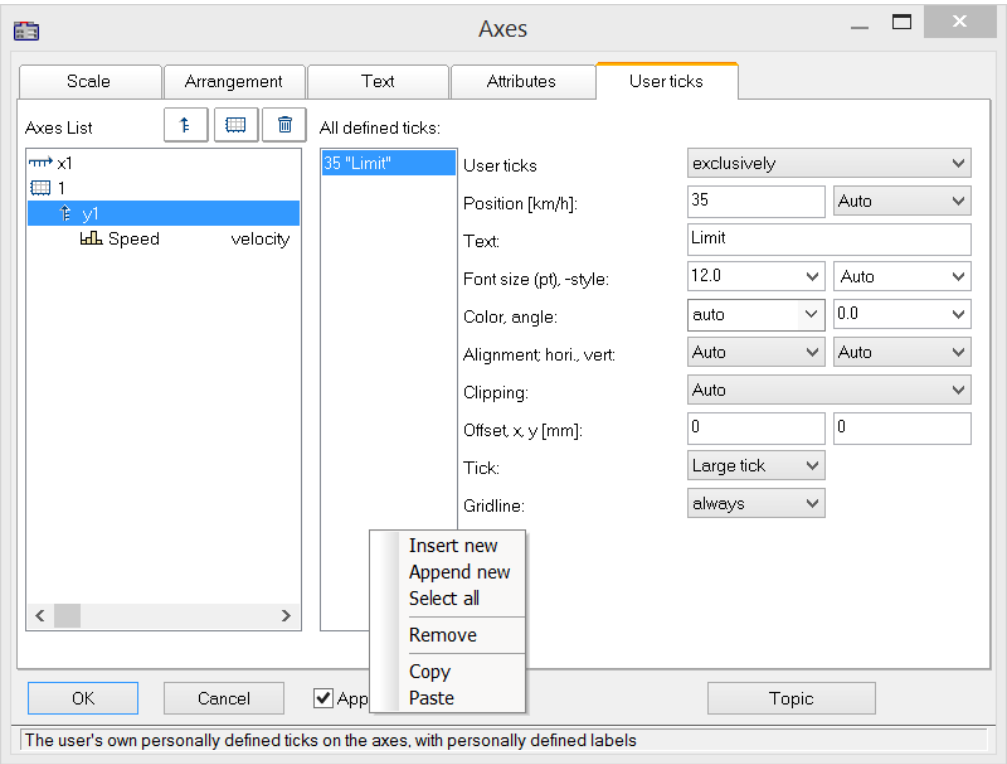
Additionally, it is possible to add an ***Axis line*** while specifying its ***Line thickness*** and individual ***Line color***.

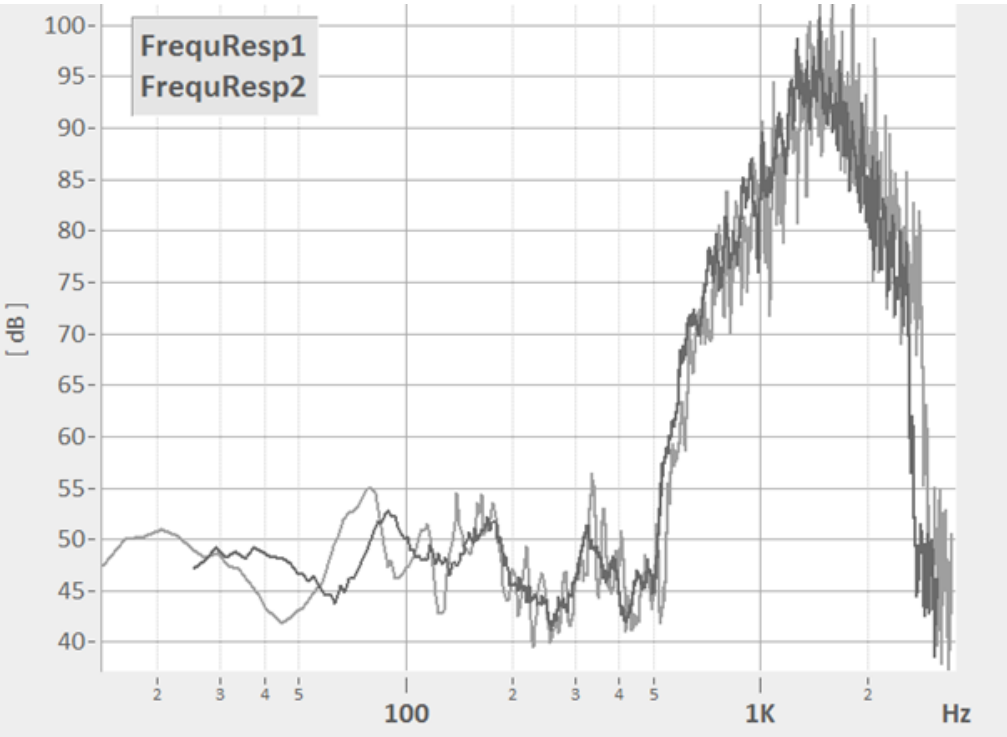
The ***Line thickness*** and ***length*** of the ***ticks*** can be set separately for both the small ticks (between numbers) and large ticks.

13.6.3.5 User ticks

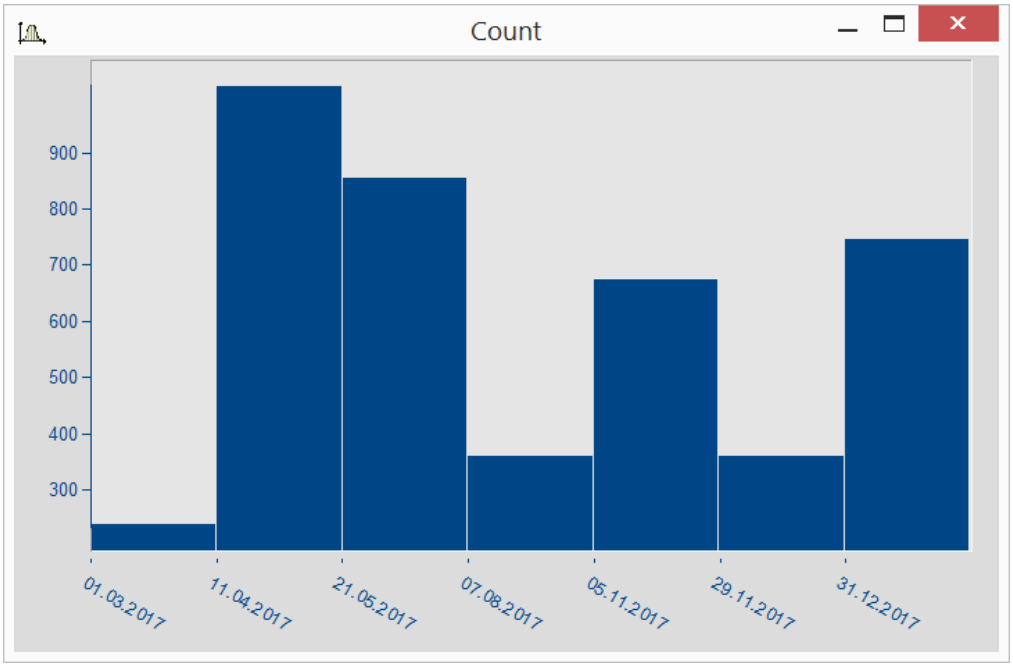


On the page "*User Ticks*", there are controls for inserting extra ticks. Using the context menu for "*All defined ticks*", you can insert, copy and delete ticks.





Any formatting, e.g. for log. Display



Date display independent of x-delta

The properties of the respective ticks assigned are listed according to the axis selected.

Settings	Description
User ticks	Replaces(*exclusively*) or complements(*additionally*) the existing scaling.
Position	Determines the position on the axis.
Font -size, -color, -alignment etc.	determine the appearance of the text. The necessary width can be set on the page "*Arrangement* 1361".
Restriction	With *Restriction* active, the text is no longer displayed if it is moved out of the visible area by scrolling the axis. Without *Restriction* active, the text is still displayed.
Shift	Moves the user ticks to the left (<0) or to the right, for instance in order to avoid overlapping with the scaling in cases of additional ticks.
Tick	Selection of the tick representation, according to the settings on the tab *Attributes* 1364.
Gridline	Shows a line corresponding to the tick. There is a choice of whether or not to make its appearance depend on the presence of a grid.
Exclusively	Custom settings for the selected tick: *Yes*= only the user tick; *No*= in addition to the scaling
From "Label for raw data"	Only visible if *Insert from "Label for raw data"* was activated with the context menu. User-defined properties of the type imc30 are imported automatically.

Reference Further texts

Free texts can be set at *Display/Text* 1402 and on the tab *Axes/Text* 1362.

13.6.3.5.1 Label for raw data

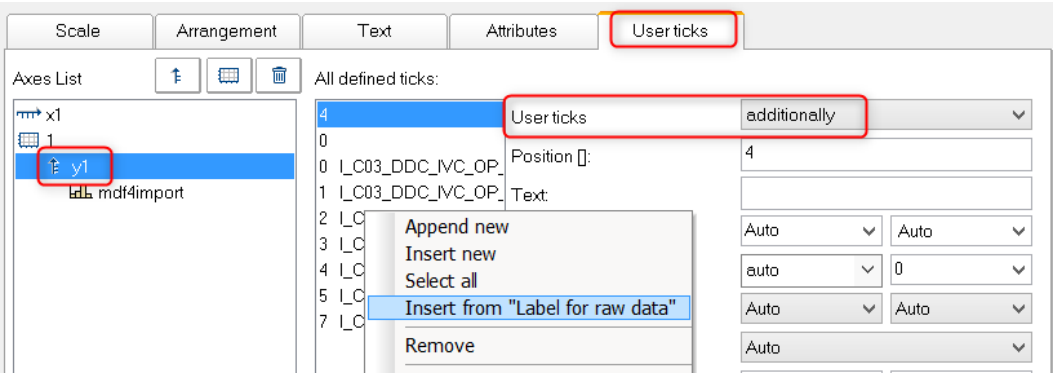
Some signals contain integers representing certain states or error codes. For instance, such a number can denote the gear in an automatic transmission (0=N, 1=D, 2=R, 3=P, etc.). Such information is typically recorded as an integer value.

imc's data format makes it possible to save such supplemental information along with the variable. One category of these "*user-defined properties*" is called *imc30* and lists texts which are provided by certain sources (MDF, CAN, etc.) as "Label for raw data[]".

When a data set contains such imc30-information, the values can be used as User Ticks.

Activating "Label for raw data"

- To import the data, open the dialog *Axes* -> Tab: *User ticks*.
- Select *additionally* or *exclusively* under *User ticks.*
- Select the affected Y-axis and right-click the mouse over the middle list, *All defined ticks.* Only when the signals has properties of the type *imc30*, the entry "*Insert from Label for raw data"* is visible.
- All *"Labels for raw data"* are imported.



Editing the presets for user ticks

As soon as *Insert from "Label for raw Data"* has been activated, the property *"Label for raw data"* is available with the following options:

no: as previously

yes: this tick is derived from a *Label for raw data[integer]*.

Template: Template, if new properties are associated with the channel.

FAMOS generates a tick without a text, at the position 0. This is used as a template. When new positions are added, new ticks are generated in accordance with these settings.

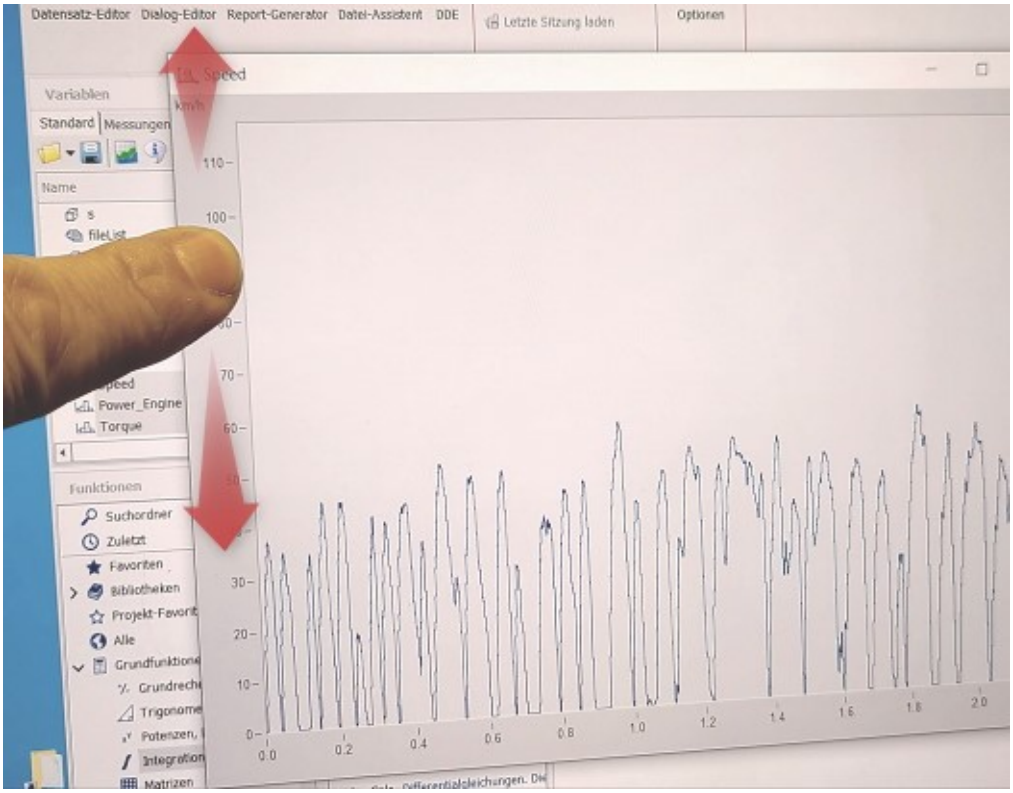
Remarks:

- When positions are discarded, the associated ticks are deleted.
- When you do not desire automatic updates, set the property *From "Label for raw data"* to *"no"*.
- *Label for raw data[integer]*: The value in brackets is a whole number in the channel's raw data, which means unscaled (whole numbers). If there are real numbers, only whole numbers are given consideration.
- *Label for raw data[integer]* always only pertain to a channel's y-values and is only suitable for the y-axis. XY-display is the exception, in which the x-axis represents the y-values of an XY superposition.

13.6.3.6 Touchscreen operation

Enhanced touchscreen operation for the curve window

For the purpose of operating the curve window by touchscreen, certain regions of the curve window are designated for performing certain actions. Thus for instance, in the upper or lower region of the curve window, you can slide the curve in the y-direction, and in the left or right margins you can slide the curve in the x-direction.



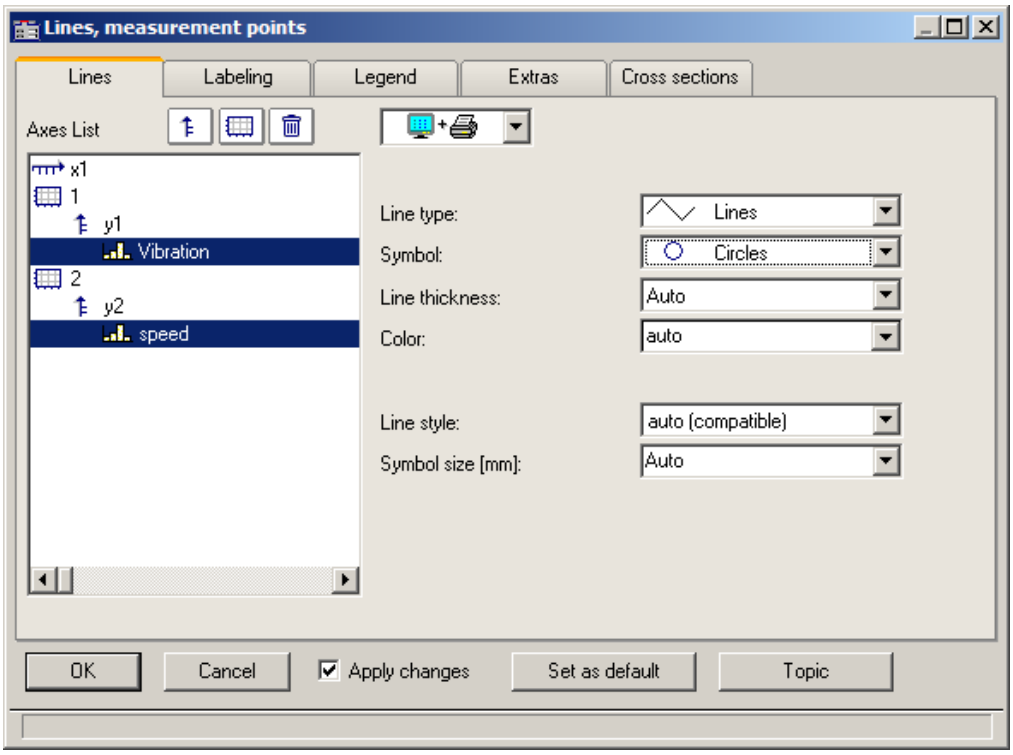
13.6.4 Line configuration

Function

Here, the type of line used to draw the curve and the symbols used to identify the sample values can be specified.

To make the settings for the lines, call the menu item *Configuration / Lines...*. Alternatively, you can switch directly from the dialog *Axes...* to the dialog *Lines* by using the button *Topic*.

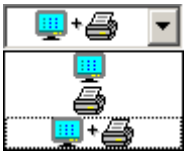
The dialog for setting line properties then appears:



13.6.4.1 Lines

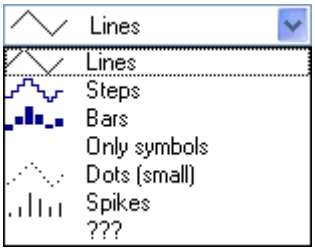
Appearance of lines in the screen display and in printouts

Effect on the printer and/or monitor screen



By making this selection, you determine whether the properties you set apply to the screen view and/or to the printer (or Clipboard). By this means, it is of course possible to make separate settings for the printer and for the screen. Not all properties can be set separately for the printer and the screen each.

Line type



Display of the samples. Normally, the samples are displayed by linearly interpolated solid *lines*. Alternatively you can select other display options from the list appearing further below.

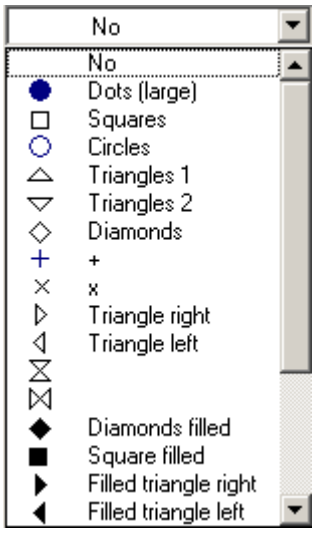
When displaying an RGB-image 1338, there is a distinction between *Lines* and *Steps*. For *Lines*, the pixels are interpolated between the samples (original image points) with color gradients. With *Steps*, the interpolated pixels remain constant. With any RGB-image, all line types other than *Lines* are treated as *Steps*. If multiple samples fall on one and the same pixel on the screen, the system takes an average.

The line type is always the same for the printer and the screen.

Line type	Description
Lines	The "*Lines*" style displays the curve as a polygon, i.e. as straight lines connecting the points of the waveform. In an RGB-image, the pixels are linearly interpolated between the original image points.
Dots	When the option "*Dots*" is selected, only the points of the waveform are displayed as dots of one pixel each.
Bars	The "*Bars*" display shows every point of the waveform as a bar rising from the horizontal axis.
Steps	When "*Steps*" is selected, all points of the waveform are connected with stair-steps, i.e. each sample value of the waveform is held until the next sample ('sample and hold' effect). In an RGB-image, the pixels are interpolated by a constant value between the original image points.
Only symbols	With the setting "*Only symbols*", there is no continuous line. At each measured point, a symbol (e.g. a square) is drawn. The particular symbol desired must then be selected under the *Symbol* control.
Spikes	"*Spikes*" are vertical lines extending from the x-axis to the height of the measured value.

Symbol

Every measured point is indicated with a symbol.



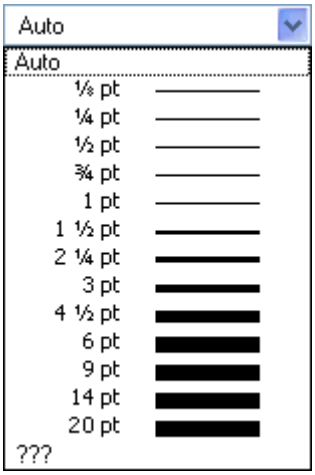
The symbol type desired is selected form this pop-down list. Symbols can be displayed either together with a line or all alone.

In special cases, not every measured point is represented by its own symbol, but rather the symbols are distributed evenly along the plot, for example, in order to differentiate between the plot lines of different channels. See the section on the menu item *Configuration/ Display* 1267, in connection with the property *Number of symbols*.

The symbol always appears the same on screen or printed out.

A fixed number of symbols can be specified on the "*Extras* 1379" tab.

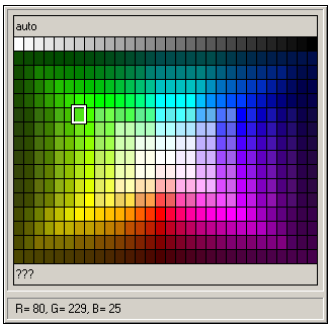
Line thickness



The thickness of lines can also be selected. It pertains no only to continuous plot lines, but also to other line types and certain symbols.

The line thickness can be different for the printout and for the screen display.

Color

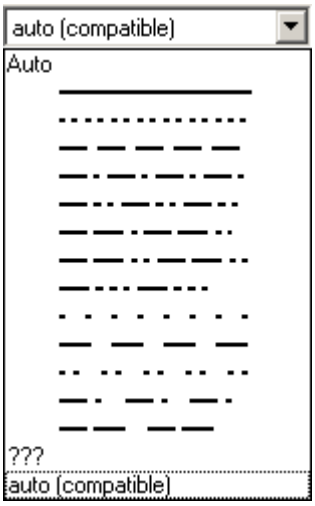


Here, the color of the line or symbol can be specified. If the color is set to *auto*, it is selected automatically by the system. In that case, a line's color depends on its order of appearance among the lines, and on the corresponding global settings for color assignments in the curve window; see the discussion of the menu item "Options / Colors 1493".

It is also possible to select a fixed color from the palette provided, which cancels all automatic settings.

The color can be different for the printout and for the on-screen view.

Line structure



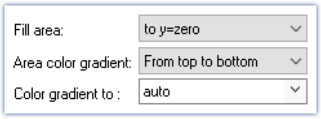
This determines the line's structure. If the structure is set to *auto*, then different line structures are assigned in succession, i.e.the first line is solid, the second dotted, the third dashed etc. For *auto (compatible)*, the global setting *Curves in structure* is observed. This mode is compatible with imc FAMOS 5.0, where the line structure was set globally in the *Colors*-dialog.

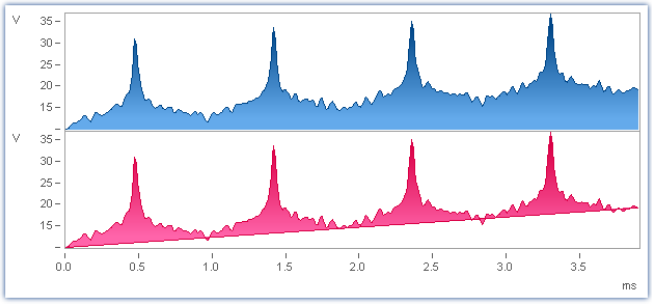
Symbol size [mm]

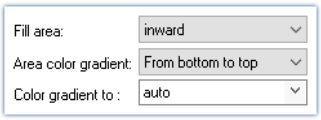
If symbols were selected for display of the measured points, the diameter in mm can be defined here. For *auto*, the global setting *Symbol diameter* is observed.

Fill area

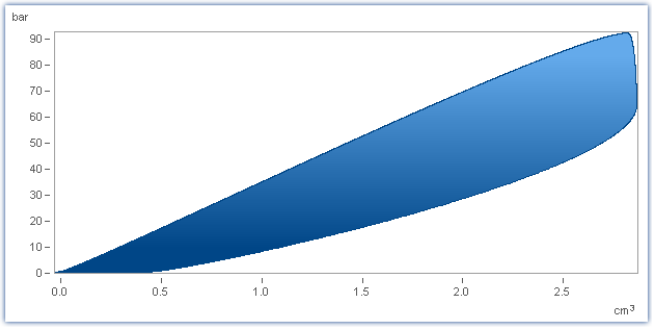
Fills the area below a line.

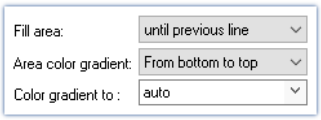




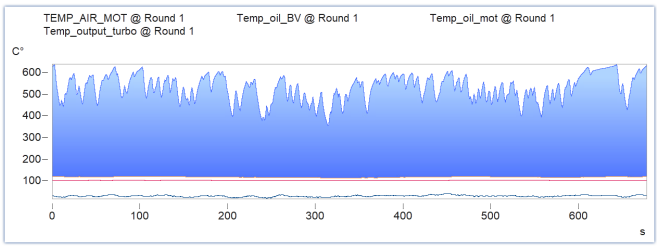


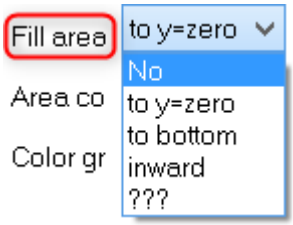
For XY data sets, the hysteresis is closed if necessary so that a closed area can be filled.





For "*Fill area: until previous line*", the area between two consecutive lines in the curve window is filled.



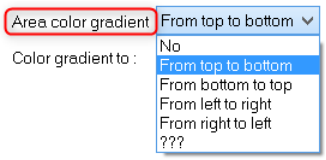


no: as previously

to y=zero: Fills area below the graph up to the zero line.

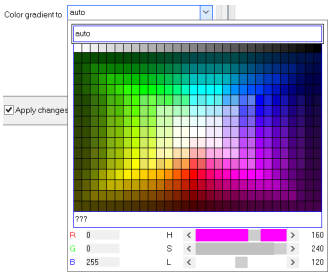
to bottom: Fills area to the bottom edge of coordinate system.

inward: encircled area. The last sample of the data set displayed is connected to the first sample. An closed line results, and the area it encloses is filled.

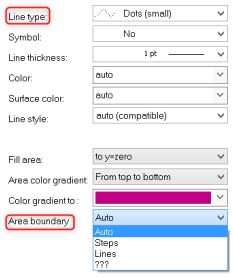


If ***Fill area*** is not ***no***:

Area color gradient: Specifies the color gradient for the filled area. without color gradients, only the one color is applied evenly. Otherwise, the color transitions from one line color to another. The second color is specified under ***Color gradient to***.



Color gradient to: auto (like the line itself) or permanently selected. For ***auto*** and color gradient, a lighter variant of the line color is drawn.



If ***Line type*** = ***Dots*** or ***only symbols***, there is the additional option

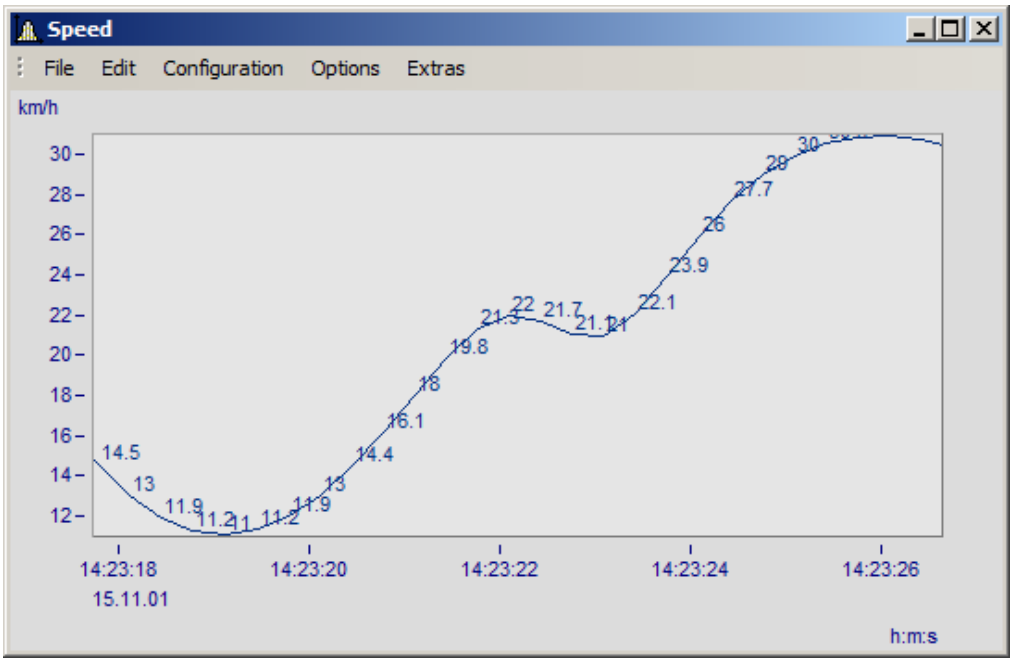
Area boundary: This determines whether the edge of the area is interpolated linearly or appears in discrete steps.

If multiple curves are displayed with overlap, the order for each curve is:

1. first the area
2. then measurement uncertainty
3. then lines

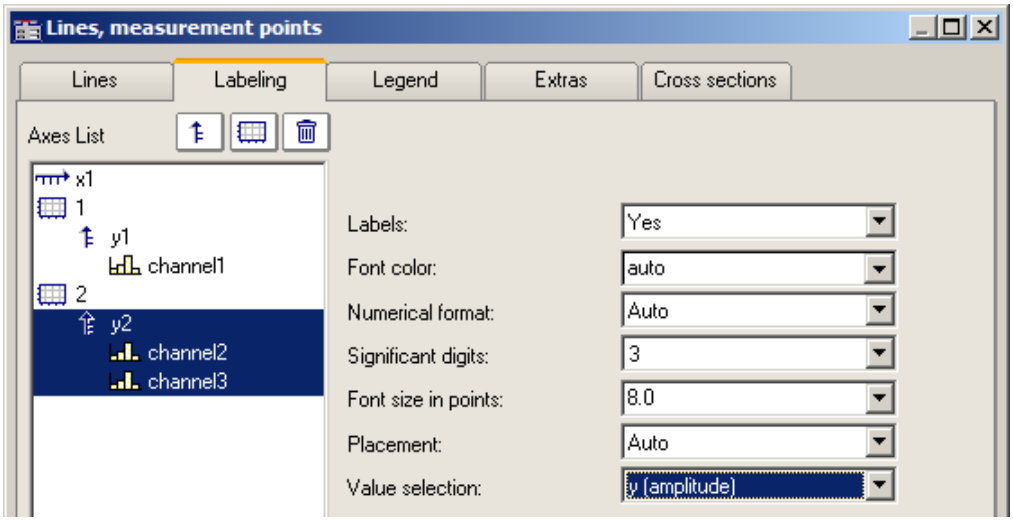
It is possible to generate a **pure** area without any borderline by setting ***Line type*** = ***Only symbols*** and ***Symbol = empty***.

13.6.4.2 Labeling



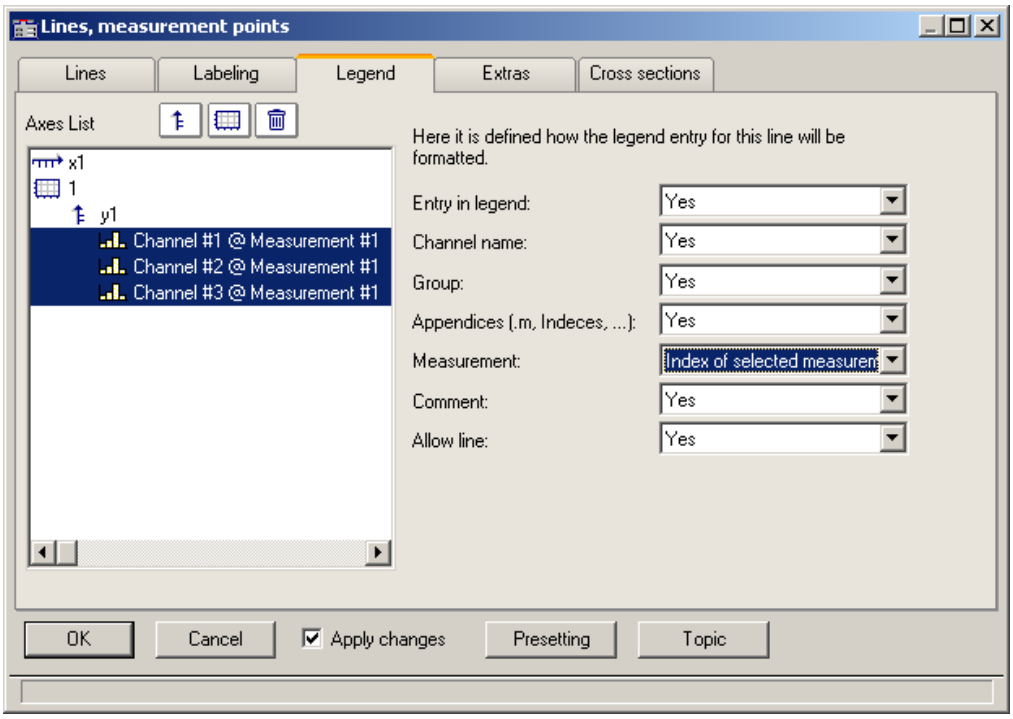
Here you can set whether or not the measurement points are labeled with text stating their numerical values.

For the ***labeling*** of the numerical values, the ***color***, ***format***, ***number of valid digits***, ***font size*** and ***position*** can be set.



Using ***Value selection***, you can set which values are displayed: y, x, parameter, magnitude and phase

13.6.4.3 Legends



Options for displaying the legend

By selecting this tab, you can select various options for displaying the legend pertaining to the line selected.

Note

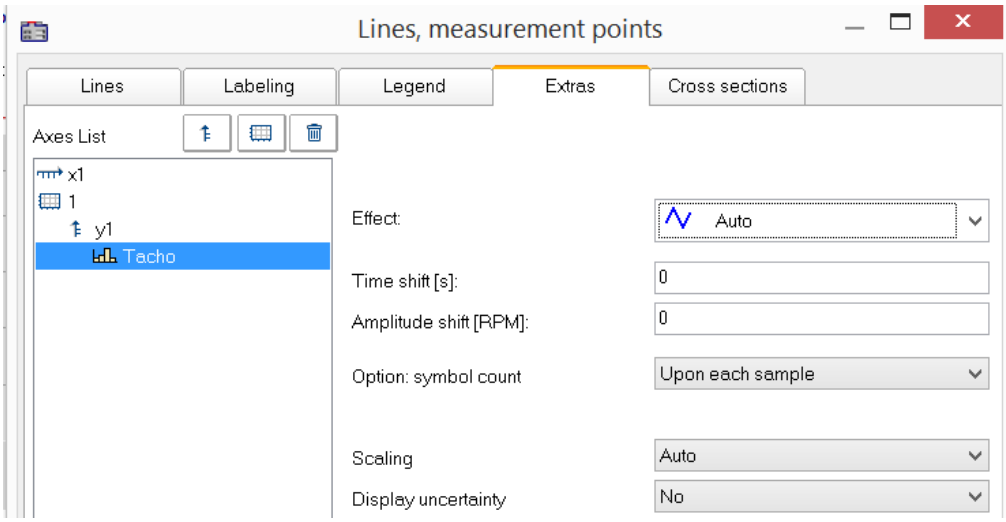
If you display only one data set in the curve window, the legend is only displayed if you have set the control for *Display\Legends\Display Legends* 1382 to *always*.

Settings	Description
Entry in legend	The option *No* means that no legend is displayed at all. Selecting *Yes* means all options are shown.
Channel name	Determines whether the channel name is displayed.
Group	Specifies the group for grouped data sets, e.g. Measurement1:Channel_01
Appendices (m, Indices, ..)	Indication of component or event number, e.g. Spectrum.b
Measurement	Here, the option *Number of the selected measurement* is especially interesting, which enables display of the measurement number instead of the measurement name.
Comment	This is the channel comment as described in the imc FAMOS manual, in the chapter Properties/ Charact. Dialog (Waveforms). For channels recorded using imc STUDIO, the comment can be entered as a channel property.
Allow line	This determines whether an example of the line's appearance is displayed to the left of the label.

Additional general legend settings are described here: *Menu Configuration / Legends* 1382.

13.6.4.4 Extras

On the page *Lines\Extras*, special display options are available:

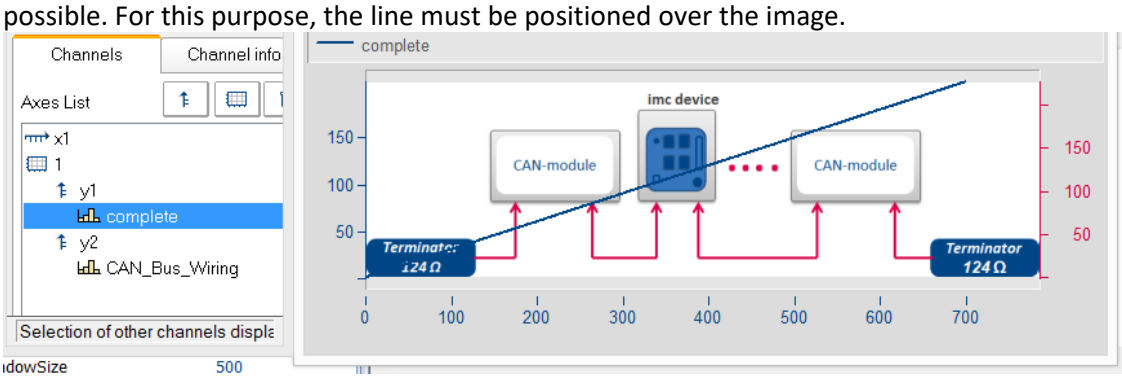


- Time-/Amplitude-Shift 1378: Shifts the line in the curve window without changing the values of the variable.
- RGB-image 1377: Display of an image made of segmented data.
- Special color map display 1378
- Reduction of the symbol display 1379
- Measurement uncertainty 1380

***Effect*: Image from RGB-values**

When the color-flag is set, for RGB variables, the parameter ***Effect*** in the *Extras* dialog is automatically set to "*Image from RGB-values*". Otherwise, it can be set manually here.

The property applies to each line separately, which means that superpositioning of an image with a curve is possible. For this purpose, the line must be positioned over the image.

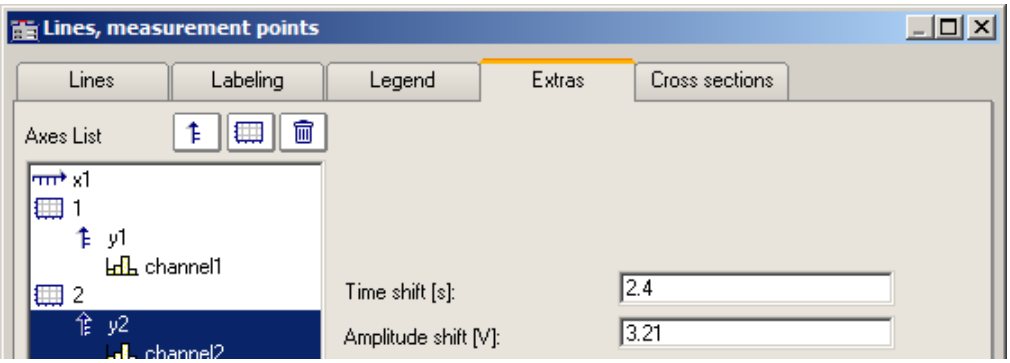


References

General info on RGB-images is presented in the chapter Display variants of the curve window 1338

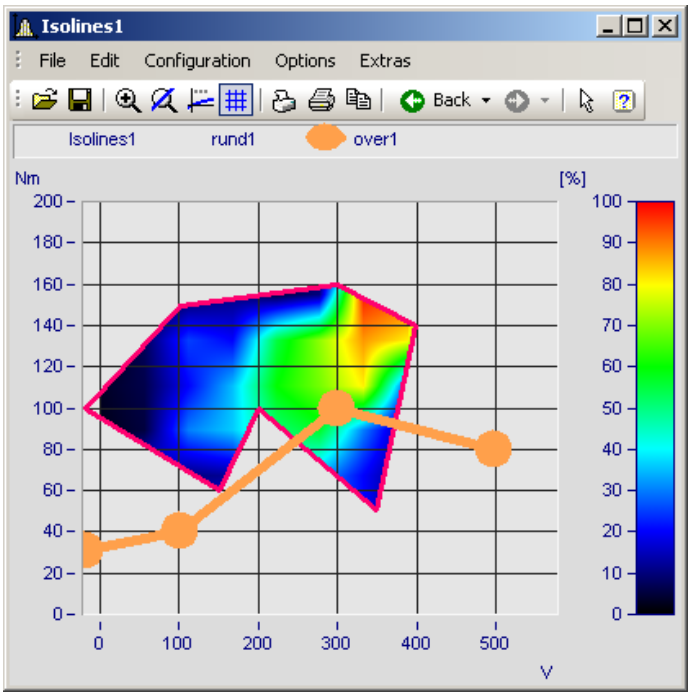
Info on aspect ratio and resolution is in the chapter Axes configuration/Arrangement 1361

In display mode *Default*, you can manually specify the parameters of the Line-Shift function here. For more info, see the section Line-Shift 1476.



Data are moved in the picture in terms of the X- and Y-coordinates.

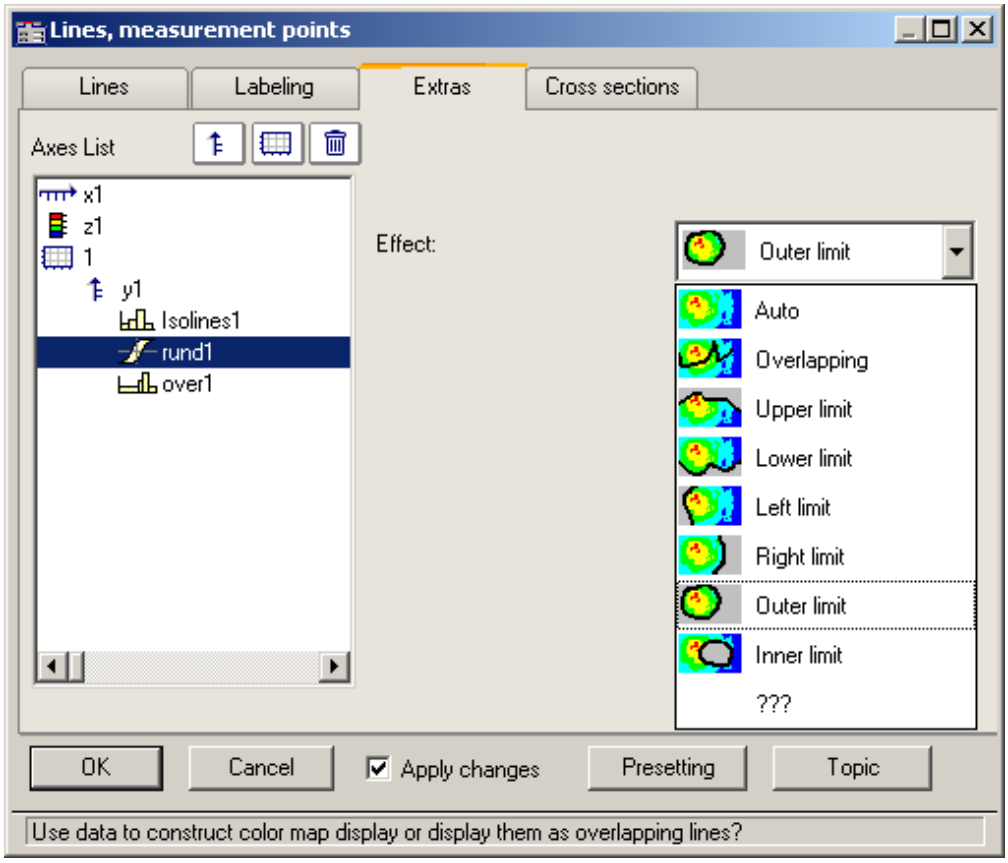
In the display style *Color map*, this menu contains the function *Effect*, with which you can use XY-data to construct the color map, or to superimpose as actual lines on the color map. This makes such displays as shown below possible.



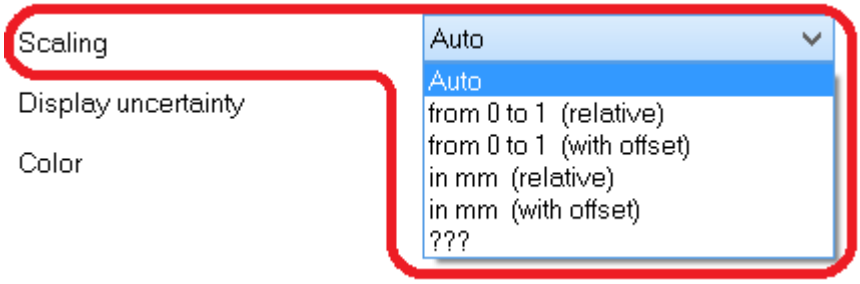
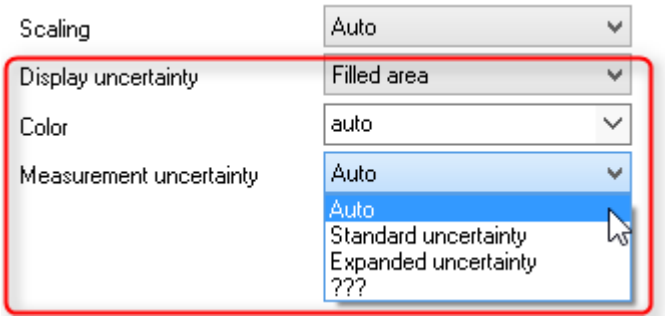
In this picture you see a function for constructing the color map, and a function for superimposing a line over the color map.

Settings	Description
Effect	The available *Effects* are 1) *Overlapping* for superimposing a line over the color map, and 2) various margin functions affecting the construction of the color map.

Effect	Description
Overlapping	An XY-waveform is superimposed as a line over the color map. For the purpose, the usual line properties are available.
... limit	An XY-waveform is inserted into the color map as a real line and frames the color map view as either the upper, lower, left, right, outer or inner margin, according to the setting for *Effect*.



Symbol	The number of symbols can be defined with "*Option: symbol count*" and "*Number of symbols*". Symbols are activated on the line tab 1371 .

Settings	Description
Scaling	 Independently of the coordinate system, the data can be scaled in millimeters. By this means, it is possible to use a data set to define a graphical object, which retains a constant size regardless of the current zoom level. Alternatively, the data can be displayed in a value rage from 0 to 1 across the entire coordinate system. The position of the data can be set to a particular coordinate by means of an offset. This is accomplished using the user-defined property *Offset X display* or *Offset Y display*
Display uncertainty (tolerance range)	 The data set's measurement uncertainty can be indicated either as a *Line* or *Filled area*. The value of the measurement uncertainty is entered as a property of the variable already when the data are captured, and it appears in imc FAMOS as a *user defined channel property*. Alternatively, it can also be set in imc FAMOS as a *user defined property* in the *channel property* dialog, or by means of the function "UserProSet(Data, "Uncertainty"...). Example: UserPropSet(Data, "imc33", 0.3, 0, 0) or UserPropSet(Data, "Uncertainty", 0.3, 0, 0) *Measurement uncertainty:* • *Expanded uncertainty*: In accordance with GUM, the expanded measurement uncertainty is a symmetrical interval around the measurement value. • *Standard uncertainty*: Measurement uncertainty expressed as standard deviation.

13.6.4.5 Cross sections

Here you can set the default appearance of cross-sections for x-, y-, and z-data sets or segmented waveforms or segmented waveforms with the respective position. For more info see the section Connect with 3D 1475.

13.6.5 Additional display options

13.6.5.1 Display

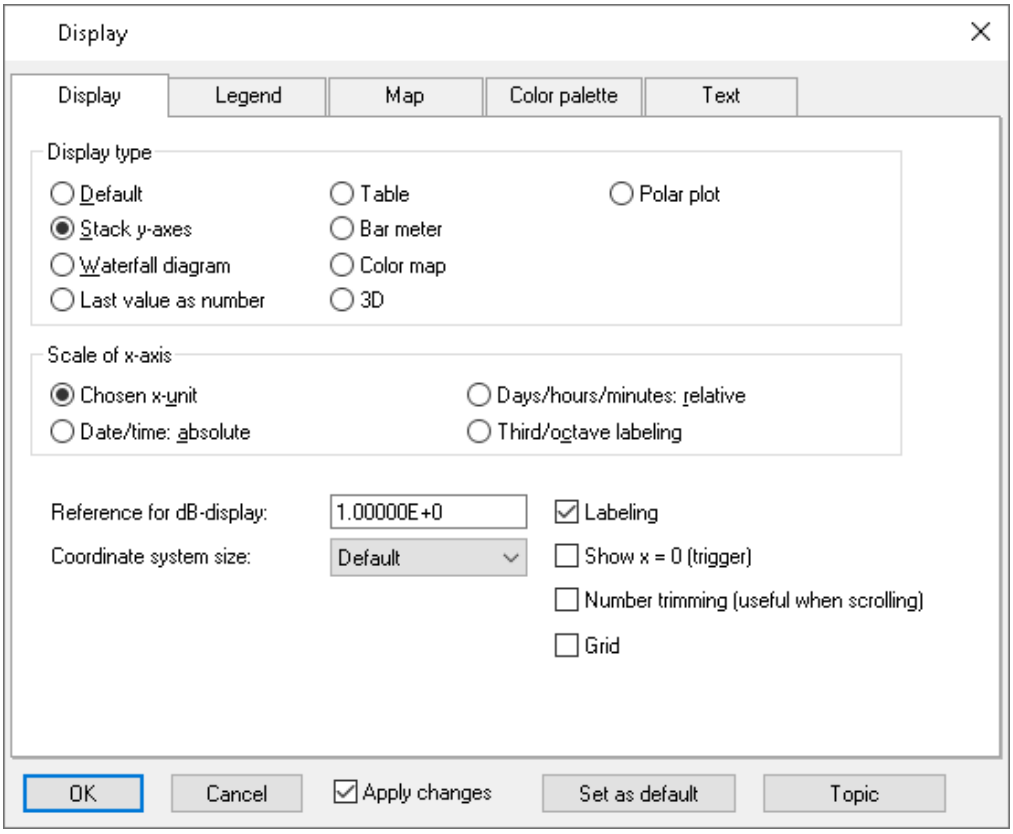
Function

The display mode of a curve window can be defined in a variety of ways.

Calling the dialog

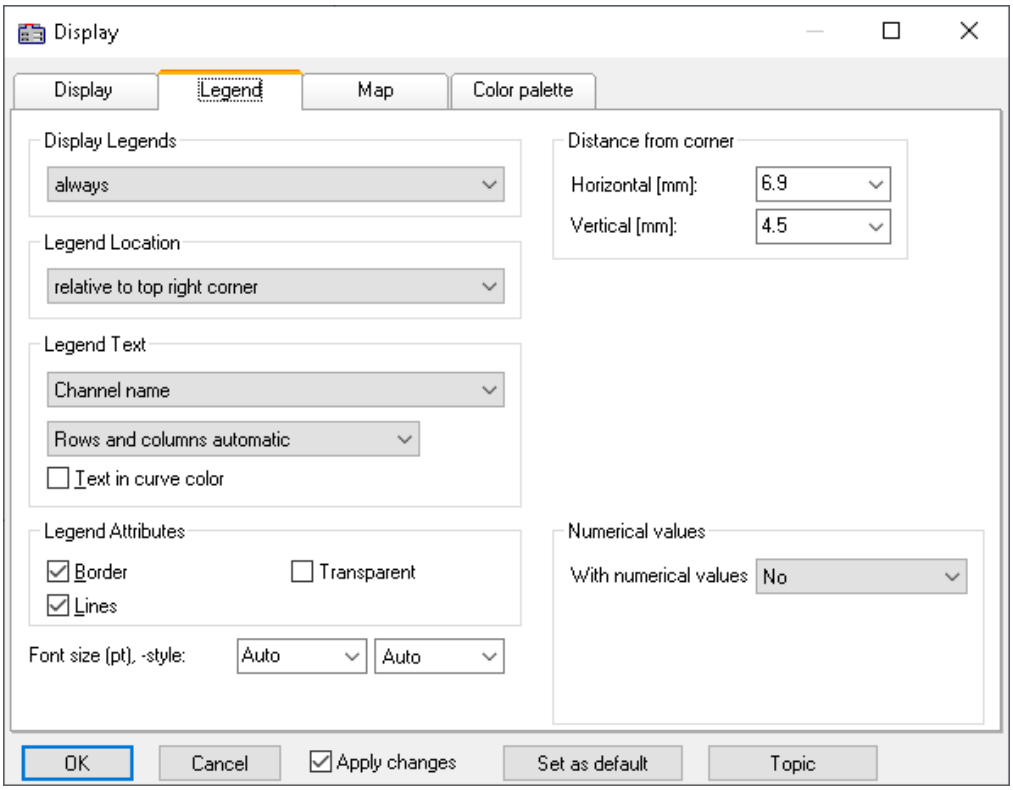
Select menu option "*Configuration*" > "*Display*" in the curve window. A dialog appears in which the curve window display can be defined.

Tap	Description
Display 1267	Display variants of the curve window
Legends 1382	All curve windows can include a legend, with the channel (variable) name, line color etc.
Map 1385	Map lets you place a picture in the background. It is additionally possible to layout the picture in specified coordinates.
Color palette 1398	A channel's color can be governed by a reference signal's amplitude.



13.6.5.1.1 Legends

All curve windows can include a legend, with the channel (variable) name, line color etc. Select "*Configuration*" > "*Legends*" in order to make the desired settings.



Display Legends

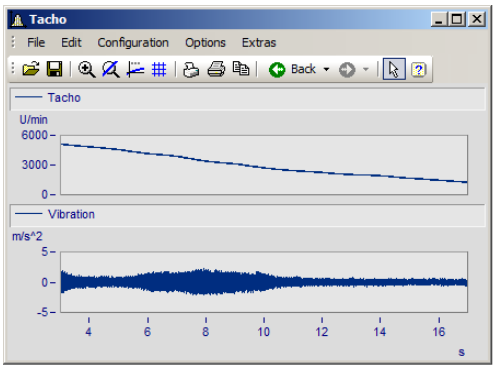
Specify here the conditions under which a legend is to appear. The following options are available:

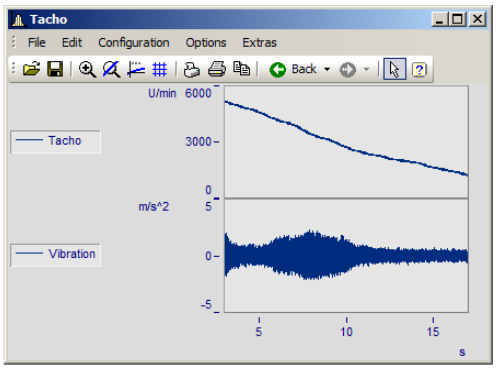
- *Automatically*
- *Always*
- *Never*
- Only if more than one curve is present

Legend Location

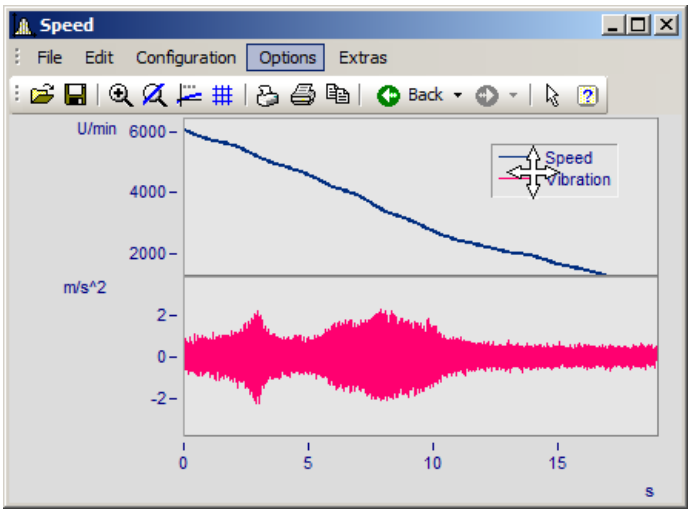
By default, the legend for all variables appears above the curve.

Alternatively, the combobox control *Legend Location* allows the legend to be placed above or to the left next to each coordinate system.





For making placements **within** the coordinate system, additional entries are available. It is possible to move the legend to anywhere within the coordinate system by means of the mouse.



Legend Text

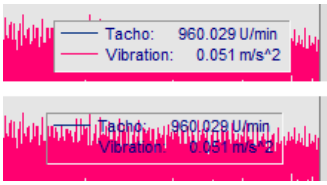
Specify here the text which is to appear in the legend, as well as the format and color of the text. The following options are available:

Settings	Description
Channel name	the variable name, with group name if available, is displayed. In the Data-Browser, affiliation to a membership is indicated by *Channel @ Measurement*; if there is no such affiliation, only the channel name is displayed.
Channel name (without group name)	only the variable name is displayed.
Channel comment	the comment entered for the channel on the "Characteristics"-file card in the imc FAMOS dialog *Variable/Properties* is displayed. See also the notes on Greek letters 1512.
Channel name and comment	the variable name, with group name and corresponding comment are displayed.
Channel name (without group name) and comment	only the variable name and the corresponding comment are displayed.
Channel name without measurement	only the channel name will be displayed. Even if the channel belongs to a measurement, the name of the measurement will not be displayed. - Only relevant in the Data-Browser.
Channel name with index of selected measurement	if the channel belongs to a measurement and if that measurement is currently selected, then the index (number) of the selected measurement will be displayed instead of the name of the measurement. If no index is available, then the name of the measurement will be used. If no measurement is available, then only the channel name itself will be displayed. - Only relevant in the Data-Browser.

The legend text can be ordered in any of the following ways:

- Rows, columns automatic
- Always row
- Always 1 column
- Fixed number of rows (enter the number)
- Fixed number of columns (enter the number)

Legend Attributes



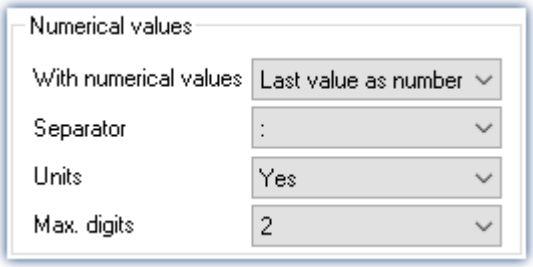
The option *Border* places the legend in a frame, which provides a 3D-effect.

Lines draws a line behind the variable's name in the color of the associated curve.

Transparent makes the curves visible behind the legend.

Numerical values

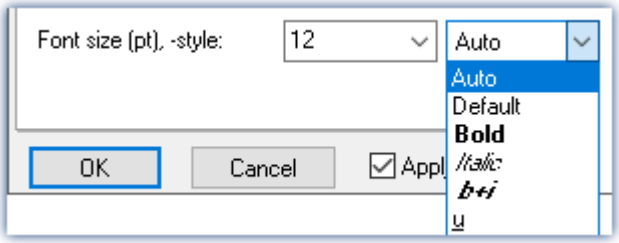
With running measurements, it is possible to display the updated values in the legend. To do this, in the box of controls "*Numerical values*", select the list box entry "*Last value as number*" for the control "*With numerical values*".



Settings	Description
Separator	*Separator* inserts a colon or equal sign.
Units	*Units* controls whether or not the units are indicated: *yes* or *no*
Max. digits	Maximum amount of digits possible. This determines the distance between the variable's name and the numerical value.

Font size (pt), -style

The font size and font style for the legend can be set manually here.



OK, Cancel, Set as default

For details on operation, see the chapter Confirmation bar 1423

13.6.5.1.2 Map

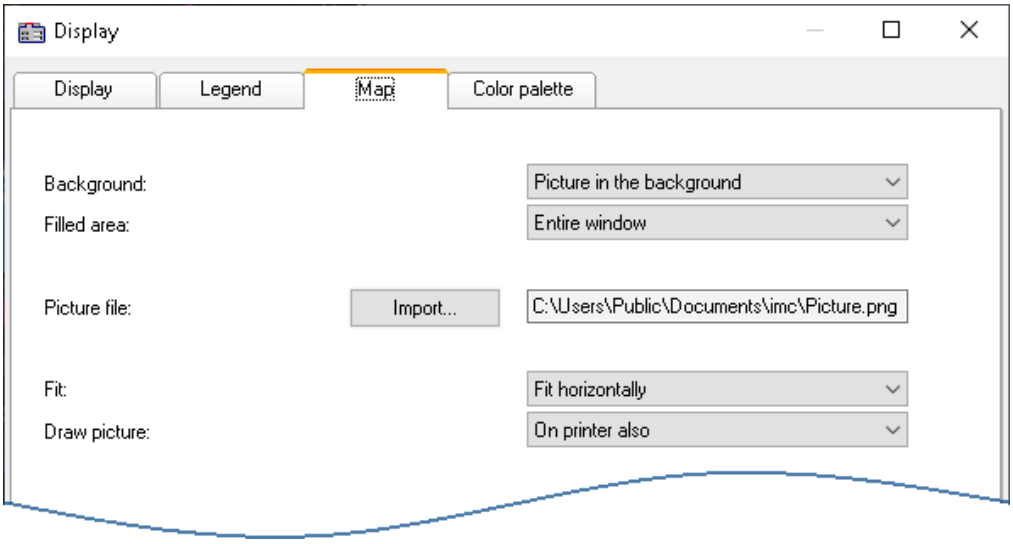
Function

Map lets you place a picture in the background. It is additionally possible to layout the picture in specified coordinates.

Mouse Operation

In the dialog called by the menu item *Configuration / Display*, select the option *Map*.

13.6.5.1.2.1 Background picture



Settings	Description
Background	Select Background to place a picture in the "*Background*"
Filled area	"*Coordinate system*" or "*entire curve window*"
Fit	"*auto*" adjusts both the picture's horizontal and vertical dimensions. "*Retain size*" does not adjust the size but centers the picture. "*Fit horizontally*" and "*Fit vertically*" each adjust the picture's respective dimension.
Draw picture	The option *Only on screen* is used to prevent the printer from printing the picture out.

Constraints

- *Background*: Only for *Default display*, *Stack Y-axes*, *Last value as number*, and *Table*. Applicable with limits to *Color map* and *Bar meter*, since the largest part is hidden. Not possible for *Waterfall* or *3D* display.
- The button *As presetting* is available in either the *Background picture* or *auto* modes. In the presettings, the filename and directory path of the picture are recorded. For any new curve window, the system first looks for this picture there.
- The background picture or the map is saved with a curve configuration file (CCV).
- .BMP files are very large, but can be adapted well to the display. JPG files normally are smaller, but are less capable of being adapted to the display.

13.6.5.1.2.2 Map

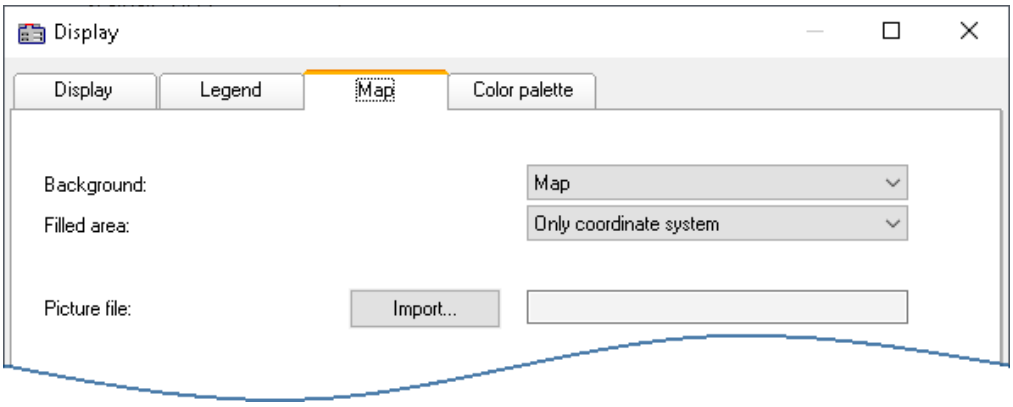
Positioning a map as the background picture

On the following conditions, a picture can be correctly displayed as a map:

- selected curve window configuration: *Display \ Default*
- linear axes
- only one coordinate system

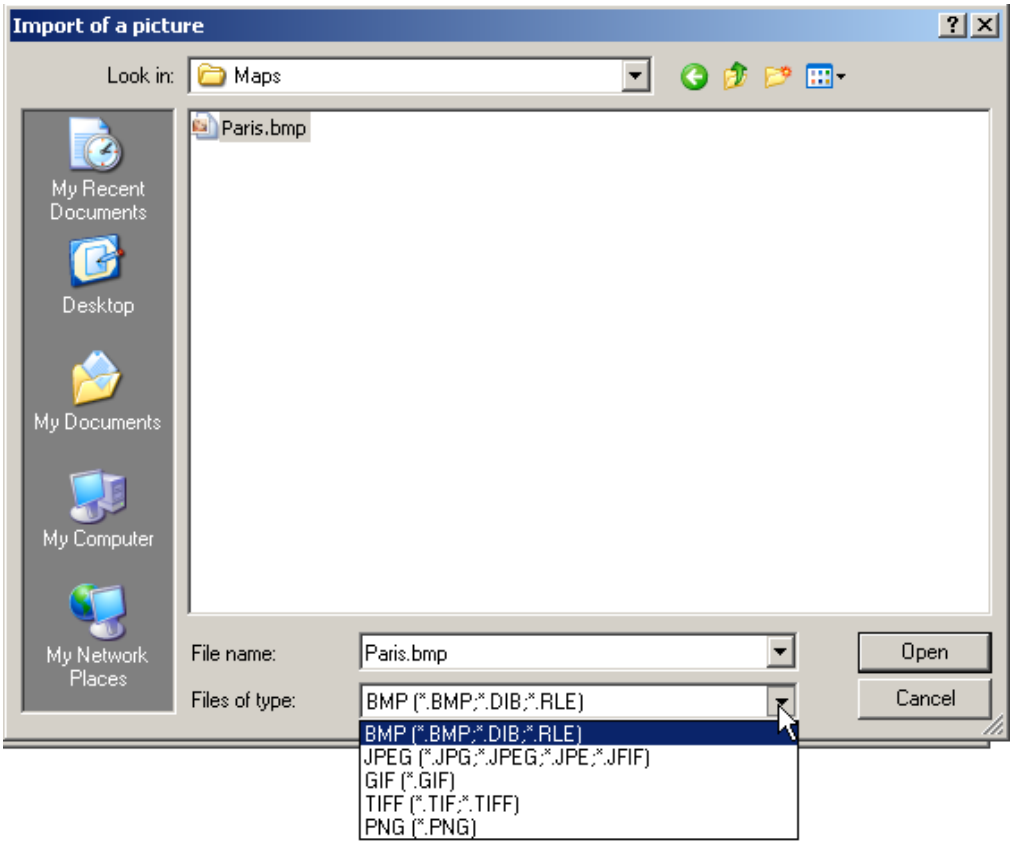
The following example illustrates how GPS data are linked with a map.

Import of a picture



Select the "Map" page and set the background to "Map"

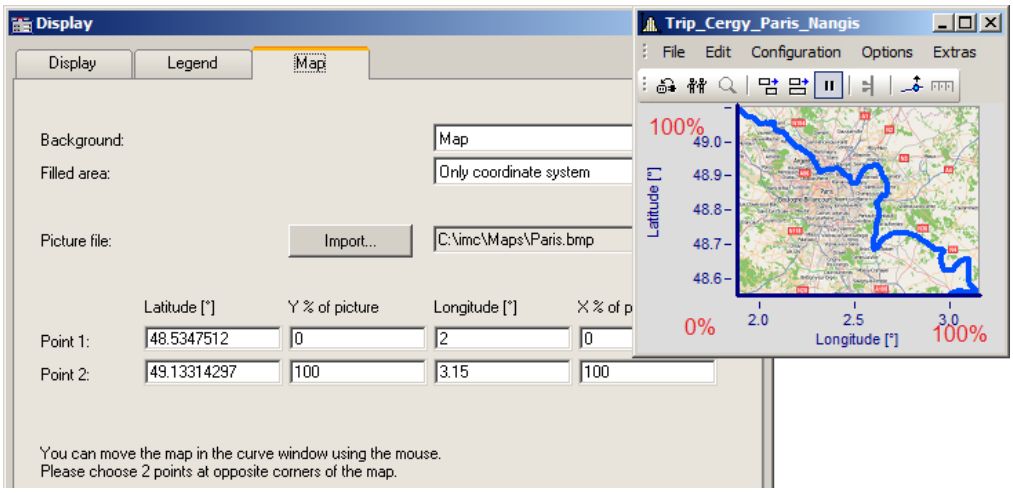
Click on the button *Import* and select a *picture file*. Specify its format.



Selecting a picture file. All customary graphics formats are available.

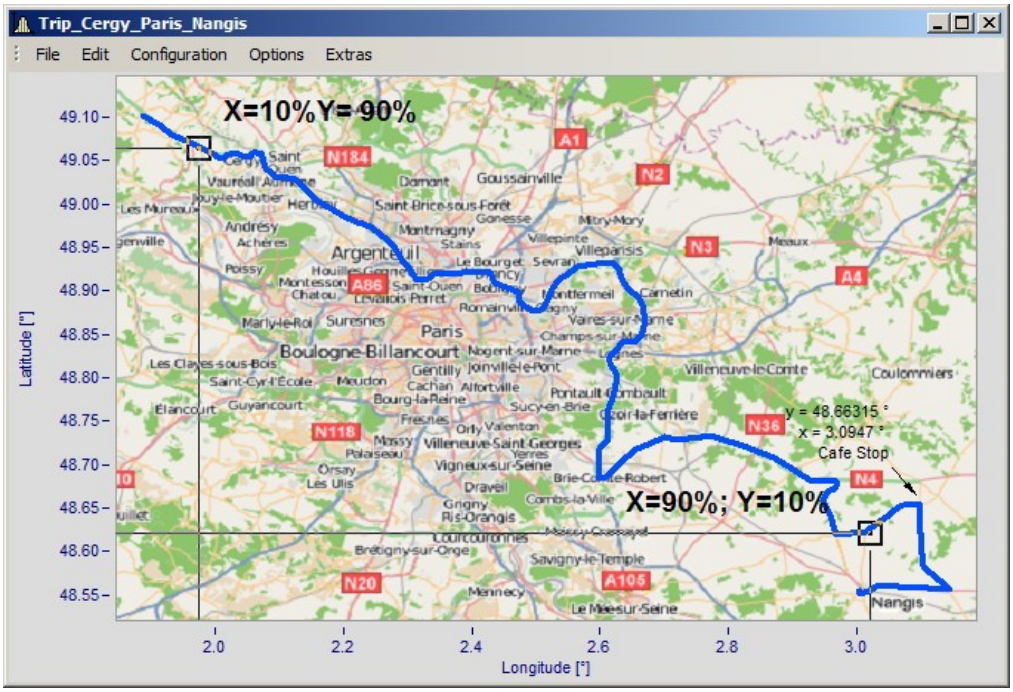
Adapting maps and Map Mode

To position a map, you must specify two points. This accomplished by making entries in the input boxes. The curve window is then in Map Mode.



Specifying points: Point 1 at bottom left, Point 2 at top right

The points are entered into the curve window while the Map dialog is open. Move the points by clicking within the squares and dragging the points using to the mouse to the correct positions.



By moving, the points' X- and Y-positions are updated.

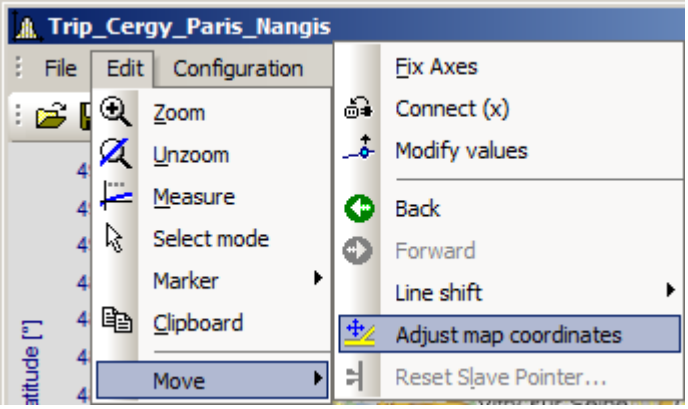
The mouse pointer changes appearance in response to the position in the curve window.

- Drag the entire map by clicking in the center of it.
- Stretch the map by clicking either at the right, left, top or bottom of the map.
- Move the position points by clicking within the squares.

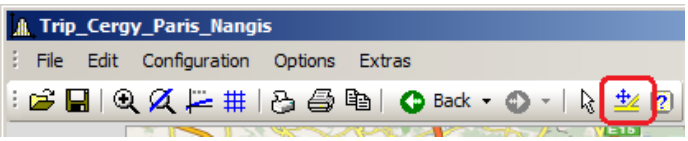
Adjusting the map and points in response to the mouse position

Activating Map mode retroactively

Once the map dialog has been closed, you can use menu commands to adjust the map: *Edit\Move\Map*.



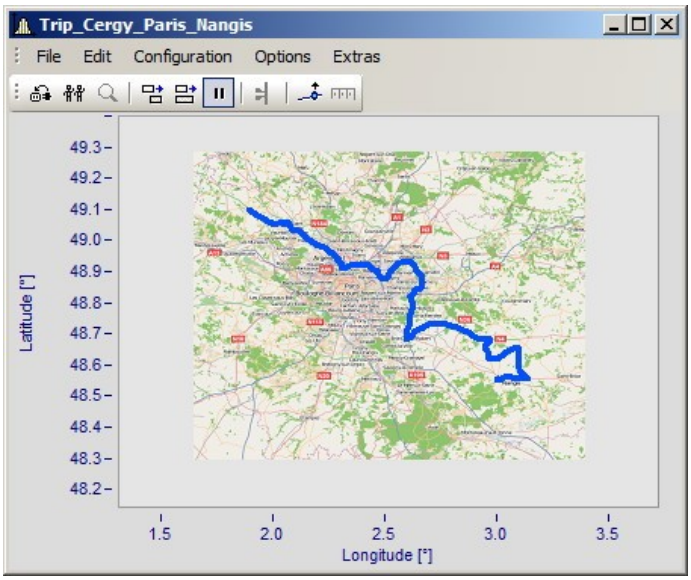
Alternatively, activate the mode by means of the toolbar. To do this, adapt the Toolbar 1505.



You can also drag this function into the toolbar.

Curve window and map

The map is dependent on the axis' coordinates:



This shows the effects of magnifying the axis width

Not that the picture is magnified, stretched or trimmed. Only the details of the original picture can be displayed. The maps used must have an appropriate projection. Constant latitude and longitude lines must be displayed as straight lines. The distances must be equidistant.

Naturally, a map extending from the north to the south pole cannot meet this requirement. The same applies to map material near the poles and the dateline.

Alternatively, you can use a static picture in the background 1385.

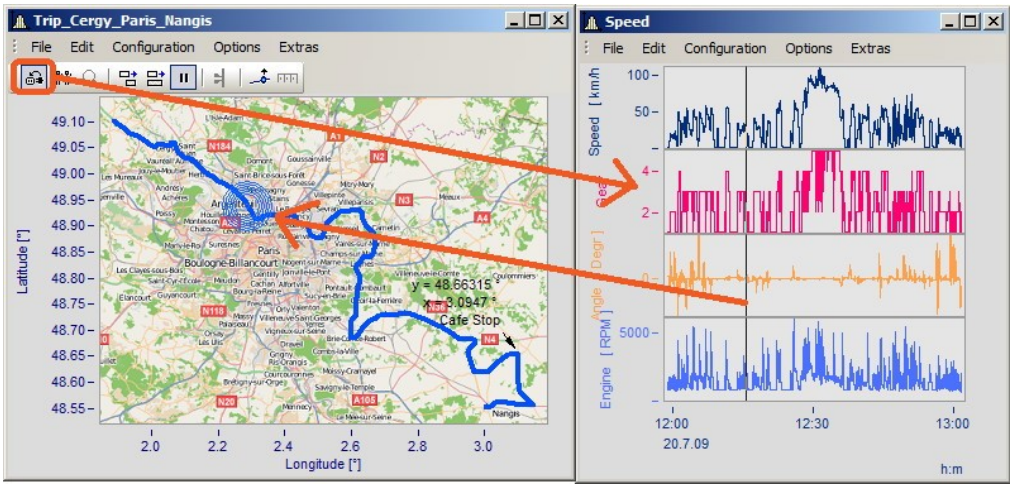
Constraints:

- *Map*: Only for Default display and Stacked Y-axes, but only for the first coordinate system and first Y-axis. All axes must be set to linear (no 1/3-octaves or absolute time).
- The button "*As presetting*" is not available in Map mode.
- The background picture or map is saved with a curve configuration file (CCV).
- .BMP files are very large, but can be adapted well in a display. JPG files are smaller but are not able to be adapted as well.

Linking a map with time data

It is possible to link a curve window displaying time data with a map display.

If the map contains an XY-channel, it is possible to link this channel with time data from a different curve window. To do this, drag the linking tool icon from the Communication toolbar to the desired curve window.



Linking the map with time data from a different curve window

You can move the position on the map in concert with the time marker line in the second curve window.

13.6.5.1.2.3 Map (from Internet)

Automatic loading of a map from the Internet

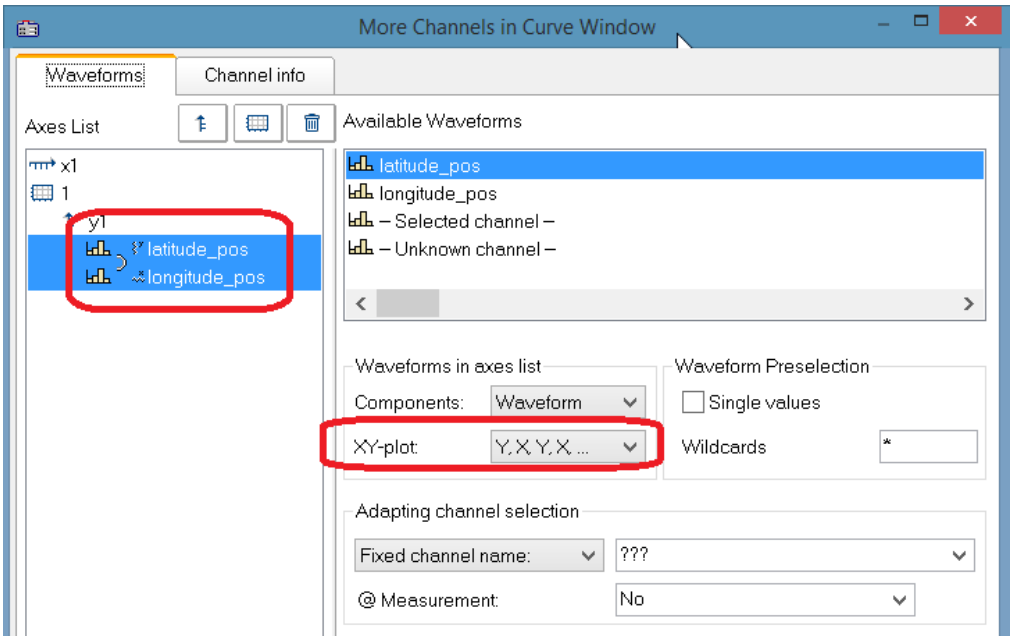
On the following conditions, a picture can be correctly displayed as a map:

- selected curve window configuration: *Display \ Default*
- linear axes
- only one coordinate system
- a data set with plausible position data is displayed.

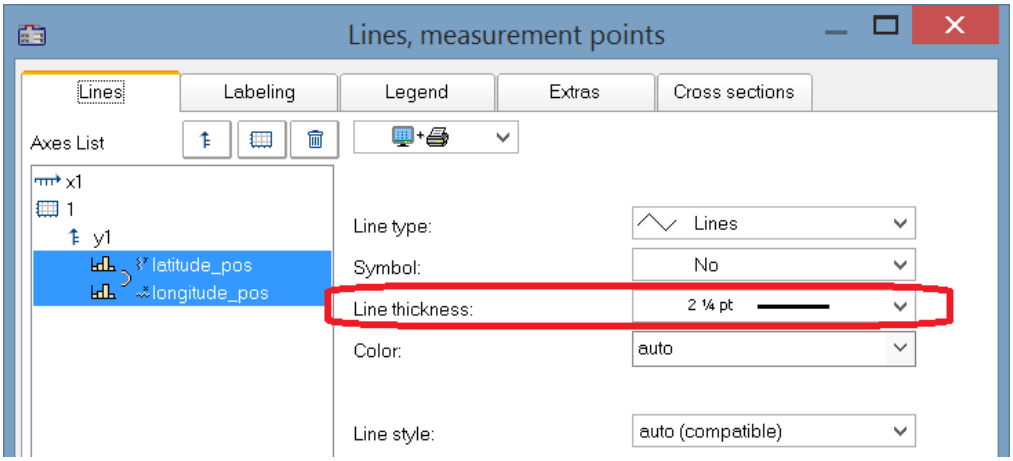
The following example shows how GPS data are automatically superimposed on a map from the Internet.

Selection and display of the GPS data as an XY-plot

Load the data sets for Longitude and Latitude and display them as an XY-plot. To do this, open a curve window with the two components and superimpose them by means of the dialog "*More Channels in Curve Window*":

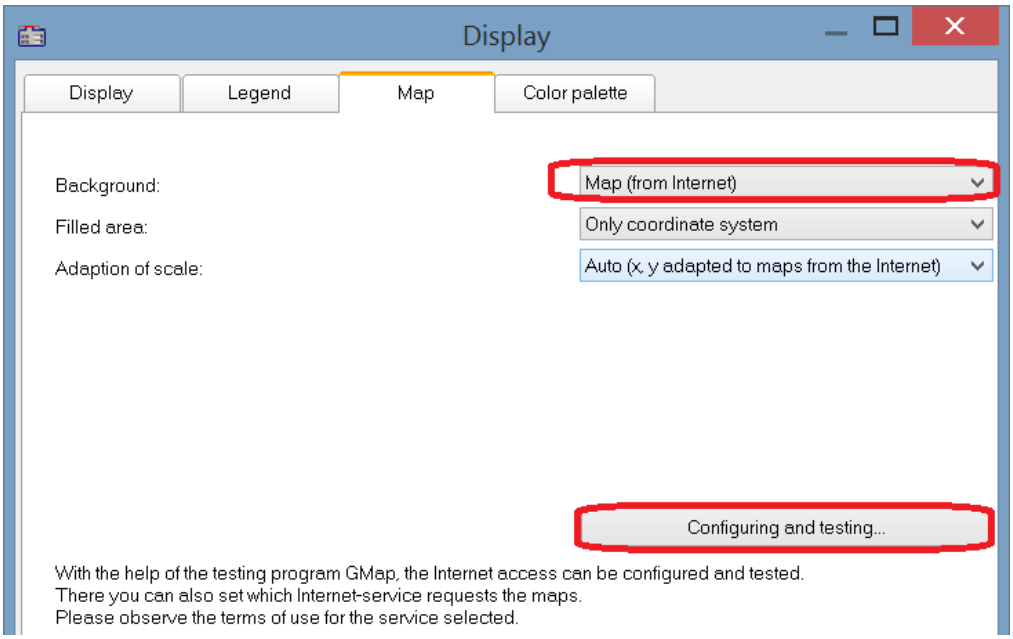


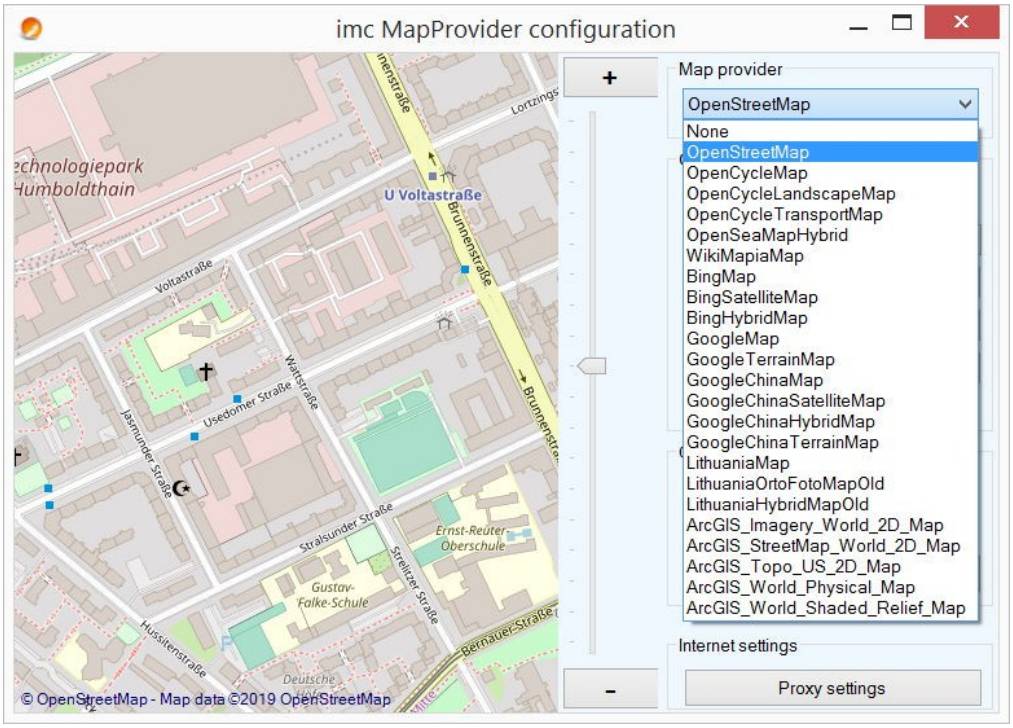
It is advisable to draw the lines more thickly:



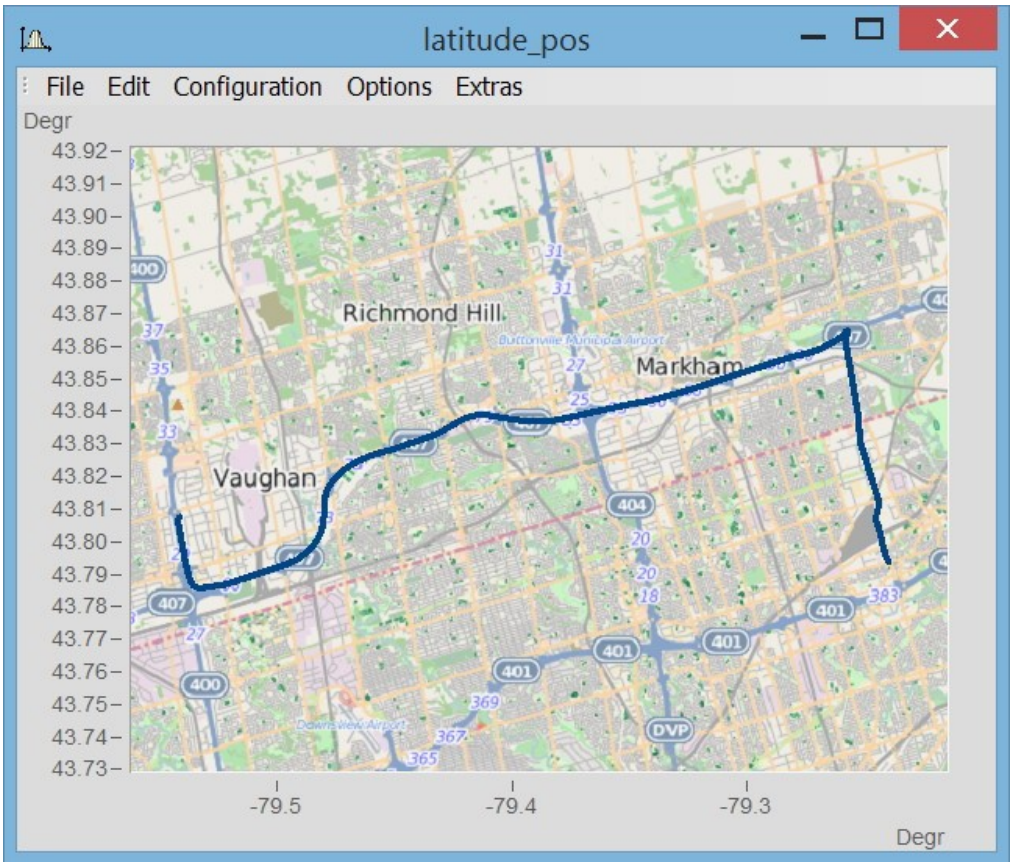
Selecting the map service

Select "*Map (from Internet)*" on the page "*Map*" in the dialog "*Display*". The setting "*Configuring and testing*" selects the Internet-service:





After selecting the map service, the map section loads immediately if an Internet connection already exists. In the curve window, this may appear as shown below, for example:



Note Changing the map-provider

After changing the map-provider, the provider's maps are not automatically reloaded. To do this, the axes of the curve window must be actively changed, e.g. by changing the zoom factor with the mouse wheel or moving the image section.

Adding a map provider

The selection of map-providers can be supplemented with the file "*AdditionalMapProvider.config*" in the folder "*C:\ProgramData\imc\Common\Settings*". If this file doesn't exist on your system, create a text file having this name.

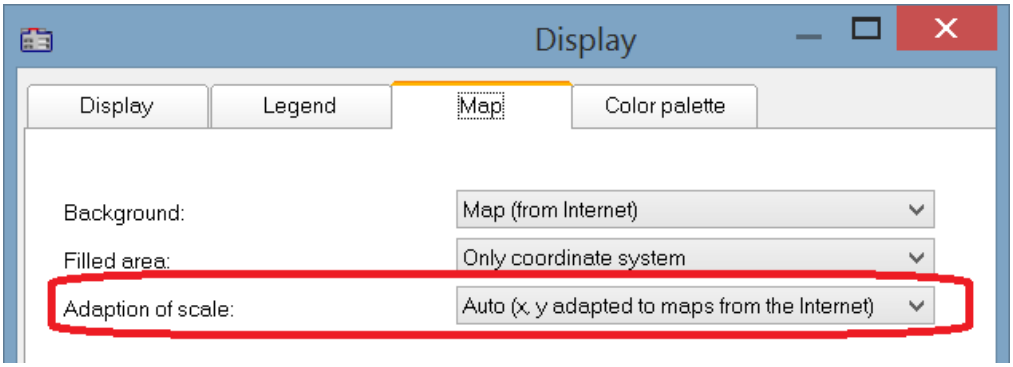
1. Open the folder. By means of *%programdata%*, Windows opens the folder *ProgramData* and then go to *imc\Common\Settings*.
2. The file *AdditionalMapProvider.config* is an XML file with the following structure:

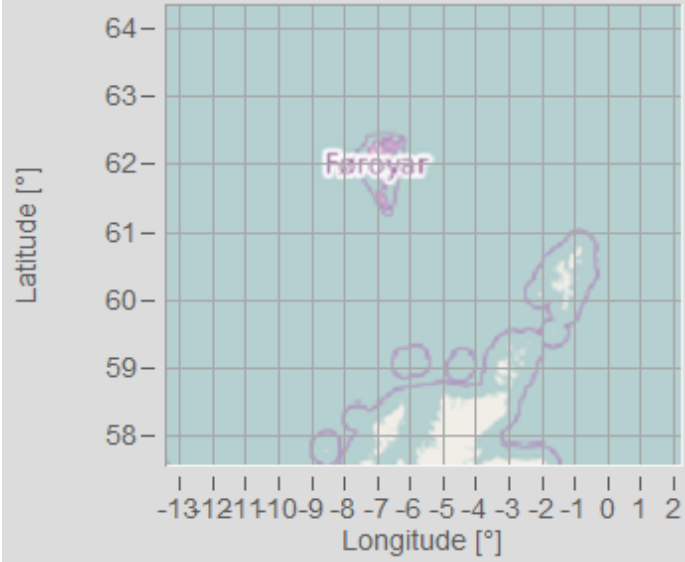
```
<?xml version="1.0" encoding="utf-8"?>
<MapProviders>
  <MapProvider Name="OpenRailwayMap-Standard" Url="http://a.tiles.openrailwaymap.org/standard/{z}/{x}/{y}.png" User="" Pwd="" TileThreads="2"/>
  <MapProvider Name="OpenRailwayMap-Signals" Url="http://a.tiles.openrailwaymap.org/signals/{z}/{x}/{y}.png" User="" Pwd="" TileThreads="2"/>
  <MapProvider Name="OpenRailwayMap-MaxSpeed" Url="http://a.tiles.openrailwaymap.org/maxspeed/{z}/{x}/{y}.png" User="" Pwd="" TileThreads="2"/>
</MapProviders>
```

Replace the lines *<MapProvider Name="... TileThreads="2"/>* with the information pertaining to your map provider.

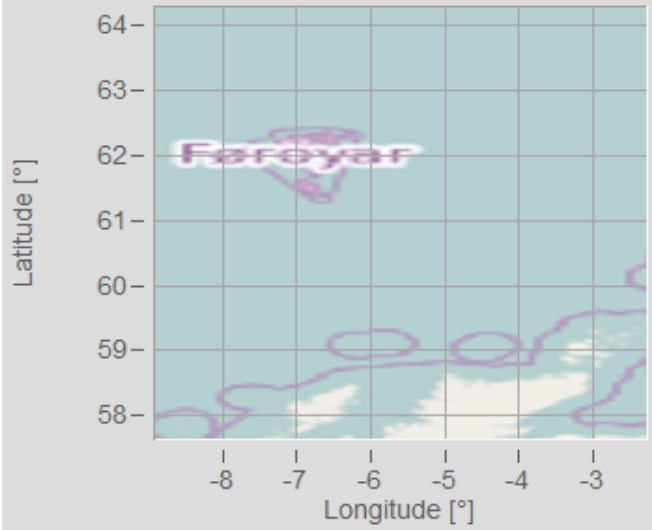
Adaptation of scale

Compensation of distortion due to Mercator projection. Maps typically display the coordinates with a distortion in the direction of the cylinder axis, in order to achieve correct depiction of angles between features on the Earth's surface. In the mode "*Auto (x,y adapted to maps from the Internet)*", the axis are stretched in proportion to the Mercator projection distortion.





Undistorted map accounting for the Mercator projection



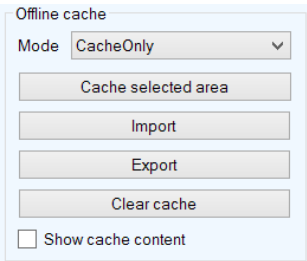
Coordinates equidistantly scaled -> Map distorted

Note

This projection named after Gerhard Mercator (1512-1592) causes imprecision in the curve window, since the coordinates are displayed equidistantly in it. The imprecision increases with the distance from the Equator and the size of the map.

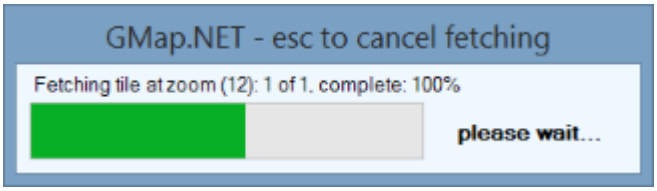
More map service option

Offline cache

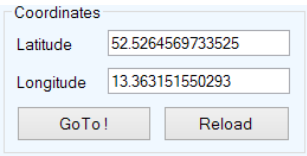


You can export, import, or save a map section locally. To do this, select "*Mode*" *ServerAndCache*" or "*Cache only*".

To make the selection, drag open a rectangular region while holding down the mouse button. When you click on "Cache selected area", the region is downloaded and saved in various zoom levels.

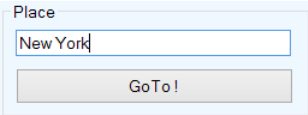


Coordinates



Select the map section by its coordinates and zoom to the desired section.

Place



Select the map section by entering a location.

Internet Settings

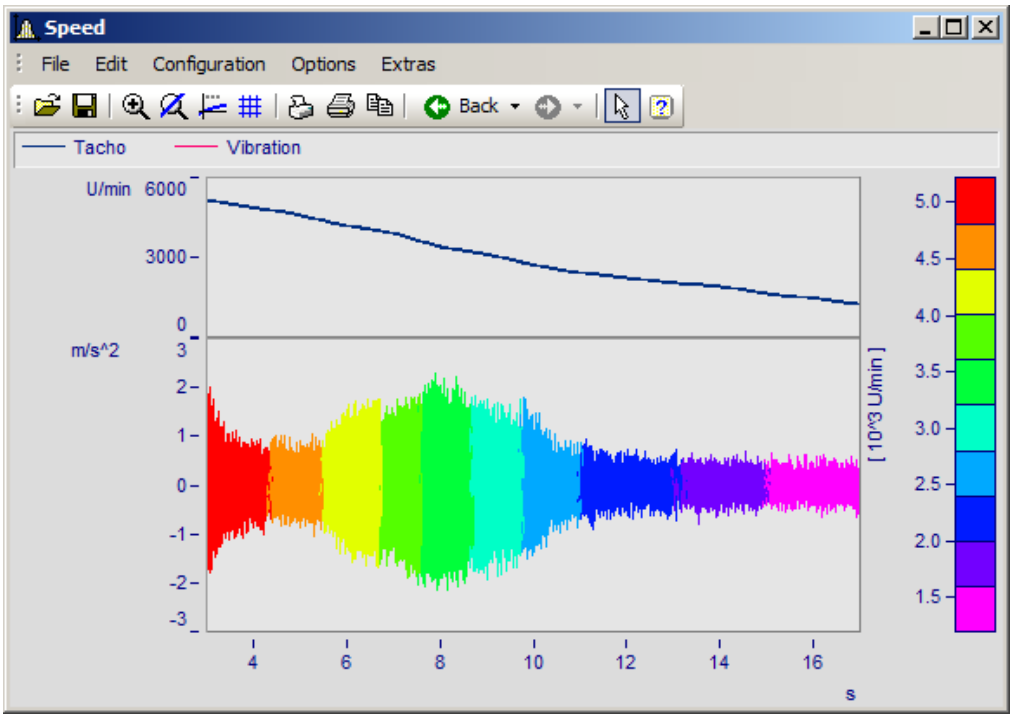
Here you can specify a proxy-server.

13.6.5.1.3 Color palette

A channel's color can be governed by a reference signal's amplitude.

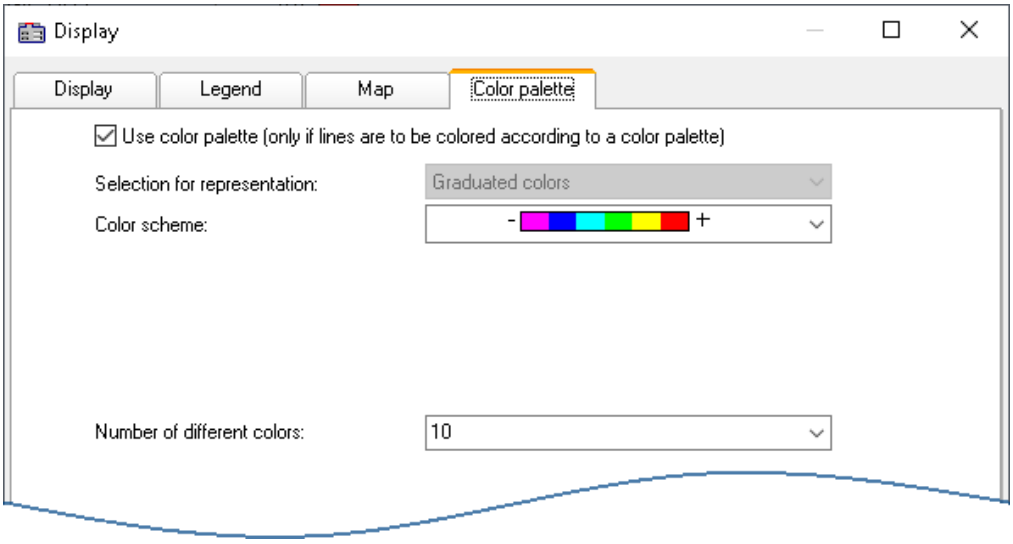
Note

This function is working only for channels using the same sampling rate (x-delta).



Line color of a graph of vibrations as a function of the RPMs

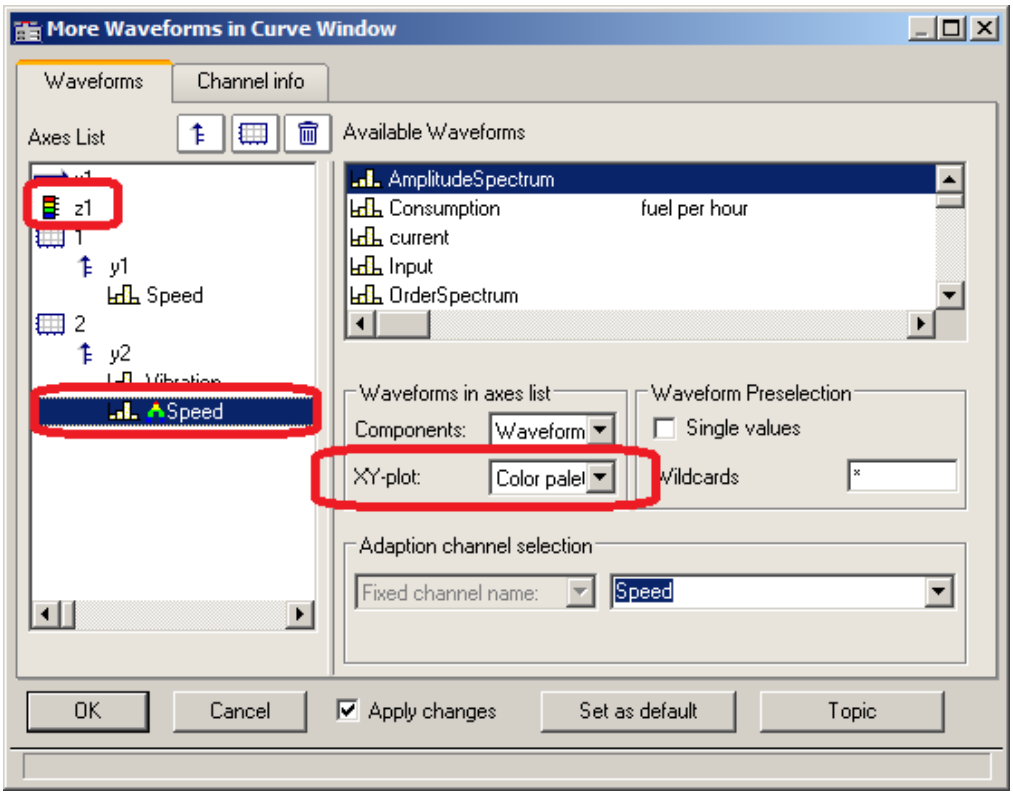
Select the page *Color palette* (not *Display type Color map*) in the dialog called by *Configuration\Display*.



First, activate the color palette by setting a check in its checkbox. In consequence, the controls for *Color scheme* and *Number of different colors*. The control *Selection for representation* is disabled, since at this moment only graduated colors are available.

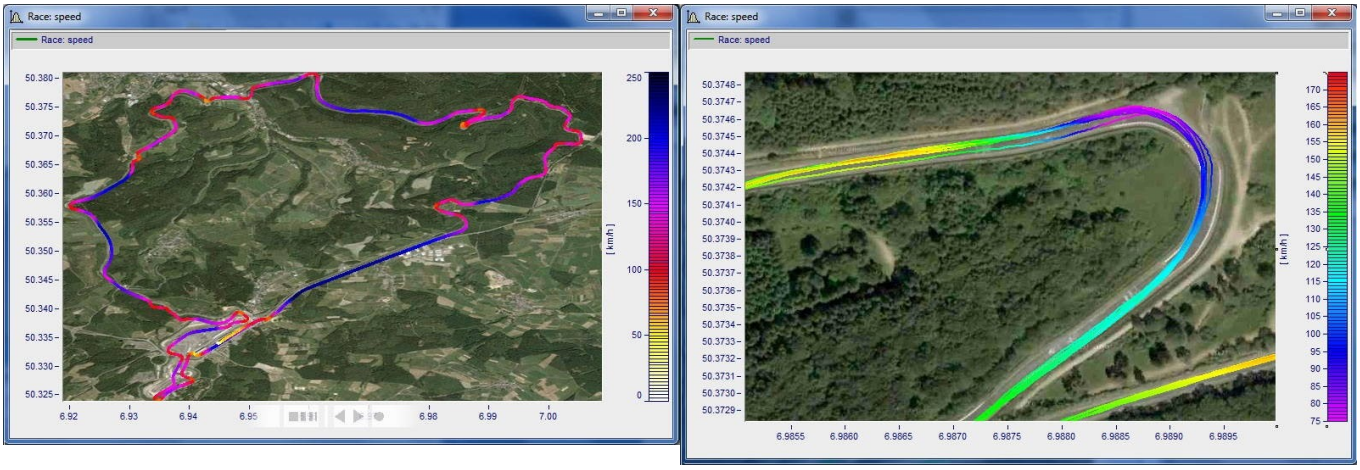
The same settings options for the color scheme and amount of colors are available as for Color map 1306.

The assignment is performed in the dialog *More Channels...,* which is called from the curve window's context menu.



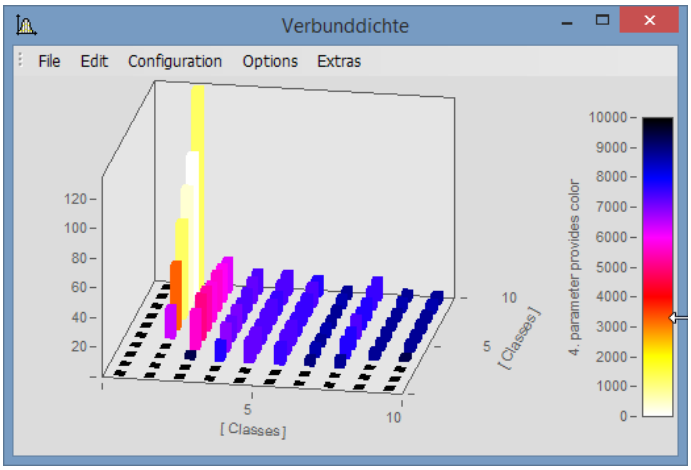
The reference channel is assigned to the same y-axis as the channel to be colored. Select a reference channel and set the *XY-plot* to *Color palette*.

Example:



Color palette as 4th parameter in 3D

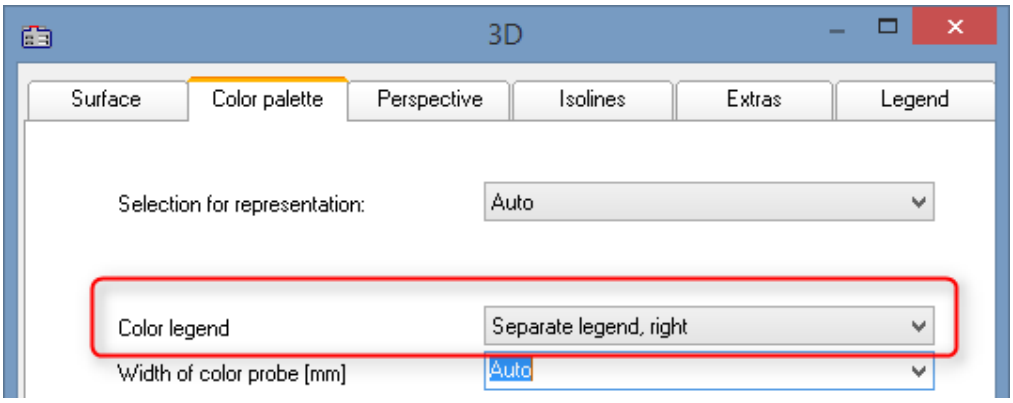
A 3D display can be colored according to a fourth parameter. Thus, 4 dimensions (4D= 3D + color) can be displayed. See also here 1409.



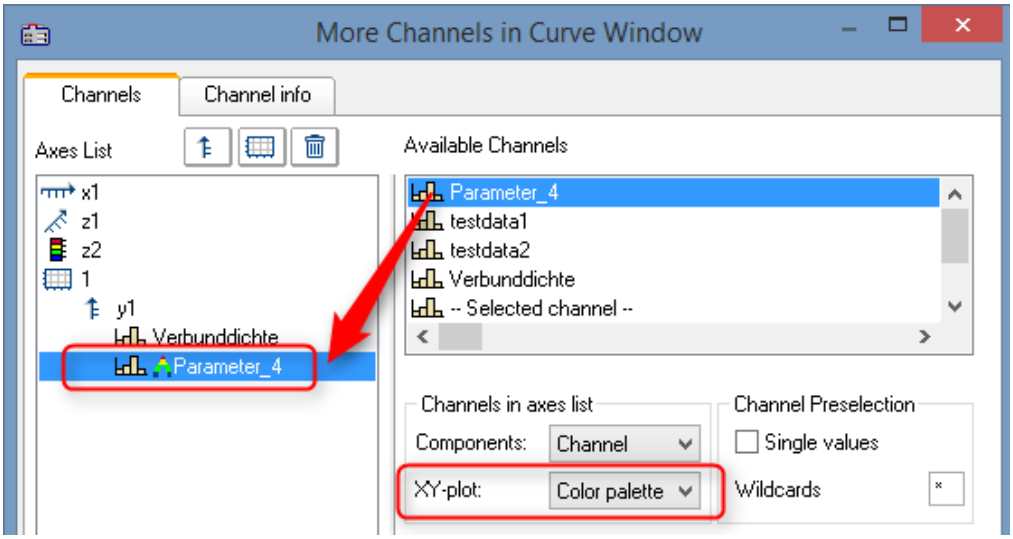
3D display with colors governed by a fourth data set

Procedure:

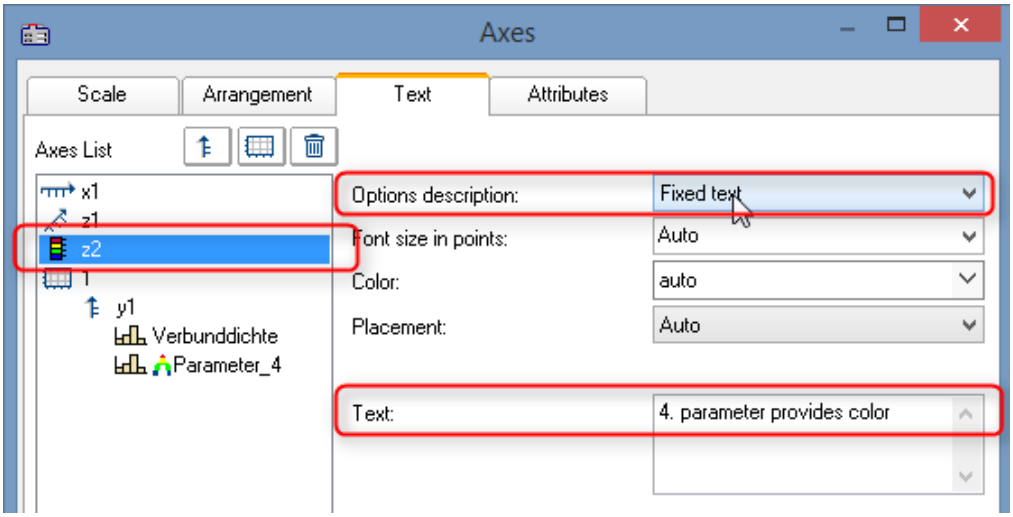
1. Set *3D* display with ***Separate legend right*** or **left** in ***Color plaette*** 1408. With the settings *Auto, No* and *Integrated into y-Axis*, it is not possible to generate a colored overlay.
2. Under *More Channels*..., place the fourth variable directly below the 3D data set.
3. Under *XY-plot*, select *Color palette*.
4. Labeling of the color palette by means of *Axes\Text.* This is necessary since the legend does not provide an indication of the color scale's origin.



Activate Separate color legend at "Configuration\3D\Color palette"



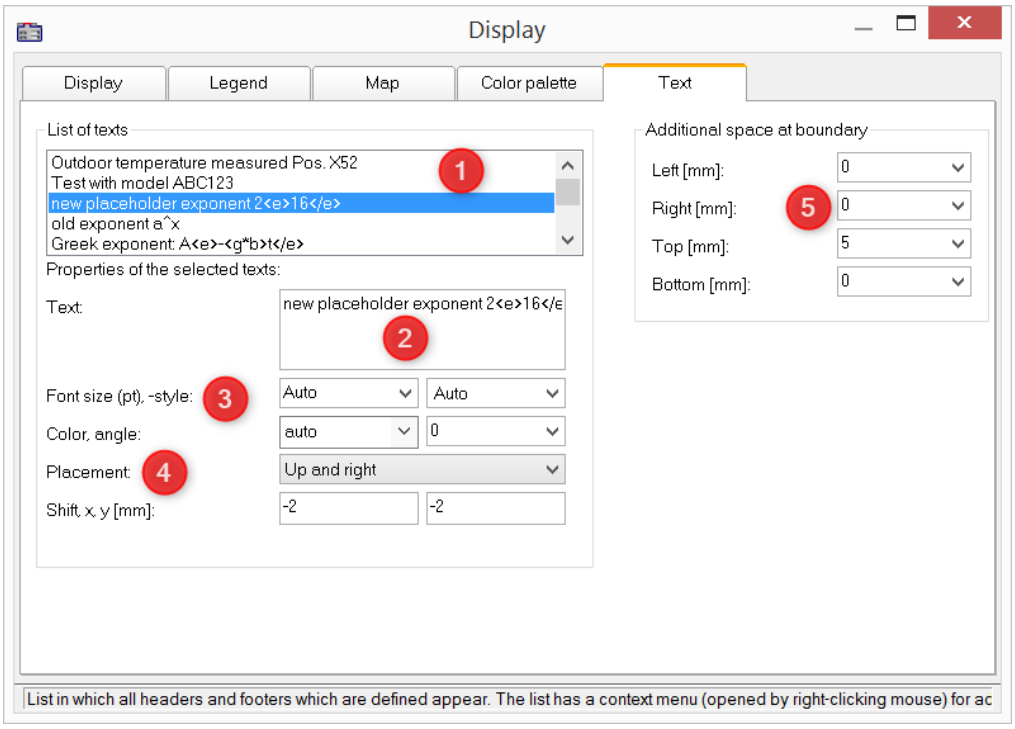
Under "Configuration\More Channels...", assign the color-coded variable

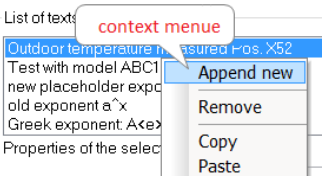


Under "Configuration\Axes\Text" with "Fixed text", state the origin of the coloring.

13.6.5.1.4 Text

The Text page of the Display dialog enables any desired texts to be positioned, for example as titles, headers, footers, or general comments.



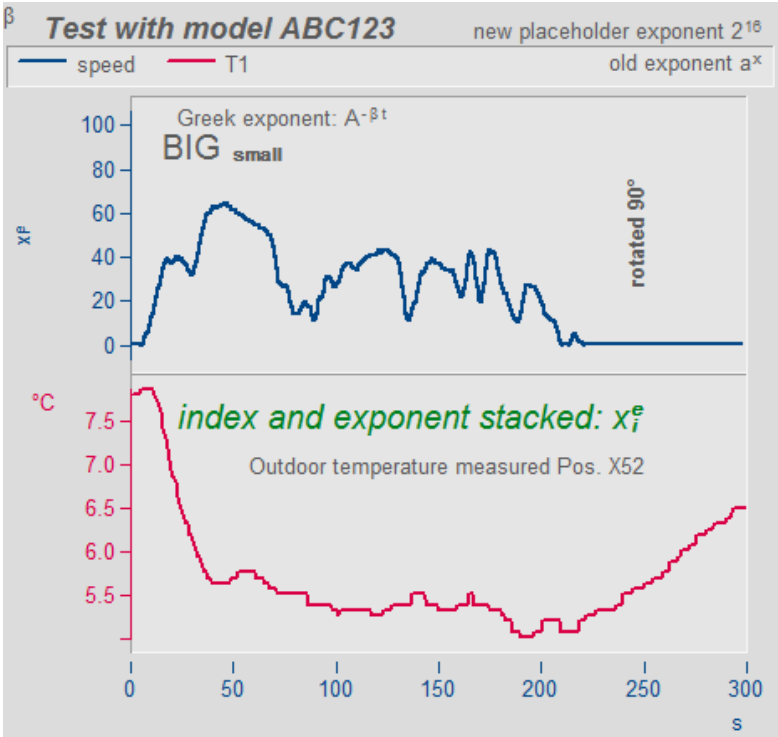


By **right-clicking** in the *List of texts* (1), you open a context menu for managing the entries.

In the *Text box* (2), up to 256 characters are allowed. Line breaks are placed by pressing the CRTL+ENTER keys.

The placeholders supported are described in the chapter *Axes configuration/Text* 1362.

The *Font, font size, -color,* and alignment (3) can be set for each entry.



The *placement* (4) is initially determined from a list and subsequently fine tuned by means of the parameters for *Shifting* in the *x* and *y* directions. When a position is selected which references a **coordinate system**, an additional parameter for specifying the number of the Y-axis appears.

Beyond the space occupied by the actual curve graph, it is possible to make room for additional texts (5).

Remote control via sequences:

The function CwDisplaySet offers some functions "*header.x*" with which to texts can be placed.

```
CwDisplaySet("header.count", 5 )
CwSelectByIndex( "header", 1 )
CwDisplaySet("header.text", "TEXTMITTE" )
CwDisplaySet("header.position", 8 )
CwDisplaySet("header.text.color", 255 )
```

The additional space at the edge is adjusted using the "*legend.x*" functions.

```
CwDisplaySet("legend.space.left", 10.4 )
CwDisplaySet("legend.space.right", 4.7 )
CwDisplaySet("legend.space.top", 10 )
CwDisplaySet("legend.space.bottom", 10 )
```

Reference Further texts

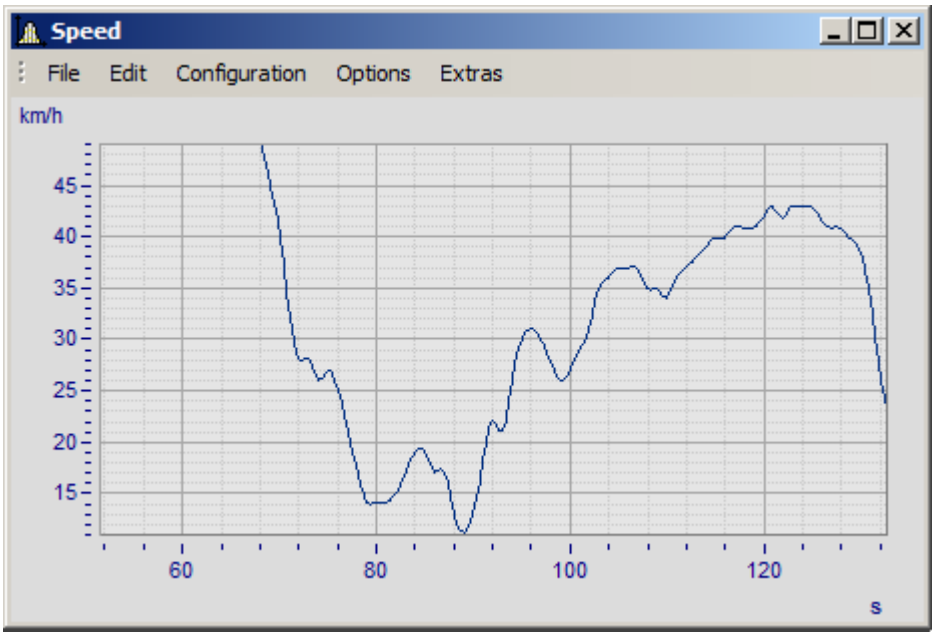
Further texts can be added via the axis dialog 1362 .
Texts can also be defined as *user ticks* 1364. Here you can also find the function "*Label for raw data* 1367".

13.6.5.2 Grid

Function

A grid may be added to the coordinate system in the curve window. The grid is composed of vertical and horizontal lines running through the ticks on the axis, with additional lines between them.

The grid consists of a main and secondary grid. The main grid lines end at the main ticks of the coordinate system axes, where the axes are labeled. The secondary grid lines end at the secondary ticks, optionally inserted between the main ticks (see menu option *Configuration/ Display* 1267). Settings for the secondary grid are made in the Axes dialog by means of the control *Small ticks* 1355.



Various line widths and types can be specified for the clipboard or print-out, see menu *Opt./ Clipboard Settings* 1489.

A grid is drawn for a logarithmically represented axis (but not in dB) so that eight lines appear between each set of markings on the axis. The eight lines are easy to interpret, since a factor of ten lies between each set of axis markings. The lines designate equidistant points, i.e. between the markings 1 and 10, the points 2, 3, 4, 5, 6, 7, 8, 9 and between the markings 10 and 100, the points 20, 30, 40,... . The lines are then spaced as on logarithmic graph paper.

Mouse Operation

- Select menu option *Grid* from the *Configuration* menu in the curve window.

Remarks

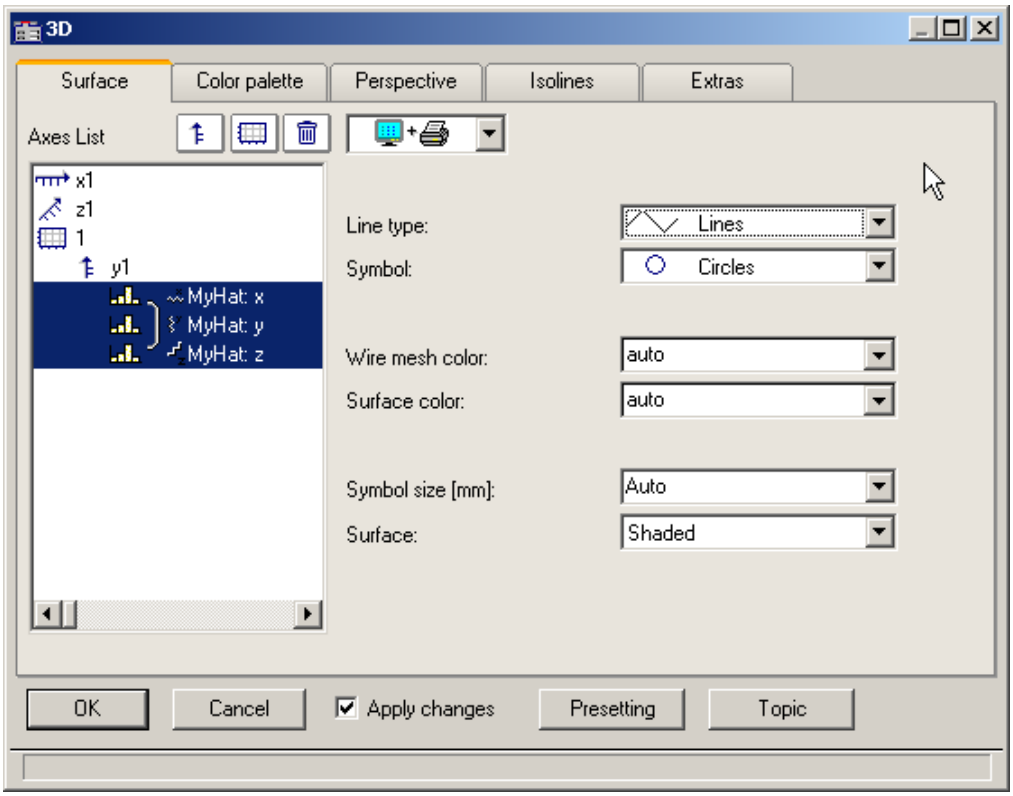
- The color of the grid lines can be determined in the *Colors...* dialog.
- A grid is not drawn if labeling is not visible.
- The menu option *Grid* is marked with a checkmark when grid display is selected.

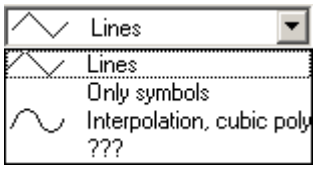
13.6.5.3 3D

You have various options in the *3D* display, to modify and optimize the display of surfaces. If you have selected the *3D* option, you achieve the setting dialog for the 3D design *Configuration / 3D*. You can also achieve this setting dialog with the corresponding button in the toolbar.

13.6.5.3.1 Surface

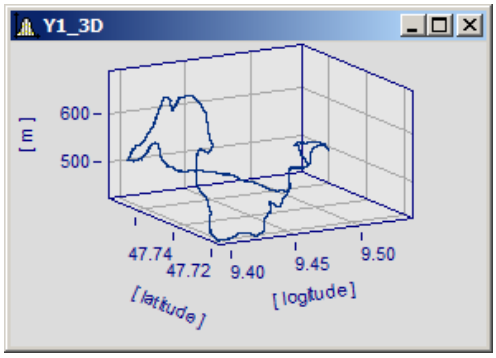
In the dialog page *Surface* there are additional settings for 3D-display besides the settings described under Lines 1370.



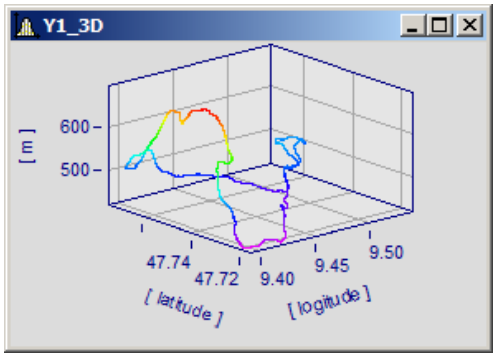
Settings	Description
Line type	In 3D-display, the only available line types are straight lines, no lines, or interpolated lines connecting measured points. For bars, see here 1407. 
Symbol and Symbol size	See the section on lines 1370.
Wire mesh color	Here you can set the color of the grid, or of the connecting lines between measured points. If you set *auto*, then the color will be set according to the global color setting (color palette 1408).
Surface color	Here you can set the color of the surface. If you set *auto*, then the color will be set according to the global color setting (color palette 1408).

Settings	Description
Surface	

Surface	Description
Wire mesh shaded	The individual measured points are connected with lines to a grid and are additionally displayed as a surface with the color/color palette set.
Shaded	The individual measured points are displayed as a surface with the corresponding selected color/color palette.
Wire mesh	The individual measured points are connected to lines in a grid.
Wire grid with color tones	The individual measured points are connected with lines to a grid and this grid is additionally displayed as a surface with the color/color palette set.
Samples	The individual measured points are displayed as points or, if selected, as symbols.
Space curve	The individual measured points are all connected in sequence with lines.
Space curve with color tones	In addition to the space curve, the line can be colored with different colors to correspond to different y-component value ranges.



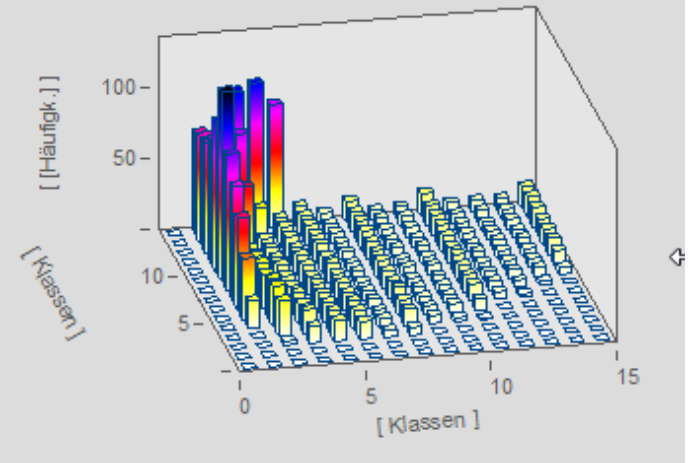
Monocolor space curve



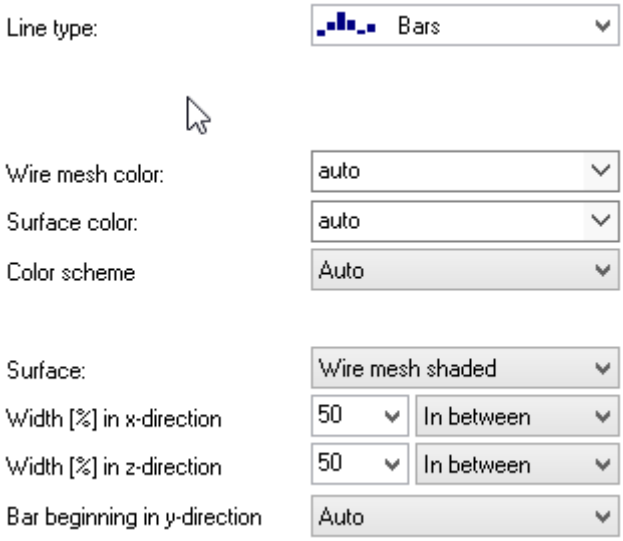
Space curve with color palette

Options for 3D bars

With *Line type: Bars*, three-dimensional bars are displayed, for which additional display parameters are shown.



3D bars



Settings for 3D-bars

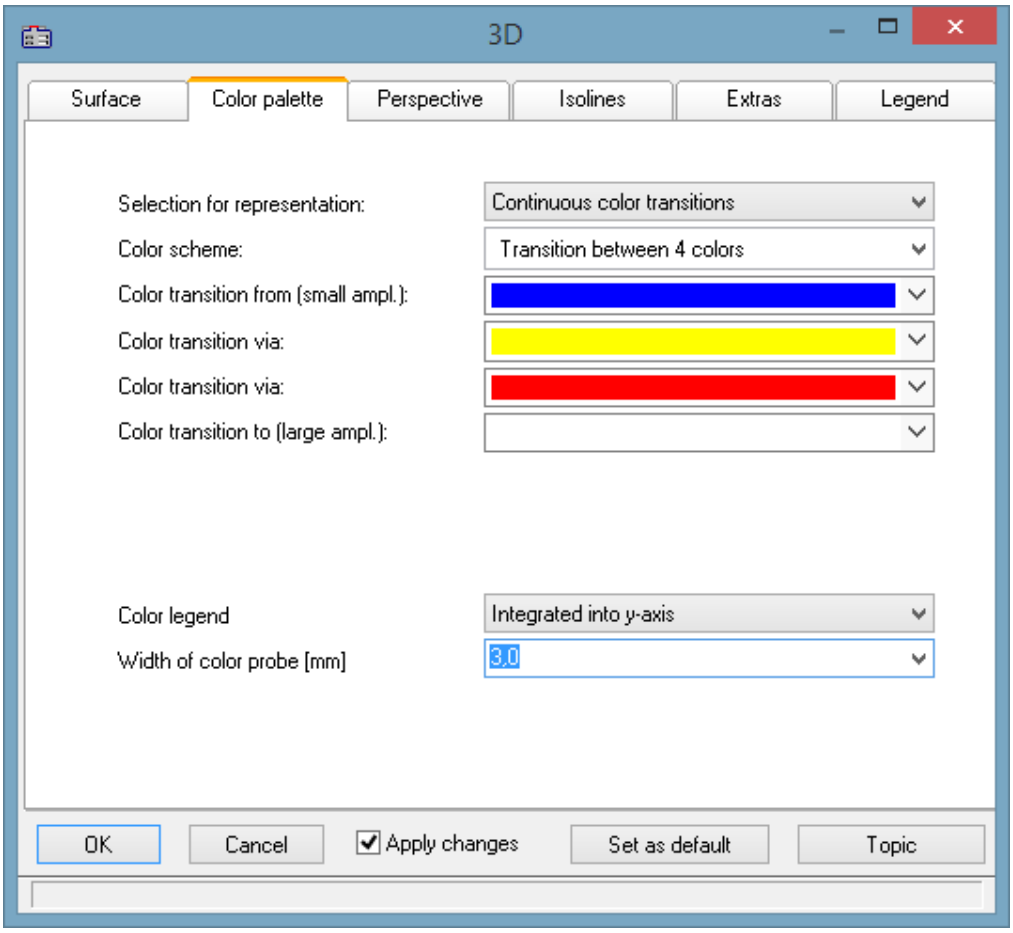
Note

3D-bars can only be set for segmented data. XYZ-plots are not supported.

Settings	Description
Color scheme	The color of the column and the color of the wire mesh (edges) can be given fixed settings. With *Surface color* set to *auto* , a color gradient is drawn if the *Color scheme* is set to *auto* or *With color gradient* is set. With *Color scheme: One color*, the complete column is colored according to its peak value.
Surface	The surfaces of the columns can be displayed in the following ways: • *Wire mesh shaded* or *auto*: colored columns, with edges • *Shaded*: colored columns without edges • *Wire mesh*: only edges • *Wire mesh with color tones*: only colored edges
Width (%) in x/y-direction	The width and depth of the bars can be set in terms of a percentage. The position may be arranged as *In between, Centered* or *Aligned*.
Bar beginning in y-direction	This sets the height of the bottom surface.

13.6.5.3.2 Color palette

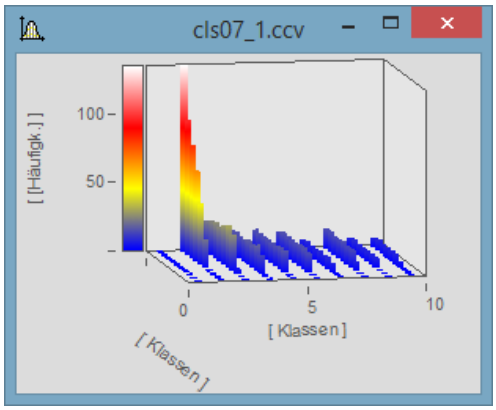
You can make a choice in the *color palette* tab to select a color for the surface. There are two illustration facilities like the continuous color transitions. You can find more information in the chapter Color map dialog, "General" 1306



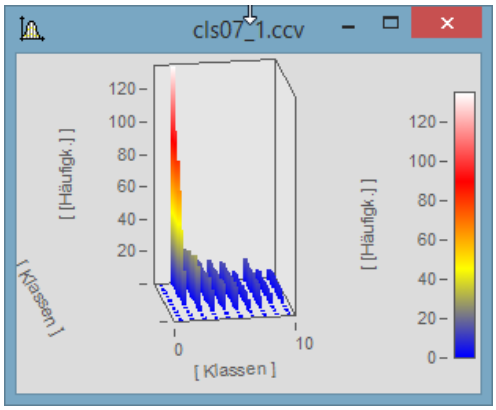
Color legend

The scaling of the colors can be indicated with a *color legend*. The following options are available:

Settings	Description
Auto, No	No color legend
Integrated into y-axis	The Y-axis is supplemented with color columns. Their width can be set.
Separate legend left, right	Separate color column to left or right of the data.



Color legend in y-axis

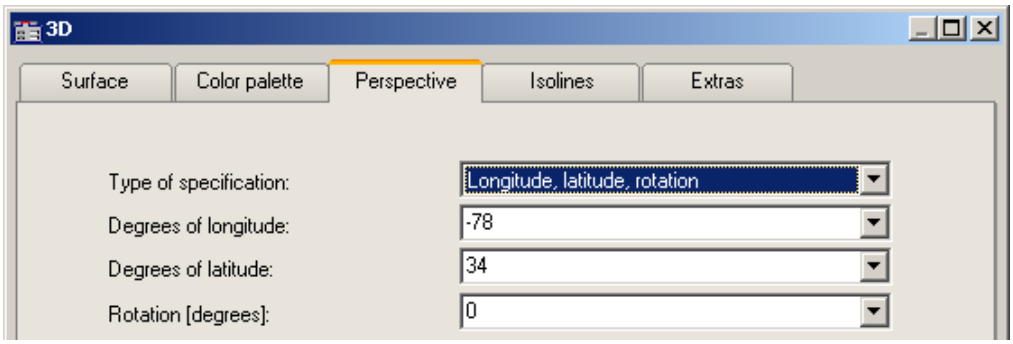


Color legend separate right

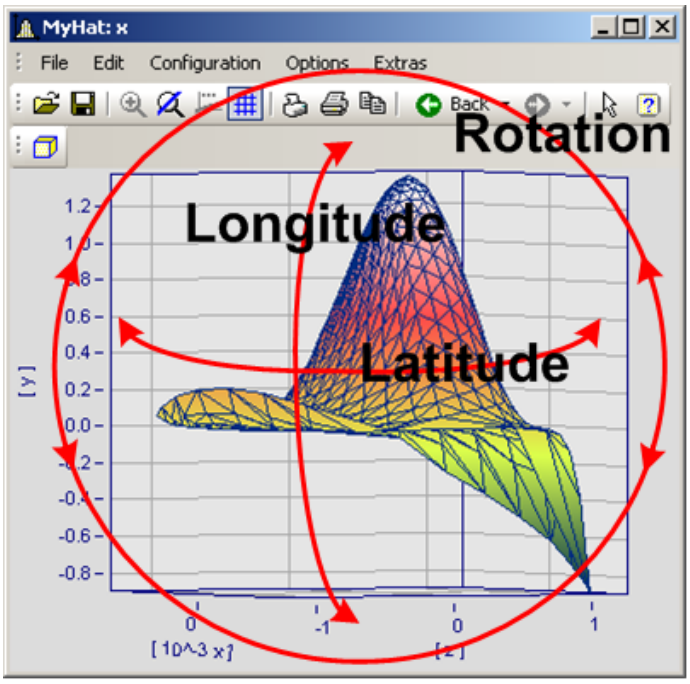
Reference

One way to color the 3D display according to a fourth variable is presented here 1400.

13.6.5.3.3 Perspective



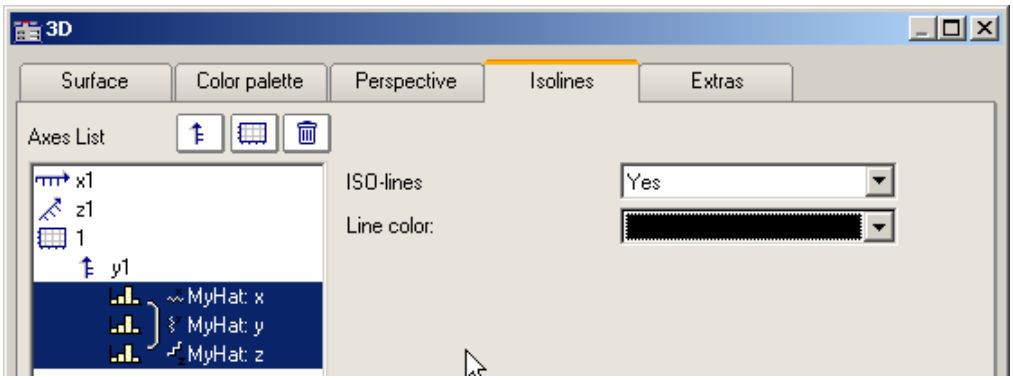
In the dialog page *Perspective*, you can set the viewing perspective as an angle between -180° and +360°. There are three ways to make the specification: *Angle of z-axis*, *Longitude* and *latitude* and *Longitude*, *latitude* and *rotation*.



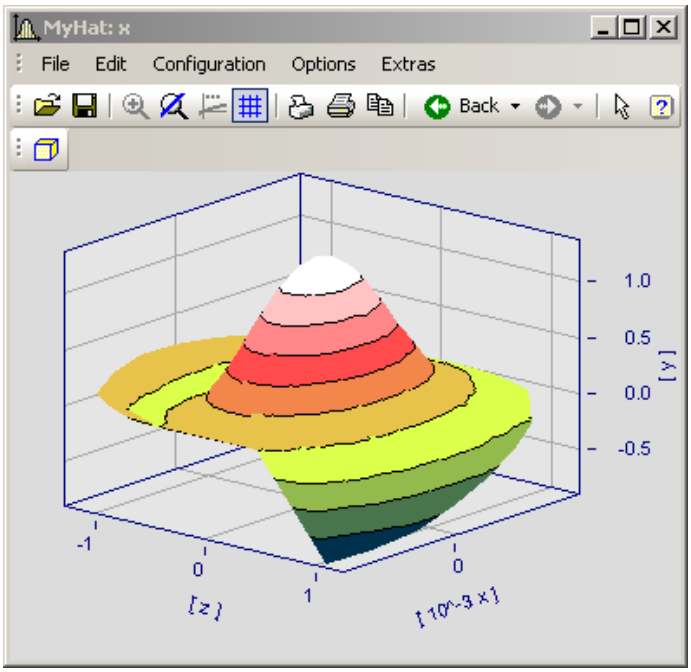
The perspective *Longitude* and *latitude* offers the best mouse control. This view behaves like a survey of the earth from outer space. This is the recommended perspective. If you wish you can additionally specify the rotation.

The three perspective parameters are related as shown below.

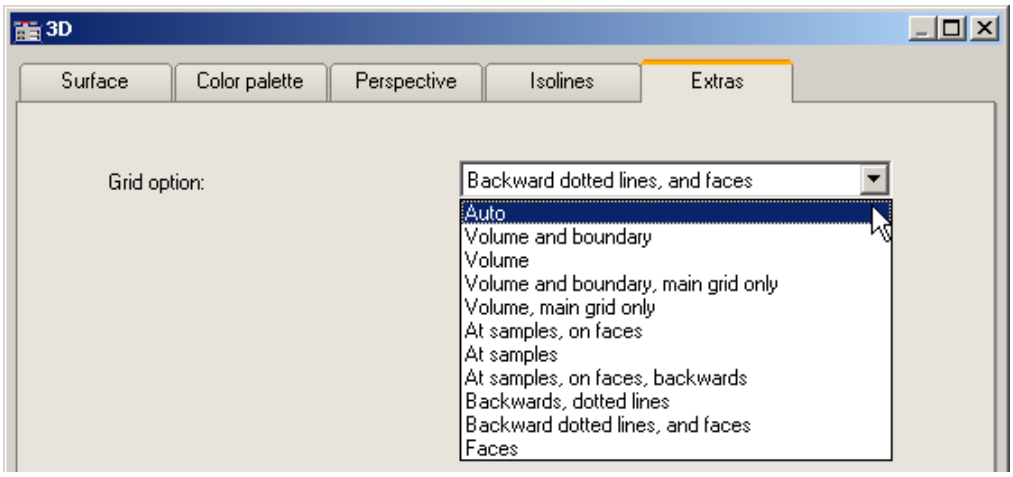
13.6.5.3.4 Isolines



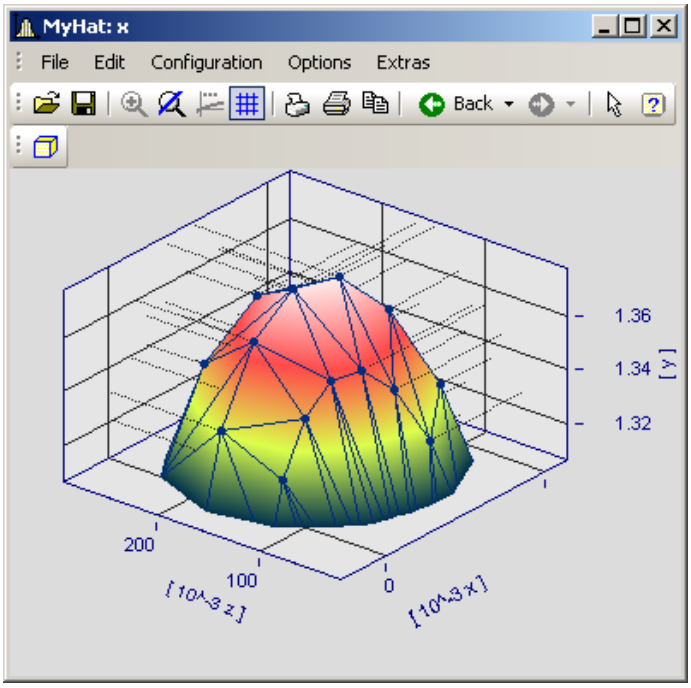
In the dialog page *Isolines* you can select whether to have isolines superimposed on the toned surface, like in color map display. This function can only be activated if graduated colors are set on the *color palette* 1408 page.



13.6.5.3.5 Extras

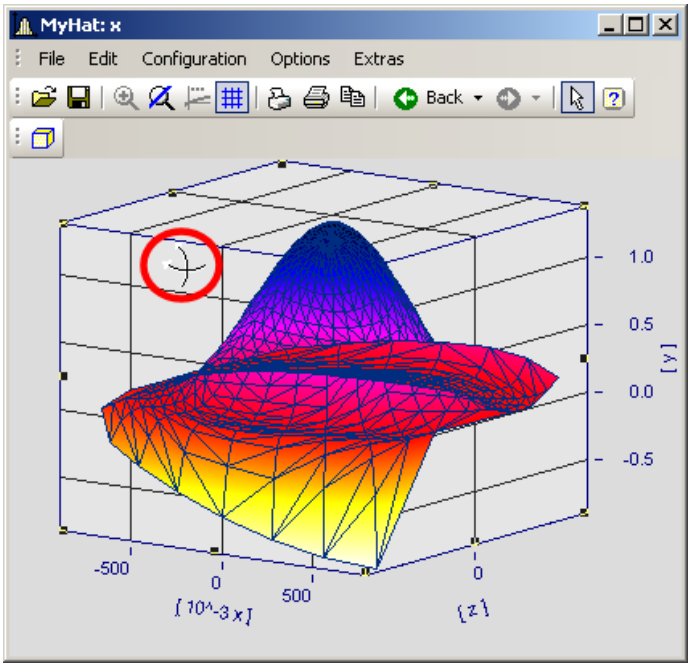


The dialog page *Extras* offers access to various grid options. This grid is then displayed in the space, in addition to the standard coordinate grid, for better legibility of a measurement point's axis values.



13.6.5.3.6 Rotate

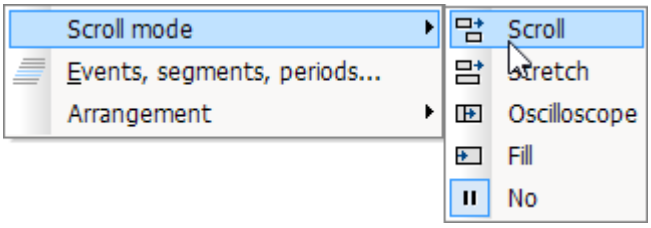
For spatial manipulation of a 3D-display, bring the mouse cursor in the coordinate system, which makes its shape change. If you then click on the mouse and start moving it, the cursor shape changes, as shown in the picture, and the 3D-display can be freely rotated by moving the mouse around.



Cursor for free rotation of a 3D-display

13.6.5.4 Scroll mode

This function is used during a running measurement, when the curve window is continually receiving new measured data to display.



Using the scroll mode, a particular segment of time-domain data can either (*scroll across*) the screen, or the segment itself, beginning from the measurement start, can continually (*grow)* or be stopped for purposes of performing analysis (*No*).

Menu item		Description
	Scroll Mode	The *Scroll mode* is set for the curve window for a running measurement. The extent of the displayed x-range remains constant; the offset of the x-axis is shifted with each new measurement value.
	Stretch Mode	Click on this button to have the curve window *stretch* during a measurement. The minimum value on the x-axis remains in effect, while the maximum is automatically set to the end of the waveform. Thus, the extent of the domain displayed increases with each new measured value.
	Oscilloscope-Mode	During a running measurement, the curve window displays a static image which, like an oscilloscope, is redrawn at intervals. The zoom width remains intact.
	Fill	Here, the curve window is filled similarly to Scroll mode. When the right edge is reached, the x-axis skips back by 75%, however. In consequence, the x-axis remains at rest most of the time, giving the user the opportunity to use a measure window 1429 during a running measurement.
	No: End Online mode	Click on this button to halt the *Scroll* or *Stretch* mode in a curve window in running measurement; this means the x-axis is frozen in its current state and the signal can then be viewed at rest as long as needed.

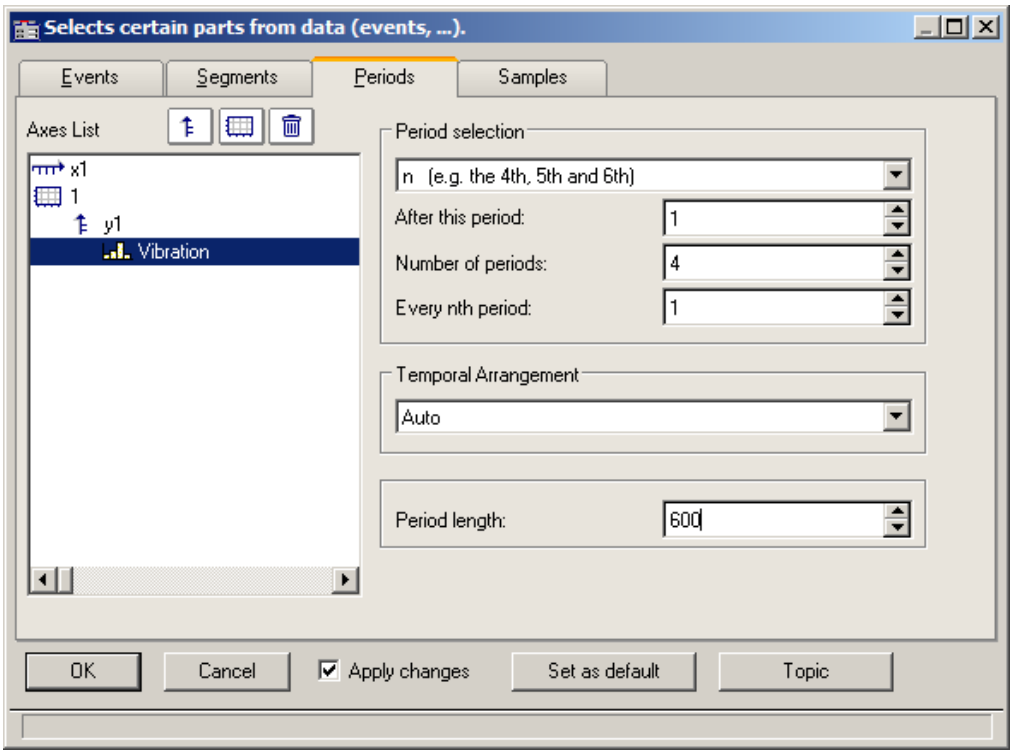
13.6.5.5 Events, segments, periods

Function

With normal waveforms, it is possible to perform a comparison of signal periods 1420 in a general way where an individual period is selected and compared with others. With **segmented** and **event-based** data, comparison of any desired portions of the data can also be carried out.

Operation

The dialog is called by selecting the menu item *Configuration / Events, Segments, Periods*.



Periods

First select the file card *Periods* from the stack (of three). Then select the channels, which are to be displayed in the period comparison, from the axes list on the left.

Next enter the *Period length* in the box on the bottom right; it is a amount of measurement values >=1.

Under **Period selection**, you specify in what way the periods are identified. The default entry is *No period comparison*. The option Auto shows all periods. Further options enable targeted display of specific periods. Select "N" periods from the list box, for example, then enter a starting period, an amount of periods, and the increment between successive periods to be compared. The first period in the waveform is denoted by the index 1.

When you have specified the amount, then proceed to the ***Temporal Arrangement***:

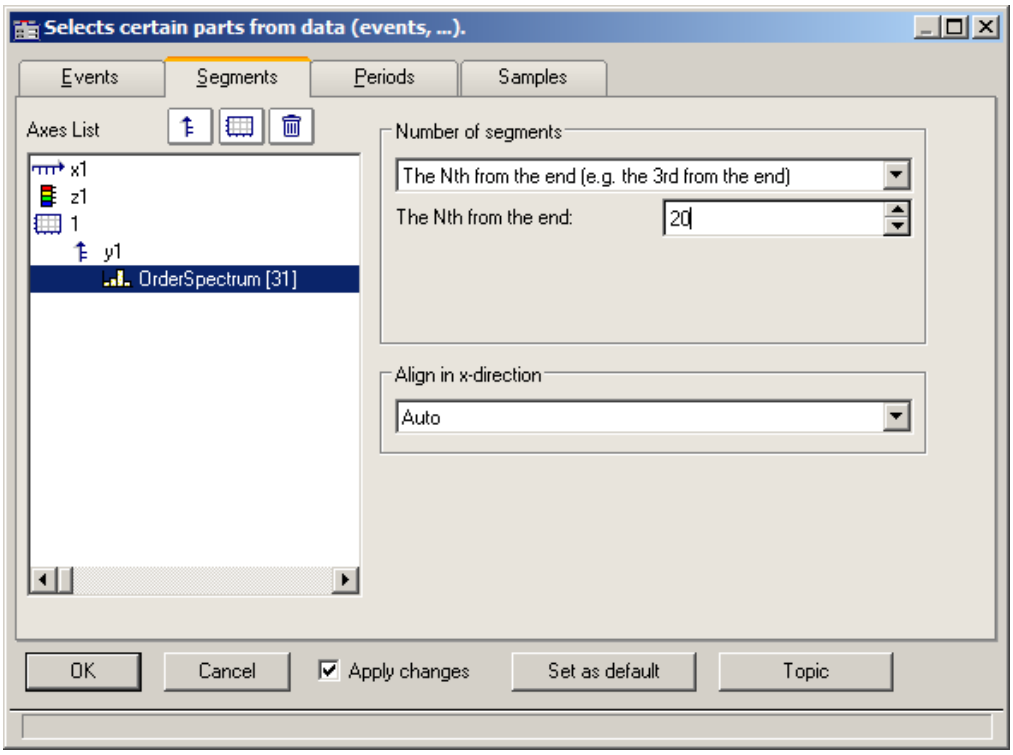
Settings	Description
Auto	The curve window determines the temporal arrangement. When in doubt, then, use this option.
x0 set to zero	The waveforms' property x0 (their relative trigger time) is set to 0 for the display.
x0 of the first period	The period comparison representation's initial x-coordinate is defined as the x-coordinate of the first measurement value belonging to the first period in the original data set.
x0 of the last period	The period comparison representation's initial x-coordinate is defined as the x-coordinate of the first measurement value belonging to the last period in the original data set.
Each period contains a unique x0	Each period has its own individual x0, which is identical to the x-coordinate of the first measurement value belonging to that period in the original data set.

© 2023 imc Test & Measurement GmbH

Segmente

For the display of segmented data. Segmented waveforms are matrices, as generated, for example, by imc STUDIO spectral calculations or class-counting.

Select the *Segments* file card:



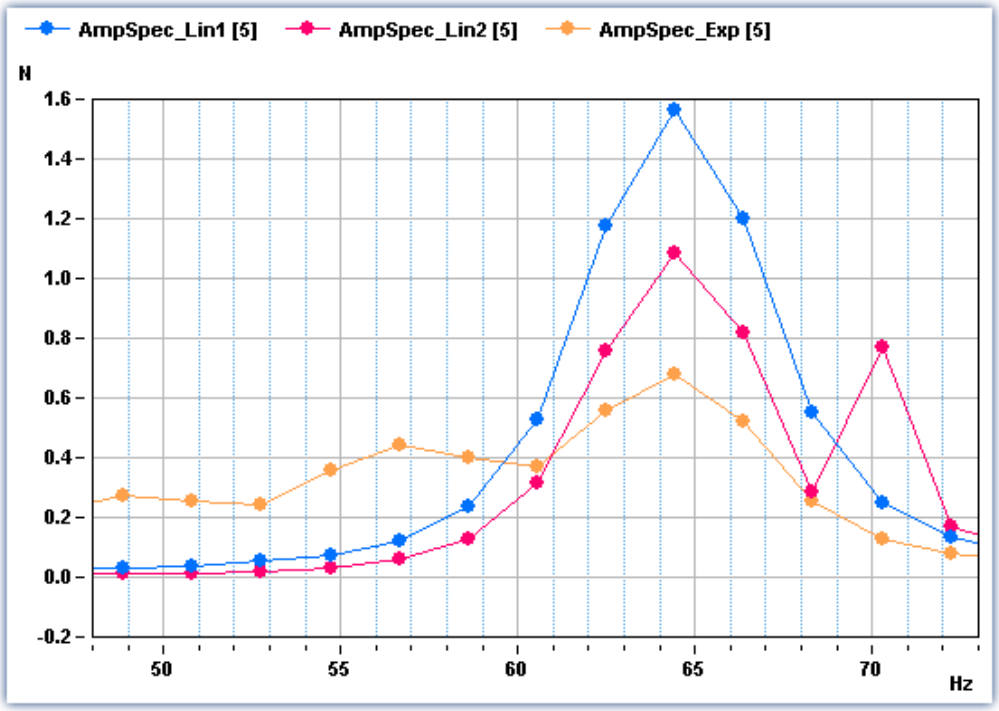
First you must make your selections in the Axes list on the left, as only the selected data indicated there are affected by settings performed with the instruments on the right.

As with *Number of segments*, select the segments to be displayed.

Then select ***Align in x-direction***:

Settings	Description
Auto	When in doubt, choose this option to at least get a view of the segments.
Retain x-coordinate	Each segment receives the same initial x-coordinate, namely, the x0 of the original data set. This is the most appropriate option for handling matrices (Rainflow,...) and multiple FFT's.
Add z to the x-coordinate	Segmented waveforms are provided with z-coordinates, which is determined from z0, dz, and a segment's index. The x-coordinate of a segment's first measurement value is in this mode the sum of x0 and the segment's z-coordinate

Example of segment indices:

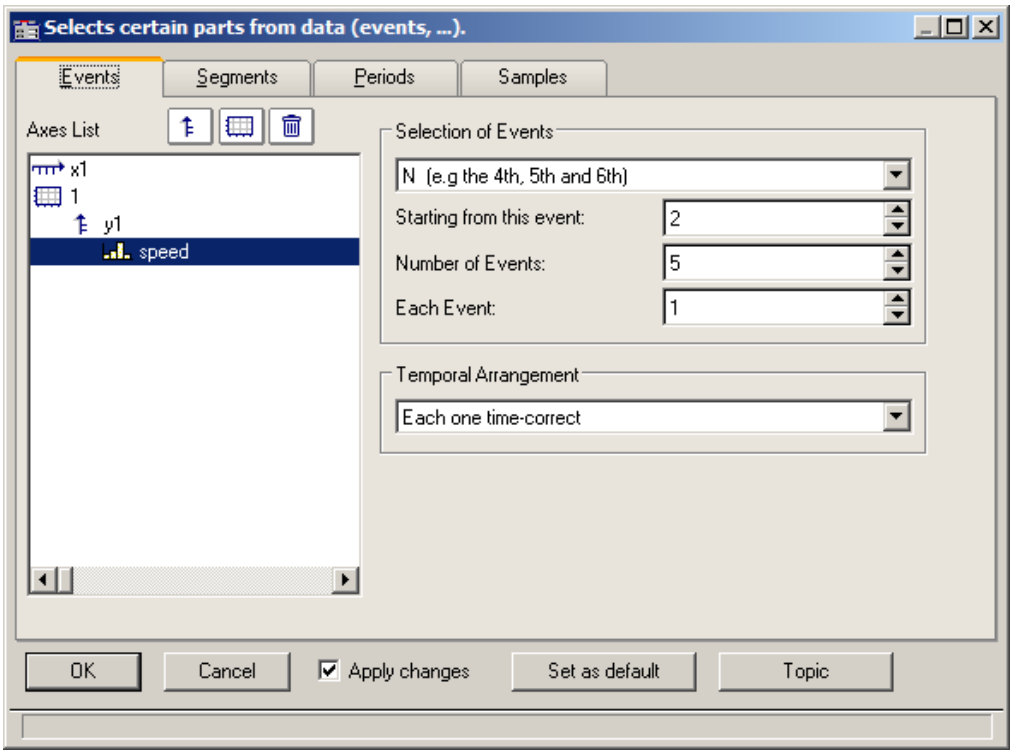


Legend with segment indices

Events

If you have recorded triggered waveforms, make the settings on the Events page. Events are triggered signal segments.

Select the page *Events*:



As before, the desired data sets are to be selected from the *Axes list* at left and the *Selection of Events* from the list box on the upper right.

Select the appropriate ***Temporal Arrangement***:

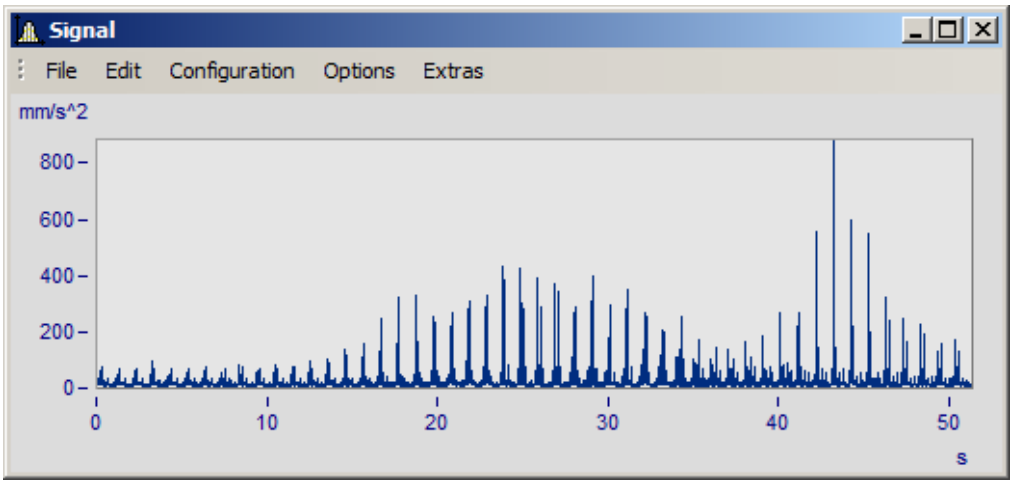
Settings	Description
Auto	This delivers an automatic representation, and is certain to achieve some result. When in doubt, then, use this option.
Each one time-correct	Each event is represented with its genuine time-coordinates. With the curve window's time representation set to "relative", the individual events receive time coordinates which correctly correlate to each other; in the "absolute" time representation the actual trigger instant for each event is indicated in absolute terms. This option is generally the most appropriate for multishot waveforms.
Trigger time of 1st is valid for all	Caution! Options which rearrange the trigger time can lead to an incorrect temporal representation. The representation may, nonetheless, be of interest for purposes of comparison. This option places the initial time coordinate of every shot at the absolute trigger time of the first event. In relative time representation there is no effect (the time-accuracy is retained); but in absolute-time mode all events are <Shift>ed to reflect the first event's trigger instant.
Trigger time of last is valid for all	As above, but based on the last event's trigger instant.
Trigger time of 1st displayed is valid for all	As above, but with reference to the first specifically selected event.
Trigger time of last displayed is valid for all	As above, but with reference to the last specifically selected event.
Trigger time difference to 1st in x0	Caution! Options which rearrange the trigger time can lead to an incorrect temporal representation. The representation may, nonetheless, be of interest for purposes of comparison. This function uses the time difference between the trigger instant of the original data set and the trigger instant of the first selected event as the x0, the initial time coordinate of the period comparison representation. This affects a representation which is made in relative time mode. The event coordinates are <Shift>ed by their genuine trigger times. In absolute time representation no effect is achieved.
Trigger time difference to last in x0	As above, but based on the last event.
Trigger time difference to 1st displayed in x0	As above, but based on the first specifically selected event.
Trigger time difference to last displayed in x0	As above, but based on the last specifically selected event.

13.6.5.5.1 Period comparison

Periodic signals occur in a variety of applications, such as rotating machinery, where the measured signal almost repeats itself at each revolution.

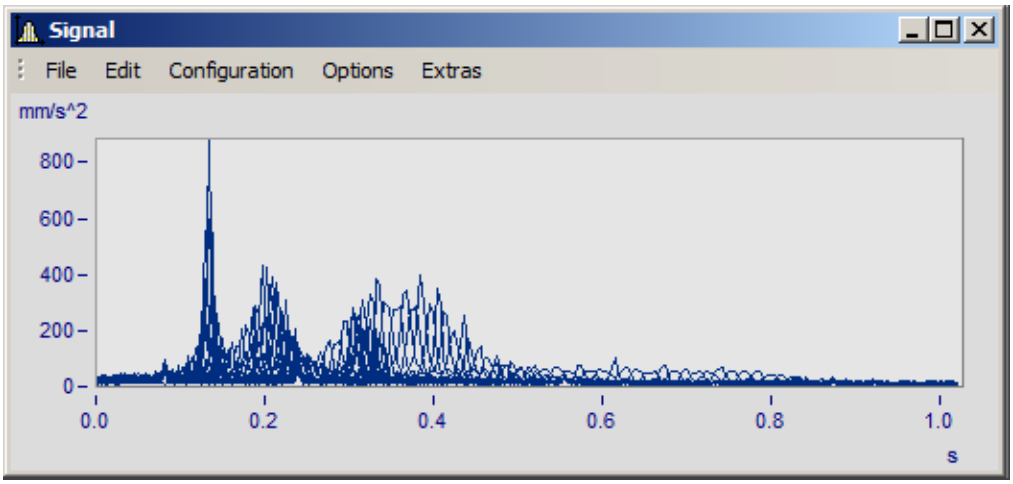
In the curve window's default display mode, entire waveforms appear plotted over time. This means that all periods are drawn in a series. In such a display, it is not always possible to compare individual periods in terms of their maximum values, for instance, or even other properties.

Example:

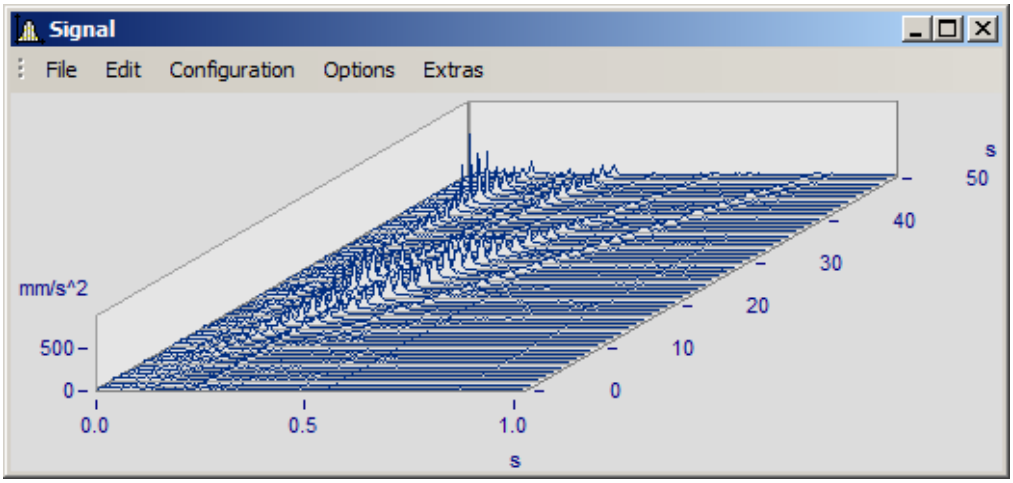


A waveform with 50 periods of 1s duration each is shown

In the period comparison mode, the waveform is divided into its individual periods. The periods are stacked over each other, enabling direct comparison. There are three ways of performing a period comparison:



Default mode. The curve plots of all the periods are drawn superimposed over each other. A dense region of coinciding signal points is visible.

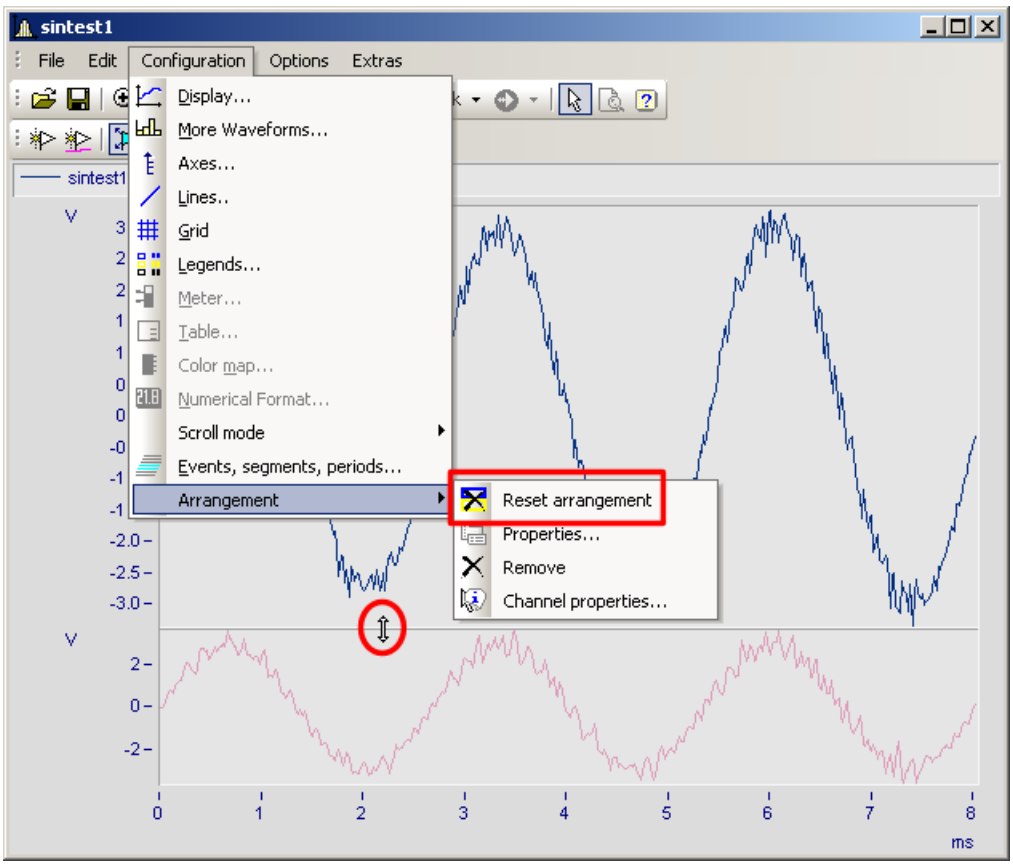


Waterfall diagram: the mountain-range-shaped surfaces clearly illustrate trends emerging across the accumulating signal periods.

In the period comparison mode, the length of a period -in data points- must be known. The amount of points reflect the amount of measured values in each period.

13.6.5.6 Arrangement

This function affects how the coordinate systems are laid out, in terms of the height of the y-axes.



In the menu *Configuration*, you have access to the same items as in the Select-mode's 1441 context menu. Additionally, there is the function *Reset arrangement*, which lets you totally undo any previous changes to the coordinate system arrangement. The relative heights of coordinate systems within a curve window can be changed using the mouse. To do this, move the mouse cursor to the separator between two coordinate systems, so that its shape changes to a vertical double-arrow. Then drag the separator, by holding down the mouse button, up or down as desired to rearrange the coordinate systems' heights.

13.6.5.7 Confirmation bar

Apply changes

When this option is selected, all changes made in the axes list are displayed immediately in the curve window. Thus, the user is always informed about the current appearance of the curve window as resulting from the settings made in the axes list. On the other hand, redrawing the curve window, especially for long waveforms and several curves, can have an adverse effect due to delays.

OK

Click on this button to close the dialog. The current settings are implemented automatically. The same effect can be achieved by closing the window, as is customary in WINDOWS.

Cancel

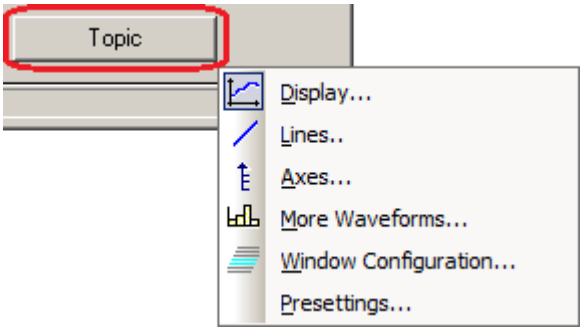
The dialog is closed and any changes made are discarded. This also applies to changes made and displayed in the curve window using the option *Apply changes*.

As presettings or Set as default

This saves the currently effective settings as default settings for future curve windows. The settings are saved to the Windows Registry. If you wish to transfer the default settings to a different computer, use the program XConfig, which is located in imc FAMOS' BIN folder.

Topic

From here, you can go directly to the properties dialogs for other objects.



13.6.6 Working with the curve window

Menu item	Description
Zoom 1425	Enlarges a section of the curve window.
Unzoom 1426	Displays the entire curve.
Measure 1427	A measurement value window and measurement cursors for measuring curves are displayed.
Select-mode 1441	In this mode, it is possible to select and edit legends, coordinate systems, axes, curves and markers.
Marker 1443	Proceeds to the list of marker functions.
Clipboard 1467	Copies the graphics in the curve window into the Windows clipboard.
Graphic-export... 1467	The graphic of the curve window is saved as image or as a PDF file.
To MS WORD 1469	The curve window is embedded as an OLE object in Microsoft WORD.
Move 1472	Proceeds to a submenu listing various curve viewing functions.

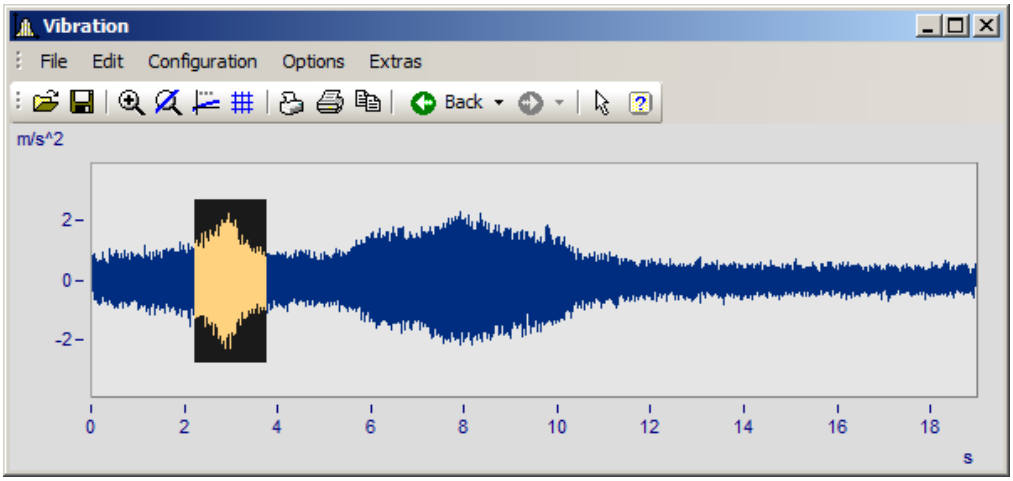
13.6.6.1 Zoom

Function

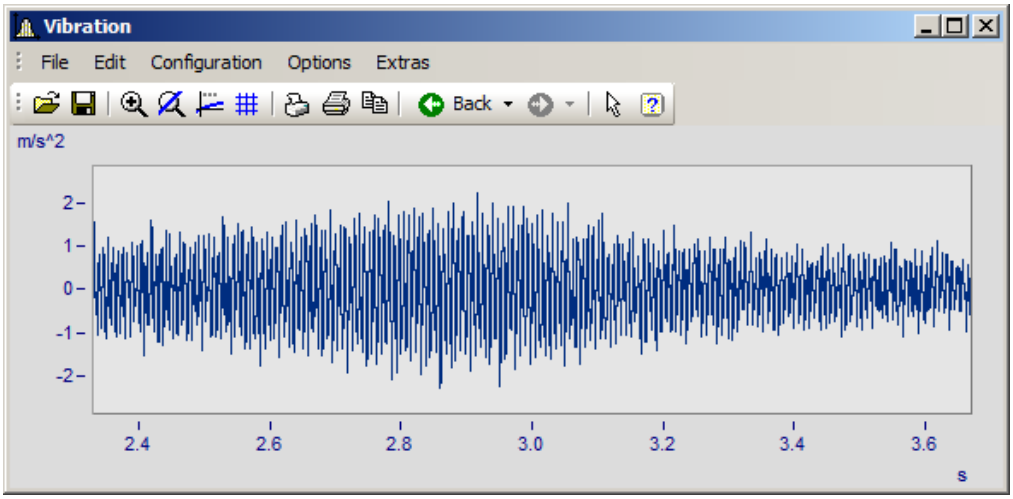
Any region of a displayed curve can be enlarged for viewing (zoomed). Zooming can be done in both x- and y-directions at the same time. The zoomed region is a rectangular area within the coordinate system, with size and position determined by the user.

Mouse Operation

- Select menu option *Edit / Zoom* from the *Scales* menu. The mouse pointer changes its shape to a vertical arrow.
- Draw a rectangle with the mouse around the area you want to see. The selected area is displayed with inverted colors.



- Release the mouse button to enlarge the area displayed in negative and end the zoom mode



Remarks

- The *Zoom* mode can be ended at any time by pressing ESCAPE.
- The mouse pointer must not necessarily be within the coordinate system when selecting the corners of the zoom range.
- A zoomed region can be zoomed further for greater detail.
- Use menu option *Unzoom* to view the entire curve.

- The zoom function can also be used when the *overview* or *measurement value windows* are active.
- After completed zooming, the axes can optionally be labeled with rounded values. In that case, a slightly larger region is usually displayed than was selected.
- Note that the option *use rounding* 1496 can be switched off in the *Presettings* menu.
- The zoom function can be regarded as a short-cut for a series of manual axis scaling operations (see menu option *Axes* 1350). If the result is not satisfactory, scale the axes manually (without rounded values).
- Note that zooming in the x-direction often does not produce the desired results, especially for x-axes displayed logarithmically. In rounding, the values on the x-axis will be shown in powers-of-ten as long as the x-range is large enough. Try zooming even more in the x-direction or scaling the x-axis manually. Then you must give consideration to the presettings for the curve window.
- When several curves are displayed in a window, all are zoomed at the same time.
- The measurement value- and overview windows are updated automatically; the positions of the measurement cursors are kept, if feasible.
- Unlimited zooming is not possible:
  - The relative resolution is 10^{-13}.
  - Note that unlimited zooming doesn't make sense, since the representation of the numbers is only precise to approximately 15 decimal places.
  - The zoom region is determined by the resolution of the monitor screen. Enlarge the curve window to make zooming more precise.
  - When the display mode *y-axes stacked* 1267 is selected, you can spread the color-negative rectangle of the zoom mode onto one single or several curves. If the rectangle covers only one curve, this curve is zoomed in the x- and y-directions; while for the other curves in the window, the y-range remains intact and the x-range is adapted to the zoomed region. If the rectangle extends over more than one curve, only the x-direction is zoomed; the y-range for each curve remains the same.
  - In waterfall diagrams, only the XY-level can be zoomed.

13.6.6.2 Unzoom

Function

Select the *Unzoom* function to return to the display of the entire waveform after having zoomed. If the curve window had customized scaling and parameters before zooming, those settings are lost after unzooming, since the scaling is then performed automatically.

Mouse Operation

Select menu option *Unzoom* in the *Edit* menu to restore the previous curve window settings.

Remarks

All y-axes present when the function *Unzoom* is selected are scaled automatically, except when they are scaled like the next axis to the left. The x-axis is scaled automatically so that the first curve in the window is displayed completely. Additional curves may extend beyond this range due to a different x-scaling.

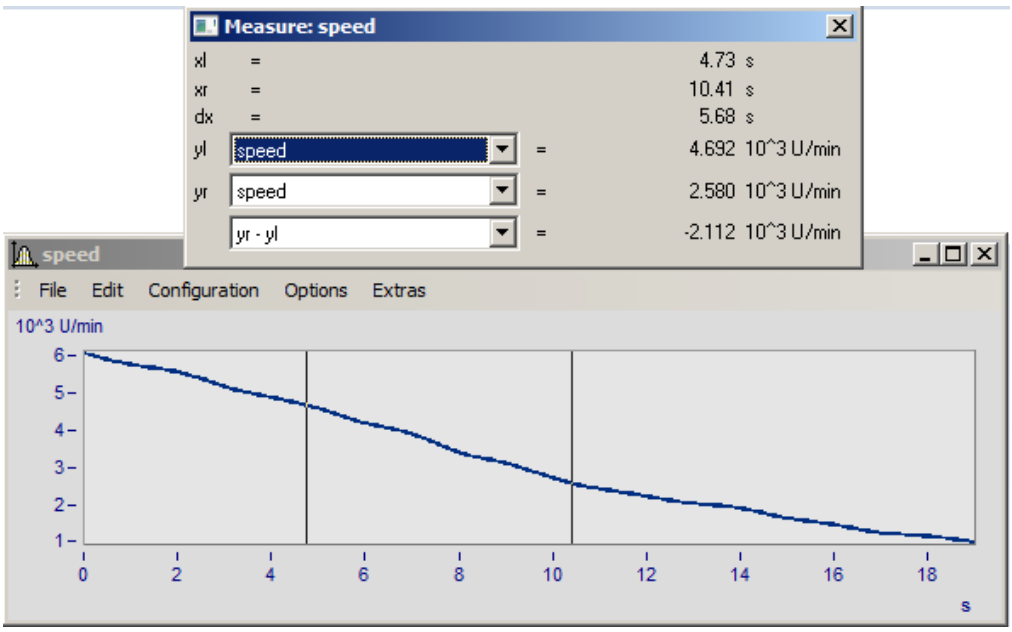
13.6.6.3 Measure

Function

Two independent measurement cursors are available for performing measurements. The x- and y-values of the intersections with the vertical cursor lines are displayed together with the curves in a measurement value window.

Mouse Operation

- Select *Edit/ Measure* and the *Measure* window appears. A measurement value window appears and two measurement cursors are set up in the curve window.
- Move the mouse pointer to over the coordinate system and hold down the left or the right mouse button. The mouse pointer then jumps to the measurement cursor. On the other hand, if you wish to place the measurement cursor at the mouse pointer's position, hold down the keys CTRL or SHIFT while clicking on the mouse button.
- The **left mouse button** controls the **left measurement cursor;** the **right mouse button** the **right measurement cursor**. If you hold down both mouse buttons at the same time, you move both measurement cursors simultaneously.
- When you click in the curve while holding down the SHIFT key, the measurement cursor jumps to the mouse pointer's current x-coordinate.
- In order to place the measurement cursor at any desired location in XY-displays, click one of the mouse buttons while holding down the SHIFT key. The measurement cursor then jumps to the closest point on the curve to the current position.



Remarks

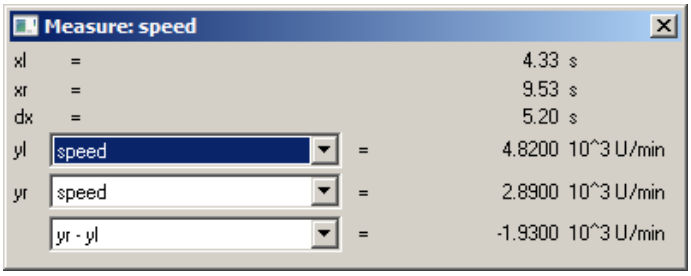
- The crosshairs may cross each other.
- In normal time displays, the horizontal line of a measurement cursor is visible only as long as the curve lies within the coordinate system and is defined for the x-coordinate of the measurement cursor.
- Zooming in XY or locus plots may result in the measurement cursors no longer being visible.
- If linear interpolation between the actual points of a curve is used to draw a graph (points connected by lines), the measurement cursors follow this path.

- To measure the actual points of the curve, select the display mode *Steps* in the *Style* submenu of the *Lines* 1369 menu.
- The horizontal measurement cursor can be switched off in the *Options/ Presettings* 1496 menu. This is sometimes useful when displaying slowly changing data or digital data consisting of steps.
- If no mouse is connected, the mouse pointer only appears whenever the curve window is active at the moment.

13.6.6.3.1 Measure window

Function

The measurement value window appears once menu option *Measure* has been activated in the curve window. Its position can be changed as desired. However, the measurement value window is always in front of the curve window to which it is assigned. The measurement value window shows the x- and y-values for both measurement points and those values' mutual differences; the display of other characteristic values is optional.

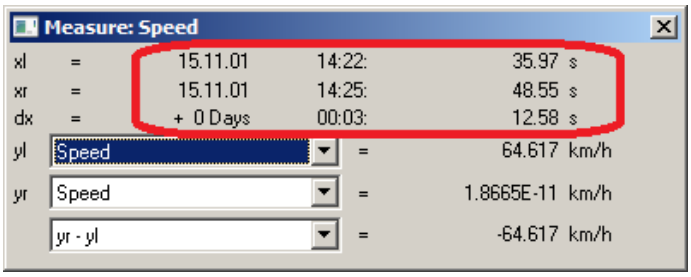


In the selection lists *yl* and *yr*", a waveform displayed in the curve window can be selected for measurement, which is accomplished using the measurement cursors. The list box at the bottom of the dialog contains the following entries as available choices of calculated values:

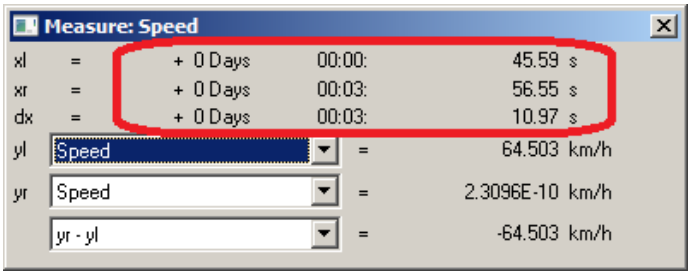
Symbol	Description
xl	x-coordinate of left measurement cursor
xr	x-coordinate of right measurement cursor
dx	Difference between the x-coordinates, xr-xl
xr/xl	Quotient of the x-coordinates of the measurement cursors
yl	y-coordinate of the specified curve at the intersection with the left measurement cursor
yr	y-coordinate of the specified curve at the intersection with the right measurement cursor

Calculations	Description
yr-yl	Difference between the y-coordinates
yr/yl	Quotient of the y-coordinates
Slope	Slope between the two measurement points, dy/dx
Slope per decade	Slope is defined in units per decade, dy/lg(xr/xl); especially useful for logarithmic display. For example, the slope of a filter is often measured in dB/dec.
1/dx (frequency)	Difference between x-coordinates (xr-xl) in frequency units, e.g. 1/ms
Surface	Integral of the data between the measurement cursors = Sum(Yn*xdelta) Available only for equidistantly sample waveforms.
y-Min	Minimum between the measurement cursors. Available only for equidistantly sample waveforms.
y-Max	Maximum between the measurement cursors. Available only for equidistantly sample waveforms.
RMS	The root of the mean value of the squares of each value between the cursors ("root-mean-square") is calculated. Available only for equidistantly sample waveforms.
Standard deviation	For the data between the measurement cursors, the deviation of each value from the arithmetical mean is squared and then all of these squared values are summed. This value is then divided by the number of these values minus 1; finally, the square root is calculated. Available only for equidistantly sample waveforms.
x-coordinate of inflection point	Inflection point according to the following formula: yl > yr -> x-position of **minimum** slope yl < yr -> x-position of **maximum** slope Available only for equidistantly sample waveforms.

When using one of the display modes *Date/ time* or *Days/ hours/ minutes*, the time data appears as follows:



Measure window with date and time view



Measure window with relative time

A negative sign before data in *days/hours/minutes* and *seconds* pertains to the entire time declaration, not only to the days.

Positioning the measurement cursors

The left measurement cursor is positioned while holding down the left mouse button, and the right cursor while holding down the right mouse button.

In normal cases the cursors move along the data line; for this reason, the *line type* 1370 applies to it.

By means of the contextmenu command "*Cursors freely movable* 1431", it is possible to cancel the linkage.

Wen you click in the curve window while holding down the Shift key, the cursor jumps to the selected x-position. Depending on the setting "*Cursors freely movable* 1431",the horizontal measurement line jumps either to the associated amplitude or to the selected position.

13.6.6.3.2 Context menu in Measure window

Move the mouse cursor to the measurement value window and right-click the mouse to prompt the following context menu:

Cursors unrestrictedly movable

If you select this function, both measurement cursors can be moved freely in the curve window without any linkage to particular curves.

Send Curve Segment to imc FAMOS!

This option sends the curve section located between the vertical lines of the measurement cursors to imc FAMOS. This new waveform receives a name which appears in the variable list and by which it is designated as per the transfer options set using the menu Options/ Transfer options. 1502.. in the curve window.

When multiple curves are displayed in the window, the section between the measurement cursors of each of them will be transferred to imc FAMOS if the pertinent transfer option is currently activated.

Note that the original data sets are overwritten if you do not specify a new name when using Transfer Options.

Note that this export function duplicates the waveform in the course of the export procedure, unlike other export functions. Shifting the position of the measurement cursors after exporting a curve section will not affect the exported waveform.

Reference

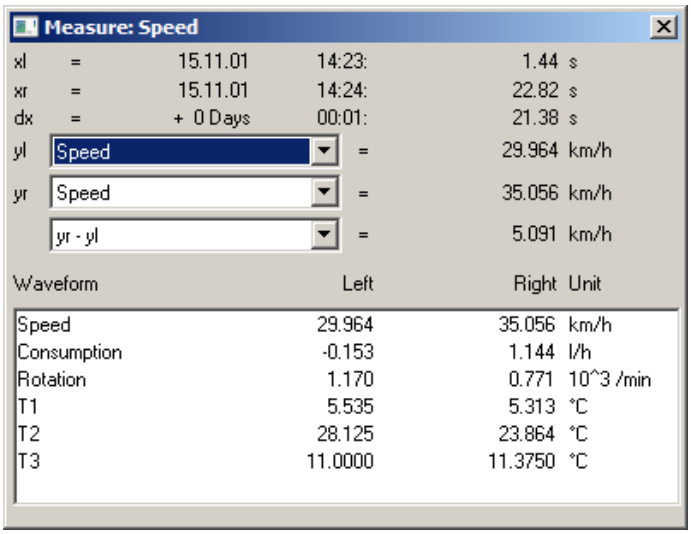
More about Transfer options 1502

Clipboard

All measurement values displayed in the measurement value window are copied to the clipboard in text format. This text is then available for use by other applications.

Channels List

When several curves in one window are compared with each other, it can be helpful to see simultaneously the measurement values of as many curves as possible. Do this by selecting the option Channels List, which displays a list of all channels. The measurement value window is enlarged and a list of all y-coordinates at each of both measurement cursors appears, irrespective of which curve the cursors are currently set to move along.



When *List of all channels* is active, the menu item is marked with a check. If you wish to close the list mode, select the same menu item again. The last state remains in operation the next time the measurement window is opened. The channel list's size can be changed. You can highlight list entries to distinguish then visually.

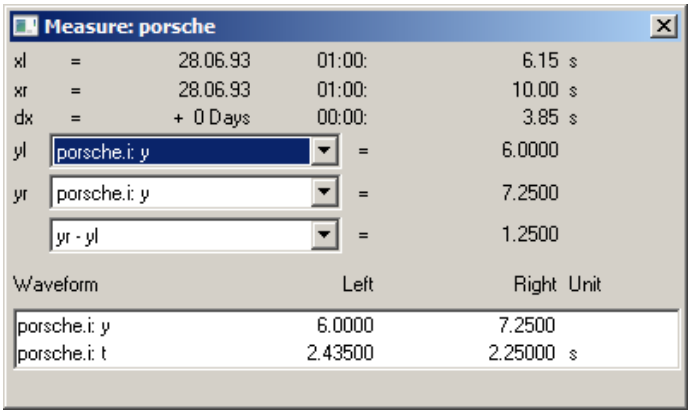
Remarks

The column *right* and *left* above the list is assigned to the measurement cursor which is controlled by the corresponding mouse button.

Expand List

When *List of all channels* is active, the menu item *Expand the list* appears. By selecting this option, separate measurement values for all XY-waveform components -if any such waveforms are present- are displayed in the list. In polar plots, the measurement values of the magnitude and phase are displayed. This option can also be activated by double-clicking on the pertinent entry in the list.

Double-clicking on a component of the waveform also deactivates the option; and only the selected component is retained as a separate entry in the list.



Reprocessing signals...

This menu item allows signals to be reprocessed, for example, to undergo smoothing or peak clipping. For a detailed description, see the separate chapter Reprocessing signals 1435.

Place marker with left click; Place marker with right click

Set a marker at the measurement cursor positions. Subsequently, a dialog for marker settings 1454 appears.

Reference

More about "Markers" 1443 and "Marker Definition" 1454

Append to measurement value file

This menu item appends the measurement values to the file of measurement values as text. The name (along with the directory path) is set in the menu under *Measurement value file name*.

The file is a normal, text file in ASCII without formatting, so that it can be edited by any word processing program. By repeatedly calling this menu item for a variety of cursor positions, it is possible to generate a list of landmark points. If the file doesn't exist, it is first created.

Note that MS-Windows applications misread *umlauts* in these files. The OEM format is used for writing to these files. For this reason, use programs (such as word processors) not running under MS-Windows to analyze these files.

Note

As of imc FAMOS 7.0 / imc STUDIO 5.0R3, this function is no longer shown.

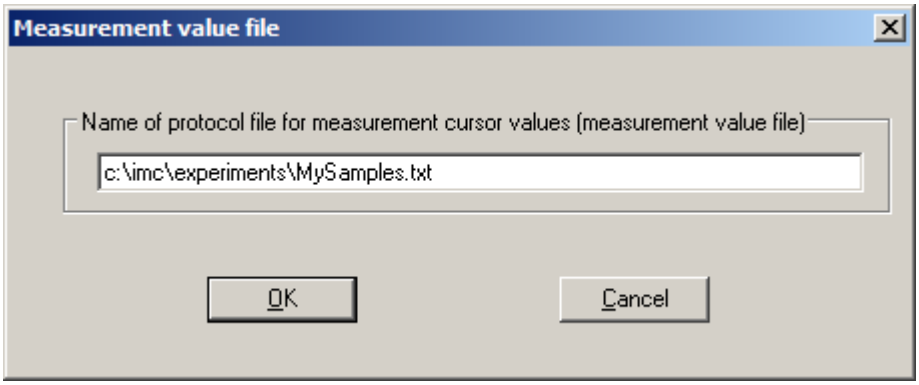
By means of an entry in the Registry, it can be made visible again:

Computer\HKEY_CURRENT_USER\Software\imc Measurement and Control\Default\CurveDataManager\Curves

If it is not already present, enter as a *New String* the text "*EnableMeasFile*". Set its content to *1*.

Measurement value filename

This menu item calls a little dialog in which the name of the measurement data file can be edited. The setting for this filename is valid globally for all measurement value windows.

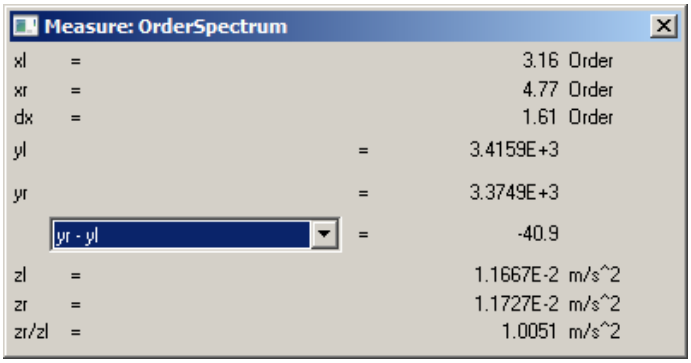


Remarks

- The *Measure* mode (i. e. Measure window is open) is ended whenever a major change in the display of the curve window is made, e.g. when numerical value display mode is selected.
- The menu option *Measure* in the curve window is checkmarked if the curve window is in the *Measure* mode. If the checkmark is removed, the measurement value window is closed.
- Each curve window can be in the *Measure* mode, regardless of the status of all other opened curve windows. It is thus possible to conduct measurements with multiple measurement value windows independently of each other. The title bars in the *Measure* windows contain the names of the corresponding curve windows.
- When measuring in all XY-representations (or polar plot), the measurement cursors move somewhat differently than in normal time-dependent representation. The movement of the measurement cursors always follows the parameter of the XY-display. Moving the mouse to the right or upward moves the corresponding measurement cursor in the direction of the increasing parameter.
- The measurement cursor speed in XY and locus plots caries depending on the point density of the curve. The measurement cursors cannot be moved in arbitrarily fine increments between samples. Zooming too strongly, therefore, can cause the measurement cursors to jump between points.

13.6.6.3.3 Performing measurements on color maps

In this display mode, the measurement cursor can be moved freely over the entire plane of the coordinate system. x- and y-coordinates are arranged horizontally or vertically. The z-coordinate is indicated along the "color axis", which extends outwards from the plane of the monitor.



13.6.6.3.4 Reprocessing signals

Function

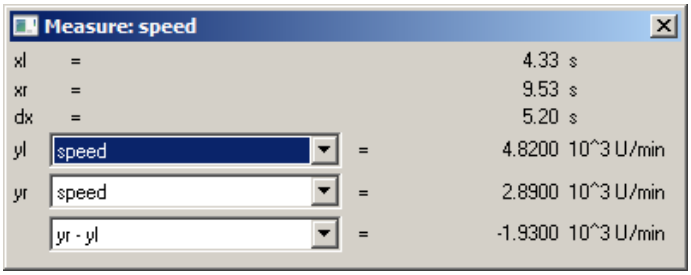
Here it is possible to edit, for instance, to smooth clip peaks from, simple signals. Thus, invalid readings can be removed from a signal and replaced with plausible ones, or distorted readings can be corrected, for example, by eliminating offsets or drifts.

These functionalities are only available off-line (in imc FAMOS).

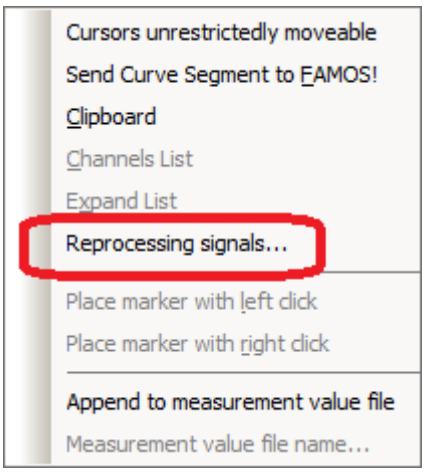
Mouse Operation

A channel is displayed in a curve window. A region is staked out using the measurement cursors, e.g. the region around a signal peak representing interference to be deleted. In the dialog called from the menu item *Reprocessing signals...*, a mathematics function is selected for applying to the staked-out region. When this function is performed, the channel's measured value are replaced by newly calculated values.

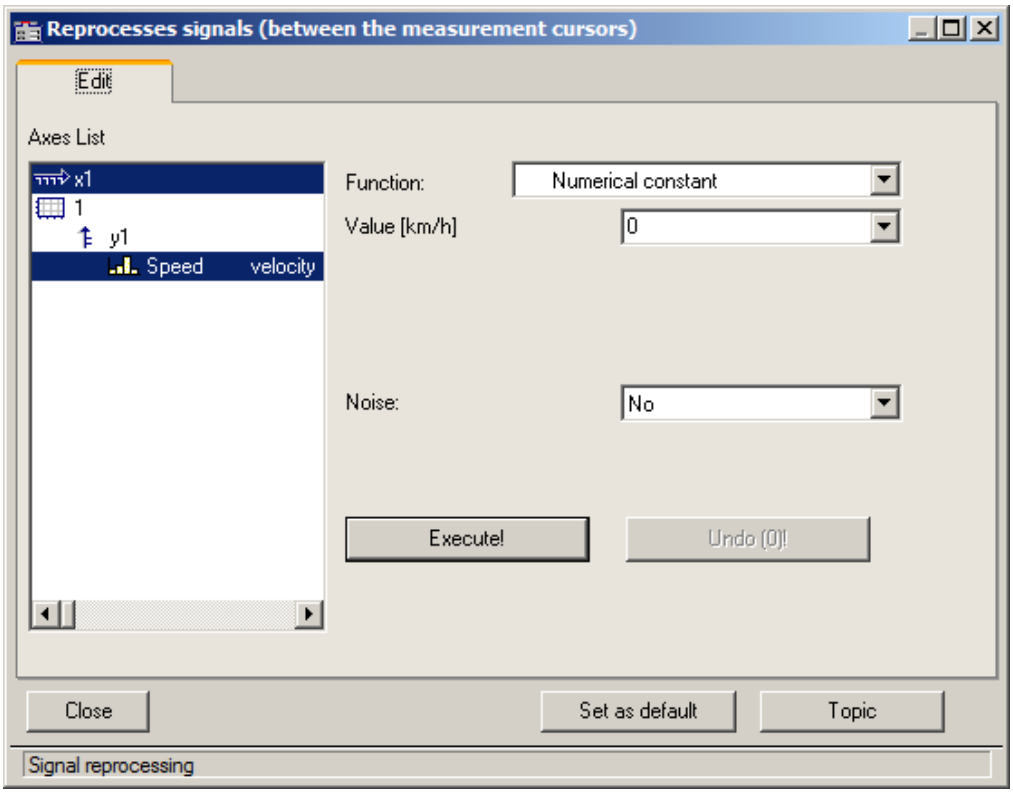
First, the Measurement value window is opened:



Next, this window's context menu is opened: move the mouse pointer over the measurement value window and right-click the mouse; the menu illustrated below then appears:



Select *Reprocessing signals*.... The following dialog is called:



Select in the left portion of the dialog the channels in the curve window to which the function is to be applied. On the right side, select the calculational function and the function parameter.

Use the measurement cursors to carefully delineate the desired signal segment.

When the *Execute!* button is pressed, the function is applied to the respective signal segment of the selected channels, which is delineated by the cursors.

The button *Undo* lets you reverse the last action taken.

Remarks

- Only simple data types can be handled in this way. These are equidistant data (constant sampling time, dx = constant) no having any additional structural pattern. In particular, it is not possible to process segmented waveforms, data with events, XY-data, data already subjected to Transitional Recording, Time-stamp-ASCII data, texts, and complex data.
- The data's original format is retained. This can mean that the value range is limited and stays that way, and that the math function can thus only be performed in approximation. For instance, if a channel's value range is -10V .. +10V with 2 Byte integer format, no value >10V can be displayed. Thus, an overdriven spline can be truncated at 10V. And it isn't possible to set the signal to 11V, instead it stays limited to 10.0V.
- Undo: Up to 30 actions can be reversed (undone).
- If invalid function parameters are specified, the function cannot be carried out. Instead, an error message appears.
- **The interval for determining boundary values may not be less than 0!**
- The dialog for reprocessing signals can only be opened for one curve window at a time. So if you wish to reprocess another curve window's signal also, first close the *Reprocessing signal* dialog in the first window.
- The dialog for reprocessing signals has no *Cancel* button. If the dialog is exited, changes made to the channels are retained. The *Undo* button is the only way to reverse changes that have been made.

- See the discussion of "Confirmation bar 1423" for more information on the operation of the *Set as default* button.
- If data in imc FAMOS are edited by means of the *Reprocessing signals*.. dialog, the display in imc FAMOS' data editor is not updated. The Data Editor should be closed during this time; it can be called again afterwards.

Description of the functions

The following is a description of all functions in the list *Function*. The functions are ordered by subject.

The subjects appear aligned to the left edge, e.g. *Adapts curve*. The functions themselves appear in the list with an indent, e.g. *Constant level based on right boundary*. Only functions can be selected.

Subjects

- *Adapt curve*: A connecting line, either a straight line or a spline, is inserted.
- *Re-define curve*: The curve is re-defined; for instance, as a specified constant level.
- *Smoothing*: e.g. low-pass filtering
- *Edit signal*: e.g. value range limiting
- *Edit trend*: e.g. add offset, eliminate trends, high-pass filtering
- *Noise*: Add noise to produce readings which "look convincing".

Please note also the description of the parameters *Noise* 1438 and *Boundary interval* 1439.

Adapt curve	Description
Constant level based on right boundary	The mean value of the right boundary interval (which see below) is taken. This value replaces all original readings in the delineated region. If desired, noise (which see below) can be added.
Constant level based on left boundary	The mean value of the left boundary interval (which see below) is taken. This value replaces all original readings in the delineated region. If desired, noise (which see below) can be added.
Connecting line	The respective mean values of the right and left boundary intervals are taken. These mean values are used as the boundary values for a straight line replacing the signal curve between the cursors (linear interpolation). If desired, noise (which see below) can be added.
Cubic spline	The respective mean values and average slopes of the right and left boundary intervals are taken. The mean values are used as the boundary values of a replacement curve between the cursors, and the slopes are used to form a cubic spline, which is a polynomial whose slope at the boundaries matches the slope of the original curve.

Re-define curve	Description
Constant numerical value	The delimited region can be replaced with a fixed numerical value to be specified. If desired, noise (which see below) can be added.

Smoothing	Description
Low-pass	Smoothing: A digital filter is applied to the readings within the delimited region, a low-pass filter of Butterworth characteristic. The filter order and cutoff frequency can be specified. The order can be set to between 1 and 10. The pop-down list presents suggestions for the cutoff frequency. The cutoff frequency must be substantially less than half of the sampling frequency for the data. The function works in a manner similar to imc FAMOS' FILTTP() function. Note that the filter has a transient oscillation and always produces a delay.

Edit signal	Description
Clip signal above	All readings in the region delineated are clipped to keep the curve below a specified maximum value.
Clip signal below	All readings in the region delineated are clipped to keep the curve above a specified maximum value.

Edit trend	Description
Add a numerical constant	Adds an offset. A fixed numerical value is added to all readings in the delineated region. This means adding an offset. To subtract, enter a negative value.
Multiplication	A fixed numerical value is multiplied to all the measured values within the staked-out interval.
High-pass	Eliminates offsets and slow drifts: a digital filter is applied to the readings within the delineated region, a high-pass filter of Butterworth-characteristic. The order can be set to between 1 and 10. The pop-down list presents suggestions for the cutoff frequency. The cutoff frequency must be substantially less than half of the sampling frequency for the data. The function works in a manner similar to imc FAMOS' FILTHP() function. Note that the filter has a transient oscillation.
Add a slope	Trend correction: a slope is added to all readings within the delineated region. The slope is a linearly increasing function. To define the slope, and amplitude is specified for the initial point (y-coordinate at left boundary of delineated region) and the amplitude at the end (y-coordinate at right boundary of delineated region). The values specified define the slope, which is then added to the readings.
Subtract linear approximation	Trend correction: a linear approximation is formed from all the readings in the delineated region. The least-squares method is used for this. The approximation The approximation is later subtracted from the curve in the delineated region.
Fade in/out	The signal can be faded in to or out from zero within the delineated region. For this purpose, the signal is multiplied with a ramp. Fading in is the typical case, from 0% to 100%. Fading out from 100% to 0% is another typical action which is possible. Both percentages can be specified. An entry of 100% corresponds to a factor of 1.0. These parameters appear in the dialog as Start, component [%] and End, component [%] as the values at the left and right boundaries, respectively. In between, the values are interpolated linearly. The two percentages supplied may not be either 0.0 or 100.0, but can take any other real value in between.

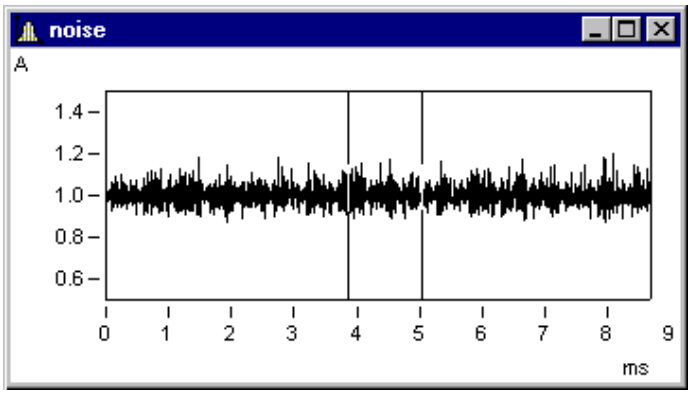
Noise	Description
Add boundary noise	The respective trend is eliminated from each of the boundary interval's signals. This is accomplished by subtracting the linear approximation calculated. What remains is interpreted as noise. This noise, cyclically repeated, is added to the readings in the delineated region. In the process, the noise at both ends merges by means of linear weighting over the width of the delineated region. The values within the boundary interval naturally remain unchanged.
Add equally distributed noise	Equally distributed noise of specified amplitude is added to the readings within the delineated region. If the amplitude is 1.0, random numbers in the range -1.0 ... +1.0 are generated and added.
Add Gaussian-distributed noise	A noise signal with Gaussian distribution (normal distribution), having a specified RMS-value, is added to the readings within the delineated region.

Boundary interval

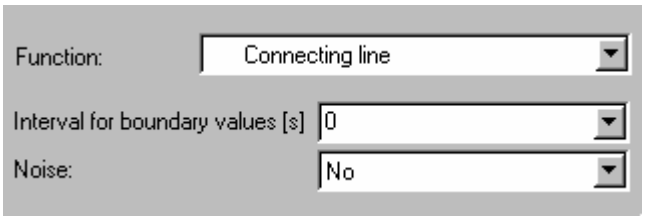
Many functions use the boundary values at the edges of the delineated region; for instance, all values within the delineated region are bridged by a connecting line or spline.

Now, it would be possible to simply use the boundary value itself for this purpose. However, this would not take the noise, which frequently is present, into consideration. The result of the function would be dependent on the noisy value at the region's particular edge, and this is largely a matter of chance. Therefore it would seem reasonable to take the mean of a small interval around the boundary, in order to reduce the influence of random noise. This interval should be, on the one hand, relatively small, but on the other hand still large enough to suppress the noise.

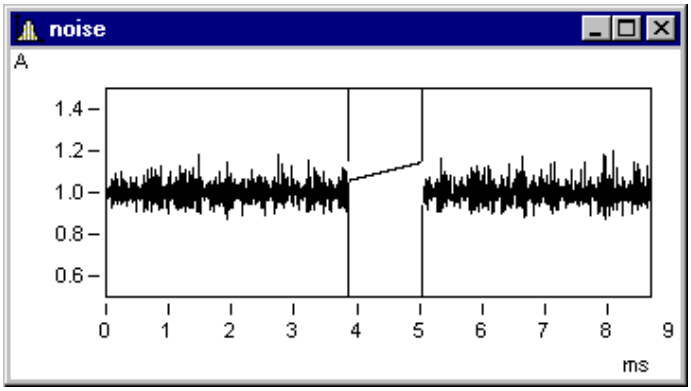
In the example below, the delineated region is to be replace by a straight line (between the boundary values):



In the dialog *Reprocessing signals*..., the following settings are made.

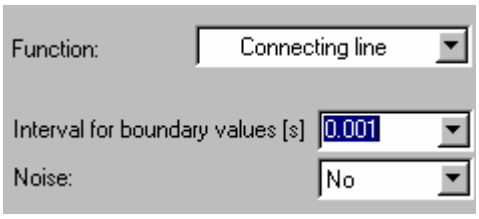


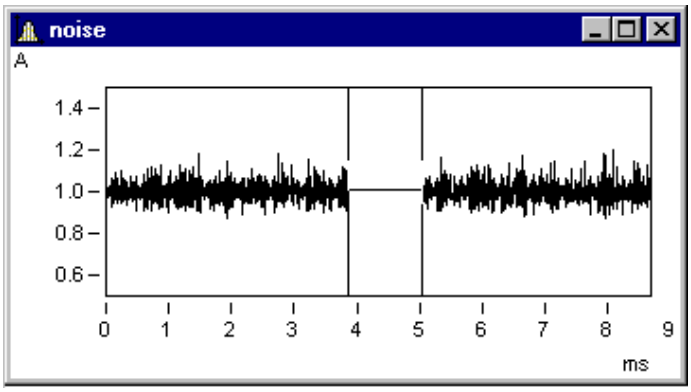
The following connecting line results:



This isn't what one would expect. By chance, the values at the edges (under the cursors) are random peaks.

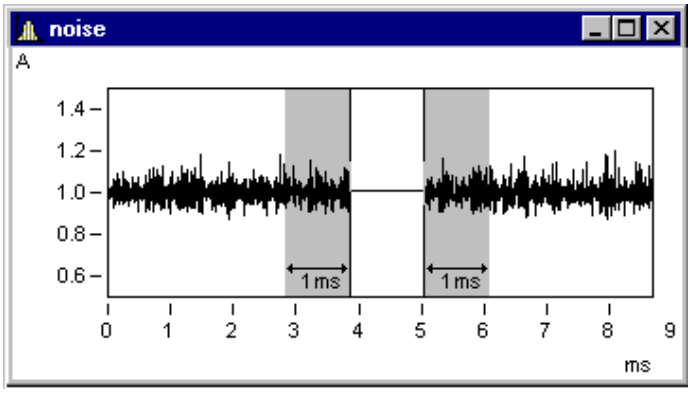
Setting a boundary interval of 1ms (0.001s !) produces proper results:





The boundary interval always is located outside of the delineated region:

Here, the boundary intervals are shaded in grey.



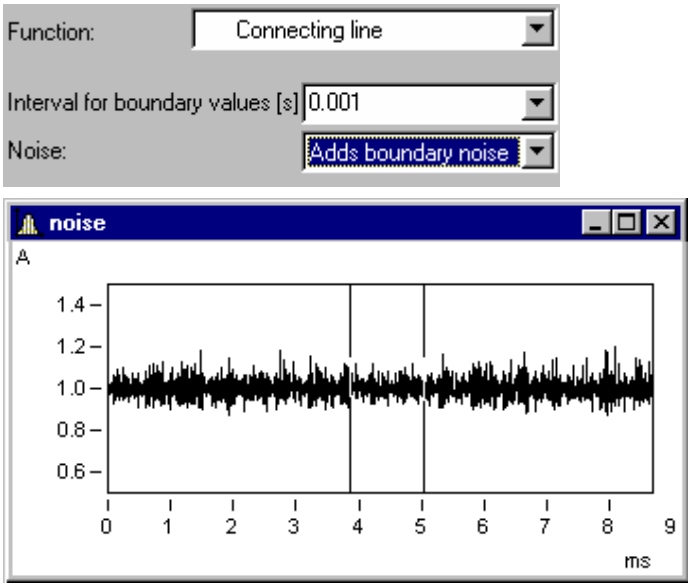
Remarks

If 0.0 is taken as the boundary interval width, the value at the exact edge is used. Thus, the minimum width of the boundary interval is 1 reading.

Adding noise

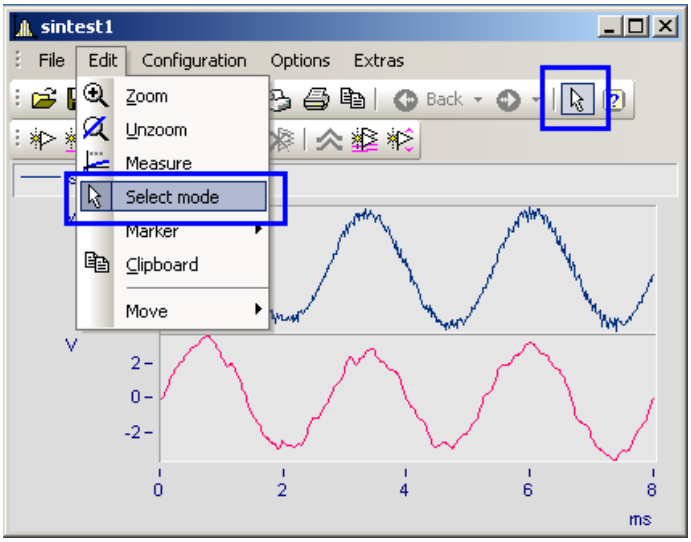
To the sample signal above, it is now possible to add noise. Toward this end, the noise in each of the boundary intervals is determined. This noise, cyclically repeated, is then added to the delineated region. Between the noise signal on the left and that on the right side, a smooth transition is accomplished by linearly increasing weighting along the width of the region.

With the following setting, noise from the boundary intervals is added to the delineated region. This produces a realistic signal appearance:



13.6.6.4 Select mode

The *Select mode* enables the user to select legends, coordinate systems, axes, lines and markers by means of the mouse. *Select mode* can be activated/deactivated either via the *Edit* menu or the toolbar. Alternatively, it is possible to activate/deactivate Select mode by double-clicking over a region where there are no objects, for instance in the middle of a coordinate system.



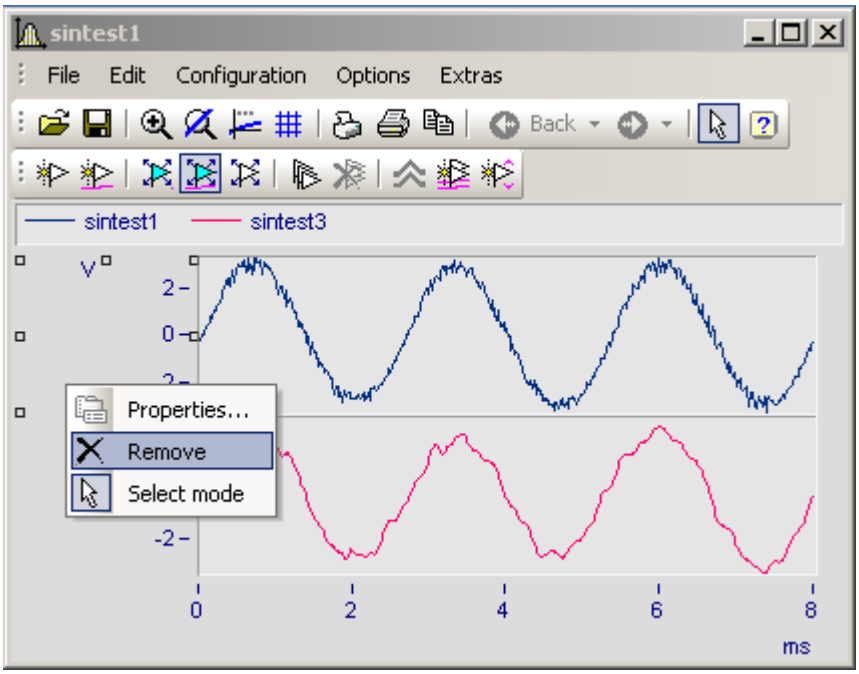
If you are in *Select mode*, the current object is highlighted. This is indicated by the presence of small empty squares at the object's edges. Any lines not selected appear dulled in color.

Multiple selection is possible by making a selection using the left mouse button while holding down the CTRL-key. However it is only possible to join up objects all of the same type (lines or axes) in multiple selection.

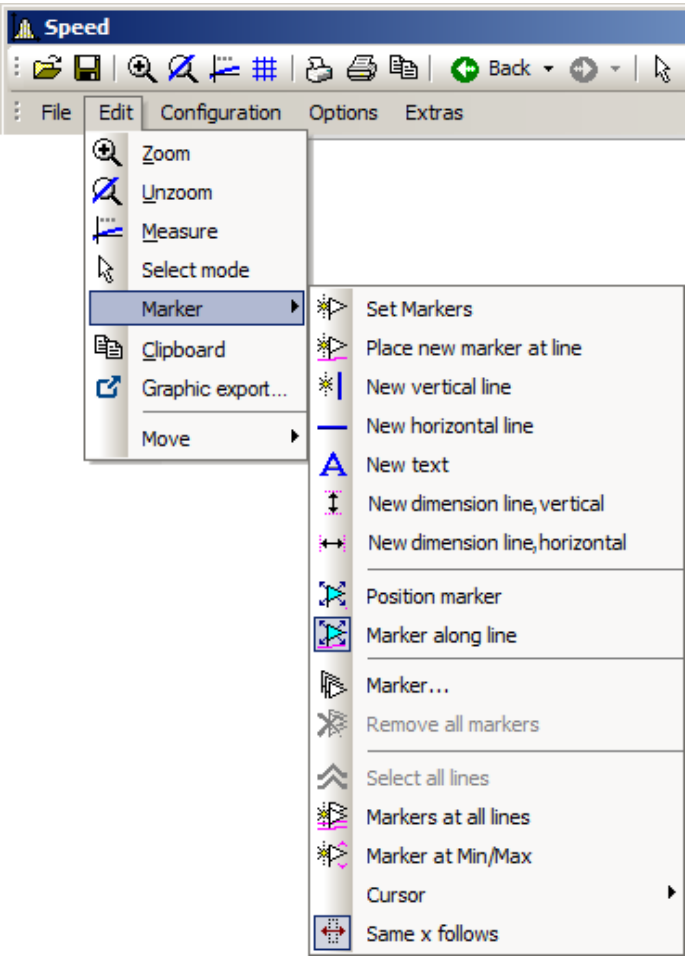
Among other things, the context menu contains the function *Delete*. This is only available if objects which can be deleted are available, such as a second y-axis. If such an object is deleted, any associated objects are also deleted. In the example, the lines in the deleted y-axis would also be deleted.

Example

You select the upper y-axis and deleted it. In the process, the sintest1 lines with which it is associated are also deleted.



13.6.6.5 Markers



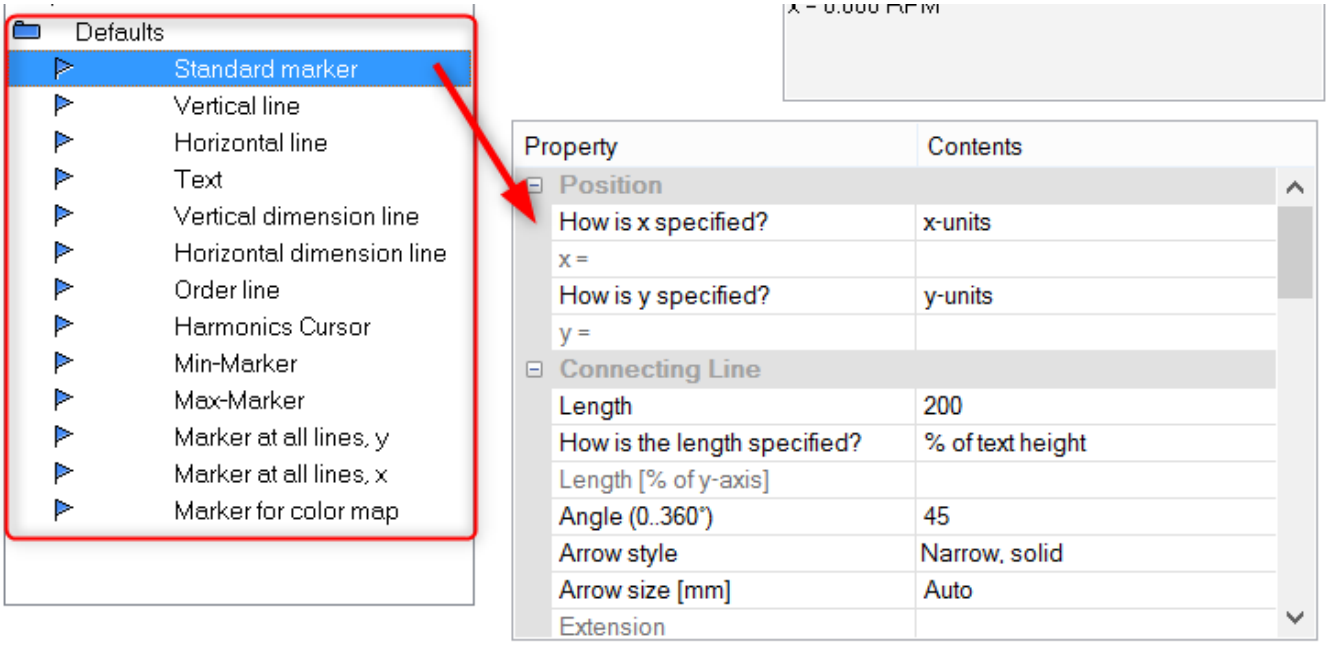
Function

A marker is a designated point in a curve window, to which a text can be assigned. The point itself is not visible, but a line with an arrow may point from the text to the point. The text can be enclosed in a frame; various characteristics can be established for the lines and font for each marker, such as color and size.

Defaults

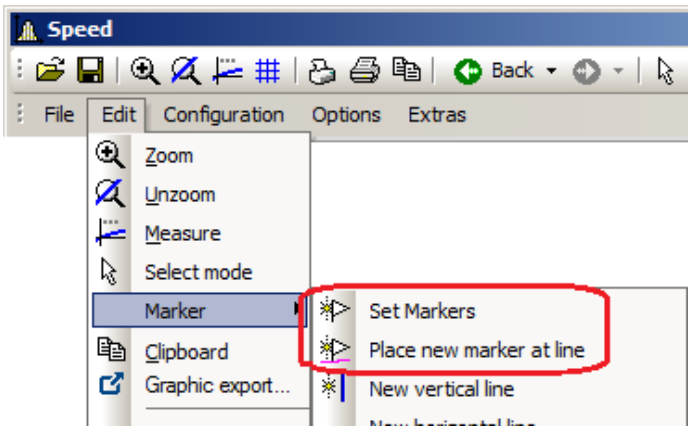
All marker types are described below. Each type has particular default settings which are applied when the marker is inserted.

You can customize these individually by modifying the parameters for the selected marker type.


Defaults
Standard marker
Vertical line
Horizontal line
Text
Vertical dimension line
Horizontal dimension line
Order line
Harmonics Cursor
Min-Marker
Max-Marker
Marker at all lines, y
Marker at all lines, x
Marker for color map
Property
Contents
Position
How is x specified?
x-units
x =
How is y specified?
y-units
y =
Connecting Line
Length
200
How is the length specified?
% of text height
Length [% of y-axis]
Angle (0..360°)
45
Arrow style
Narrow, solid
Arrow size [mm]
Auto
Extension

13.6.6.5.1 Set markers

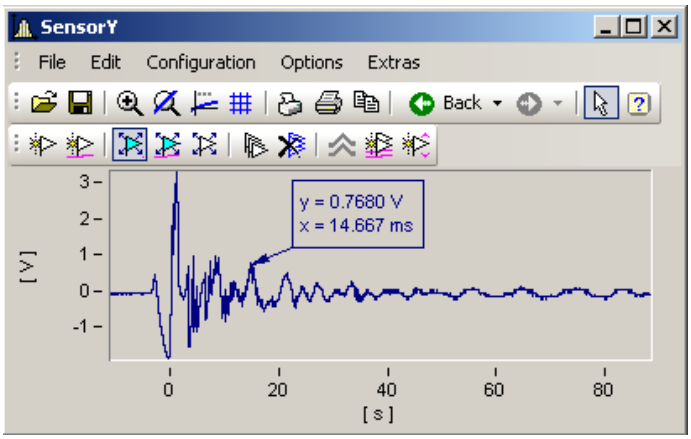
There are two ways of setting a marker, which are accessed either from the Marker toolbar or via the menu item *Edit/ Marker*.



Set Markers ( )

This function lets you generate markers at any desired position. If the function is selected, the cursor is represented by this symbol ( ), which disappears again after the marker is set.

Click to the point in the curve window to be marked. A dialog box is promted in which marker characteristics can be established. If no settings are made and the dialog is ended with <OK>, the x- and y-values of the marked point are used as the marker text. More about Marker Definition 1454.



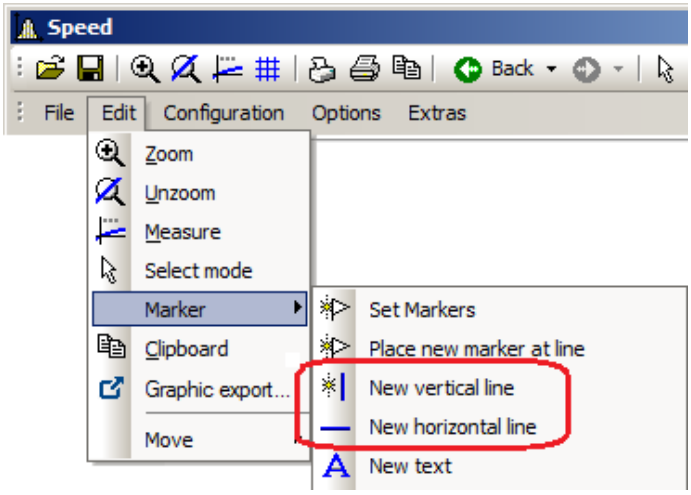
Markers can be placed anywhere in the curve window. Zoom more intensely if a marker should be placed near a line. Accuracy depends on the resolution. To place a marker exactly at a certain point in the curve, select the function *Place new marker at line*.

Place new marker at line ( )

This function only helps to generate markers at lines. If the function is selected, the cursor is represented by this symbol ( ), which disappears again after the marker is set. The marker automatically snaps to the next closest line and then moves in response to cursor movements only along the line, until the cursor approaches another line.

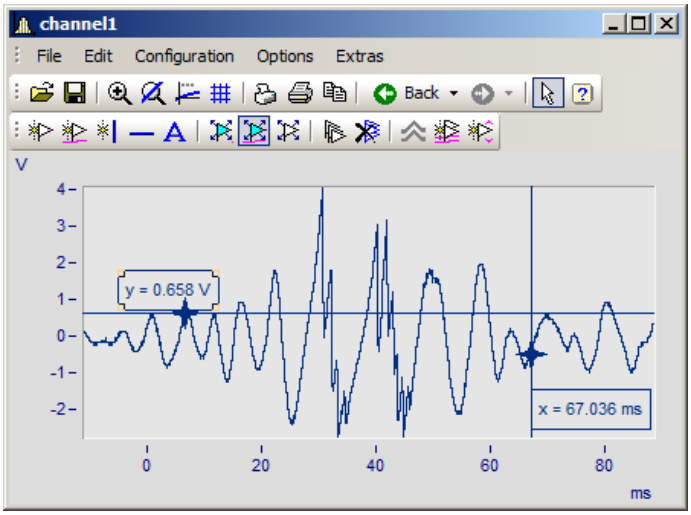
Both modes can be exited by a click of the right mouse button.

13.6.6.5.2 New line

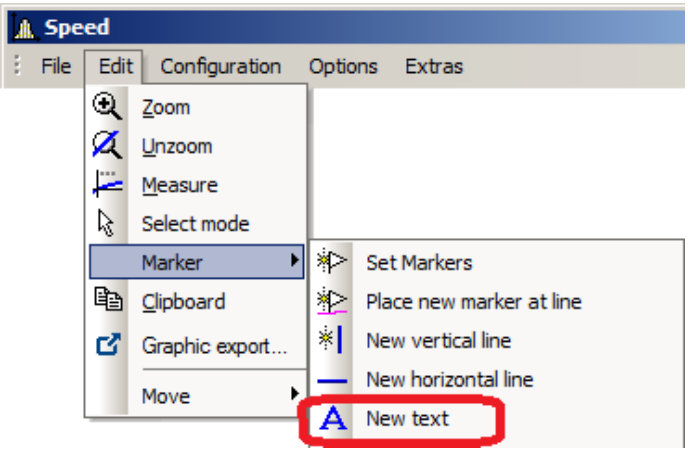


Vertical line new and **Horizontal line new**

Use these to create vertical or horizontal lines. The location is displayed as an amplitude or x-position in the text box. If the text box attaches directly to the line, it can be moved along the line. Drag the box further away from the line, it can be placed freely at any location.



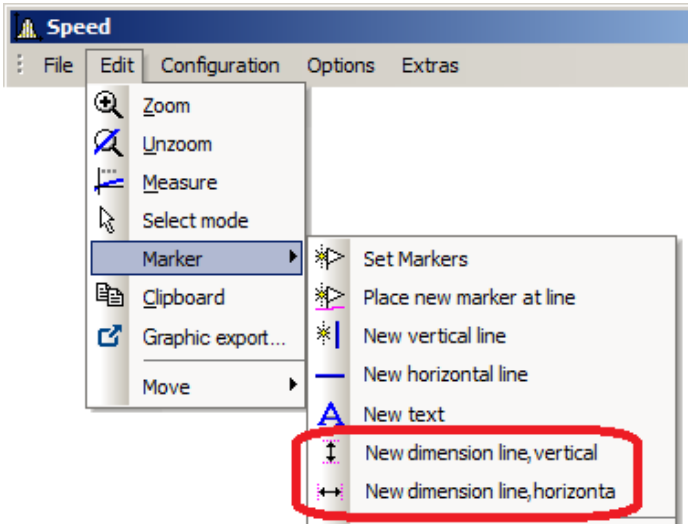
13.6.6.5.3 New text



New text

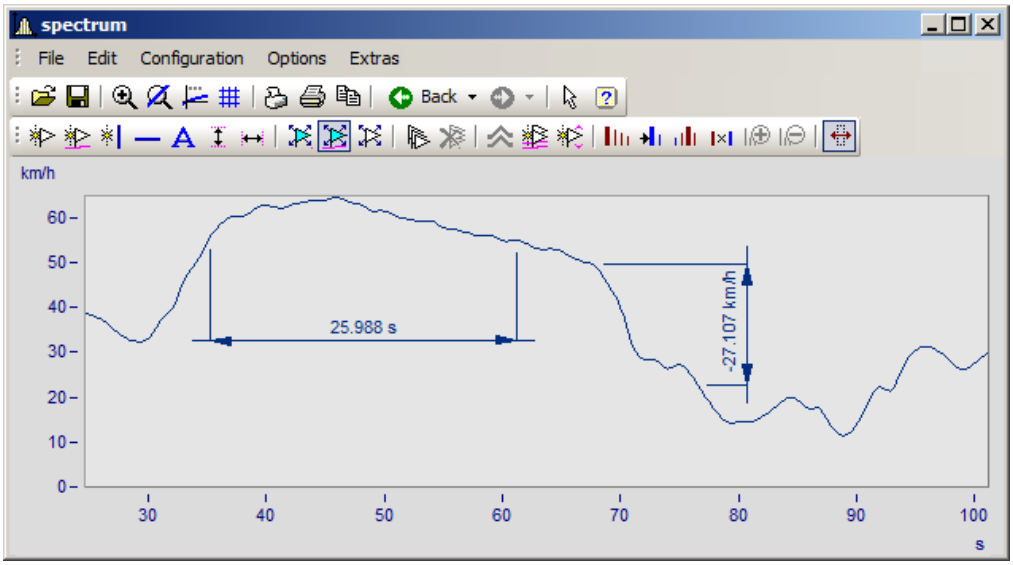
A text box without an arrow can be placed freely. Otherwise, the same settings are available as for markers.

13.6.6.5.4 New dimension lines

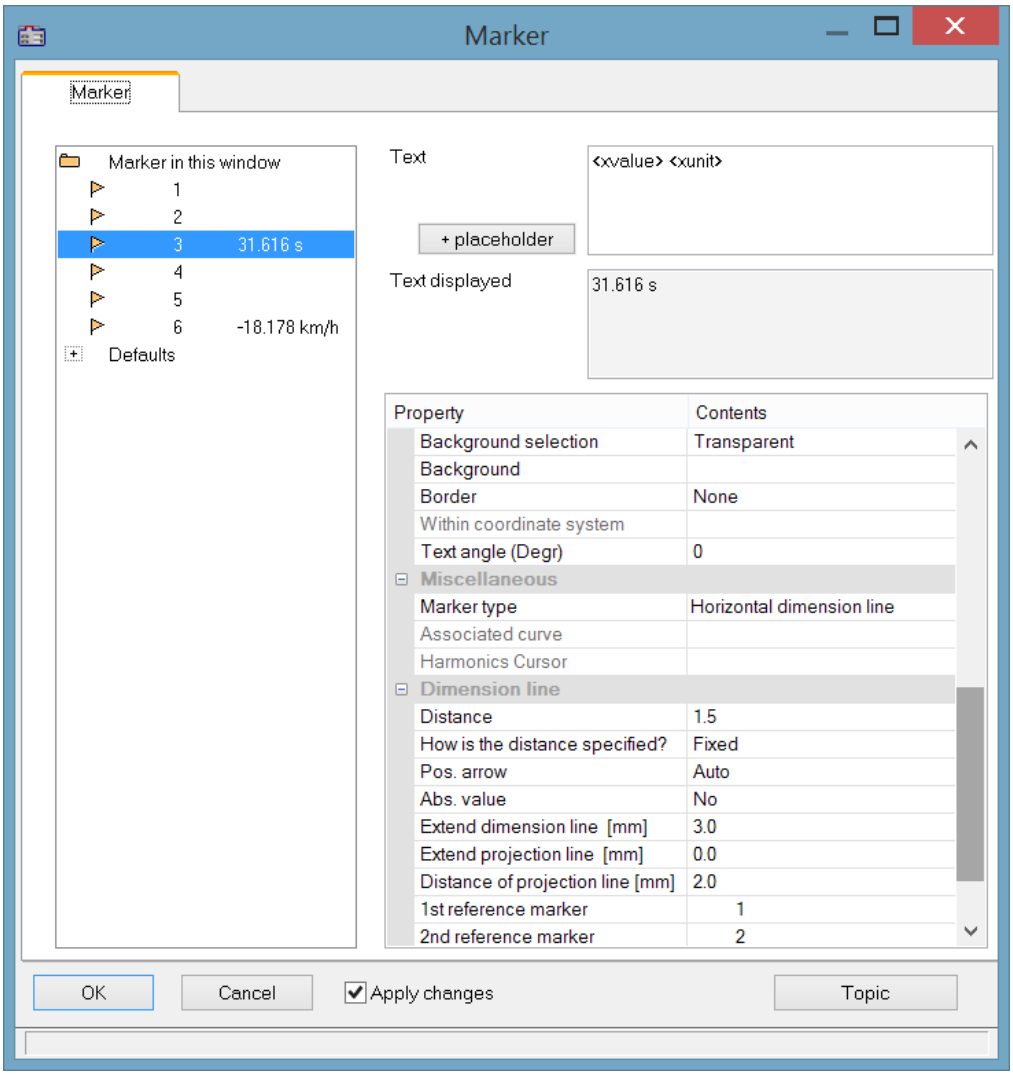


New Dimension Line: vertical and New Dimension Line: horizontal

Select this entry to create a vertical or horizontal dimension line between two points in the curve window where you click the mouse.



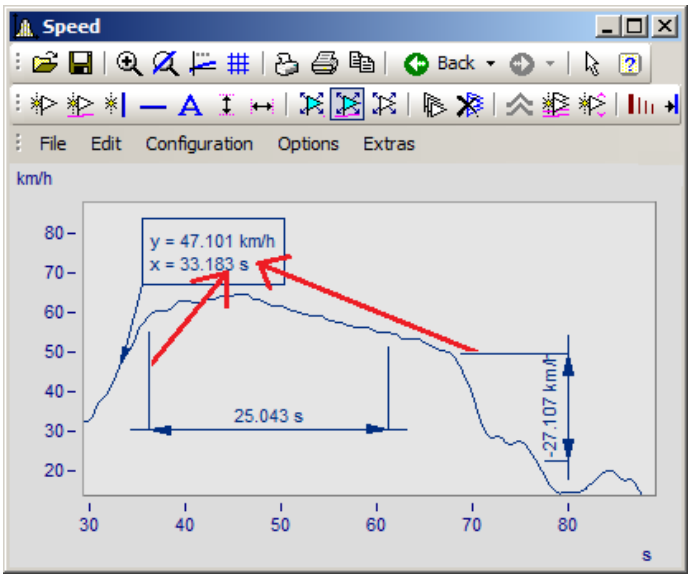
You can change the width of the dimension lines using drag&drop. The placement of the legend as well as of the line is positioned by drag&drop. Making settings for the markers is done as shown below:



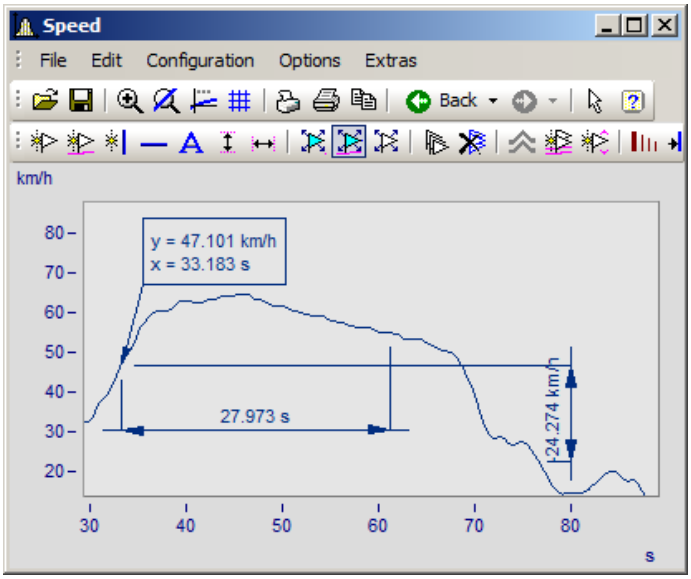
Position

Reference marker: A dimension line is defined by three entries for the marker: two reference markers (1, 2) and the actual line arrow and legend. If other markers are already in the curve window, these can also be used. Assignment of the markers is made by means of the list boxes for the 1st and 2nd reference markers. Alternatively, use drag&drop in the curve window to move a dimension line's extension lines to the existing marker.

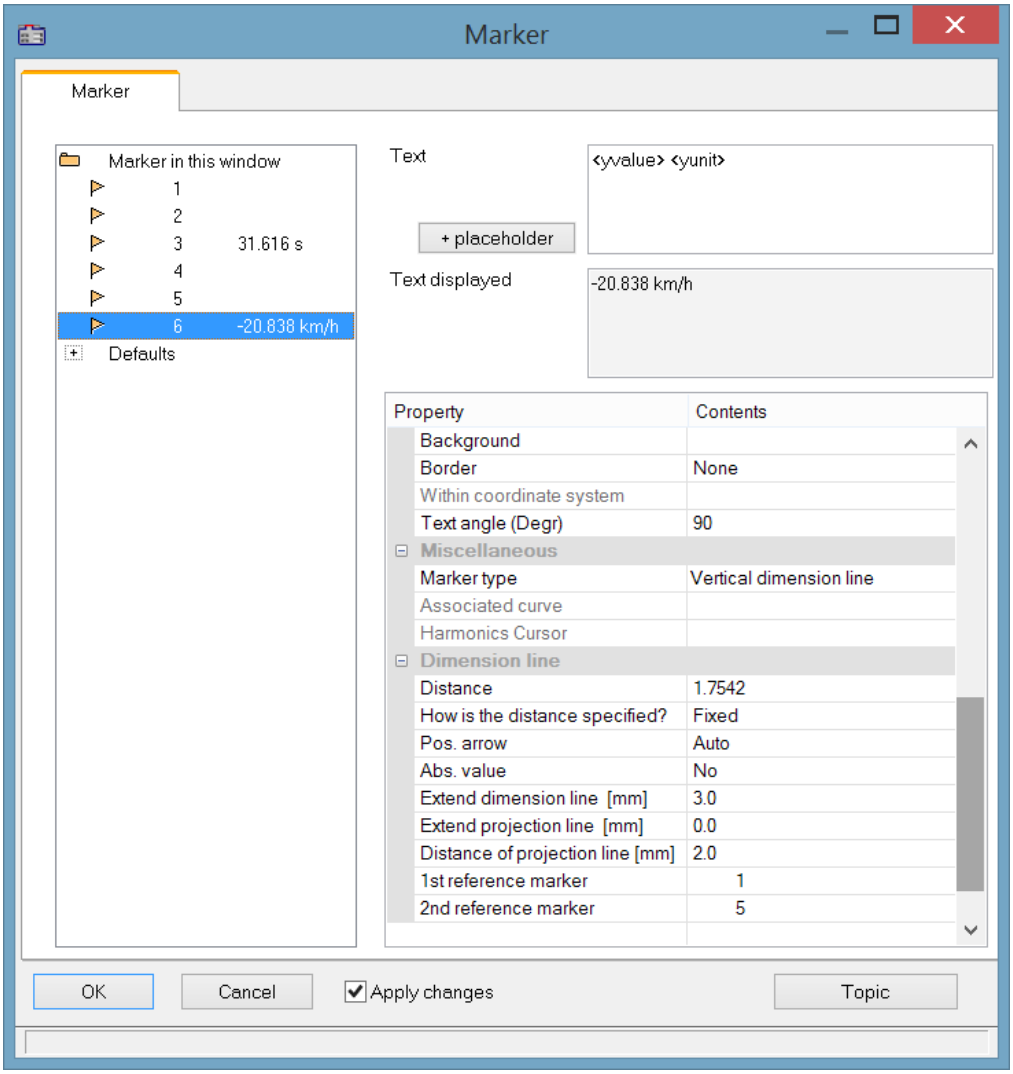
The example below shows how two dimension lines are linked to a marker. When the marker is subsequently moved, the dimension lines are adapted automatically.



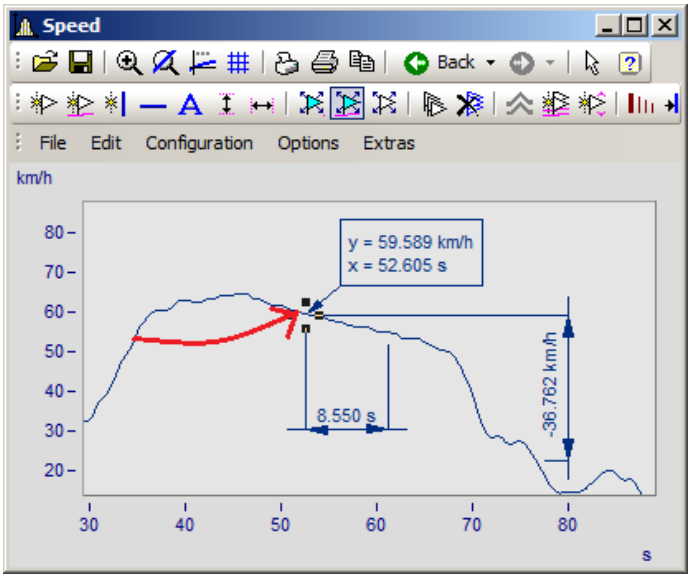
The marker and dimension lines are not yet linked. The extension lines are dragged to the marker.



Now the marker and the dimension lines are linked with the marker.

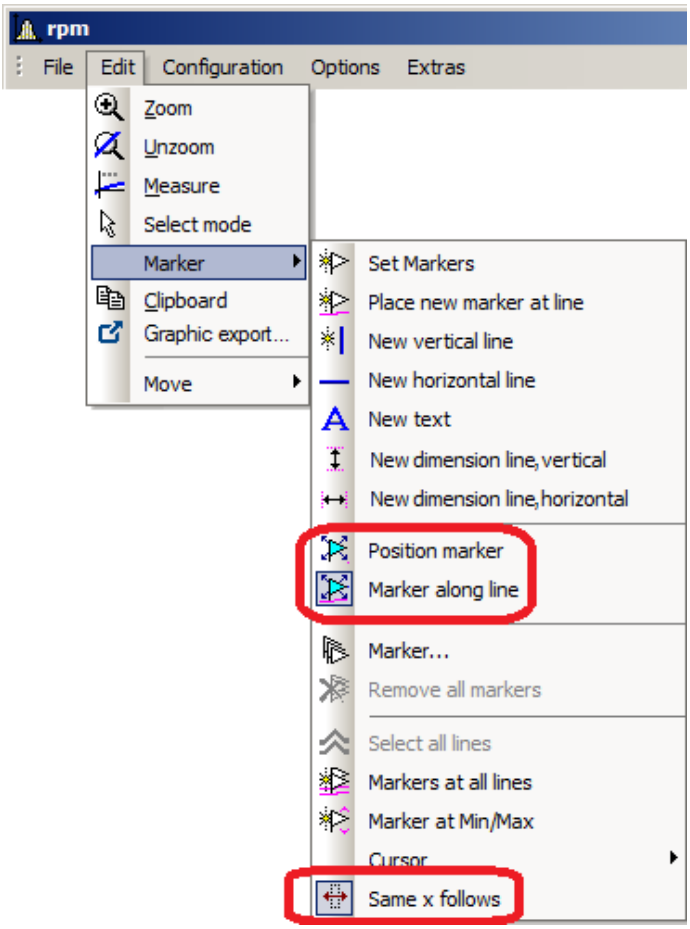


In the definitions dialog, the reference marker for the dimension lines is now linked with the marker.

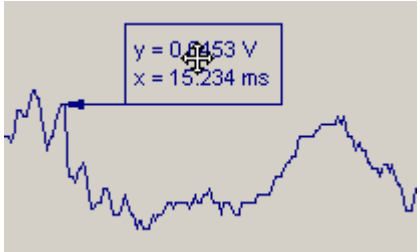
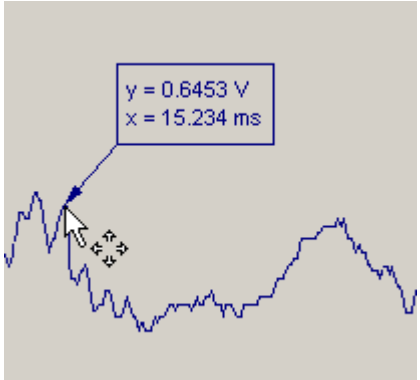


Now, when the marker is moved, the dimension lines are updated.

13.6.6.5.5 Move markers



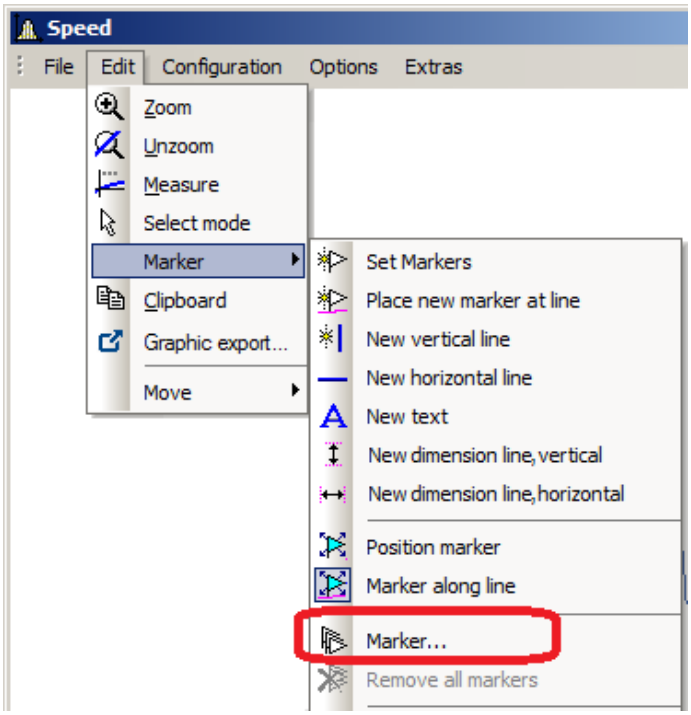
There are different ways to change the marker position:

Action		Description
	Position Marker	This function lets you move a marker freely, while the marker text is automatically updated to reflect the respective new position.
	Marker along line	This function lets you move a marker along a line, while the marker text is automatically updated to reflect the respective new position. If there are multiple lines in a curve window, the marker snaps to the line closes to the cursor's location.
	Move label	In order to move a marker's label, bring the cursor near the label or the arrow, so that the cursor changes shape as shown below. 
	Move marker	Existing markers can be moved subsequently. To do this, use Drag&Drop to move the arrowhead point. 
	Same x follows	Markers assigned to the same x-position can be moved jointly by means of this function. To do this, move a marker. The markers having the same x-position are moved along with it automatically.

Remarks

- Markers are assigned only to the curve windows, not the waveforms.
- In order to save markers, the curve configuration must be stored as a CCV file.
- Markers already established for a curve window may become meaningless if additional curves are modified or the XY-display is redefined. These markers can be deleted

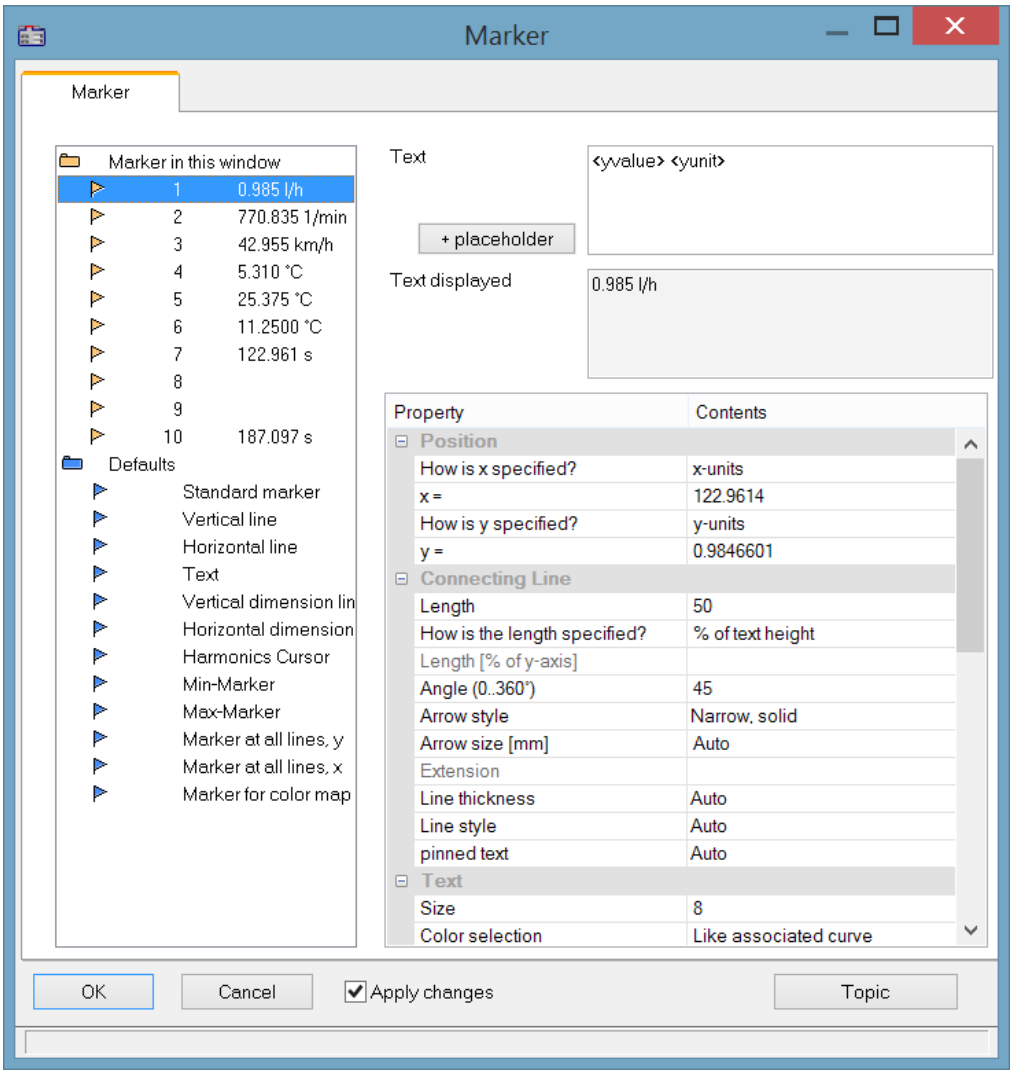
13.6.6.5.6 Marker definition



Function

Displays the properties of the markers created.

The following dialog appears if a marker was set in a curve window:



The list of markers, in which all markers in the curve window are listed, is found in the upper left box in the dialog.

Select an entry from the list to change the settings for the corresponding marker. Several markers can be selected by dragging the mouse across the entry or clicking several times on the entry while holding down the <CTRL> button.

The Marker-**properties** are listed at right in a table and are set using the column at right.

Marker type

Along with the usual markers having arrows and a text box, the following marker types are available:

- Vertical and horizontal lines
- Text
- Vertical and horizontal dimension lines

Normally, the type is determined already when the marker is set. By means of this combo-box, it can be changed subsequently.

Position

Determines the marker's position. By default, the position is specified in the coordinates of the curve to which the marker is assigned. In addition, the numerical value can also be specified in x- or y- units, or in percentage of the x- or y-axis. Select the corresponding entry in the combo box to the right of the number boxes. When specifying the numerical value in percentage of the axis, 0% is at the lower left, 100% at the upper right of the coordinate system. This allows markers to be defined so they are always visible in the curve window, regardless of the scaling. This is useful for comments, which needn't necessarily be assigned to a certain line in the coordinate system.

When changing from dB display to linear display, the axis coordinates change and the markers slip from their position. The scaling should be determined before the markers are specified.

Connecting line

The connecting line between text and marker is arranged in this group.

Settings	Description
Length	The length of the connecting line between the text and the marker. It is specified either in x-units, y-units, in percent of axis length or in percent of the text height. The last option is recommended. All value specifications should be in scientific notation with units, meaning that the exponent must be indicated. For a length of 0, no line is drawn. If an arrow is set, the length of the line is the entire length for line and arrow. If the line is shorter than the arrow, the arrow is drawn completely.
Angle: (0...360°)	The angle of the connecting line can be set between 0° and 180°. The line is horizontal if an angle of 0° is specified. This means that the line can be set to run in any direction.
Arrow	The line between marker and text may have an arrowhead at the end of the marker. The size and type can be selected: • none • narrow, solid • Point • wide • large • Diagonal dash • narrow • large, solid • Star • wide, solid • circle • Default The size of the arrow is oriented to the size of the symbol, see also "Clipboard settings 1489".
Line thickness, Line style	Settings like for Lines 1370. At printout, the line appears in the thickness of the cursor-line, see Clipboard Settings 1489.
Text corner	Position of the text box (upper right, upper left, lower right, lower left)

Text

The text specified in the text window is assigned to the marker, and can consist of several lines. Press <CTRL> and ENTER to enter a new line. If a marker is created, this text is already defined as the x- and y-measurement values. This text can be overwritten.

Placeholder

The edit box initially contains the placeholder *<auto>*, which by default represents the y and x-value. This entry can be supplemented/replaced with the following placeholders:

Placeholder	Description
<xunit>, <yunit>, <zunit>	Display of the x-, y, or z-unit
<xvalue>, <yvalue>, <zvalue>	Display of the marker's x-, y, or z- component. For xvalue, mostly the time value in the x-axis' formatting.
<name>	Display of the variable's/channel's name
<comment>	Comment on the variable
<xtimeofday>"	Display of the marker's time. For this purpose, the x-axis must represent absolute time.
<xdate>	Display of the marker's date. For this purpose, the x-axis must represent absolute time.

Placeholders for values, with specified precision level

Placeholder	Description
value**:fx** e.g. <yvalue:f2>	Num#ber of digital positions (0..15); in the example 5.34211 -> `5.34`
value:**fxpy** e.g. <yvalue:f2p3> <yunit>	Specified number of decimal places and order of magnitude as exponent of 10. The exponent of 10 is only displayed along with the unit. In this example, 3556.23 becomes `3.55 10^3 RPM`.

Note

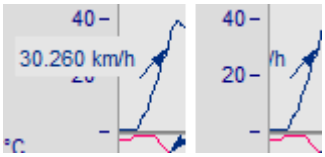
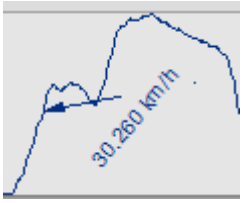
- Greek letters 1512 can also be displayed.
- Numbers without a unit sometimes come to a misleading representation:

Example:

31.000.000 represented with <yvalue: f0p6><yunit> is displayed as 3110^6

A multiplication sign or blank space creates clarity:

<yvalue: f0p6>*<yunit> -> $31*10^6$

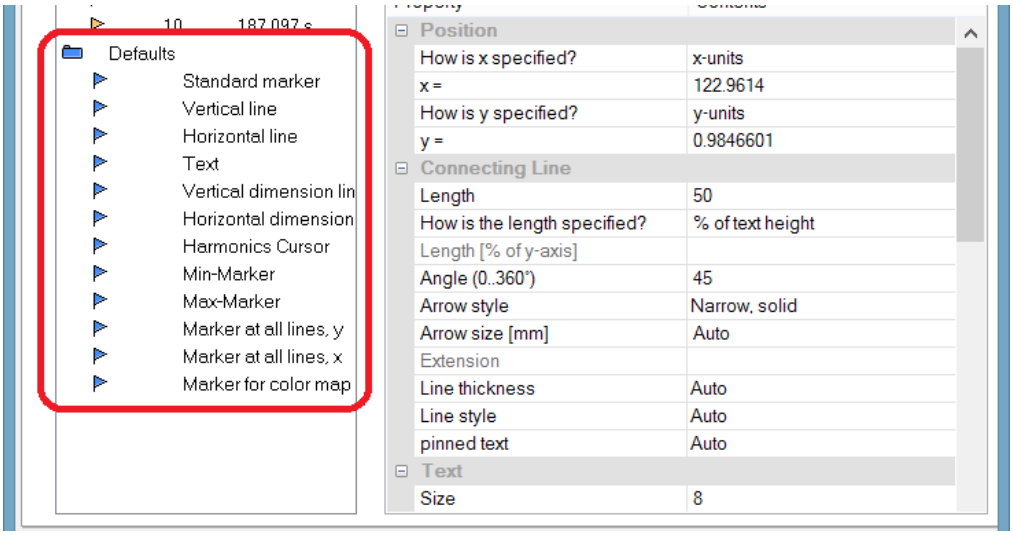
Settings	Description
Size	The size of the text can be defined in points. In general, a 12-point font can be read quite easily. The font specified in "Clipboard settings..." is the font used for printing. A true-type font should be selected. The standard font set for the curve window is also used for the monitor screen (see "Presettings 1496").
Color	The color can be selected as absolute (e.g. red) or relative. A relative color definition refers to a color already defined for the curve window. For example, if the color of the first curve in the window is selected, it is possible to select another color for the printout and the screen.
Background	The background of the text can be in the same manner as the text color. A transparent background is also possible, so that lines behind the background are not completely hidden.
Border	The text may have a border around it. Various options are available for the border joining the connecting line with the marker: • *none, simple, pointed (trapezoid-shaped), double border*
Within coordinate system	When the text box is located at the edge of the coordinate system, you can decide whether it overlaps the edge or whether it is clipped. 
Angle text (Degr)	Angle of the text box up to ±90 degrees. 

Delete

All lines selected in the marker list are deleted.

Edit defaults

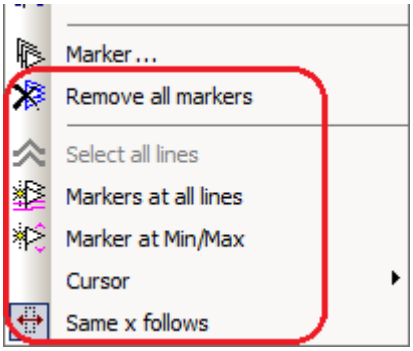
The ***Defaults*** for the *Marker types* are listed at left links and can be edited at any time using the Properties table.



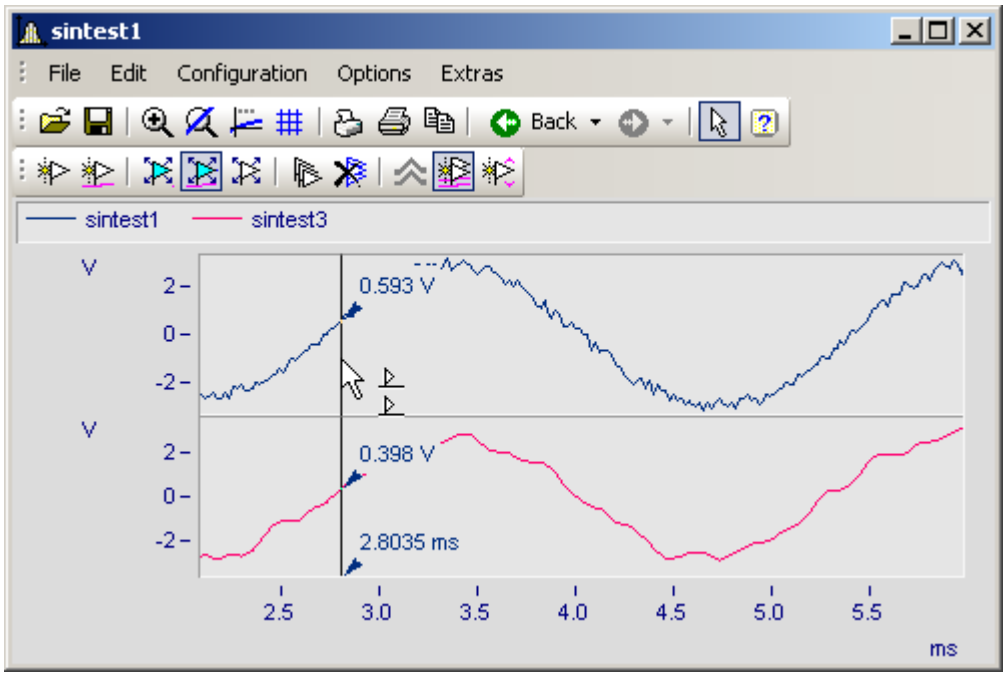
Remarks

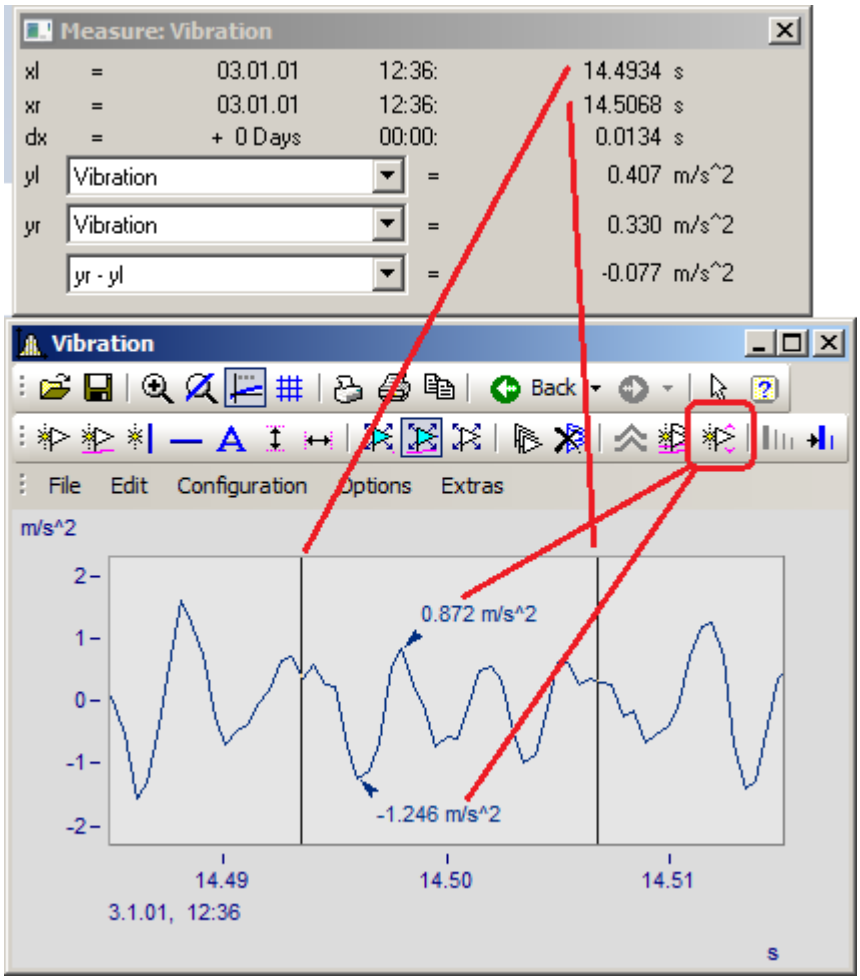
- If several markers are selected in the list, the characteristics for all of the selected markers are indicated. If a characteristic differs among the separate markers, this is indicated by "???" in the corresponding box. If a characteristic is changed, this change is effective for all selected markers.
- An inapplicable entry in a text box is ignored.
- When changes are being made in a dialog, they are implemented immediately and displayed. They can be seen immediately when the dialog is <Shift>ed next to the curve window. If the curves are very long and the on line-update is too slow, the curve window should be placed so that the graphics can no longer be seen when working in the dialog. It is recommended to use a transparent display for the representation in a waterfall diagram.
- Markers already defined for the curve window may lose their meaning if the curves are modified or XY-plots are redefined. These markers should then be deleted.
- The definition of markers is recorded when the curve window configuration is saved. The markers are also reloaded when a configuration is loaded. However, the markers may not be visible in the window because the settings do not fit the window or the curves.
- To design markers which are not device-specific, the colors should not be specified as absolute. Use instead a default or a color in the curve window (e.g. the color of the first curve). The default itself can be dependent on the device. The same is true of the font size and the line characteristics.

13.6.6.5.7 Extra marker functions

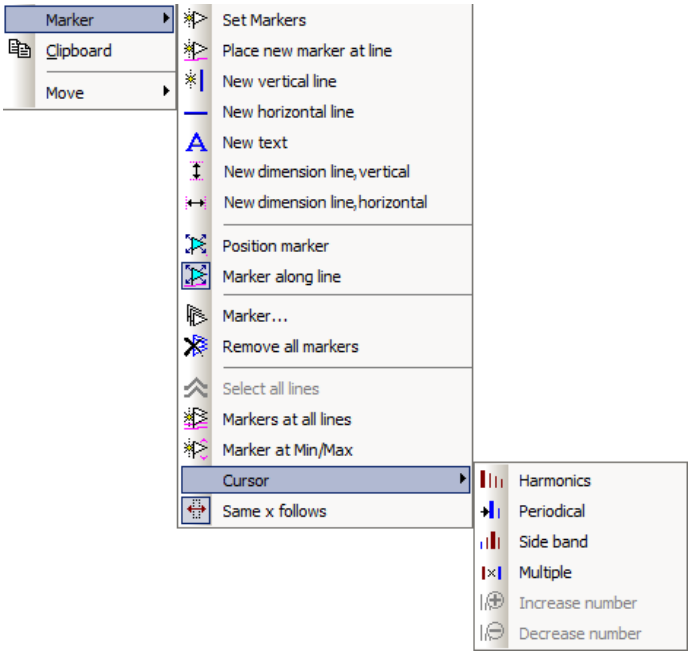


There are four additional marker functions available:

Menu item	Description
Delete all markers	With this function, all markers set in the curve window can be deleted at once.
Select al lines	With the Select mode active, this function selects all lines in the curve window.
Markers on all lines	Set markers at all lines or at all lines selected in the Select-mode. Select the lines desired an click on the symbol in the toolbar. A vertical line appears which extends across the entire curve window, even when multiple y-axes are stacked one above the other. The line denotes the y-value and the x-value on the y-axis. Simply move the line with the mouse and confirm setting of the marker at the desired position by clicking the mouse. The position of the markers as well as their labels can be moved as desired, just as described in the section "Set Markers 1452".  **Note** Markers on all lines If two lines are close together so that the positions of the markers almost match, only one marker is created for this position.

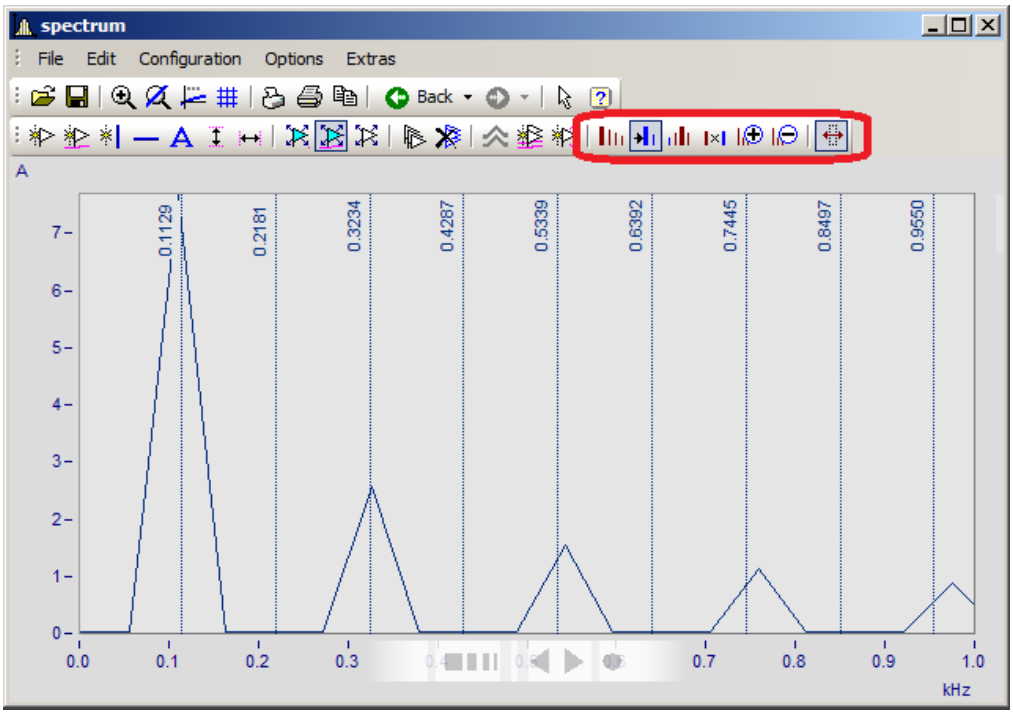
Menu item	Description
Markers at Min/Max	Markers are set for the maximum and the minimum of the line selected. If multiple curves are in a curve window, the function only applies to the line selected. As described in the section "Move Markers 1452", the marker's position and its label can be moved as desired. The function only affects the selected time segment. For this reason, zoom on to the region before setting the marker. If the "*Measure*" function is also active, the system finds and denotes with markers the minimum and maximum just between the two measurement cursors. 

13.6.6.5.8 Harmonics cursor

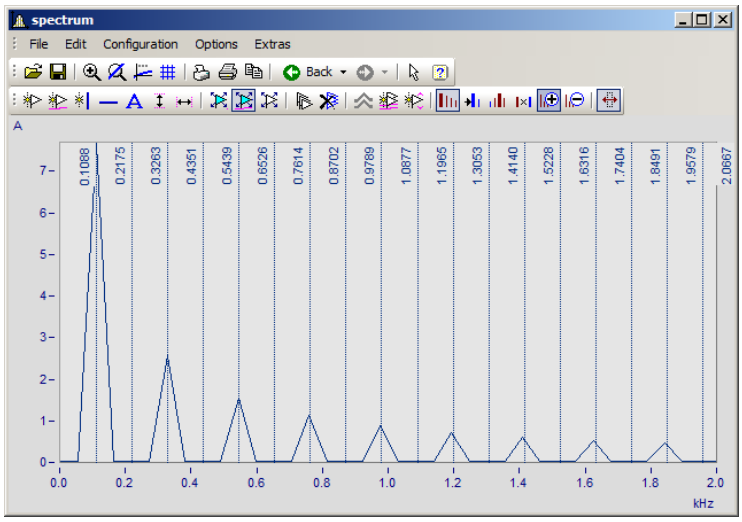
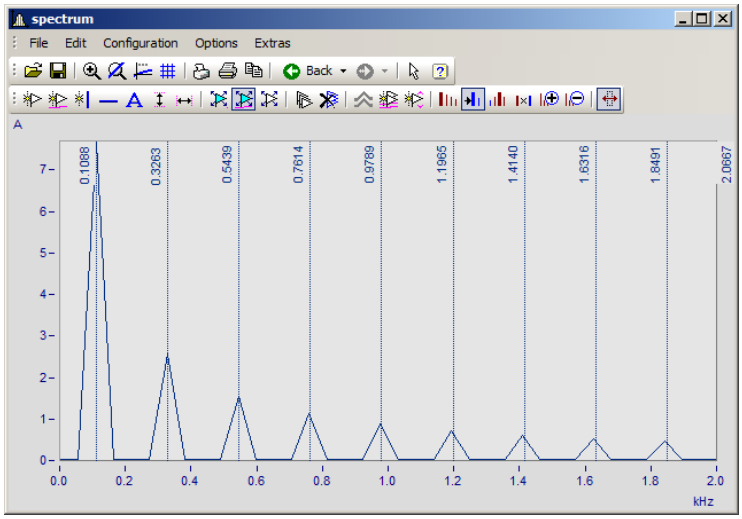


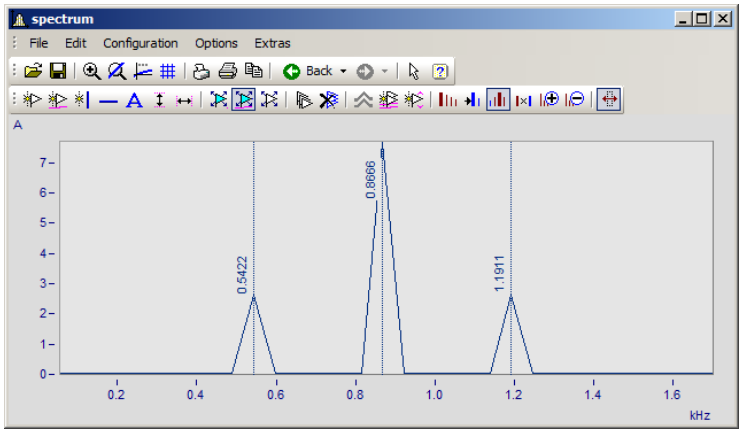
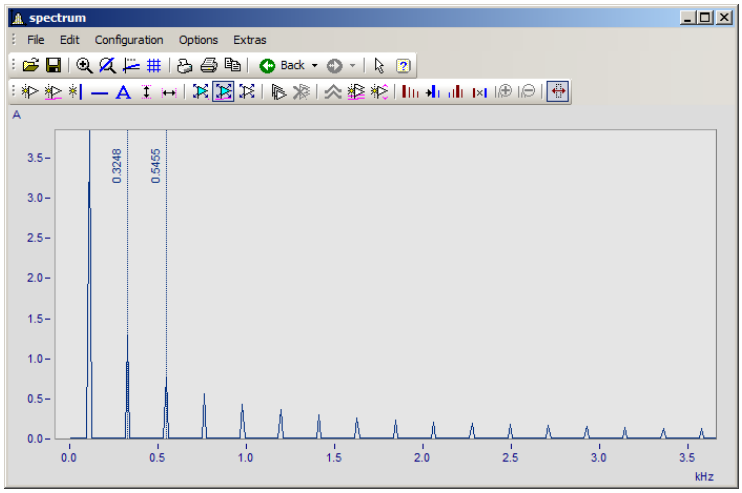
Function

Harmonic cursors indicate periodic multiples of a fundamental frequency. Set the marker at the position of the fundamental oscillation. The position can be readjusted subsequently, if desired.

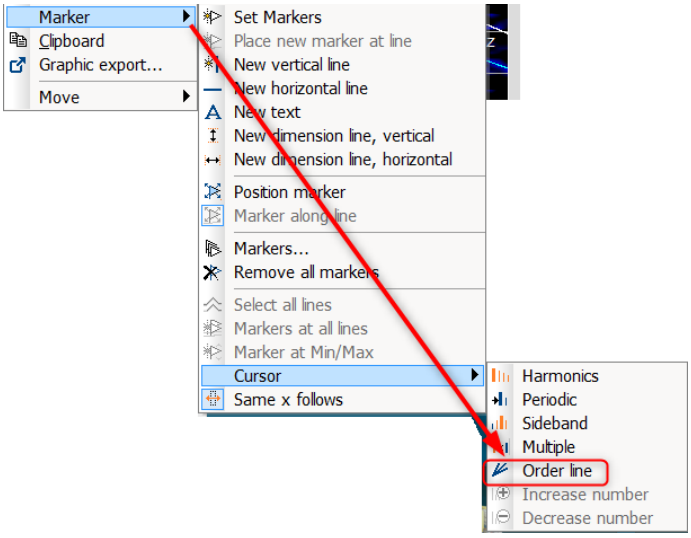


The following types are available:

Menu item	Description
Harmonics	Fundamental oscillations with harmonics. 10 equally spaced markers are inserted. Place the fundamental oscillation. The subsequent harmonics follow the fundamental oscillation by the respective multiple. The fine adjustment is performed by moving the upper harmonics. 
Periodic	Periodic processes with arbitrary start time. The fundamental frequency can be freely placed. The distance between the harmonics is independent of the fundamental oscillation. 

Menu item	Description
Side band	Fundamental oscillation with side bands. 10 equally spaced markers are inserted. Position the fundamental oscillation. The harmonics which follow the fundamental oscillation by the respective multiple. Fine adjustment is performed by moving the upper harmonics. 
Multiple	Distances at a fixed ratio. Two frequency lines which can be freely positioned. 

13.6.6.5.9 Order lines



Calling the Marker dialog for order lines

Function

The amplitudes of RPM-dependent vibrations can be displayed by means of various color diagrams.

When the amplitudes are plotted over the RPMs, the correlation becomes visible as straight **order lines**, which begin in the origin of the coordinate system. Frequencies which are independent of the RPMs appear in this diagram as horizontal lines of constant frequency.

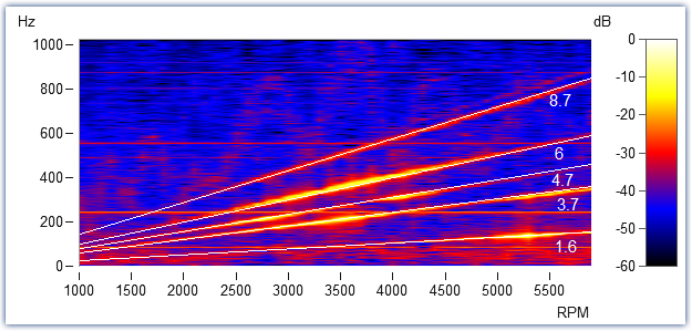
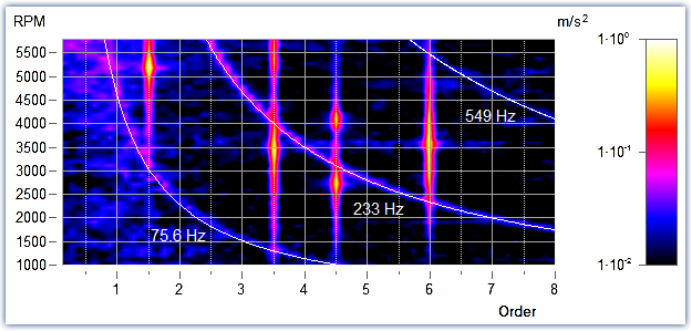


Fig. Amplitude spectrum with frequency over RPMs

In the Markers dialog, these order lines can be highlighted and labeled. Clicking on the marker symbol "*Order line*" creates the first marker, applying the presets for calculating "*Order line in the RPM-spectrum*". Subsequently, you open the marker definition dialog by double-clicking and you can design the display.

Calculating the diagram over these orders, they appear as vertical lines. Fixed frequencies are distorted to **hyperbolas** by these conversions.

For the example shown here, setting for the calculation in the marker definitions must be "*RPM and Order*".



RPM and order with hyperbolas

As with all markers, the text belonging to the line can be designed with placeholders, freely defined texts and custom formatting.

The line's position can be moved manually using the mouse. As a snap grid, the parameter *Multiple of* in the section *Order line* is used.

New hyperbola markers get the value of the highest frequency line x120% as initial frequency - and this to the next grid *Multiple of.*

The most important properties of this *Marker-type: Order line* are:

Property	Contents
⊟ Miscellaneous	
Marker type	Order line
Associated curve	At the first y-axis
Harmonics Cursor	
⊞ Dimension line	
⊟ Order line	
Parameter	6
Calculation	Order line in the RPM-spectrum
Start of the line [%]	0
End of line [%]	100
Labeling position [%]	95.079
Multiple of	0.1

Order line

Calculation with unit over unit

Order line in the RPM-spectrum

RPM over frequency

Frequency over RPM (=default)

Angular frequency over frequency

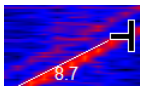
Frequency over angular frequency

Hyperbola in the order spectrum

RPM and order (=default)

Angular frequency and frequency

Parameter is the value of the order or of the constant frequency which appears as a hyperbola. This value determines the position and is represented by the placeholder *<yvalue>* and formatted accordingly, e.g. *<yvalue:f1p0> Hz*



Start/End of line [%]: Determines whether the line is to be drawn up to the edge of the coordinate system (0-100%) or should keep some distance from it (e.g. 5-95%). The ends of the line can also be edited graphically using the mouse.

Labeling position [%]: This value specifies the position in percent of the visible order line length. The text can also be positioned manually using the mouse.

Multiple of: When the order line/hyperbola is moved, the position snaps to multiples of this value. When changing *Multiple of*, the parameter in the curve window is updated immediately, but not in the properties table. This allows further editing of the value in the properties. With Ok of the properties or click on the marker list the parameter is accepted.

Additionally, the section *Text* offers the usual formatting properties for texts, and in the section *Connecting Line* there are parameters such as *Line thickness, -style* and *Text angle*.

Note Order lines

- The first order line applies the presettings for the marker-type Order Line 1443. You are able to format these according to your wishes. If any order lines are added, the settings for the previous order line are applied.
- It is possible for a marker to be inserted in the range which is not visible, e.g. if an order spectrum was calculated only for a range of 2000-4000 RPM. In this case, the marker must be given a parameter value in the visible range by means of the marker definition.

13.6.6.6 Export

13.6.6.6.1 Clipboard and Graphic export

The graph displayed in the curve window can be sent to the Windows clipboard. The clipboard is a Windows tool which allows applications to exchange data in any form, e.g. text or graphics.

With the menu item *Edit\Graphic-export..* 1487. the graphic can be saved as a file in an image or PDF format. The default settings 1501 are taken into account here. For example, it is possible not to overwrite an existing PDF file, but to add the new graphic to the document.

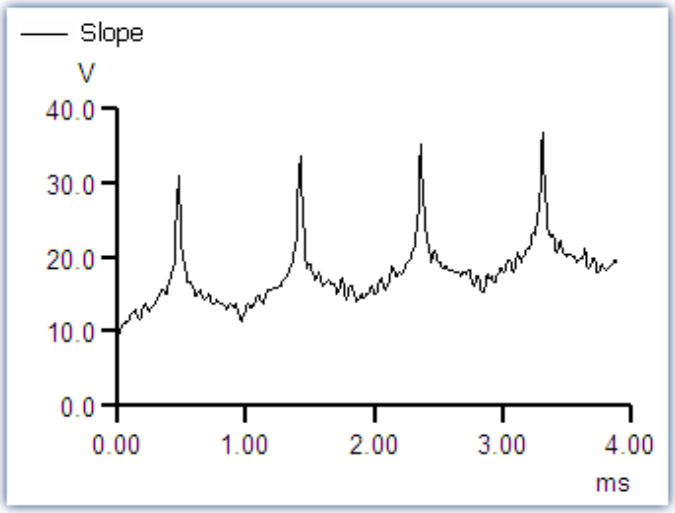
When one application sends graphics to the clipboard, another application can read and retrieve these graphics from the clipboard. In general, data are not damaged when they are read from the clipboard, which means that applications can retrieve the same graph as often as desired. Existing graphics are replaced when an application sends new graphics or text to the clipboard. Only the last information sent to the clipboard can be read.

In the case of curve windows, graphics are sent to the clipboard in the form of a memory-resident metafile. A meta file is a standard format defined in Windows and is supported by many applications. The meta file does not contain screen pixels; instead it describes the graphics using a multitude of vectors. This creates a high-resolution graphic display which can be arbitrarily scaled up or down.

Graphics sent to the clipboard can be read by text processing or desktop publishing programs. In that environment the graphics can be combined with text and other graphics to create documents to be printed by any printer or plotter supported by Windows. Graphics are printed in the best resolution possible for the output device.

Graphics are created in the fashion specified in the menu option "Options/ Clipboard settings...". The specified size, font and line width are used to format the graphics.

Graphics can be output to the clipboard in color or in black and white. Have regard for the color setting possibilities, especially the printer's. The curve window copied to the clipboard can, for example, have the following layout:



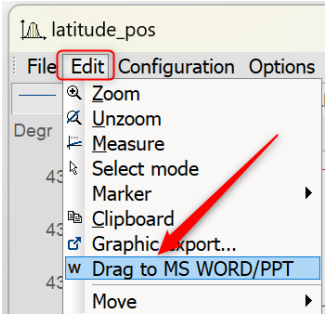
Remarks

- Letters may look distorted, lines may intersect and overlap and details may disappear when a meta file created from the curve when another program is loaded. The image can be improved by enlarging the display on the screen. Meta files are designed to optimize the image output by printers or plotters, and not for ideal display on the monitor.
- The fonts used for the meta files are selected from the available fonts for the printer (see "Printer Setup 1484 "). Not all of these fonts are easily legible on the screen or available in the necessary size.
- A meta file can only be created when sufficient memory is available. The clipboard remains empty or is emptied if the available memory is insufficient.
- In some extreme cases, it may not be possible to create a suitable meta file. Such situations can occur when several curves are displayed, or in connection with dotted, bar-graph or XY-representations. It is often helpful to display a smaller section of the waveform, to change the display mode, to cut out a section of the waveform or reduce data using mathematical functions.
- When axes scaling is set to automatic, the axes labeling on the screen and on the printer will be calculated differently for the current dimensions. Specify a set scaling and a set number of markings to achieve clearly defined proportionality.
- In printing, texts may overlap if the selected font is too large.
- If a waveform does not have a "created.."- time, the current time is automatically designated as such.
- Images of curves sent to the clipboard can also be incorporated in reports using the imc Report Generator. However, a direct transfer of the curve window is recommended, whereby objects pose as curves. See chapter 'Report Generator 1482' in this document.
- The time format (time and date) can be changed using menu item "Country settings..." in the Windows Control Panel.
- Even the best and most popular text processing and desktop publishing programs contain minor flaws. Although the graphics in most programs appear to be correct, one of these programs does not correctly interpret the offset of the curves, which is present as a command in the meta file. As a result, when graphics are inserted in this program, the curve lies next to the coordinate system. In another program, the clipping (fading-out) of curve sections extending beyond the coordinate system is ignored. Thus, the curve may extend well into the labeling or generally beyond the coordinate system. It is recommended to transfer the graphics to the Report Generator or to a text processing program directly, without modifying the graphics.

Reference Clipboard Settings

Section 'Clipboard Settings 1489'

13.6.6.6.2 To MS WORD (OLE)



Users with an **imc FAMOS Enterprise** or **Runtime** license can embed the curve window into a WORD or PowerPoint document.

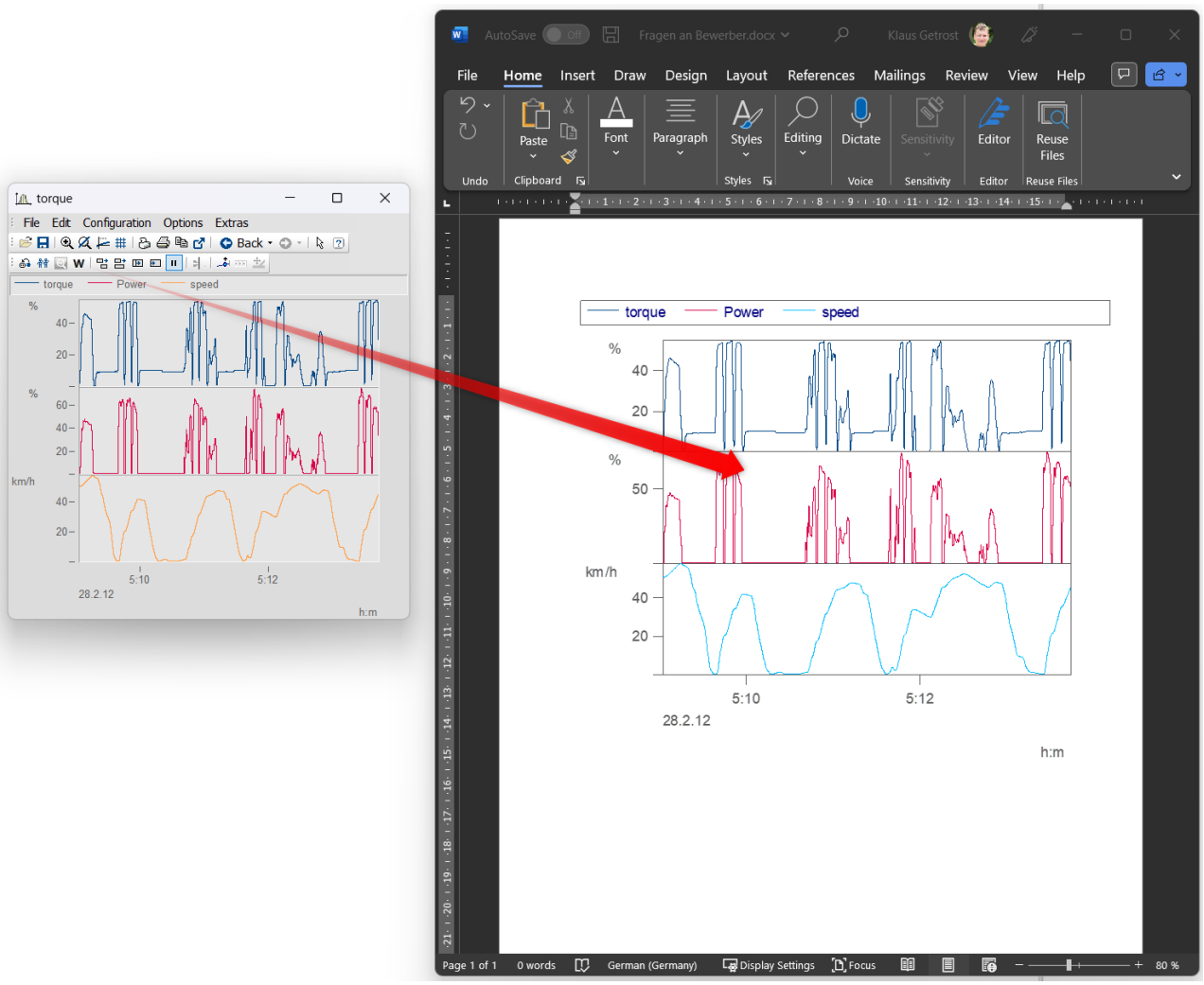
The **transfer** 1469 is done by **Drag&Drop** via the menu of the curve window or its toolbar.

Note

- With this action, the **data** is also transferred. This affects the file size of the Microsoft document, whereby the data is saved compressed. If the data exceeds a certain size, the transfer is prevented! This depends on the PC and the installed Microsoft version.
- **On the target system**, the curve window in the Microsoft document can only be edited 1470 if at least an **imc FAMOS reader** is installed. Otherwise, the curve window is displayed as an image and cannot be edited further.

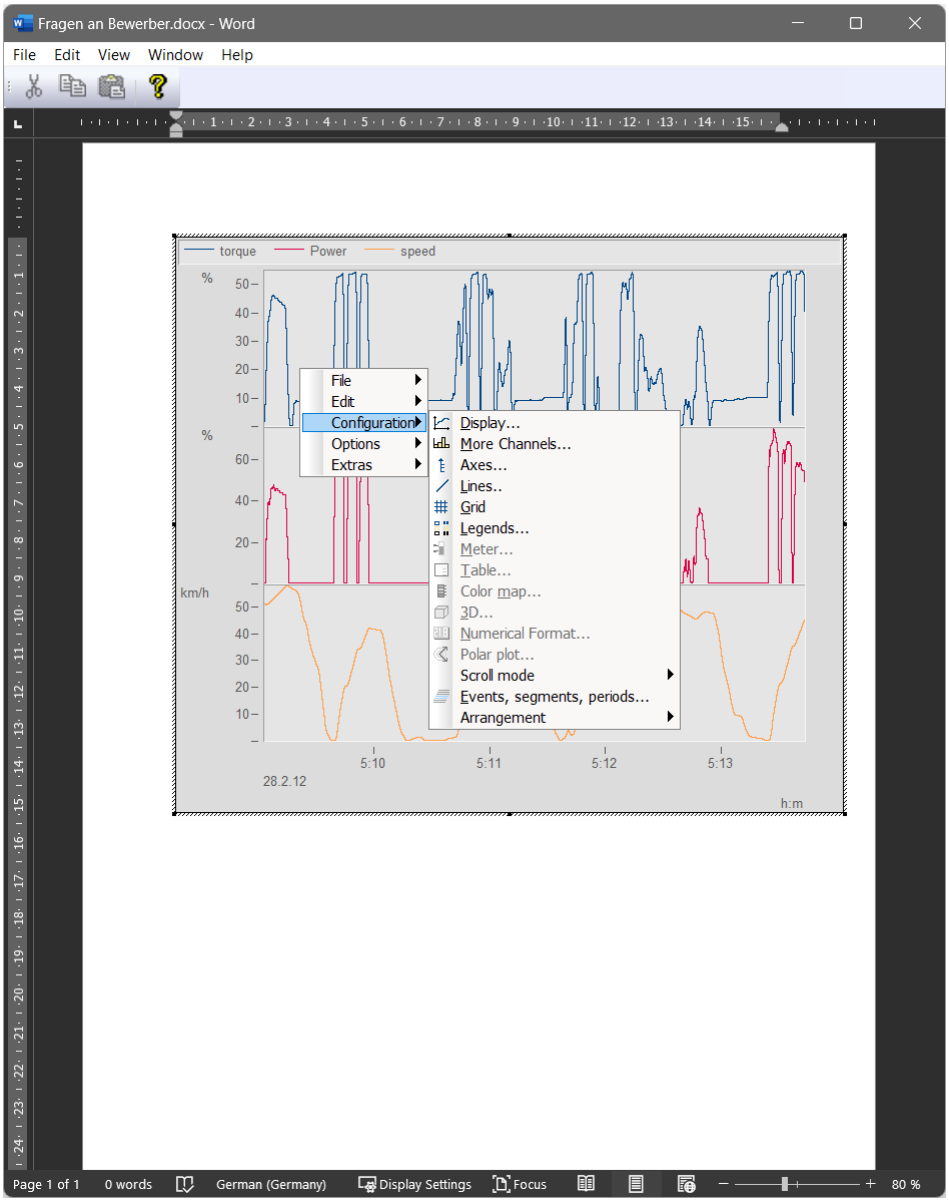
Transfer

The curve window is transferred to the Microsoft document by dragging and dropping from the *Edit\Drag to MS WORD/PPT*, or via the communication toolbar with the button W :



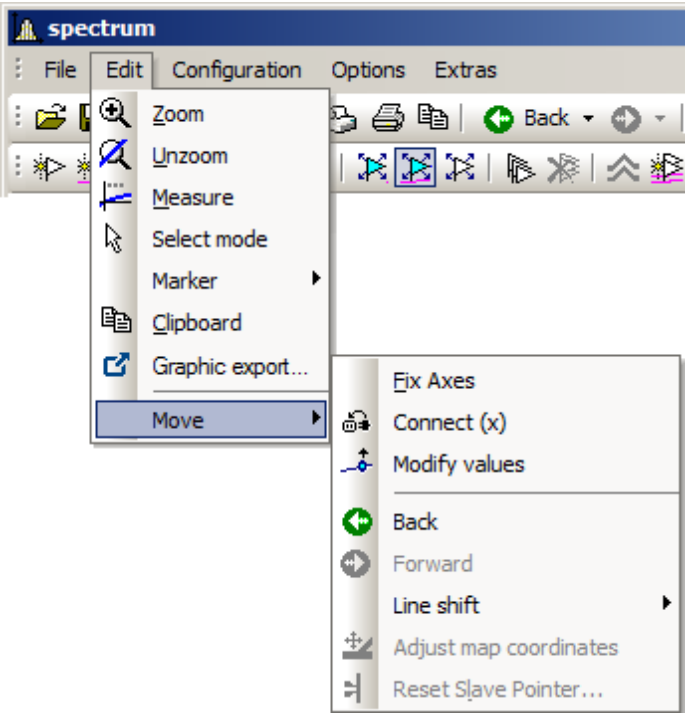
Edit

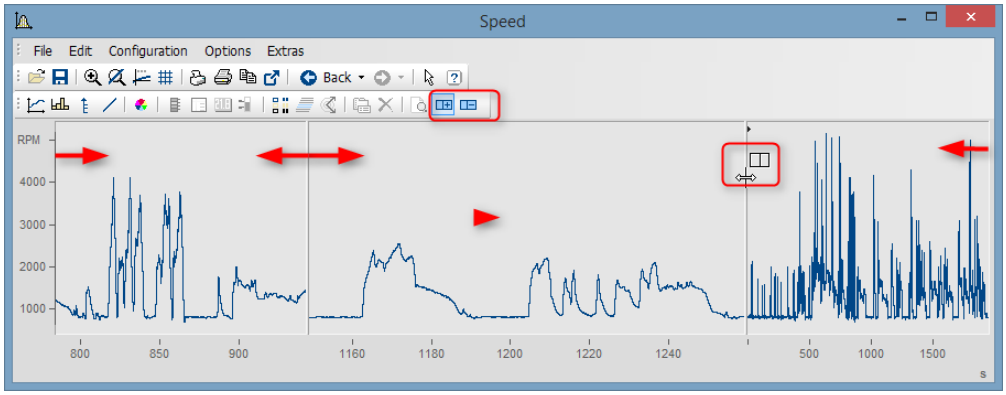
Double-click to set the curve window into edit mode. This is left again by clicking in the area outside the curve window. For editing, at least the imc FAMOS reader must be installed.



13.6.6.7 Move

The menu item *Move* contains various curve window functions for working in the curve window:

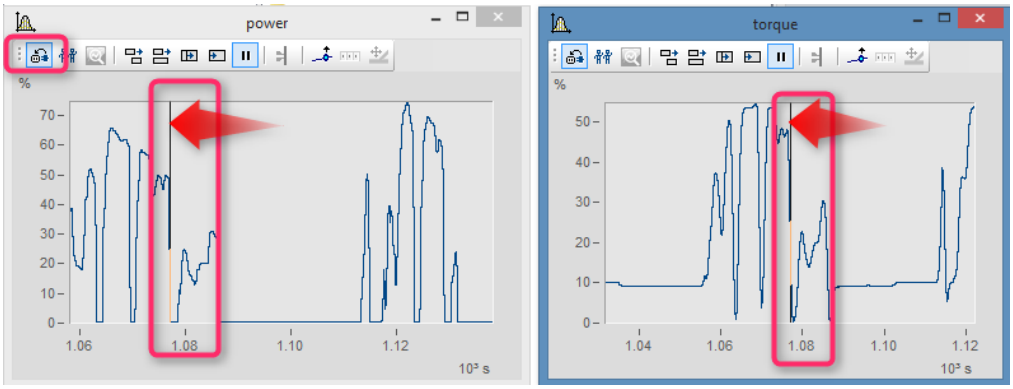


Menu item	Description
Fix Axes	This lets you fix the current axes so that the curve window does not automatically adapt to any newly added waveform, but rather keeps the previously existing axis scaling.
Connect	For connecting curve windows. For more on this topic, see the section Connect 1473.
Modify values	This function lets individual measured points from a data set be moved in the y-direction, thus changing their values. These changes are applied directly in the data set and stored. Caution: These changes can no longer be reversed by means of the *Back*-function. There will also not be any prompt for confirmation of the changes to the data set.
Back and Forwards	Step-by-step reversal of the most recent changes, or conversely repetition of the most recently reversed changes. With these two functions, changes in the curve window view can be reversed, or reversed changes can be restored. In the toolbar, the *Back*-function is additionally provided with a dropdown history, so that every change to the curve window view can be reversed separately. The exception is the function *Modify values*, since it directly changes the data set and not the curve window view.
Line-Shift	For moving selected lines in horizontal and vertical direction. More on this topic is presented in the section Line-Shift 1476.
Reset Slave Pointer...	For resetting the slave pointer. For more on this topic, see the section Bar meter 1294.
Split-Mode	*Split-Mode* subdivides the curve window's X-axis. This makes it possible to observe the same data at different points in time and at different zoom levels. With *Split-Mode+*, the curve window can be split an arbitrary amount of times. With *Split-Mode-*, the subdivision is canceled.  *Split-Mode*

13.6.6.7.1 Connect (Link)

Use this button to create a link between a curve window and either another curve window or a imc FAMOS table window. The connection is a x- (or scroll-) connection. Two windows connected in this manner have the same x-value at a defined position in their display ranges. Scrolling in one curve window prompts synchronous scrolling in the connected window. Several windows can be connected to form a linkage chain. No matter in which window the displayed x-range is changed, all connected table or curve windows are automatically updated to reflect the change.

To create a connection, click on the designated button and hold down the mouse button. The mouse pointer changes to a "blocked" symbol. If the mouse pointer is now moved over a curve or table window, it changes its shape back to a cursor. As soon as the mouse button is released, the windows are connected. The button appears to remain pressed and thus indicates the connection.



The x-axes of both curve windows are linked.

A vertical (curve window) or horizontal (table window) reference line indicating the x-position of the linkage is shown in the window. The connected windows each have the same scaling value in the x-direction at this line. The behavior desired can be specified using the *Options* dialog opened via Options\Presettings*Settings > What is influenced in the Link and This window follows* 1496.

This reference line first appears in curve windows in the middle of the x-axis, but can be moved to any position desired using the mouse. Move the mouse to this line; when the mouse pointer then changes its shape, hold down the mouse button and the line shifts as you move the mouse.

All connected windows should have the same x-axis mode (relative x-axis or display with absolute time).

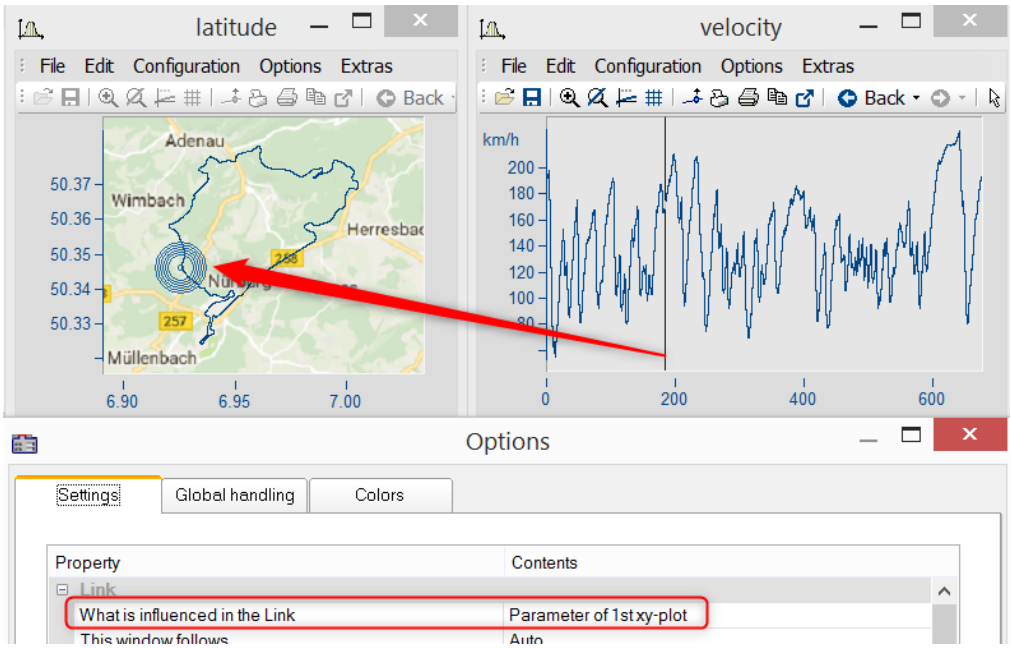
To undo an x-linkage, simply click on the button again.

Connect with XY

When linking XY and normal data sets, it may be necessary to specify which component of the XY data is to be linked.

This is configured under "*Options*"/"*Presettings*"/"*Link*"->"*What is influenced in the link*". In the following example the GPS channels Longitude and Latitude are used to display the speed with the position on a map.

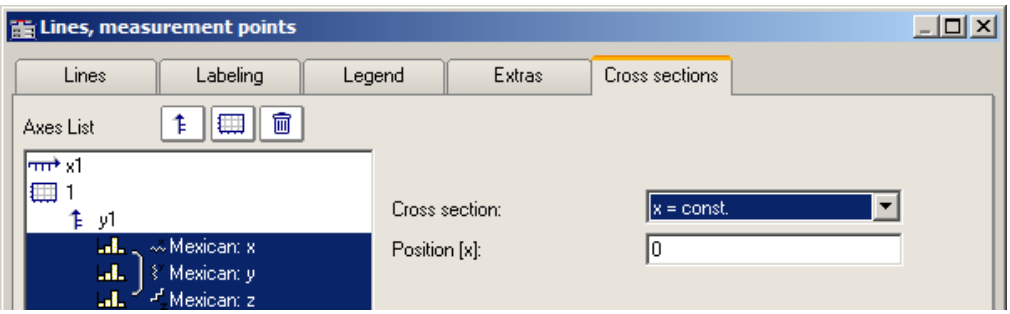
For this you have to set "*What is influenced in the link" -> "Parameter of 1st xy-plot* 1497".



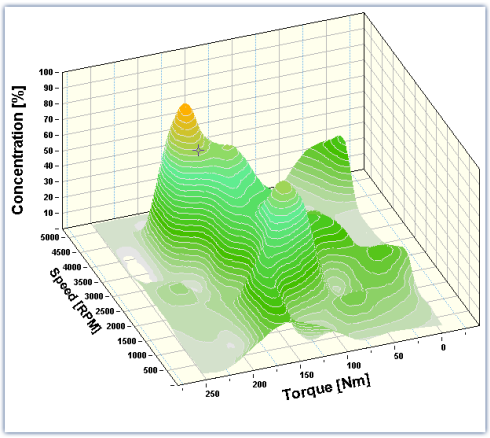
Connect with 3D

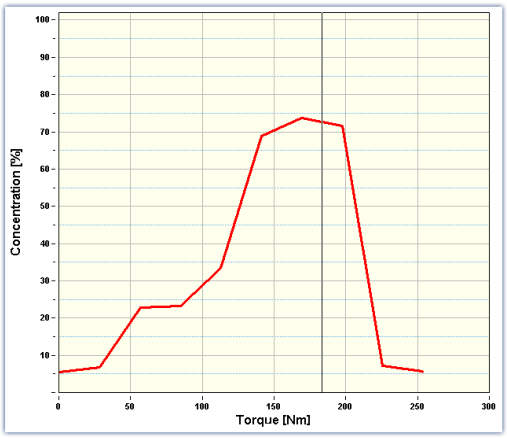
The ability to link curve windows together is also provided for 3D-displays. In this case, a 3D data set can be simultaneously displayed in 3D-display and as a cross-section with x or z as the constant, and both curve windows can be linked together.

To do this, load the 3D data set in a curve window with standard display style, parallel to the curve window with the 3D-display, and then open the *Lines*-dialog either from the coordinate system's context menu or as the menu item *Configuration / Lines*. On the page *Cross sections*, you can select from the dropdown-list whether to keep the x- or the z-coordinate constant. Then confirm by clicking *OK*.



In one of the two curve windows, activate the *Connect*-function, so that the cursor changes shape. Then move the changed cursor into the other curve window's coordinate system and click on the left mouse button. In the curve window with the 3D-display, a crosshairs appears at the upper left edge. Move the cursor to the crosshairs, click the mouse on it and move the mouse while holding down its button. In the curve window with the standard display and the cross section, there is a vertical line. The position of the vertical line changes in response to the position in the 3D-display.

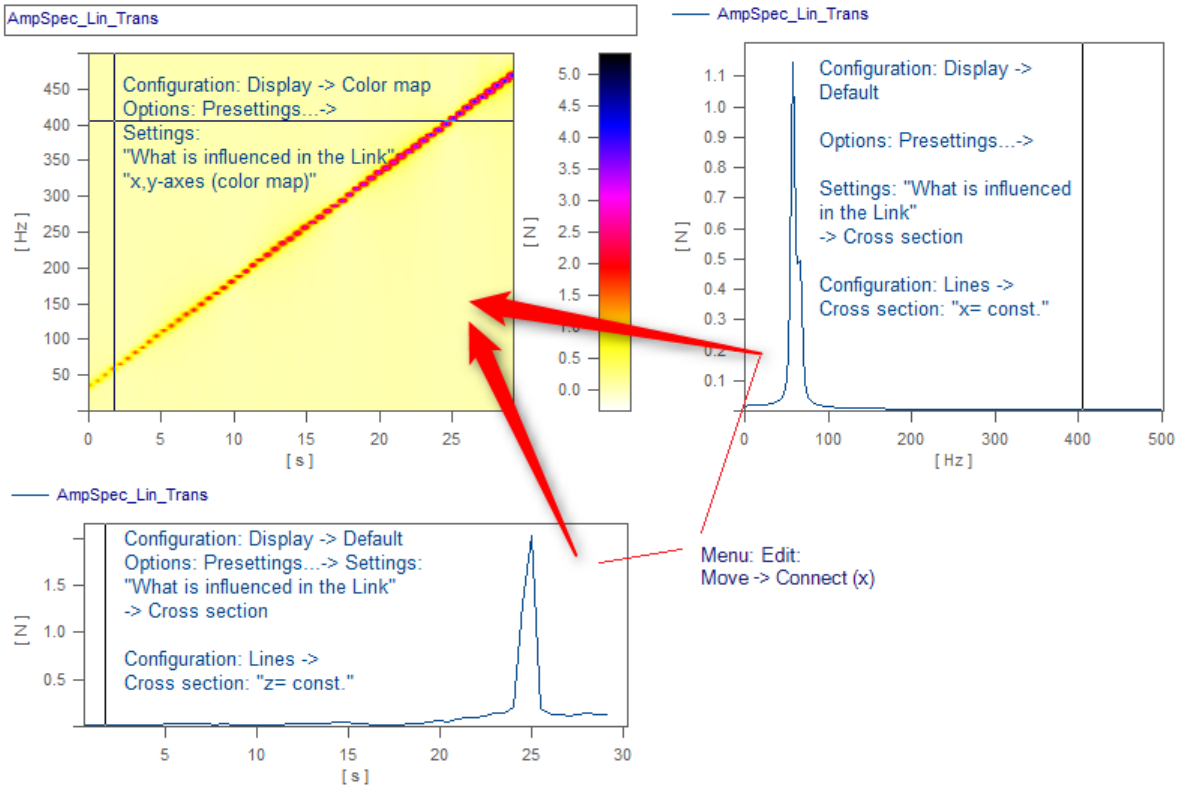




Example Example with two cross sections

Color maps can also be linked in the y-direction. For this purpose, in the Options for the curve window 1496, under *Settings*, it is necessary to set the *Link*-property "*What is influenced in the Link*" accordingly:

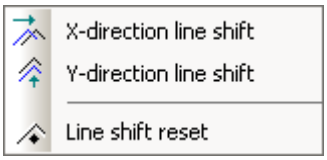
- On the color map: "*x-, y-axis (color map)*"
- On the right and bottom curve window*: "Cross section"*
- All curve windows show the same amplitude spectrum from example project "FA70 Spectral analysis"



The linkage is made by means of Edit> Move> Connect (x)" by using Drag&Drop in succession from the lower right window into the color map.

13.6.6.7.2 Line Shift

Using this function, any lines selected in the Select-mode, or all lines at once can be moved in the either the x- or y-direction.

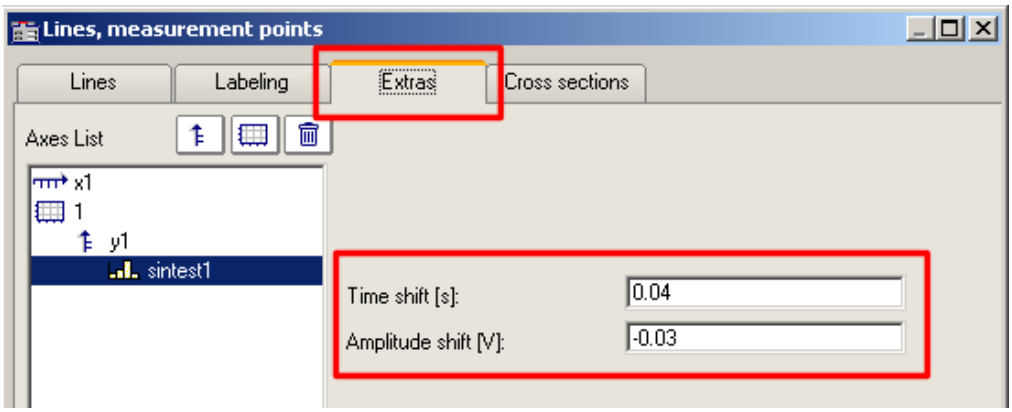


To do this, select one of the two functions either using the menu *Edit/Move/Line-Shift* or from the *Line-Shift toolbar*. ´

With the cursor in the pertinent coordinate system, and its shape changed to a double arrow, the lines can be moved either horizontally or vertically, depending on the function, by moving the mouse while holding down its button. With this function, only the view changes, i.e. the data set remains unchanged and any changes made with *Line-Shift* can be changed back either using the *Back* function or using *Line-Shift Reset*. With *Line-Shift Reset*, all changes are reversed at once, while *Back* reverses the actions one step at a time.

Line-Shift of a selected line in x-direction

The parameters for Line-Shift can also be changed directly in the Properties dialog for the lines, under *Extras*. There, too, the changes can apply either to individual lines or to all of them. With linear scaling display, the parameters are simply added. By contrast, with logarithmic scaling display, the parameter's value is interpreted as a factor, where the value 1.0 stands for no shift, a value of 10.0 stands shifting by tenfold, and a value of 0.1 shifts the lines downwards one-tenth.



Changing the Line-Shift parameters

Reference

The Time-Shift settings can be saved with the ccv file, if the curve window option"*Time-Shift in der CCV* 1498" has been set to "*yes*".

13.6.7 Ribbon

13.6.7.1 "File" menu

Menu item		Description
	Load 1479	Loads a curve window configuration from a file.
	Save as 1481	The curve window configuration (e.g. the axis scaling) is saved in a file.
	Transfer to imc FAMOS! 1482	The waveform in the curve window is transferred to imc FAMOS.
	Report Generator 1482	Calls the Report Generator. The report's layout can have your own personal, creative design.
	Print 1482	Prints out the curve window.
	Printer setup 1484	Sets up the printer used for printing out the curve window's contents.
	Overview Window 1485	Displays an overview window.
	Twin Window 1486	Generates a new curve window with the same appearance.

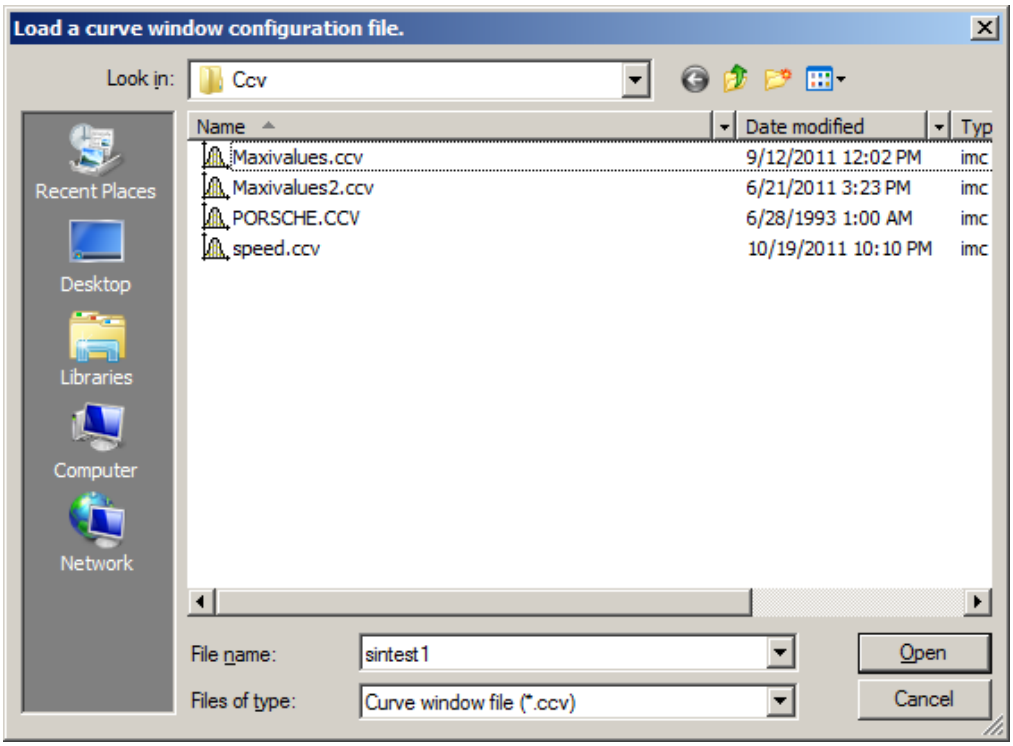
13.6.7.1.1 Load configuration

Function

You can a curve window configuration directly from a file. A curve window configuration includes various attributes of the window such as size, axes scaling and names of additional curves in the window.

Mouse Operation

- Select *File/ Load*...
- A dialog appears in which the file can be specified whose configuration should be loaded.

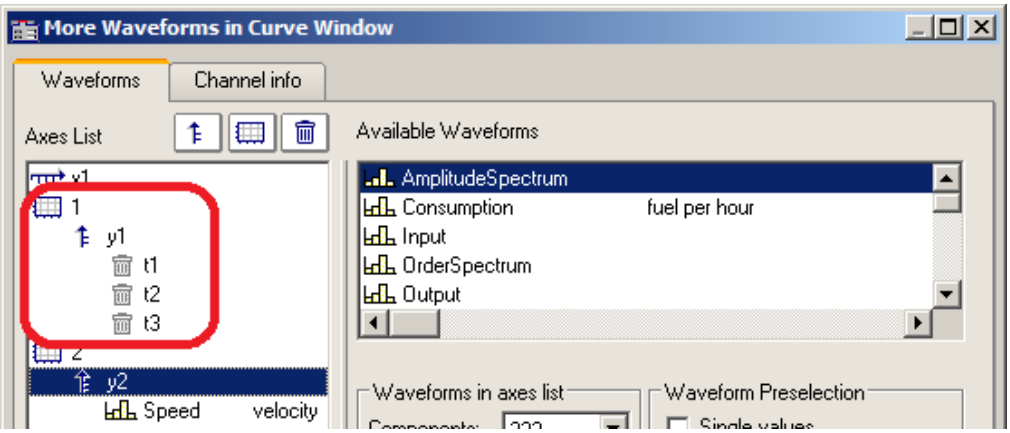


- Select file format "CCV".
- Select the desired file and end the dialog by clicking on *Open*.

Remarks

- The standard file name extension is "CCV".
- The directory used for loading configurations is read from the Windows System Registry when the program is started. In imc FAMOS, the folder can be handled via the menu item Extra / Options... .
- Once a file has been saved or loaded successfully, the directory involved is used for any subsequent loading or saving operation.
- The curves displayed in the window and their corresponding data sets are not contained in the configuration.

- When a curve configuration is loaded, an attempt is made to find all of the waveforms which were saved to the configuration. Any of these which are not found are marked as invalid:



- The names provide an indication of what waveforms are expected for the display. (When loading configurations from imc FAMOS 2.0, the system behavior of that time is reproduced and the waveform is deleted.) Configurations having multiple curves should be used only for stock tasks, where the names of the waveforms stay the same. For making changes, see Chapter 4, kit function *CvReplaceChannel()*.
- The name of the "basis waveform" (i.e. the first data set to be opened in the window) of a configuration is also saved.
- The position of the window does not change when a configuration is loaded manually. In automatic loading using a program or a sequence, the window is moved to the position saved in the configuration file. This does not apply to curve windows in dialogs.

Note

By means of Copy & Paste 1515, you can directly transfer a configuration from one curve window to another.

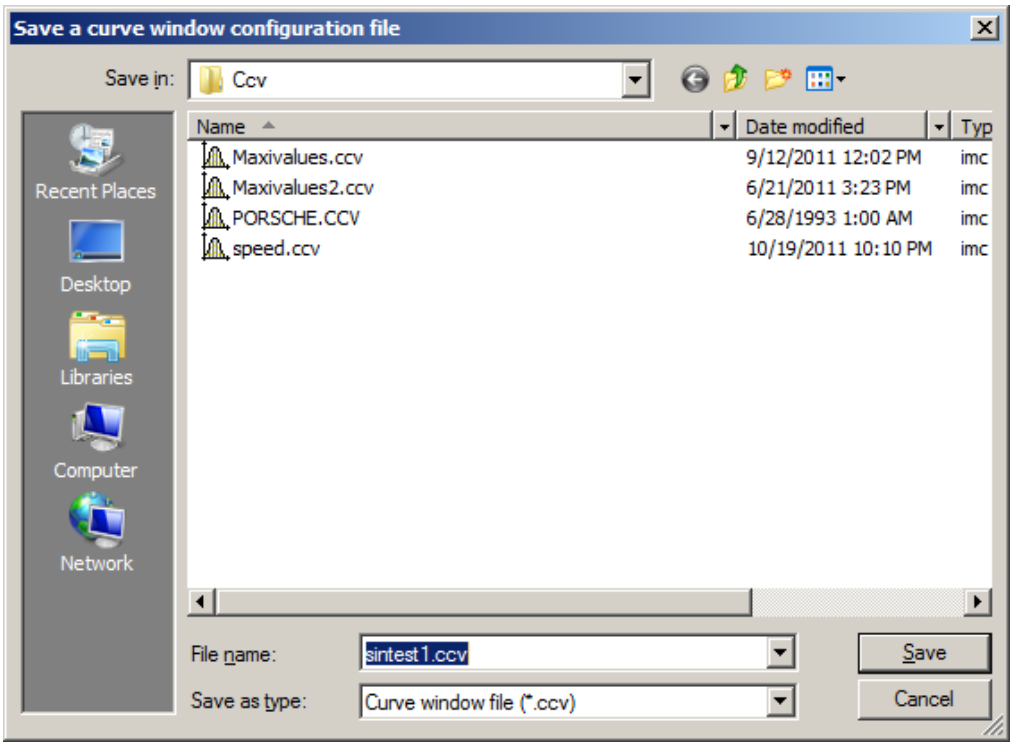
13.6.7.1.2 Save configuration

Function

The curve window configuration can be saved in a file. This configuration includes curve window attributes such as window size, axes scaling and the names of additional curves in the window.

Mouse Operation

- Select menu item *File / Save as...* in the curve window menu.
- A dialog appears for selecting the file in which the configuration should be saved.



- Enter the name of the file in the input box.
- Select the file format "CCV".
- End the dialog by clicking on *Save*.

Remarks

- If the file into which you wish to save the configuration already exists, the program asks if it should be overwritten.
- When a configuration is saved with automatic axes scaling set, different values may be written to the axes when this configuration is loaded for a new waveform. An appropriate scaling is defined automatically for the new waveform.
- It is useful for some applications to save configurations with automatic axes scaling. This is recommended especially when the expected signals fluctuate strongly in their range of values.
- The standard file name extension is "CCV".
- The directory path for projects is the project folder, or the default CCV folder from imc FAMOS Options.
- Once a file has been saved or loaded successfully, the directory involved is used for any subsequent loading or saving operation.
- The waveforms displayed in the window and their corresponding data sets are not contained in the configuration.

- The name of the "basis curve" of a configuration is also specified.

Note

By means of Copy & Paste 1515 you can directly transfer a configuration from one curve window to another.

13.6.7.1.3 Transfer to FAMOS!

"*Transfer to imc FAMOS*" creates a copy in imc FAMOS of the data sets displayed in this curve window. After the transfer, the data sets appear in the imc FAMOS Variables list, where they are available for working with. The transfer is normally only helpful if the curve window belongs to another imc program than imc FAMOS, such as imc STUDIO.

The settings made in the dialog "*Options*" > "*Transfer options* 1502" for naming variables in imc FAMOS are also valid for this operation. A command can be specified to be executed after the variables have been transferred to imc FAMOS, for example, a command to call a sequence for analyzing the transferred waveform.

Note Variables will be overwritten

Existing variables in the Variable list will be overwritten without notification.

Reference

- Transfer of waveforms or sections of waveforms can be made from a measurement value window. Refer to the section "*Context Menu in Measure Window* 1431" for more information.
- Waveforms can also be transferred to imc FAMOS using Drag&drop 1515.

13.6.7.1.4 Report generator

This lets you open the Report Generator, which can be used to assemble your curve windows in a report. For more information, please see the Report Generator documentation.

13.6.7.1.5 Print

Function

Hardcopy of a curve window can be created on any printer or plotter supported by Windows. The hardcopy is always made to an entire page. The graphics are centered on this page.

The graphics are sent to the printer which was specified in the dialog belonging to the curve window's menu item *File/ Printer Setup...* Various printing preferences can be specified in the *Opt./ Clipboard settings....* The size, font, width, etc. correspond to these settings.

Mouse Operation

To print the curve window, choose *File/ Print.*

In consequence, a small info-dialog appears which indicates that the printout is in progress. The process can be canceled by clicking on the *Cancel* button.

Normally, you have to allow for a delay before a print job is actually stopped.

Remarks

- Use the MS-Windows Control Panel to make settings for the printer.
- Additional parameters can also be specified for the printer here, e.g. printer resolution, paper size and format (portrait/landscape), printer memory allotment, etc. To do this, use the curve window's menu item *File / Printer setup*...
- The quality increases with the resolution. Graphics only look good, of course, when they are printed with high resolution. Note however that the higher the resolution, the longer it takes to print. In fact, the time to compute a printer job increases with the square of the resolution.
- Select a low printing resolution if you value speed of printout more than quality.
- The current time is used as the "Created/ Modified" time if the waveform does not have one already.
- To combine text and graphics in a desktop-publishing program, send output to the Windows Clipboard instead of using the Print option. See "Clipboard 1467".
- If the printer is either not correctly defined in the MS-Windows system or switched off, or if it has no paper, an error message will appear.
- If you wish to create a more detailed layout, use the imc Report Generator. See chapter "Report Generator 1482" in this document for more information on this powerful program.
- The time format (date and time) can be changed in the Control Panel under *Regional Settings*.

Reference

See "Clipboard Settings 1489", "Printer Setup 1484"

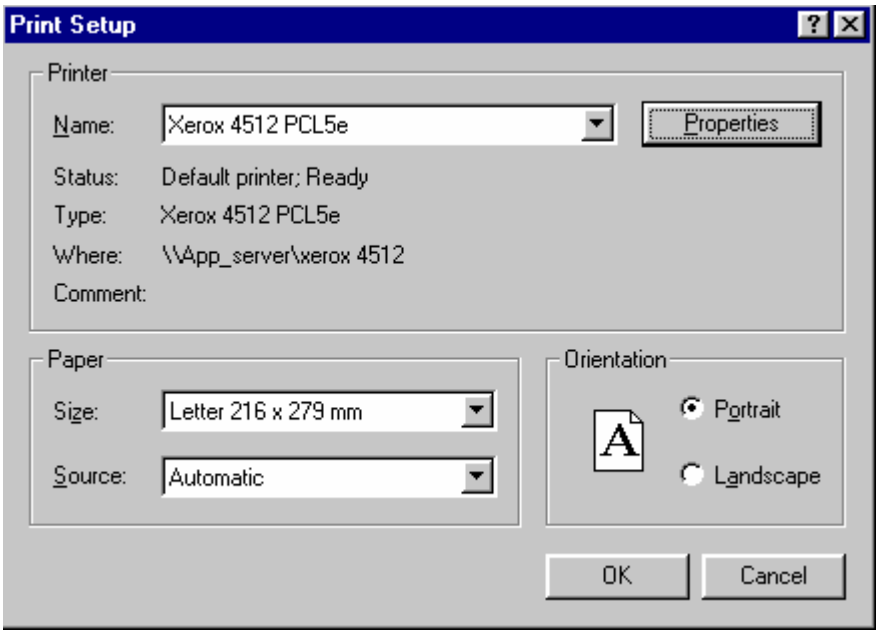
13.6.7.1.6 Printer setup

Function

This is where the printer is configured, which is used to produce the hard-copy output from the curve window.

Operation

Select menu item *File/ Printer setup*... in any curve window. The standard Windows dialog for selecting and configuring a printer.



This dialog's appearance varies according to what Windows version or printers are involved.

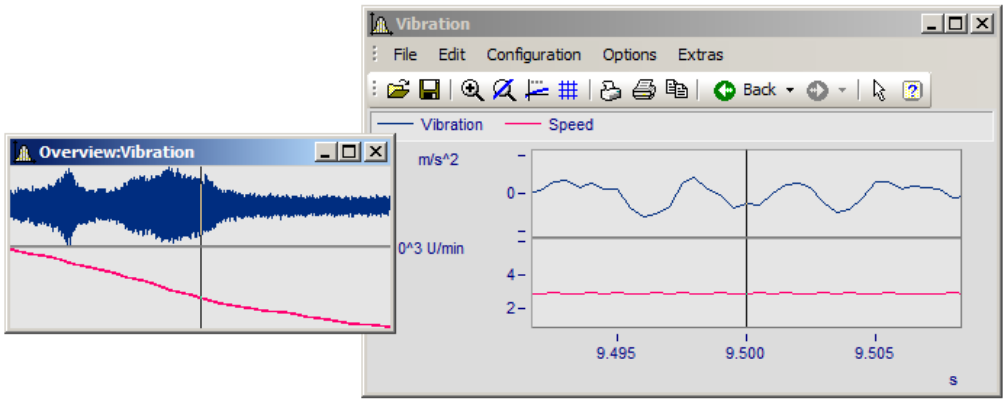
The settings apply to all of the computer's imc curve window's and remains operative upon the program's next start.

13.6.7.1.7 Overview window

Function

Whereas a curve window can display any portion of a waveform, an overview window, which is shown in conjunction with a curve window, will only display the entirety of any waveforms which are represented at all in the curve window. The curve- and overview window are linked with each other in the x-direction, in that the reference line of each window marks the same x-coordinate. The overview window is especially useful for:

- Keeping track of a position in convoluted or long curves (especially during zooming), or distinguishing between different axes lengths when several waveforms are displayed together.
- correctly associating a displayed section with its entire curve when scrolling in the curve window in the x-direction.



Mouse Operation

- To open an overview window, choose *File/ Overview Window*. A check mark appears beside the selected option.
- Zoom in on one region of the curve window and navigate through the signal using the reference line in the overview window.
- Click again on this option or select *Close* in the system menu of the *overview window* to close it.

Overview windows are independent windows which behave almost like curve windows. However, each overview window is assigned to a curve window and cannot exist without the curve window; the overview window also contains the same menu as the curve window. All settings for configuration and display of waveforms in curve windows can be made for the overview window. It is thus possible to display a curve together with variously strongly zoomed views by opening an additional overview window.

Remarks

- The overview window is only recommended if the range displayed in the curve window can be displayed fully in the overview window.
- When the curve window is reduced to an icon, the overview window is not visible.
- The display style used in the overview window can be changed and zoomed as desired.
- If an overview window and a measurement value window are displayed for the same curve window, the measurement value window should be closed when the zoom range is moved in the overview window. This reduces flickering while the curve window is being updated.
- The title of the overview window is composed of the string "Overview:" followed by the name of the corresponding curve window. This allows easy identification of the windows.

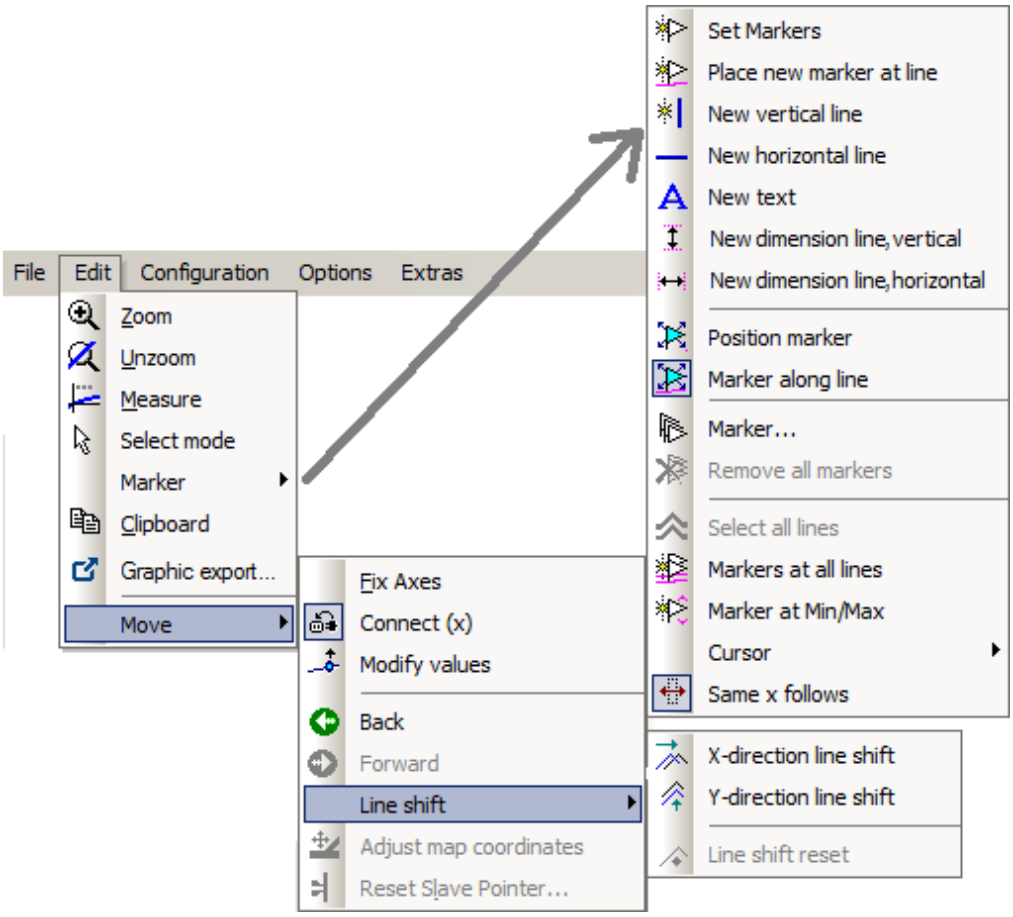
13.6.7.1.8 Twin window

Using this item you can create an identical copy of the curve window. Subsequently, both windows can be configured independently of each other.

Possible applications:

- During an online measurement, the measured data can be displayed both as a curve plot and as numerical values (Last Value as Number 1283).
- Display of a Waterfall graph in 3D, or a color map.
- Simultaneous display of the exact same data both in an overview and zoomed.

13.6.7.2 "Edit" menu

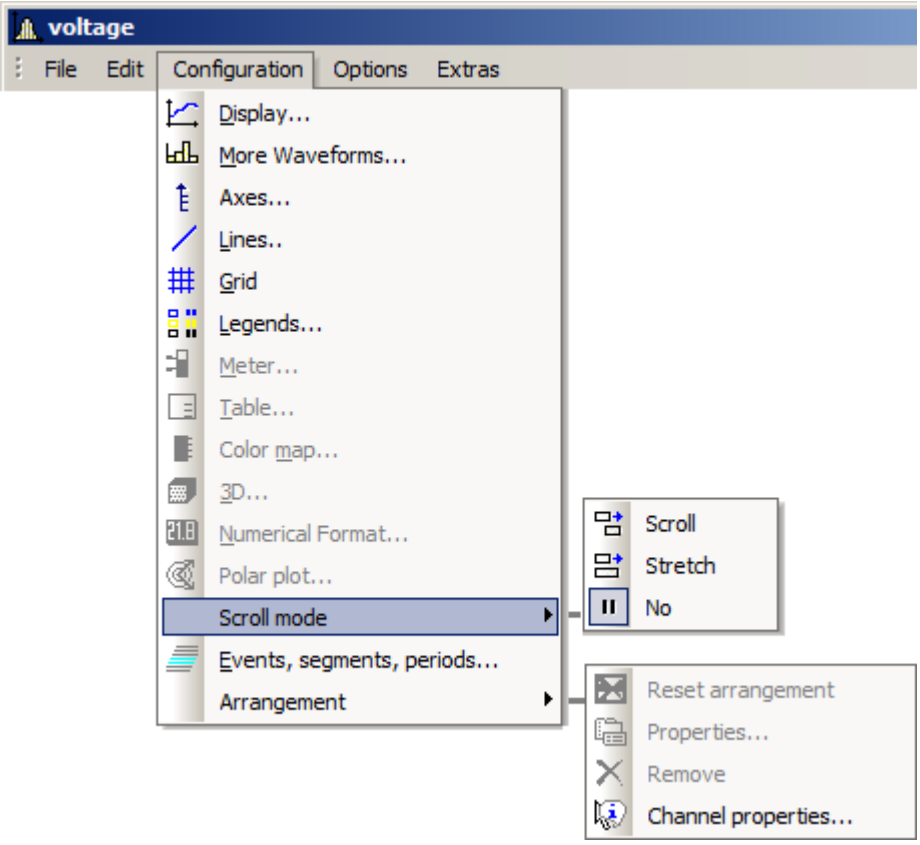


Menu item	Description
Zoom 1425	Enlarges a section of the curve window.
Unzoom 1426	Displays the entire curve.
Measure 1427	A measurement value window and measurement cursors for measuring curves are displayed.
Select-mode 1441	In this mode, it is possible to select and edit legends, coordinate systems, axes, curves and markers.
Marker 1443	Proceeds to the list of marker functions.
Clipboard 1467	Copies the graphics in the curve window into the Windows clipboard.
Graphic-export... 1467	The graphic of the curve window is saved as image or as a PDF file.
To MS WORD 1469	The curve window is embedded as an OLE object in Microsoft WORD.
Move 1472	Proceeds to a submenu listing various curve viewing functions.

Reference

Another possibility to navigate through the curve window is provided by the 'Axes Navigation Bar' 1516.

13.6.7.3 "Configuration" menu



Menu item	Description
Display 1267	The structure of the curve window and other attributes, such as the "Freeze" mode and time/date display can be selected here.
More Channels 1339	Selects additional waveforms to be displayed in the current curve window.
Axes 1350	Sets x- and y-axis parameters.
Lines 1369	Sets line (measurement cure) parameters.
Grid 1404	A grid can be added to the coordinate system.
Legends 1382	Sets the display of legends in the curve window.
Meter 1294	Sets properties of bar meter display.
Table 1288	Sets properties of tables display.
Color map 1303	Sets properties of color map display.
3D 1315	Sets properties of 3D display.
Numerical format 1283	Sets properties of the display mode "Last value as number".
Polar plot 1315	Sets properties of polar plot.
Scroll mode 1414	In the curve window, a particular section of the data set can either "scroll past" or continuously "grow" from the beginning.
Events, segments, periods 1415	Selects the display of individual events (in imc STUDIO: trigger events), individual segments (e.g. spectra or rows in a matrix) or even periods (period comparison).

Menu item	Description
Arrangement 1422	Restores the curve window arrangement, access to properties and deletion of selected objects.

13.6.7.4 "Options" menu

Menu item	Description
Clipboard settings 1489	Settings for graphics created for output or export with the clipboard are specified here.
Colors 1493	Changes the colors in the curve window.
Presettings 1496	Presettings such as the standard configuration directory and drives for storage of temporary files are made here.
Printer preview 1503	This function makes it possible to switch between the Print Layout view and Normal view. It is used exclusively in curve windows which are embedded in the Data-Browser's Report view.

13.6.7.4.1 Clipboard settings

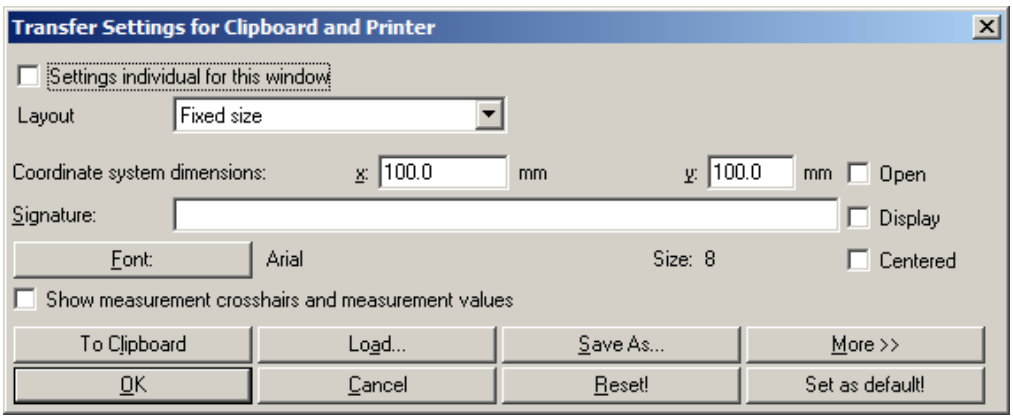
Function

The graphics in the curve window can be documented in professional quality. Graphics can be sent to the clipboard as vector graphics, for further processing by any text processing, graphics- or desktop publishing programs. The graphics can also be sent to the Report Generator for output in a report designed by the user.

The user can specify how graphics are generated by selecting the font and line width in the clipboard settings.

Operation

Call menu option *Options/ Clipboard settings*... in the curve window menu. The following dialog appears:



The settings are applied when the item *Clipboard* 1487 in the *Edit* menu is called.

Settings	Description
Settings individual for this window	Here you can set whether the clipboard settings only apply to the current curve window or for all curve windows.

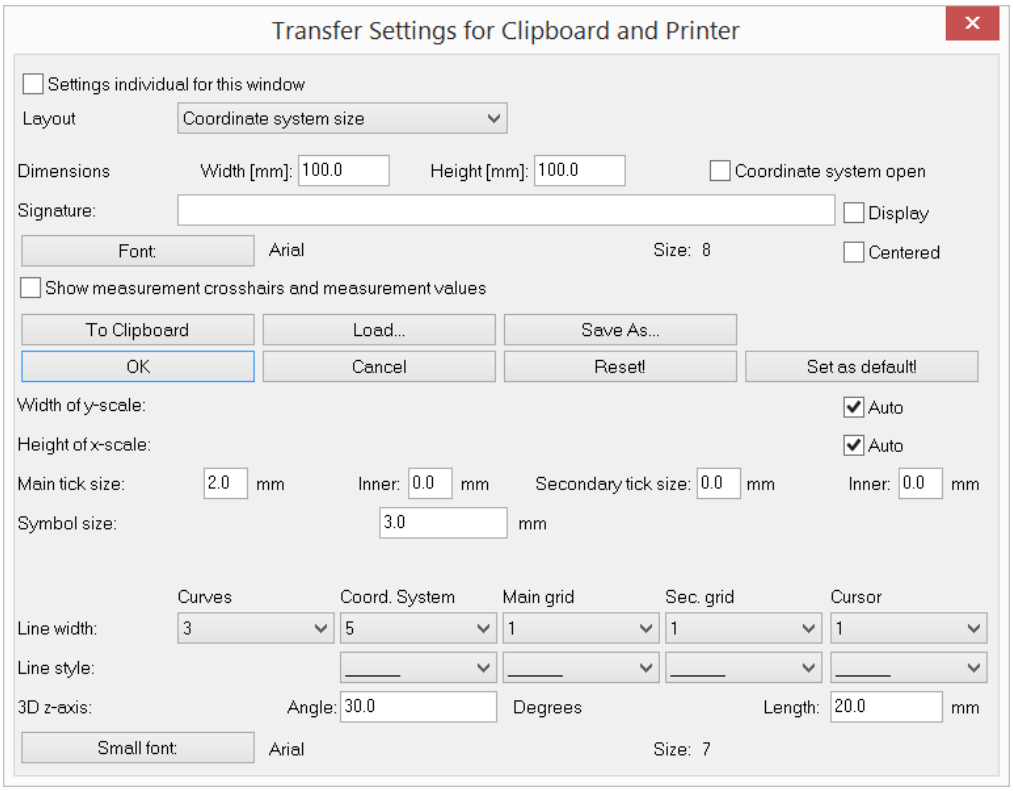
Settings	Description
Layout	***Proportions as on screen*** (the default setting!) saves the curve window to the clipboard with the proportions which accord with the settings currently applied on screen. The layout ***Coordinate system size*** creates a graph in which the **coordinate system** exactly applies the *x* and *y* values specified by the setting *Dimension*. Observe also the notes on the Table 1490. ***Total size*** creates a graphic in which the **outer edge of the curve window** is determined by the x and y values.
Coordinate system	The ***width*** and ***height*** of the coordinate system, stated in mm, are set in the text boxes appearing in the dialog line *dimensions*. If the lines are particularly thick, the size is calculated from the middle of one line to the middle of the next line. Axis labeling of the coordinate system is not included in these dimensions, they are drawn outside of the actual coordinate system. Dimensions should be at least several mm, but no larger than the page.
Open coordinate system	This option causes curves in a standard display (y-axes not stacked) to be drawn in an open coordinate system, with no upper and right-hand borders. The curve is thus less crowded in the display. When this option is not selected, all lines in the coordinate system are drawn.
Caption	The user can specify a text to appear below the x-axis. This text may contain up to 60 characters. It will be displayed, along with the date and time, only when the option *Show* is selected.
Centered	When this option is selected, all labels (as much as possible) are centered under the ticks. The effect is most noticeable for the labels at the edges of the axes, at the far right and left of the x-axis and at the top and bottom edges of the y-axis. When the type is not centered, these labels are often drawn so that they end at the edges of the coordinate system.
Show measurement cursors (crosshairs) and measurement values	When a measurement value window is present when graphics are generated, and this option is active, the measurement cursors are included in the coordinate system display. The measurement values of the cursors are displayed below the x-axis.
Font	Select here the font for the axes scaling. The MS-Windows standard dialog appears when the "*Font*" button is selected. The dialog shows all fonts available for the printer currently set for the curve window. Select a font, the size in points and any other features desired, e.g. bold type. A font size of 10 to 12 points is generally very legible. TRUETYPE fonts are preferred since they can be scaled more easily.
To clipboard	Click on this button to copy the contents of the curve window with the current settings to the clipboard.
Reset!	When the *Reset*! button is selected, the controls of the dialogs are set to default values. Controls accessed with the *More >>* button are also reset.
Save as	A dialog appears for saving the contents of the *Clipboard settings* in a file.
Load	Once the contents of the dialog have been saved in a file, this file can be loaded to initialize the dialog with the settings in the file.

Note

It is possible to achieve an exact scale of units/cm by means of the combination *Layout:* ***Coordinate system size*** and an appropriate count of *Ticks* 1353 along the X- and Y-axes. Note that due to particular presets in the target program, the size of the graphic may not be equal to 100%. For example, after insertion into a WORD document, it is necessary to specify 100% for the *Size and Position* (in the graphic's context menu).

More >>

The dialog box is enlarged when this button is selected. Additional, less frequently used settings for the graphics configuration are made here. The dialog appears as follows:



Settings	Description
Width of y-scale	The width of the y-scale is set automatically, if the checkbox appearing after *y-scale width* is marked. To specify a different width manually, deactivate the check box *auto*. A text box appears in which the width should be specified in mm. Make sure that the specified width is large enough for the selected font. The width is calculated from the middle of the line enclosing the coordinate system.
Height of x-scale	The height of the x-scale is set automatically, if the checkbox appearing after *x-scale height* is marked. To specify a different height manually, deactivate the check box *auto*. A text box appears in which the height can be specified in mm. Make sure that the specified height is large enough for the selected font. The height is calculated from the middle of the line enclosing the coordinate system.
Size of ticks	Specify here the length of the *main* and *secondary ticks*. The length is composed of sections: length within and outside the coordinate system. Specify the total length and the portion *inside* for the main ticks and the secondary ticks, respectively. The lengths are specified in mm with up to one digit after the decimal; a length of zero is also possible.
Symbol size	Lines can also be distinguished by accompanying symbols, see the menu items *Configuration/ Axes* 1350 and *Configuration/ Display* 1267. The size of the symbols is specified here. Enter the diameter of the symbols in mm, with up to one place after the decimal.

Settings	Description
Line width	The width can be specified for the following lines: • curves, coordinate system, main grid, secondary grid, measurement cursor Any specified line thickness 1370 for an individual waveform overwrite the settings saved in the Clipboard. Specify the line width with a setting between 1 and 100. A very fine line width can be achieved with the lowest settings. Line width depends on the resolution of the output device. Since the width is always entered in pixels (units of resolution), only a whole number of points of the output device can be specified. A line width of 1 may be a practically invisibly fine line on a high-resolution laser printer, but may appear like bold type on a simple matrix printer. Line widths are generally set to between 1 and 5.
Line style	Various line styles can be selected for the following: • coordinate system, main grid, secondary grid, measurement cursor The following line types can be selected: • broken, dense dots, coarse dots, dense dashes, coarse dashes, alternating dots and dashes The various line types help differentiated between main and secondary grids when both are drawn with thin lines, so as to avoid distracting from the displayed curves.
z-axis angle	The angle of the z-axis can be set to between 1 and 89 degrees for the waterfall diagram. The larger the angle, the steeper the z-axis climbs. An angle of 30 degrees is recommended.
Length of z-axis	The length of the z-axis can be specified in mm for the waterfall diagram, with up to one place after the decimal.
Small font	The font for the small print in the 1/3-octave/octave labeling of the x-axis can be specified. Click on the button *Small font* to prompt the MS-Windows standard dialog for selecting the font.

Remarks

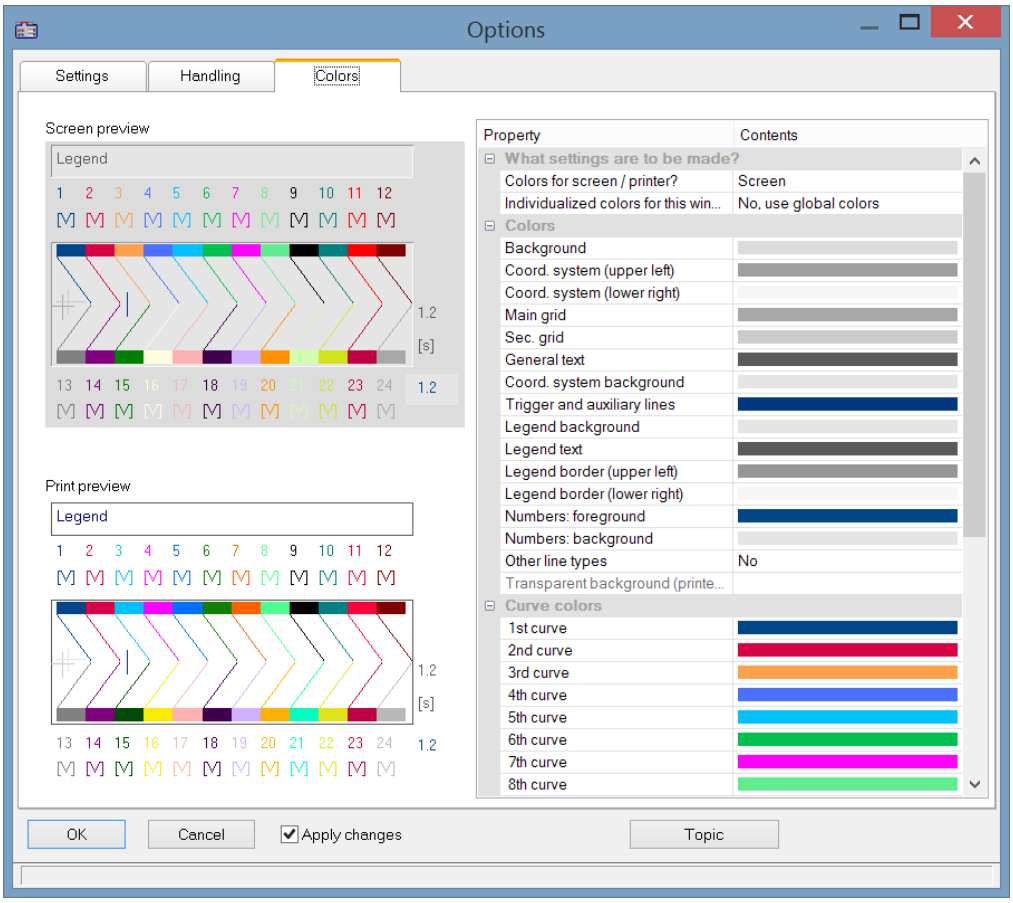
- The settings in this dialog apply to all curve windows.
- The fonts available for selection depend on the printer set for the curve window.
- When graphics are transferred to the clipboard, it is important that the printer set for the curve window is also the printer with which the graphics will later be printed. The meta file for the clipboard is created specially for the selected printer; if graphics are outputted on a different device, some fonts may not be available or scaling may change and then the graphics will appear in poor quality. Format settings such as landscape or portrait and the size of the page should also correspond. The metafiles are in some measure adapted to the output device.
- A metafile is created with the selected fonts and size for the curve window graphics when they are sent to the clipboard. If this metafile is later changed to a different size (e.g. because the graphics were shrunk by the text processing program after they were inserted in the clipboard), the font proportions may no longer be correct. The font may become too wide or too high. Use of truetype fonts may also have this result, since the Curve Manager only dictates the height of the font in points. Always create graphics in the size in which they will later be printed. If this is not possible, at least try to comply with the height-width proportions.

13.6.7.4.2 Colors

Function

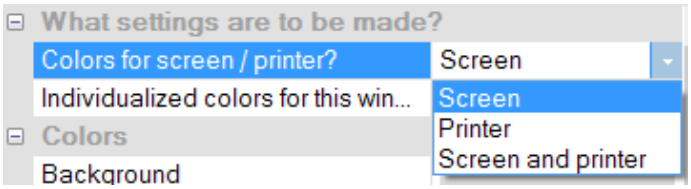
All colors used in the graphic interface of the curve window can be defined as desired in a special dialog. Single windows can use an individual color scheme.

The following dialog is used to define the colors:



Screen/Printer

A combobox is found at the top edge of the dialog, which contains the entries *Screen, Printer* and *Screen and printer*.



The colors can be changed from the default settings. The default colors for the printer are black/white. However, the screen colors can be applied for printouts, see below under ***Copy..*** 1494.

The printer settings additionally offer the option of leaving the background transparent. Besides saving ink, this has the advantage of keeping graphics objects located behind the curves visible.

Individualized colors for this window decides whether the colors are adopted as the default setting for all windows or not. This option can be set separately for the **screen** and **printer** selection. So it is possible to set colors for the screen individually but to use the global default settings for the printout.

Color graphic elements

A curve window contains the following graphic elements:

- background
- coordinate system and grid ...
- coordinate-system background
- general text
- legend...
- numbers...
- trigger line, auxiliary lines
- curves 1 to 12

Operation

Select menu option *Colors*... in the *Options* menu in the curve window.

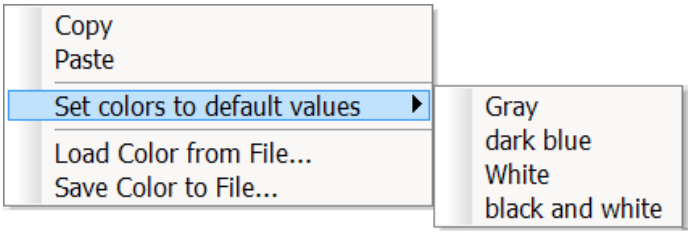
A dialog appears in which the colors can be defined. This dialog contains a list of the graphic elements in the curve window.

A display is found in the center of the dialog showing the colors currently defined for icons of the graphic elements. Colors are defined for one element at a time.

Select a graphical element from the list and set its color.

Context menu

Right-clicking the mouse in the dialog box calls the context menu:



Menu action	Description
Copy	The table including colors is copied to the clipboard.
Paste	The table values are pasted in the selected curve window in order to adjust the screen colors to the printer settings or for an individual window.
Set colors to default values	The colors are reset. There are serveral color schemes to choose from.
Load Color from File	The color settings for the monitor and for the printer are loaded from a file. This file must previously be created by using the menu option *Save Color to File*.
Save Color to File	Stores color settings in a file.

In the sequence, the colors can be loaded using the Curve Kit's function `CwGlobalGet("colors.printer.pattern")` or `CwGlobalGet("colors.screen.pattern")`.

Remarks

- None of the colors produced by mixing and blending may be selected for lines and texts. If other color intensities are selected, the program will use the closest true color. This also applies to the coordinate system background color.
- It is recommended to select always colors which produce a good contrast. Yellow curves on a white background would be a very inappropriate combination.
- Background colors should not contain any noticeable pattern, to guarantee good contrast to the curves and lettering.
- Do not select a dark background color for printing. Laser printers are not designed to print large dark areas. A white background would be appropriate.
- Colors set in this dialog are effective for all curve windows.
- The colors for printing are also used when the graphics are transferred from curve windows to the MS-Windows clipboard, and when curves are sent to the Report Generator.
- The selected colors settings remain intact even after the program has been ended.
- Colors cannot be selected individually if more than twelve curves are displayed in the curve window. The colors are then repeated cyclically.

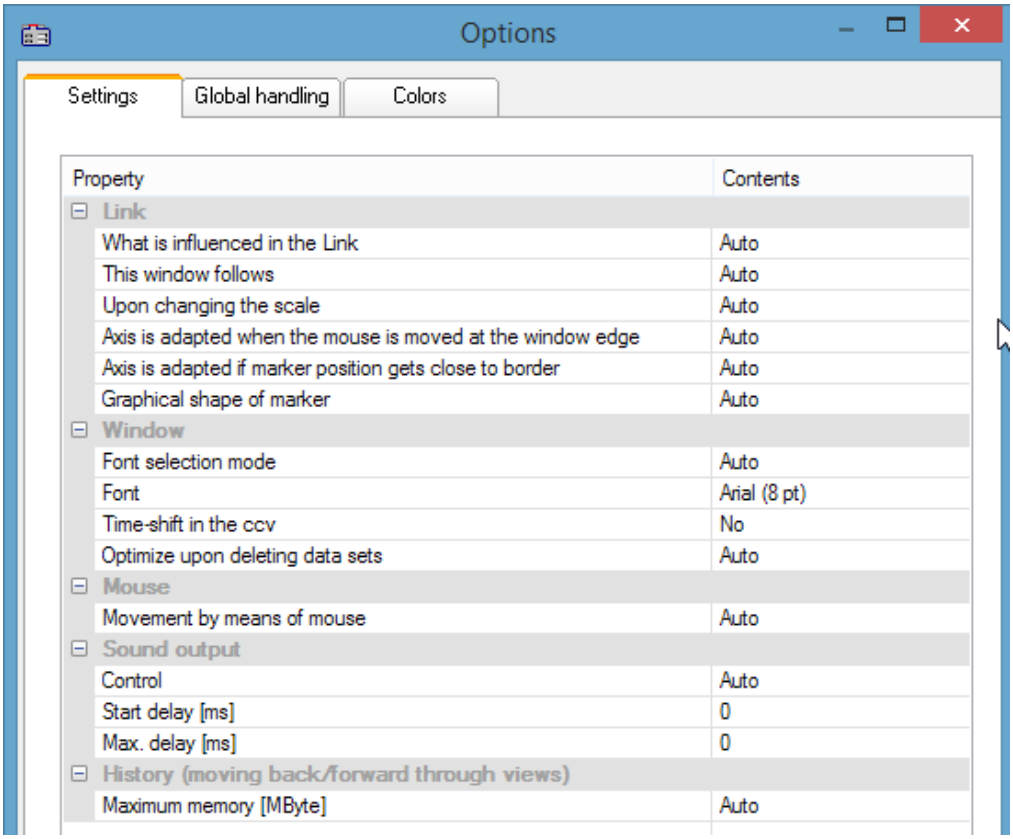
Reference See also

"*Curve window colors on the Panel page* 1518" - Which colors are used for the on-screen display and which for the printout.

13.6.7.4.3 Presettings

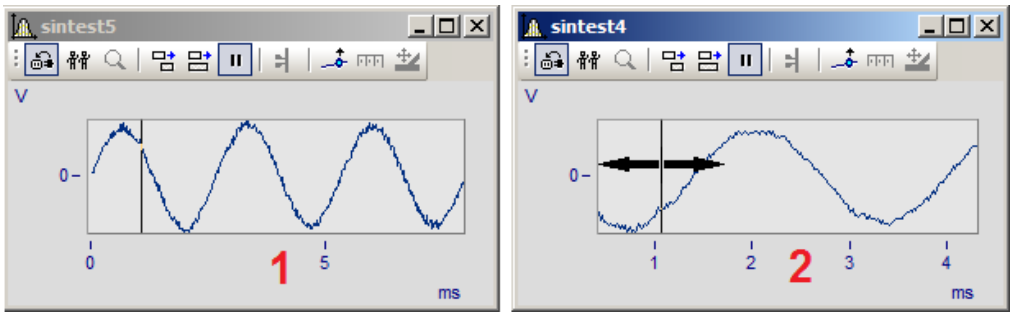
Here you find more controls for the curve window's default settings, such as the font, axes, linkages to the curve window, etc.. By clicking on the column Contents, you can select from the possible options; see the arrow below.

Settings



Presettings - Page: Settings

Link: The settings for the link affect how one curve window is linked to another one, for instance, one curve window's waveform in the time domain could be linked with second curve window showing GPS positions.	
What is influenced by link	**auto:** depending on the display type, one of the following options is used **x-axis:** usually default **Parameter of 1st XY-plot:** e.g. for the position on a *map* 1392 **x-, yAxis (color map):** both the x-, and the y-direction are linked 1473. **y-axis:** follows the y-direction in color maps **Cross section:** creates a cross-section referenced to the linked axis on the color map.
This window follows	**auto:** mostly Line follows **Axis follows:** The curve and the axis are moved, but the link pointer remains stationary. **Line follows:** The curve and axis remain stationary while the link pointer moves.
Upon changing the scale	**auto:** Stretching and compression of the x-axis; mostly: line remains at screen position **Line follows:** Line remains at screen position **Line remains at position on screen:** curve moves away behind it.
Axis is adapted when the mouse moved at the window edge	Affects curve windows where the line moves (in picture 2 below) **auto:** mostly "no", except with map 1392 **yes:** the axis of the curve window whose line is moved is compressed. **no:** no changes.
Axis is adapted if marker position gets close to border	Affects curve windows which move due to linkage (in picture 1 below) **yes:** Axis of linked curve window is compressed
Shape of position marker	**auto:** always line, except with map 1392 where a circle is used . **Line:** Vertical line



Link options

Window: In this section, set the **font**.	
Font selection mode	**Auto:** the font which is principally used whenever the curve window is opened. **Individual for this window:** Only affects the current curve window.
Font	All fonts used by Windows are available here.
Time-Shift in the ccv	**Yes:** Any shift made using the Time-Shift or Line-Shift function will be saved with a curve configuration file (ccv).
Optimize upon deleting data sets	With "*No: Lines and axes remain intact*", only a variable will only be removed from the curve window. The structure remains intact. This parameter can be set by the function `CwDisplaySet("opt.on.delete", 0)` as well. With "*Auto*", the system behaves the same way as before, which means the associated axes are removed. If the deleted data set is the only variable in the curve window, the curve window closes.

Mouse: Behavior of the curve window when dragging the mouse while holding its button	
Movement by means of mouse	**Auto:** Moves the location of the data in the curve window in the X and Y directions, similarly to moving a map. **Only in x-direction:** Like *Auto*, only in the x-direction. **No:** The display is not moved. Instead, the data can be moved into a different coordinate system or curve window by means of Drag&Drop. (Default behavior up to imc FAMOS Version 7.0)

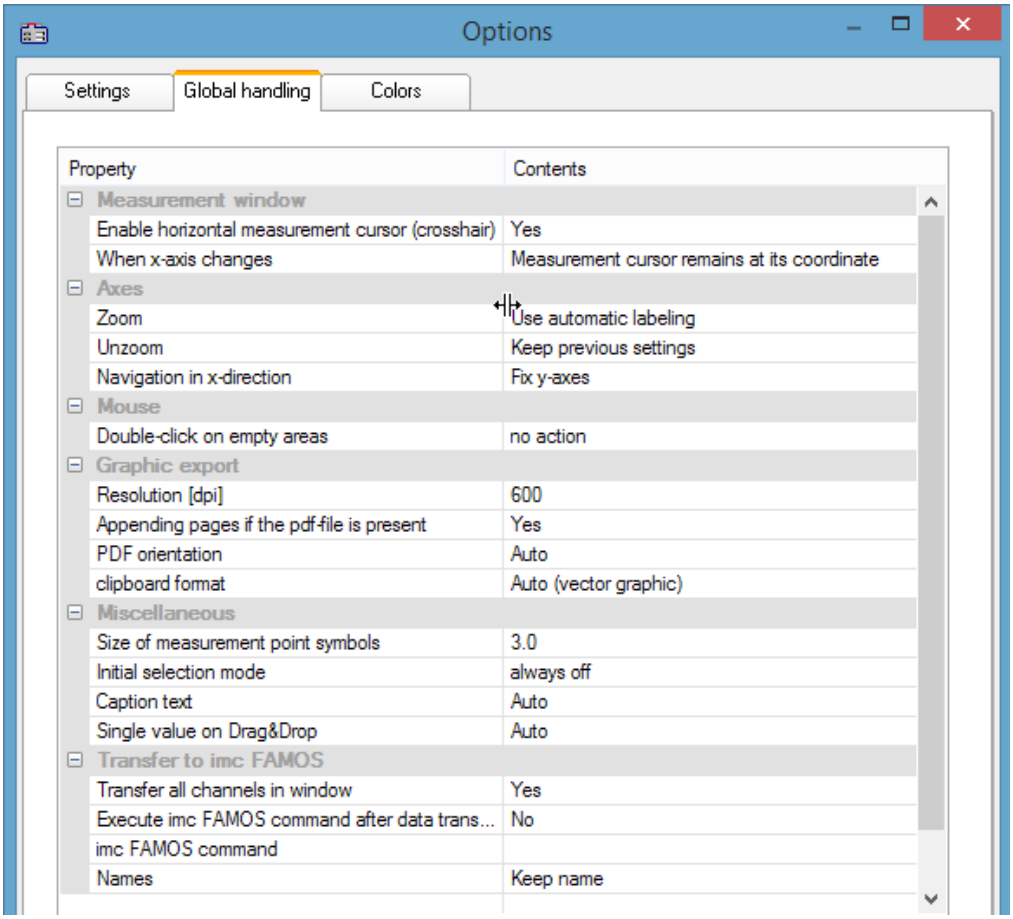
Sound output: Options for playback of a data set by means of the audio output.	
Control	Either the data is played from the player position (*auto*) or the last, currently recorded data is played back(e.g. during the measurement with imc STUDIO).
Start delay [ms] Max. delay [ms]	Options for synchronization of online streaming data with imc STUDIO. More info is presented here 1511.

History: Determines the maximum undo buffer depth for changed made.	
Max. memory [MByte]	Here you can determine the maximum memory allocated to the history of changes, which can be reversed by means of the undo 1473 command.

Scroll mode: Options for automatic scrolling for streaming data with imc STUDIO	
Allow smart scrolling	*Yes/No*
Smart scrolling as of width	Stated in milliseconds. With this setting, you determine the scrolling behavior with streaming data during a running measurement. In particular with the output of sound, these settings can be used to avoid lapses.
Time lag with smart scrolling	
Jitter on the right edge	

Graphic export: Sets the maximum memory depth for the changes made.	
Export optimization	**Bitmap:** A bitmap is created containing the **interior of the coordinate sys**tem, with its curves and graph lines. This bitmap is then exported as a PDF. In consequence, when the graphic is complicated, having very many measurement points, graphics elements are not generated as a vector graphic, but only as a bitmap. Markers, axis labels and legends are not affected and continue to be processed as text elements as vector graphics. **Vector graphic:** The interior of the coordinate system generates a vector graphic. **Auto:** Set to either "*Bitmap or Vector graphic*", depending on the implementation. This setting may change in future versions. • The setting "*Export optimization*" is only useful if overall a vector graphic is to be generated in Global handling 1501. If the setting there is for export of the curve window as a Bitmap to PDF 1501, this setting provides no benefit. • This option is applied, besides when **exporting the curve window as PDF**, also when **printing out a Panel**, exporting the **Panel as a PDF** and when transferring to the **Report Generator**. • **In a Report Generator printout file**, the setting Bitmap can cause very large DRB files, since bitmaps are not compressed there. But subsequently, compact and easily handled PDFs are generated from the DRB. • This option applies **individually to each curve window** or ccv-file. It can thus be used to provide individualized modification of the behavior for particular curve windows, e.g. ones with complex graphics. • When transferring a graphic to the **Report Generator**, any printer currently set for the Report Generator is taken into account. Thus, the coordinate system's bitmap is appears in shades of gray when a B/W printer is used. If this is not the desired behavior, because actually only a PDF is to be generated, a color printer (or if not available the XPDS printer) should be selected.

Global handling



Presettings - Handling

Measurement window	
Enable horizontal measurement cursor	*yes/no:* display of the horizontal measurement cursor (yes: crosshairs; no: vertical line)
When x-axis changes	*Sets whether the measurement cursor remains at the data set coordinate or at the pixel position when the zoomed region is changed.*

Mouse	
Double-click on empty areas	Enables *Select-mode* 1441 to be activated by double-clicking on an empty space in the curve window.

Axes: Here, settings are made which affect the ticks on and at the ends of the axes, once zooming and or re-zooming has been performed.	
Zoom Rezoom	*Use automatic labeling; Place labels at end, use rounding;Place labels at end, use no rounding; Keep previous settings* *See Axes Scale* 1351
Navigation in x-direction	Determines whether after navigating 1516 the Y-axis is permanent or is scaled automatically.

Graphic export: Settings for exporting the curve window as a graphic to the Clipboard	
Resolution [dpi]	Dots per Inch (150, 300, 600, 1200)
Appending pages if the pdf-file is present	If the PDF document already exists, it can be overwritten or supplemented with the curve window.
PDF orientation	Portrait or landscape
clipboard format	Vector- or pixel graphic
Create PDF files	**Bitmap:** The entire graphic is converted to PDF as a bitmap. Resolution in dpi. (Default before Version imc FAMOS 7.3) **Vector graphic:** All graphics elements which are not available as a bitmap are embedded in the **vector graphic in the PDF**. For color maps and 3D, the resulting bitmaps have 300 dpi, which are retained. However, the text elements in the curve window are processed to vector graphics. The system can do a search through the PDF for these by text search. The vector graphic as significantly better resolution at lower memory requirements for normal curve plots. BUT: for graphics with many vector elements (e.g. 10000 big solid dots), the PDF becomes unwieldy large an export is slowed down by an extreme amount. In this case, it is preferable to use the Bitmap setting. On an individual basis, the interior of the curve window can be generated as bitmap and the text elements as vector graphics. To do this, use vector graphics here and for *Export optimization use "Bitmap" under Settings* 1499. For vector graphics, the **Windows XPS printer** is used. This is installed by the operating system and must be operational. Otherwise, the printer will need to be reinstalled via the PC's control panel settings. Here, a resolution of 600 dpi is always assumed; the stated resolution is ignored! **Auto:** When exporting from the Panel or Report Generator, imc FAMOS checks which method is more advantageous. For specific conditions, always select either Bitmap or Vector Graphic. From directly in the curve window, *auto* is either set to *Bitmap or Vector Graphic*, depending on the implementation. The settings are global and can also be set in imc FAMOS, under the menu item: "*Extra*" > "*Options*" > "*File - Save/Export: PDF*".

Miscellaneous	
Size of measurement point symbol	*0.5 -10* mm. Size of the symbols 1371 representing the measured points, e.g. squares, circles...
Initial selection mode	Default setting for *Select-mode* 1441.
Caption text	**Auto:** Name of first dataset. **Filename:** The filename of the loaded ccv file can be shown in the caption of a free-floating curve window.
Single value on Drag&Drop	With the option "*As horizontal line*", single values are always displayed as a line, e.g. for display of value limits.

Transfer to imc FAMOS: A detailed description is provided here 1502.	
Transfer all channels in window	You can transfer either all of the data sets or only the first one which has the window's title.
Execute imc FAMOS command after data transfer	The function/sequence set under "*imc FAMOS Command*" is executed.
imc FAMOS command	Command which is executed once the data have been transferred if the option "*imc FAMOS command after data transfer*" is activated.
Names	Variables names can be changed in order to avoid overwriting existing variables.

Remark: The presettings remain in effect after the program is closed.

Reference: Colors 1493

Transfer Options

Waveforms displayed in the curve windows can be transferred directly to imc FAMOS. It is often useful to transfer data from imc STUDIO or user-made programs which use the curve window to imc FAMOS for special evaluation.

But even within imc FAMOS, the measurement window 1431 enables signal portions located between measurement cursors to be copied. Note that there is a way (Name 1502) to automatically rename the portions, in order not to overwrite the original data.

Waveforms are transferred one by one to imc FAMOS. Once a waveform has been successfully transferred to imc FAMOS, it appears in the variables list.

For the purpose of transferring to imc FAMOS, it is also possible to select options via the presettings 1502.

Execute imc FAMOS Command after Data Transfer:

Immediately after all selected waveforms have been transferred, a imc FAMOS-command can be optionally executed. Any command which can be executed in imc FAMOS can be transferred. For example, a sequence call can be transferred, which imc FAMOS uses to execute the sequence. When the option "*Execute imc FAMOS command after data transfer*" is not selected, it is irrelevant what the user enters in the input box.

Names

Use this option to decide under which file names the data are stored in imc FAMOS after transfer. The following options are available:

Options	Description
Keep	The name of the waveform is accepted unchanged. This option is useful when data are transferred from another application to imc FAMOS and the same name should be used. **Attention:** Be careful when transferring measurement intervals of imc FAMOS curves!
'_' instead of first character	The first character in the name is replaced by '_' (underline).
'_' precede	A '_' is placed **in front of** the name. If the name is too long, it is truncated at the end.
'_' append	A '_' is added **to the end** of a name. If the name is too long, the last character of the original name is dropped.
Fixed Names	User-defined names can also be specified. This option is useful only when just one waveform is transferred. If the user attempts to transfer multiple waveforms with the same name, each waveform will be overwritten by the next one.

The option most appropriate depends on the application and the particular names. The options '_' *precede* or '_' *append* are recommended for transferring measurement intervals within imc FAMOS. Any significant characters in the name should not be changed. The transferred data should have distinct names.

Mouse Operation

- Select menu item *Options/Transfer options*... to set the options for transferring data. A dialog appears in which settings are made.
- Select menu option *File/ Transfer to imc FAMOS* in the curve window to transfer data to imc FAMOS using the specified options.
- In order to transfer a curve window section defined by the measurement cursors, select the context menu item *Send curve segment to imc FAMOS* 1431.

Remarks

- If the imc FAMOS application hasn't been opened yet, it is automatically started when transfer to imc FAMOS is executed. The imc FAMOS.EXE file is searched for in the same directory as the IM7CUDAM.DLL file.
- If waveforms are to be transferred to imc FAMOS, it is recommended to end any sequences running in imc FAMOS before starting transfer.
- The transfer options remain intact after the program is ended.
- Transfer options are valid for all curve and measurement value windows.

13.6.7.4.4 Printer preview

This item is only enabled in the Data Browser's Report view.

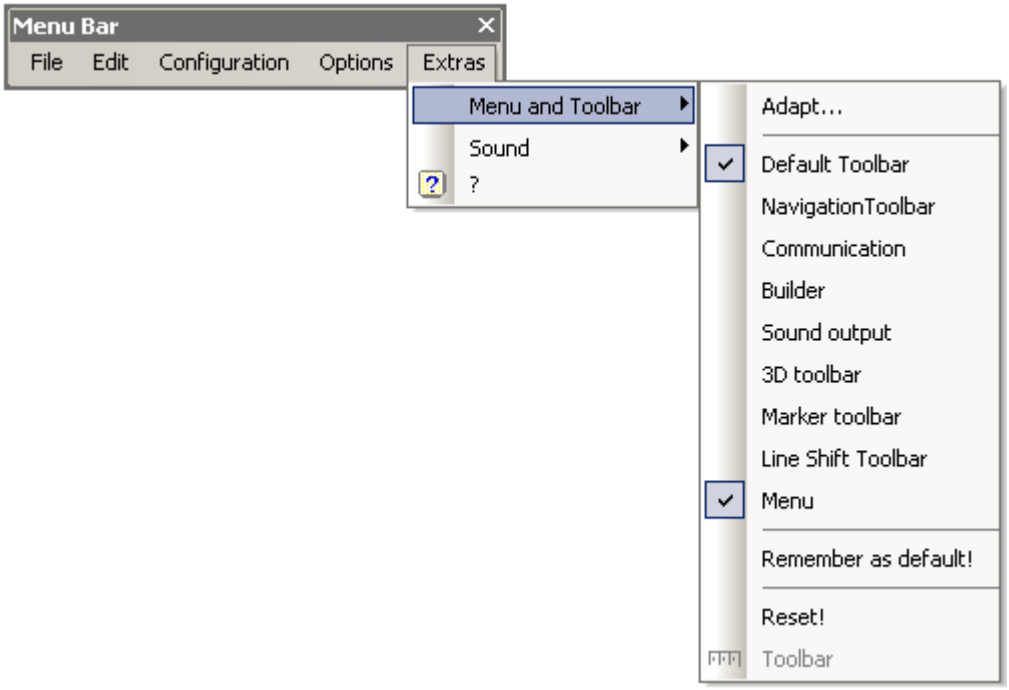
The *Print Preview* is activated by default for curve windows which are incorporated into a report in the Data-Browser. This means that the colors are displayed which the user set in the *Colors* 1493-dialog in the *Options* menu as the color scheme for the printer.

With this option, it is possible to switch to the monitor screen's color scheme.

13.6.7.5 "Extras" menu

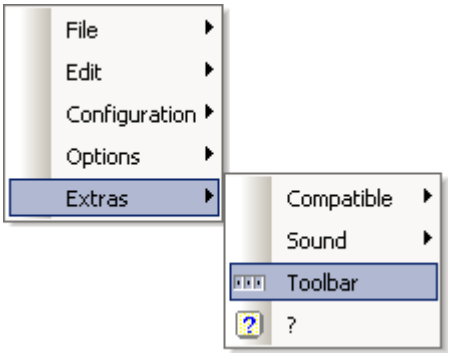
Menu item	Description
Menu and Toolbar 1504	Options pertaining to the menu and toolbar.
Sound 1507	Activates the sound output and access to the functions of the Sound-Toolbar.
?	Displays the curve window's Help.

13.6.7.5.1 Menu and Toolbar



Menu item	Description
Adapt... 1505	Dialog for adapting (making one's own arrangement of and settings for) the curve window's menu and toolbar.
Standard Toolbar, Navigation Toolbar, ...Menu	These toolbars or even the menu bar can each be activated separately.
Remember as default!	The arrangement of the menu and the toolbar is saved as the valid default arrangement. In this way, the same settings are used even for a different, new curve.
Reset!	The menu and toolbar are reset to their initial, factory-set state.

Toolbar

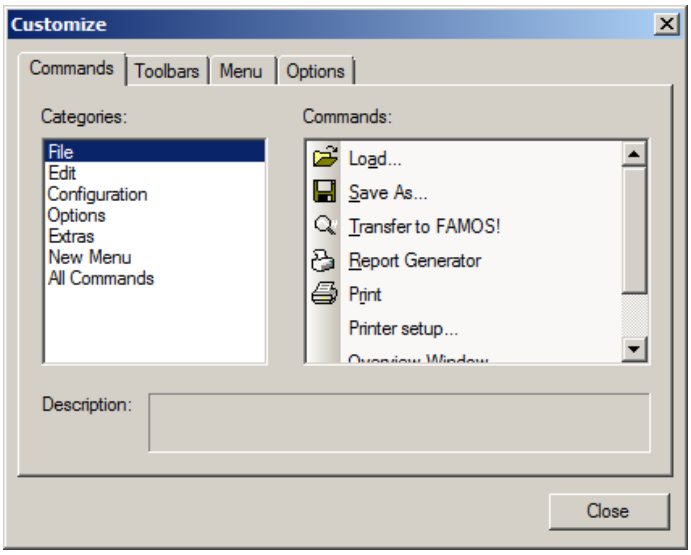


As a curve window integrated in the Data-Browser's Report view, the entire Toolbar only appears when a curve window is selected. To cancel this behavior so that the Toolbar does not appear, the function "Toolbar" can be selected. This is only available from within the Data-Browser. To make the toolbar visible again, select "Toolbar" under "Extras" in the curve window's context menu.

13.6.7.5.1.1 Adapt / Customize

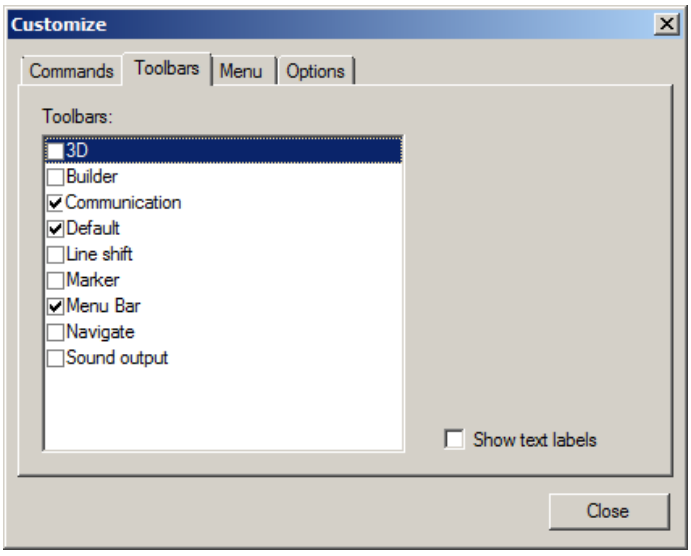
The curve window's menu and toolbar can be adapted. Right-clicking in either the menu or the toolbar opens the context menu. Alternatively, select the menu item *Extras / Menu and Toolbar*.

The menu item *Adapt*... opens the following dialog:

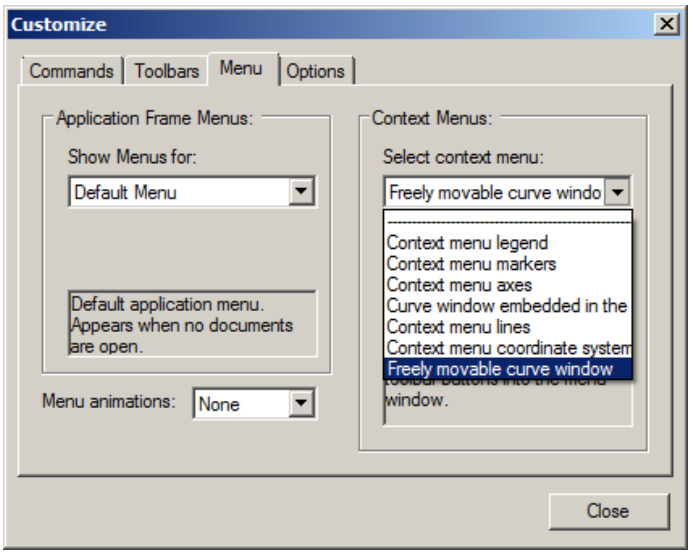


A command can be dragged from thee list at right and dropped on a menu or toolbar. Pop-down menus open by themselves when the mouse is moved over them. The list at left contains the different command topics.

The dialog below offers the additional option of showing or hiding the various toolbars:



The curve window has a context menu. This, too can be edited:



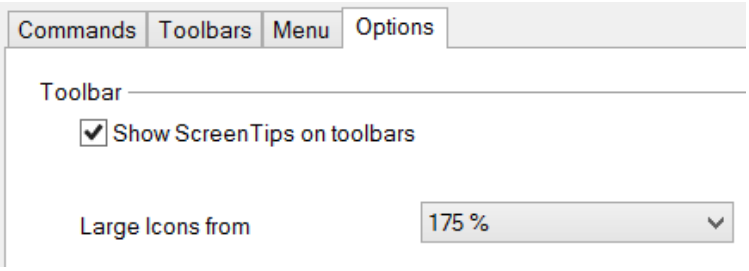
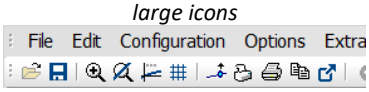
Select the display type which you wish to process.

- normal curve window which is free-floating and freely movable and can also be maximized (popup-window)
- a curve window without title bar (child window) embedded in a dialog.

While editing menus, the curve window can't be operated.

While the dialog *Customize* is open, all menu items as well as toolbar elements can be moved around by the Drag&Drop technique. Note that moving an element to the "outside" means deleting the element.

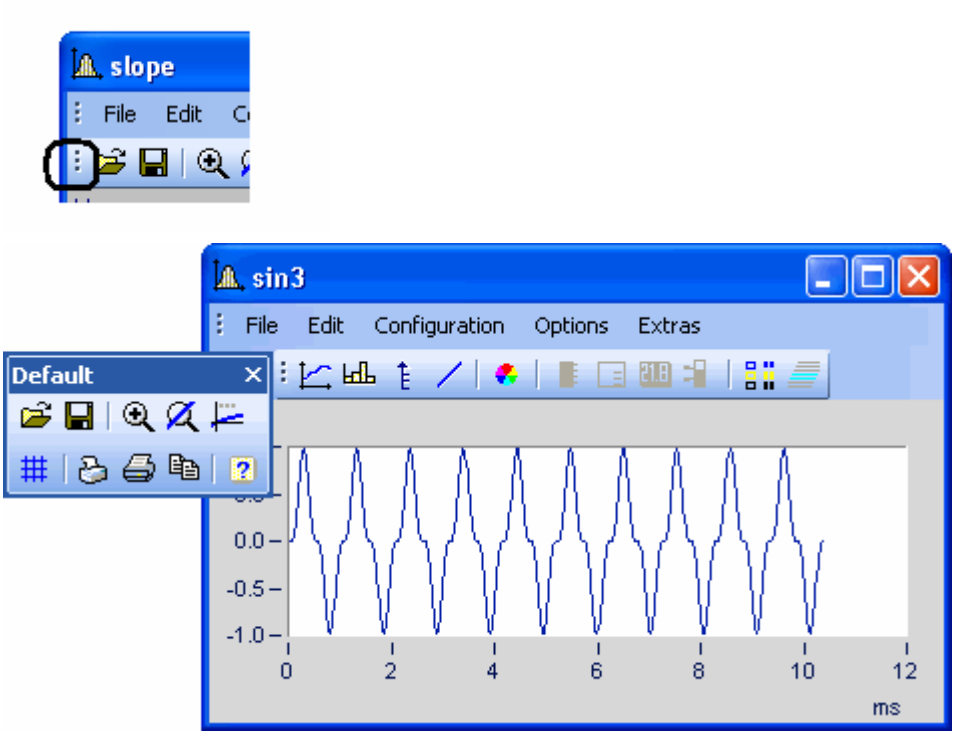
On the page "***Options***", there are two options for the display of the *Toolbars:*

Options	Description
Show tips on toolbars	A description of the icon is displayed when the mouse is moved over the window.
Large buttons	For clearer display at fine screen resolutions such as 4K, the icons can be automatically enlarged from a certain scale.  *small icons* *large icons*  *small icons*

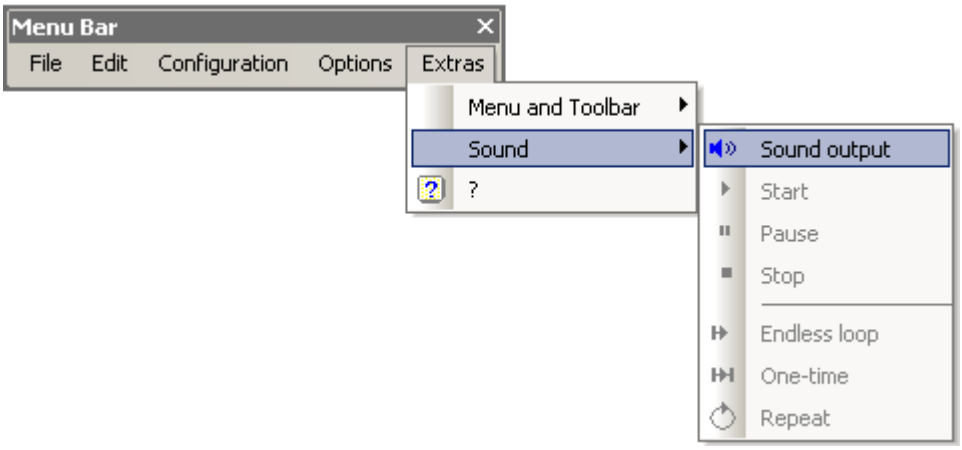
Changes of the menu are permanently saved when the dialog Customize is closed. The particular toolbar arrangement saved depends on the display type:

- Default
- Last value as number
- Overview window
- Table

The toolbars can also float freely away from the window. To accomplish this, drag the toolbar away by its left edge:



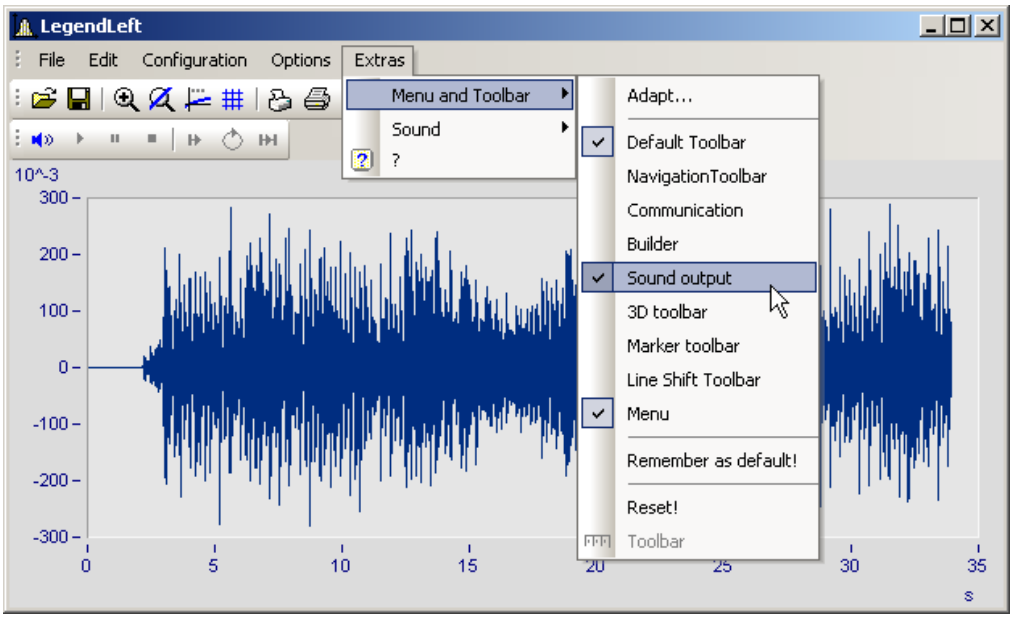
13.6.7.5.2 Sound



The Sound Toolbar lets you make measured data sets audible. For this purpose, a variety of functions such as cutting, endless looping and repeat are available. The sound output always only applies to the first data set in the first coordinate system and is only available in the Standard view, or with multiple y-axes.

13.6.7.5.2.1 Sound toolbar

For working with the sound output, you should select the associated toolbar from the menu *Extras / Menu and Toolbar / Sound output*.



Sound output toolbar

Under Sound Output, the following functions are available:

Menu item		Description
	On/Off	This function lets you switch the sound output on and off. A vertical bar marks the current playback position within the curve window (playback marking), and a horizontal bar denotes the selected playback range.
	Start	With this function you start output of the sound, and the selected range of the data set is played back in real time. The playback's starting and ending points can be freely selected and the position of the resulting playback time window can be moved as desired (see Cut Sound 1510).
	Stop	This function stops playback of the data set. The playback marking is set back to the start of the selected playback region.
	Pause	This button pauses playback of the data set. The playback marking remains at its current position. By clicking on it again, playback can be resumed.
	Loop from loop start to end of waveform	When you choose this operation mode, the final position for the playback is automatically set to the end of the waveform; see Cut sound 1510)
	Loop (repeat) from loop start to loop end	Repeated playback of loop (so-called Repeat-mode).
	Loop (single) from loop start to loop end	One-time playback of the loop

Volume

The volume is set by setting the scaling of the y-axis. The range between y-min and y-max is interpreted as the maximum volume of the sound card. Thus for example, if the signal extends from -5 to 5 (units) and the y-axis extends from -5 to 5, the maximum volume is achieved. If there is an additional signal extending only from -0.5 to + 0.5, and if it is displayed with the same y-axis settings, the amplitudes from the sound card will only be 1/10 as high. I.e. the signal will be played back correspondingly softly.

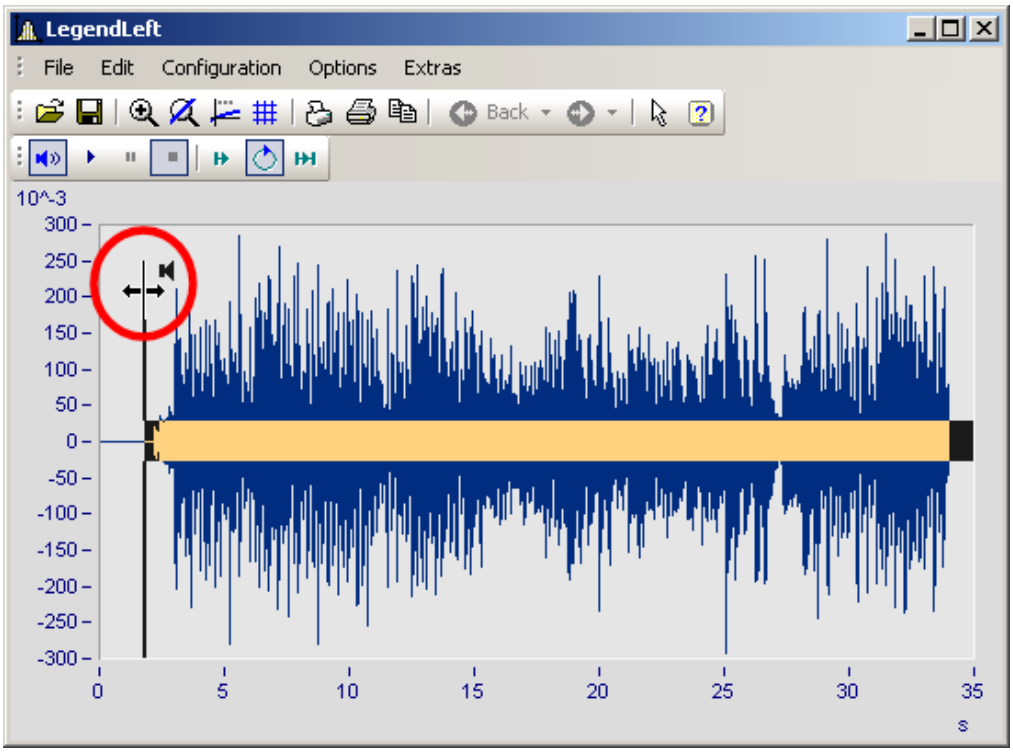
When the volumes of various signals are to be compared, all y-axes must be set to the same scaling. If the y-axes are scaled automatically, all signals have the sound card's maximum level.

The Sound Kit functions have not settings options for volume.

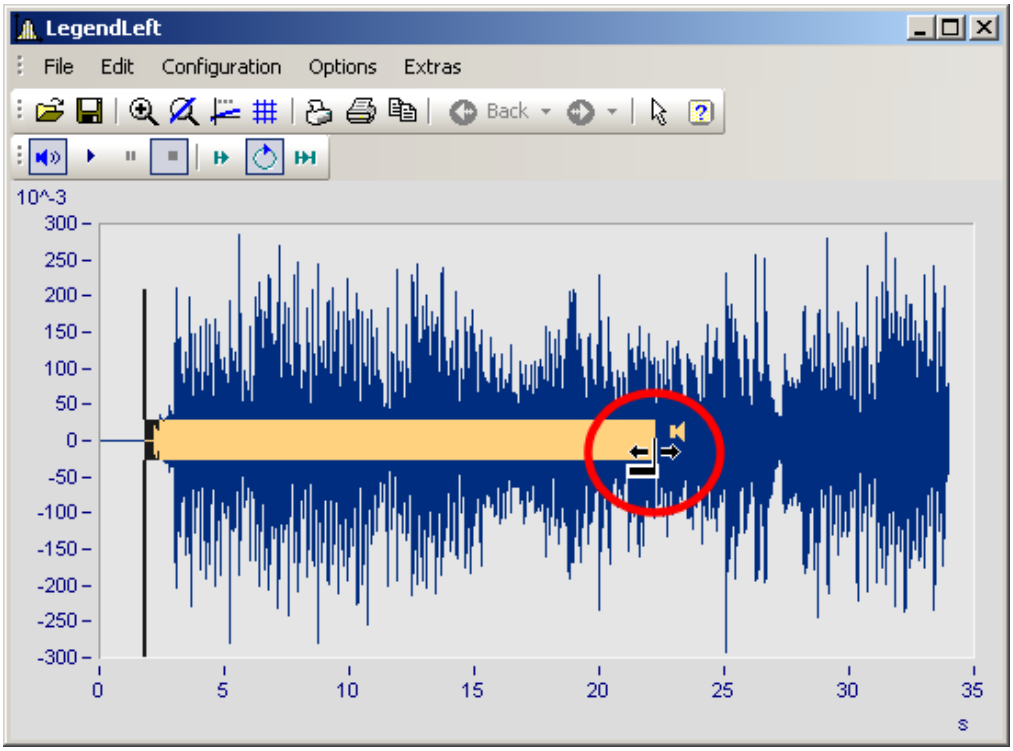
13.6.7.5.2.2 Sound clip

In sound output, it is possible to freely select (take a clip from) the playback region.

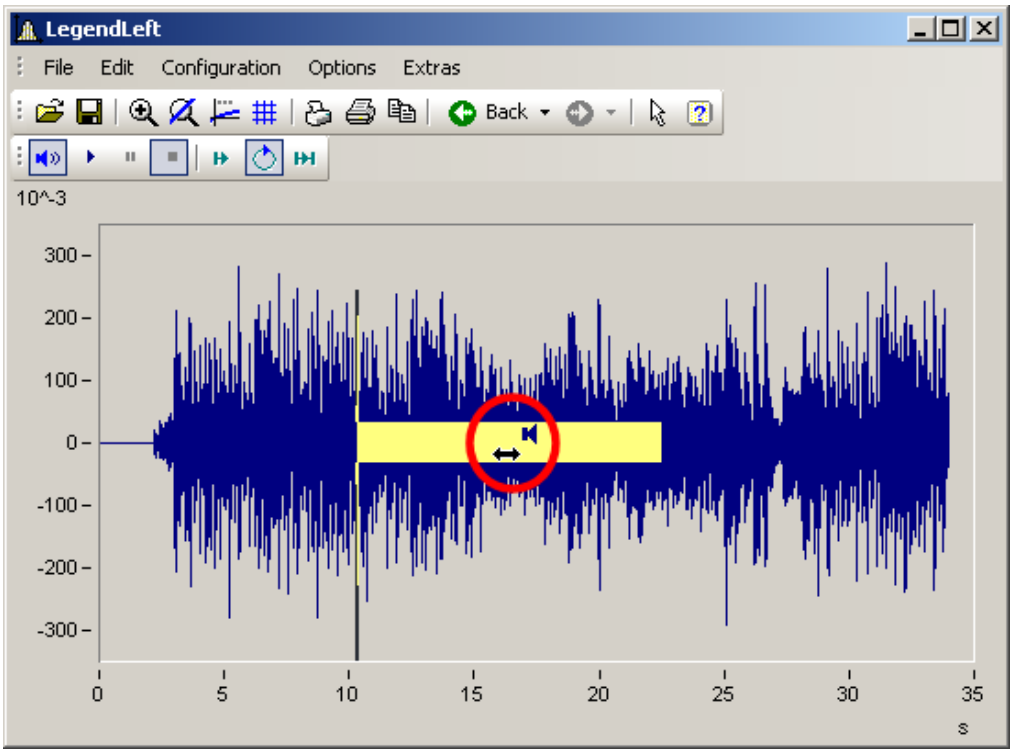
Grab the start position at the beginning of the Playback strip and move it to the position desired.



The final position for playback can also be changed. Grab the end of the playback strip and move it.



You can also move a selected playback region forwards or backwards along the data set. Click on the strip and move the mouse while holding down its button.



13.6.7.5.2.3 Audio output directly during measurement

During a measurement with imc STUDIO /DEVICES, it is possible to use the curve window to listen directly to the streaming data. Since the data are transferred block-by-block and sporadically, the output must be delayed in order for the playback to be seamless.

In the curve window settings 1498, you will find the two entries "*Start delay*" and *"Max. delay".*

At the beginning, as well as after losing synchronization, the *"Start delay"* takes effect. By means of *"Max. delay",* it is possible to compensate for asynchronization between the measurement device and the PC. Both values are stated in milliseconds.

13.6.8 Information and tips

13.6.8.1 Greek texts in comments, markers and axis labels

Axis labels and comments on the variables can be supplemented with Greek letters.

This is accomplished using the statement <g*Placeholder>.

Example

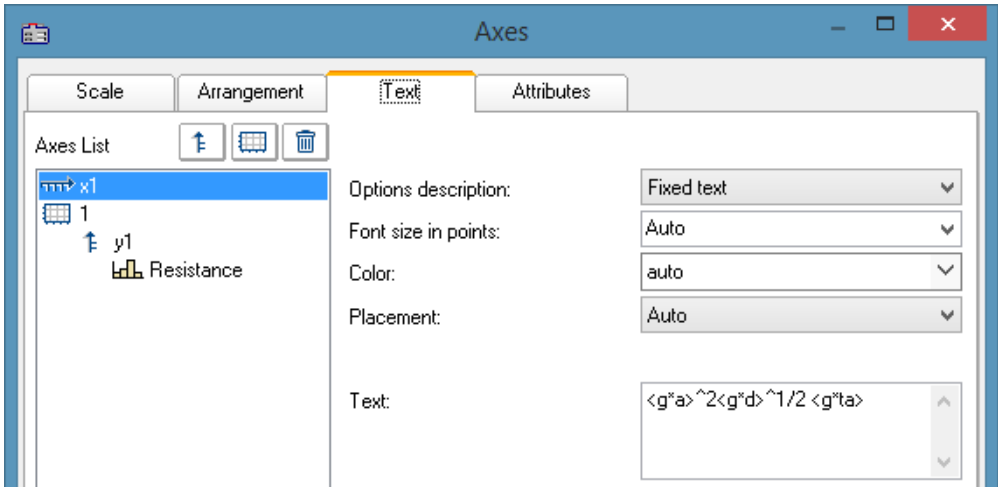
<g*a> ^ <g*a> is displayed as α^{β}.

The following table lists the available Greek letters with their respective placeholders. Here, the Unicode is only provided as a reference; it can't be used in the statement.

Placeholder		Meaning	Placeholder		Meaning
'a'	0x3b1	Alpha	'm'	0x3bc	My
'b'	0x3b2	Beta	'n'	0x3bd	Ny
'g'	0x3b3	Gamma	'x'	0x3be	Xi
'G'	0x393	uppercase Gamma	'X'	0x39e	uppercase Xi
'd'	0x3b4	Delta	'p'	0x3c0	Pi
'D'	0x394	uppercase Delta	'ph'	0x3c6	Phi
'e'	0x3b5	Epsilon	'ps'	0x3c8	Psi
'et'	0x3b7	Eta	'P'	0x3a0	uppercase Pi
'z'	0x3b6	Zeta	'Ph'	0x3a6	uppercase Phi
'th'	0x3b8	theta	'Ps'	0x3a8	uppercase Psi
'ta'	0x3d1	theta, (customary) notation	'r'	0x3c1	Rho
't'	0x3c4	tau	's'	0x3c3	Sigma
'Th'	0x398	uppercase Theta	'S'	0x3a3	uppercase Sigma
'k'	0x3ba	Kappa	'ch'	0x3c7	Chi
'l'	0x3bb	Lambda	'Ch'	0x3a7	uppercase Chi
'L'	0x39b	uppercase Lambda	'o'	0x3c9	Omega
			'O'	0x3a9	uppercase Omega

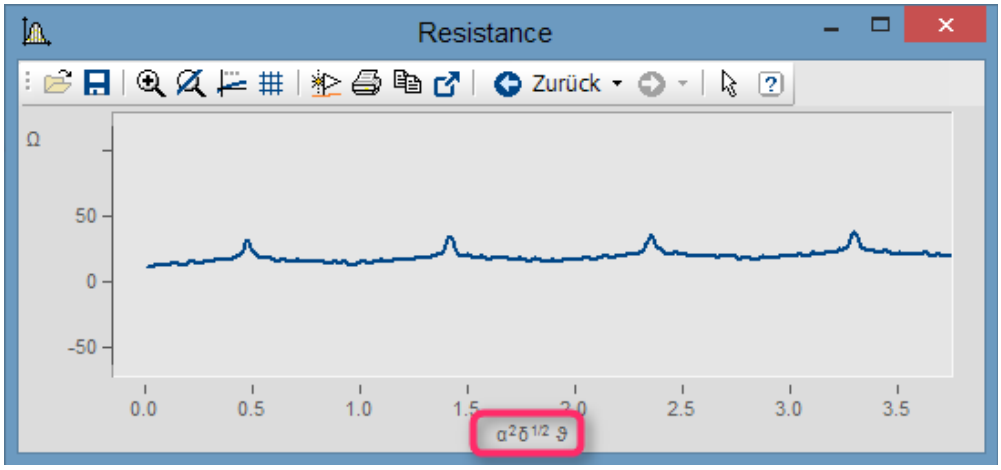
Example curve window

with



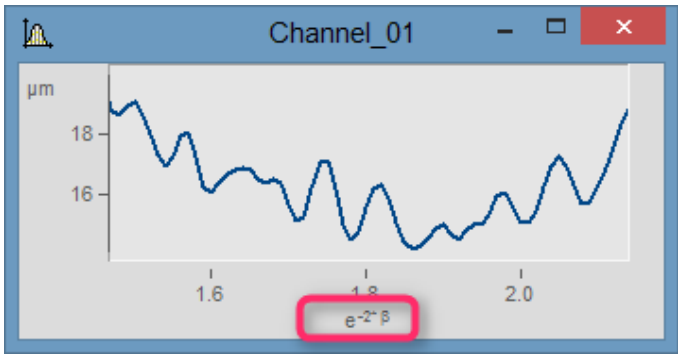
*Greek characters in "Axes\Text": <g*a>^2<g*d>^1/2 <g*ta>*

the result is:



Greek characters as text at the X-axis

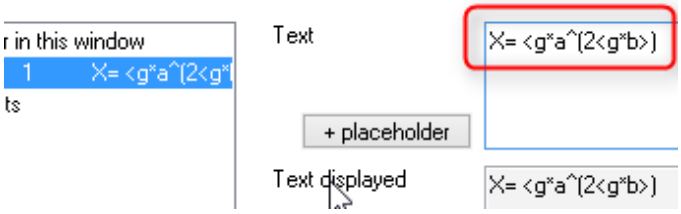
Exponent with multiple characters. From e^(-2*<g*b>) we obtain:



Exponent of placeholder in parentheses: e^(-2<g*b>)*

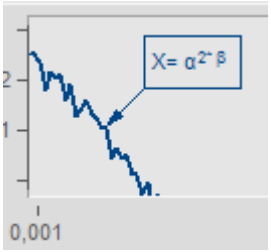
Example markers

with



Greek characters in marker-definition:

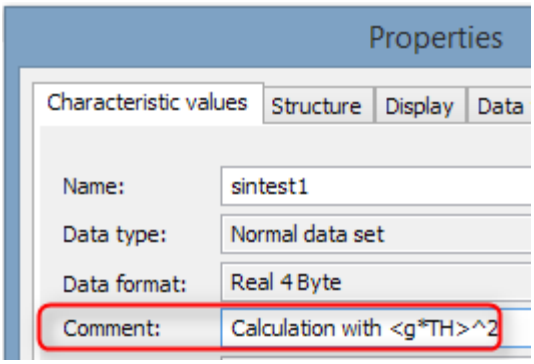
the result is:



Greek characters in markers

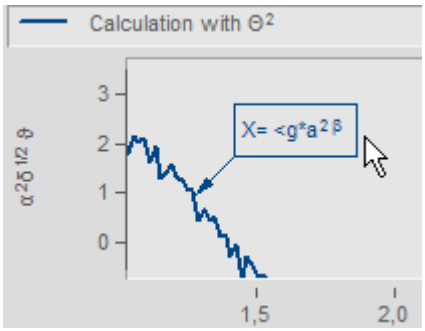
Example comment on variables

with



Greek characters in variable "Properties\Comment"

the result is:



Greek characters in "Axes/Legends"

Orders of magnitude and units

Specified orders of magnitude with units are recognized and displayed appropriately. For example, the unit 'mm' is corrected to the SI unit 'm'. 10.000 'mm' becomes 10 'm', 0.01 'mm' becomes 10 'µm'. The condition is that *"auto"* is set for *"Order of magnitude* 1354*"* under *"Scale"*.

13.6.8.2 Curve window context menu

Function

Click with the right mouse button in the curve window to prompt a context menu, which offers quick access to further functions in the curve window. The context menu depends on the cursor's position and on whether or not you are in Select-mode.

13.6.8.3 Copy&paste - Transfer configuration

With Copy & Paste, you can transfer once curve window's configuration to another. To do this, copy the configuration of the selected window using ***CTRL-c*** and transfer it to another one using ***CTRL-v***.

With embedded curve windows, it is additionally necessary to hold down the *SHIFT*-key (curve window in the Panel).

13.6.8.4 Drag&drop, mouse wheel

Moving waveforms

Drag&drop allows the user to shift or copy waveforms in curve windows. To apply this technique within a coordinate system, hold down the CTRL-key. Use drag&drop for the following:

- **Drag** a waveform **from** the **imc FAMOS variable list to** the **curve window**. This allows the waveform to be displayed in the curve window as well as indicated in the list.
- Within a curve window: While the **CTRL-key is pressed**, **all data can be moved to an axis or to a coordinate system**.
  Note: The old behavior without the CTRL-key is made available by means of the Registry entry:
  `[HKEY_CURRENT_USER\Software\imc Measurement and Control\Default\CurveDataManager\Curves];` String value: `"dd63"="1"`
- Transfer **all data** for an axis, a coordinate system or an entire curve window **to another curve window** for display.
- Move **all data** concerning an axis, a coordinate system or a curve window **to** the **imc FAMOS** variable list. If the data do not belong to imc FAMOS (i.e. are from another application, e.g. imc STUDIO), they are copied to imc FAMOS.
- With maximized curve window display 1276 (coordinate system = outer frame), Drag&Drop also starts when an axis is moved to the bottom. In that case, the mouse pointer symbol changes.

Displayed region

- Using ***drag&drop*** within a curve window moves the axes.
- By using the ***mouse wheel***, the region around the mouse cursor's current position is increased/reduced. The increments of size change are smaller/larger when pressing the SHIFT-/CTRL-key, respectively.
- If the **mouse wheel** is located over either the **X-** or the **Y-axis**, then only the respective axis is altered.
- Pressing the **CTRL-key** causes the **effect of the mouse wheel** to be **increased**, while pressing the **SHIFT-key** causes it to be **diminished**.

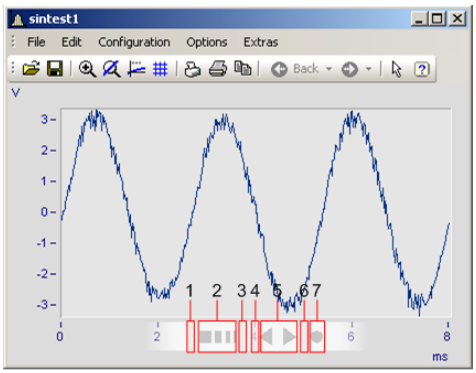
Remarks

- If the mouse is released over a y-axis, the data are displayed on this axis. Release the mouse over a coordinate system, and the data are displayed with a new y-axis for this coordinate system. Otherwise, a new coordinate system is opened.
- When data are moved to the variable list in imc FAMOS, the waveforms are actually copied. In all other cases, the same waveforms are simply relocated.
- The cursor changes its shape during "Drag". If a symbol appears indicating an invalid operation, "Drop" cannot be executed, only Cancel is possible.

13.6.8.5 Axes navigation bar

For quick navigation through the curve window, each axis has its own navigation bar. The controls appear when you move the mouse cursor square over the axis labeling. The navigation bar offers three basic functions:

- You can magnify or reduce the displayed region, while leaving the value in the middle of the x- or y-axis, or z-axis remains unchanged. This function is provided by the button denoted as "2" in the picture. Alternatively, you can change the axes' range using the mouse wheel.
- You can shift the region displayed to the right or the left. This function is provided by the button denoted as "5" in the picture.
- Any changes in the magnification, moving or zooming can be reversed and thus return to their original appearance.



Dragging		Description
1 <- [2] 3	Dragging from 2 to 1	Magnification
1 [2] -> 3	Dragging from 2 to 3	Shrinking
4 <- [5] 6	Dragging from 5 to 4	Move left
4 [5] -> 6	Dragging from 5 to 6	Move right
[7]	Click on 7	Rezoom the X-axis. Displays all data

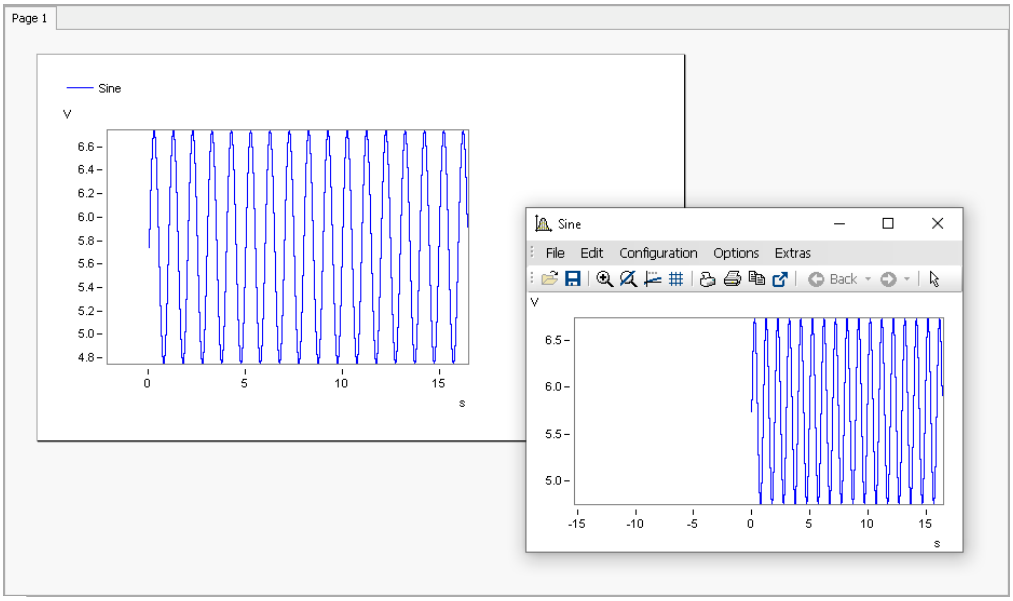
Use of the keyboard arrow keys

While holding down the left mouse button with the pointer in the Navigation pane, you can also move the region by using the keyboards arrow keys.

13.6.9 Curve windows in imc STUDIO

13.6.9.1 Curve window - "free-floating"

The "**free-floating**" curve windows are associated with the experiment. They remain open until closed (X) and are displayed even when the Panel is not open.



Example: curve window - embedded and "free-floating"

The curve window can be opened from different plug-ins:

Setup:

- Open the context menu for the desired channel in the Channels table
- Click on: *Show in curve/values window*

Panel:

- Double-click in the tool window Data Browser on the desired channel
- or select the desired channels and click on the *Show* 1222-button (/) in the symbol bar of the tool window Data Browser.

Reference See also

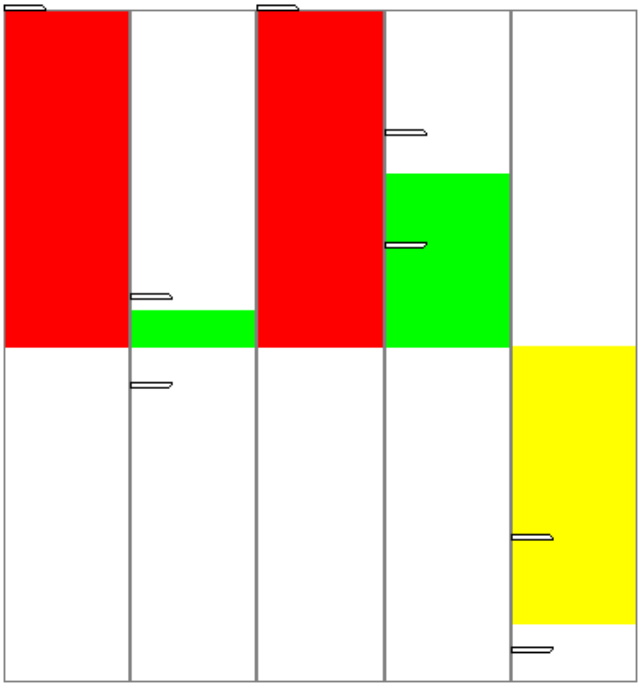
Setup - Device Configuration > "*Import of an imc DEVICES Experiment* 283": How do I get the free-floating curve windows from imc DEVICES into the imc STUDIO Panel pages?

13.6.9.2 Level indicator

The Widget is a pre-configured curve window, suitable for channels having a measurement range; e.g. the analog inputs.

The level indicator displays a bar chart representation to indicate how far the current measurement value is from the measurement range. The channel's respective display range automatically adjusts itself to the measurement range set (this is applied upon performing the action "*Process configuration*").

The current signal level is represented by a three-state color display.



The axes are configured as follows:

- **Range:** ±110% of the measurement range
- **Warning color yellow:** 80-100% of the measurement range (for both upper and lower level)
- **Warning color red:** beyond 100% of the measurement range (for both upper and lower level)

Note — Taring and bridge balancing action before and during running measurement:

Upon starting the next measurement after a bridge balancing action, the level indicator automatically adopts the channel's value range boundaries. If the balancing is performed during the measurement, the Widget does not display the correct value level since it only knows the previous range boundaries. The indicated measurement range in the file also retains the previous state.

13.6.9.3 Curve window colors on the Panel page

The curve window colors can be configured differently for the display/operation pages and for the Report pages. The Panel pages have their own "skins", which specify the applicable colors. However, you are also able to personally specify the colors for each curve window.

As described in the chapter "*Colors* 1493", there are colors for the on-screen display, and colors for the printout. Typically, the result should appear the same whether in the printout/PDF or in the monitor screen display. The colors are applied as follows:

Page	Colors
Dialog-page	• On the monitor screen: Screen color • In the printout: Screen color

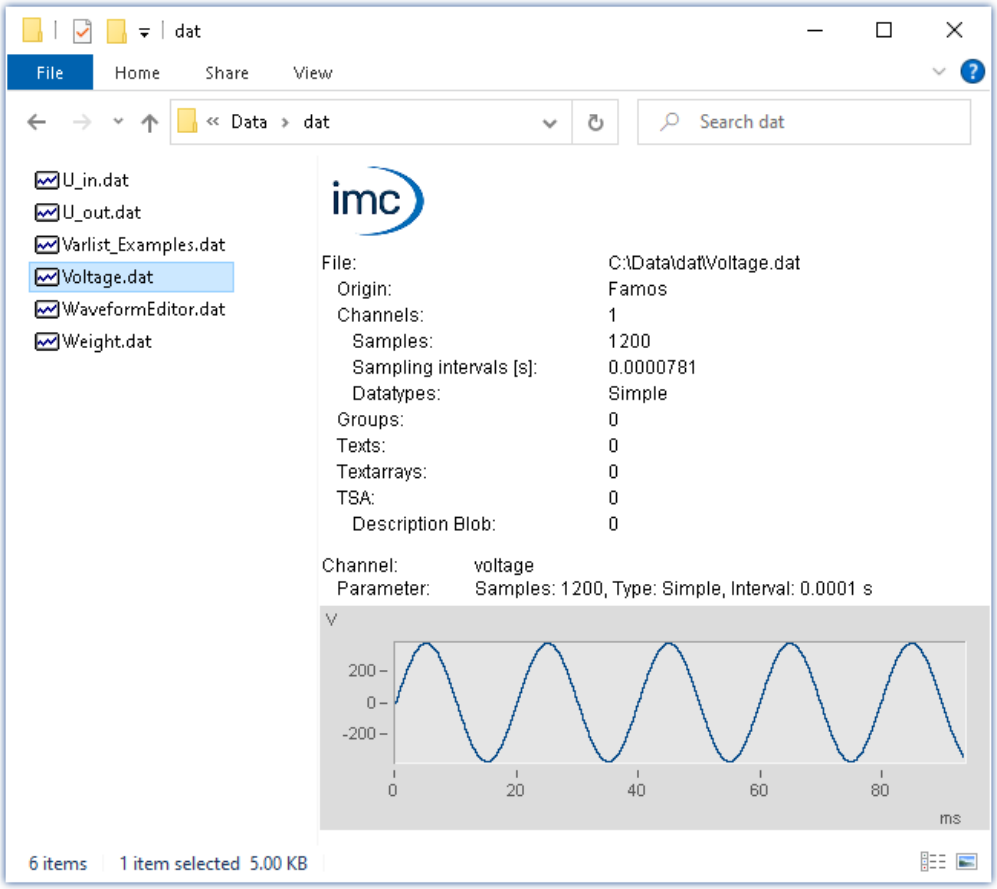
Page	Colors
Report-page	• On the monitor screen: Printer color (The Report-page is designed for the printout or for the PDF. In consequence, the printer colors are displayed here.) • In the printout: Printer color

Note Curve window on Report pages

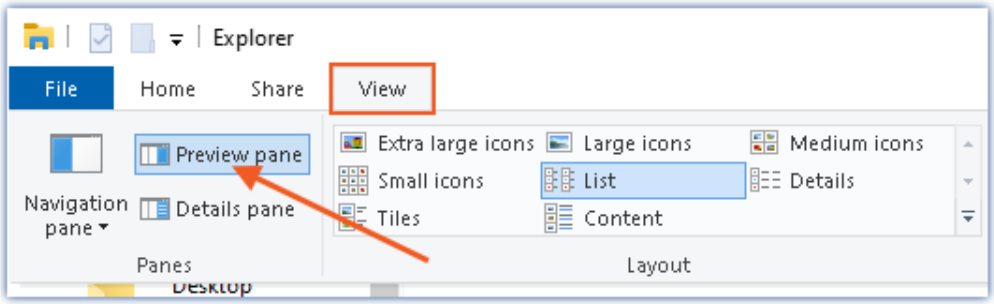
Curve windows on Report pages are set so that the global colors are used for the printout. Furthermore, the background is always white, even when a different color is set.

13.6.10 Preview in Windows Explorer

imc FAMOS files can be displayed using the Microsoft Windows Explorer's preview function.



Activating the Preview window



Activating the Preview window in the Windows Explorer's View menu

The following properties are displayed:

Area	Description
Upper view region	• A roundup of the data regarding the channel count, texts, text arrays, log channels, bus channel descriptions and groups. • For channels, the lowest and the highest channel counts, the sample interval and the data types present are additionally listed.
Lower view region	Infos: • Output of the displayed file set's name • Roundup of channel parameters, sample count, interval, data type • if applicable: additional channel properties group association
Curve window, text contents, array contents	Only **one** data set is always displayed. When multiple files are selected, only the first is displayed. The curve window comes with a limited scope of the imc curve window's functions, including the ability to zoom in on a specific region.
Scroll bar	When the file contains **multiple channels**, a scroll bar appears, by means of which it is possible to navigate through the various channels.

13.7 Special Widgets

Here, special Widgets are described.

13.7.1 Text input for report channel

This Widget currently has no function.

13.7.2 Clock

With its various design and formatting styles, the clock provides a variety of ways to visually illustrate a report. It can indicate how long a measurement has been running or when a measurement ended. The current time and date can be displayed. For these purposes, a variety of formats are available.

Here are a few examples:

Variants of the clock

Selecting the time display style

A variety of time readouts are available, depending on the linked variable.

Variable linkage	Description
SystemClock - PC time	Current PC clock time - Default option when the Widget is set up.
SystemClock - System time	Current clock time of imc STUDIO ("VRTC 307"). This time is used for all components of imc STUDIO.
Linkage with a channel	Displays the channel's elapsed measurement time, meaning the amount of time which passed since the trigger fired, or the measurement was started. The final time display reflects when the measurement concluded or the triggering event ended. When the trigger fires again, the display time returns to "0 s" and restarts.
Linkage with a trigger-variable's "*Trigger time*"	Displays the moment in time at the last change of the state: Armed, Triggered, Stopped. I.e. when: • the measurement is started, • the trigger fires, • the trigger event ends and • the measurement is stopped. Thus, stopping the measurement also resets the clock after a previously concluded triggered measurement.

Clock properties

Properties	Description
Appearance	Here you specify the appearance of the clock; e.g. with **clockhands**, or a **digital readout**.
Mask	There are various ways to display the time or date. The output format depends on the system and on its setting for the Region (Windows "Clock, Language & Region"). E.g. in the format "Long date" in one region may contain the weekday, while in a different region it does not. **Mask** — **Description** Long time — Example: 08:11:45 Short time — Example: 08:11 Long date — Example: Wednesday. April 17, 2019 Short date — Example: 17/04/2019 Date and time — Example: 17/04/2019 08:11 UTC (IRIG Format) — Example: 107:08:11:45:261 The first position indicates the current day's number within the calendar year. Thus the date 1/5/2019 is day "5" in this system. This format provides no indication of the current year's number.
Time zone	In imc STUDIO, it is possible for differing time zones and clock readings to coincide. • The external clock (e.g. GPS-mouse) may indicate different time than the PC. • The devices may be located in different countries. • On a business trip, the laptop's clock time may reflect a different time zone than the device's. The clock-Widget can be set for different time zones. For this purpose, there is a new selection available: "*imc STUDIO-Time zone*". With this selection, the clock automatically uses the device's time zone. With the selection: "*Local time zone*", the PC's time zone is used.

13.7.3 Image

Using the Widget, you are able to display a logo or a project graphic, or a background which shapes the page. There are various ways to apply the Widget.

Reference	Background image

If you wish to display a background image over the entire page, you can use the page's property: "*Background image* 1539".

Properties

Properties	Description
image	Available file types: png, jpg and bmp If an image is selected, the text: "*(Bitmap)*" appears under this property. In order to delete the image, delete the text. Be aware that large pictures place high demands on the PC's memory and performance capacity. So if possible, first modify the image to the desired size.
Size mode	The image can be stretched or displayed without any zoom factor:

Size mode	Description
Auto and Normal	The image is displayed in its original size within the Widget. The upper left corner is the reference position. If the Widget is too small, only a section is displayed.
Stretch image	The image is stretched to completely fill the Widget. The aspect ratio is not retained.
Automatic Widget size	The image is displayed it its original size. The Widget adjusts itself to the image's size.
Center image	The image is displayed in its original size within the Widget. The center point is the reference position. If the Widget is too small, only a section is displayed.
Zoom image	The image is stretched in such a way as to touch the edges of the Widget on at least two (opposing) sides. The aspect ratio is retained. At the other two edges, there are resulting empty strips ("*Background color*").

Background color	The color displayed wherever the Widget is not totally filled by the image.

13.7.4 Graphical switch

The graphic switch can be used as an operator control and as a status indicator. Depending on the switch state, it can have **different designs** or **colors**. Thus for example, the **positions of movable elements** can be represented schematically. As well, **rotation of the image** can be achieved by means of an additional variable.

left: switch without any rotation
right: switch with 15° rotation

Properties	Description
Zones	In addition to the color, an image can be defined for each zone 1252. Thus, various states can be represented by a color, or movable objects by multiple images.
Rotation: Angle	The angle can be set to a fixed value or to a variable. The value is stated in degrees (0-360). Values outside of this range are converted accordingly: -90 = 270; 450 = 90.
Rotation: Center point	The center point around which the switch is rotated.
Rotation: Angle factor	This factor affects the angle. Example: Factor "90" → for angle values of 0, 1, 2, 3, the switch rotates by 0°, 90°, 180°, 270°.
Rotation: Zoom factor	Within the Widget, the image is shrunk by this factor so that it can be displayed completely even when rotated. As soon as any angle is entered, the Zoom factor is automatically set to 0.7. This value can be changed.

13.7.5 Execute menu action

This Widget lets you execute a menu action. A list contains the actions available in the menu ribbon. When this Widget is run, it replicates clicking the mouse over the corresponding button. This widget can be helpful when the Panel is running in fullscreen mode. The menu ribbon is not present then.

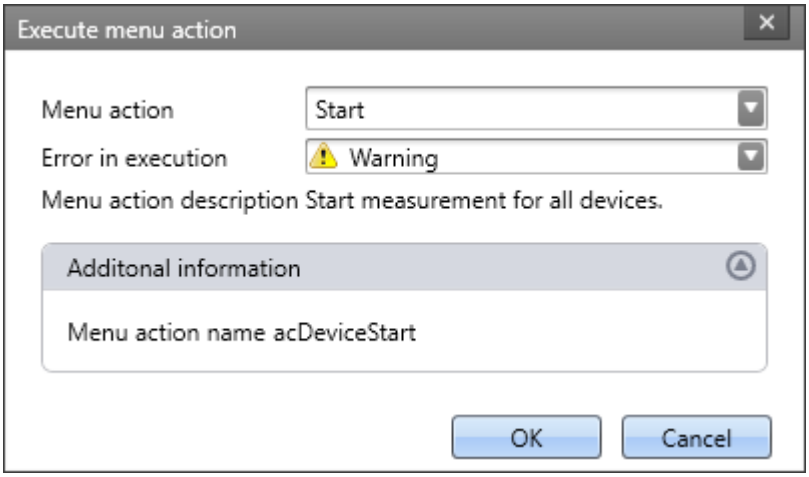
For touchscreen displays, the Widget comes in very handy. In order to make it easier to operate, it can be magnified for display. The icons adjust to the size of the Widget.

Various menu actions

This Widget adapts itself to the status of the menu action. E.g. when the action is currently not present, the button can not be clicked. Some menu actions are attached to specific user roles, device components or loaded imc STUDIO-components.

Reference Command - Execute menu action

For Sequencer-routines, there is a command of the same name 1756. This command is often linked with switches on the Panel.



Configuration: Execute menu action

Parameter	Description
Menu action	Here, select the desired menu action. Open the drop-down list. The menu actions appear in groups. Scroll to the appropriate group and select the desired action. Filtering: You can enter the first letter of the action's name in the input box. Then the list will be restricted to actions starting with that letter.
Error in execution	If this action returns a fault condition, there are various responses available. • Error: The action returns an error message in the logbook. • Warning: The action returns a warning message in the logbook. • Ignore: No info appears in the logbook.
Menu action description	The action's description text is displayed.
Additional information	The menu action's internal designator is displayed.

13.7.6 Table

The table has a wide variety of applications. It can contain single values as well as whole channels and segments. For each cell, you can define various editors such as lists, switches and sliders. You are able to edit a channel's individual points or add or delete points 1528.

If a **value can't be displayed correctly**, the cell is filled with hash characters "###" as an immediately visible indication of the problem. A truncated number is not displayed.

Structure of the properties

The properties are divided among **three areas**: "*Table*", "*Column*" and "*Cell*". Depending on the selection made, the associated properties are displayed. This means that, for instance, if only the table is selected, only that one area is displayed. There is a fourth area when "*embedded Widget* 1531" is present.

Inherited properties: Some properties can be specified to apply throughout a higher area; for instance, you can define "*Zones*" for the table. In each column or cell, you can decide whether the zones are "*inherited*" from the level above, or whether to define specific ones.

Design and properties of the table

Most of the actions below are only possible in Design mode.

Action	Description
Linkage with variables	Either by means of Drag&Drop or of the Properties, you can link individual elements of the table with variables. • Linked with a cell **Single values**, **channels**, **texts**, ... can be linked with any cell. The respective momentary value is displayed. • Linked with a column You can link **channels** with a column. The cells are then filled with the variable's values. Using the property: "*Column*" > "*Behaviour*" > "*Direction* 1528", you can specify whether the channel's first or last values are displayed. • Linked with the table **Segmented channels**, **matrices** can be linked with the entire table. The number of columns is determined by the number of segments (max. 20) Useful property: "*Automatic table length* 1528"

Action	Description
Adding/removing columns and rows	By means of the properties, you can specify the column and row counts. If you wish to add/delete a targeted **column**, open the context menu at the desired position in the **column header** and select the respective action. If you wish to add/delete a targeted **row**, open the context menu at the desired position in the **table** and select the respective action.
Moving columns	Use Drag&Drop to move the columns to the desired position
Table length in relation to the channel length	By means of the property: "*Table*" > "*Behaviour*" > "*Automatic table length*", the table length is automatically assimilated to the count of points in the channel; e.g., you can display imc FAMOS results in the table. In response to each additional result in the channel, the number of rows increases. When multiple channels are displayed simultaneously, please ensure that the point counts and update rates of the different channels are the same. Do not use another column to display single values.
Order/read-out direction	If the column is linked to a channel, use the property : "*Column*" > "*Behaviour*" > "*Direction*" to specify whether the first or last value is displayed at the top. By clicking the mouse button over the arrow icon next to the column header you can switch the read-out direction. Using the property: "*Table*" > "*Behaviour*" > "*Allow a change of direction*" you can deactivate the change at a click of the mouse.
Start only at x-th value	Using the property: "*Column*" > "*Behaviour*" > "*Offset*", you can set for a channel that, for instance, the values starting from the 5th value on. 1st value = 0; 2nd value = 1; ...
Insert/delete values for a channel	Add values to a channel or delete some values. By means of the context menu, you can insert a value either above or below the selection: "*Insert sample before*", "*Insert sample after*", "*Delete sample*"
Entry-editors	Instead of entering text and numbers from the keyboard, you also can use other methods to make entries. See "*Entry-editors for the cells* 1529"

Titles, captions and texts

Text-settings	Description
Change text of column header (title)	Change the title using the property: "*Column*" > "*Appearance*" > "*Title*". When the title is changed, the source 1246 is automatically set to "*User-defined*". If the column is linked with a variable (e.g. channel), the variable's properties such as its name, can be used as the title (see "*Source* 1246").
Hide column header (title)	Set the visibility using the property: "*Table*" > "*Appearance*" > "*Column captions*". In this way, you can design your own headers with other elements.
Hide table header (title)	Set the visibility using the property: "*Table*" > "*General*" > "*Title*". By means of the position selection, you can hide the title by clicking on the box "*Title*".
Show title column	For each title, you can show a title column appearing to the left of the column. Property: "*Column*" > "*Appearance*" > "*Title column*". By means of the associated property: "*Column title*", you can change the title of the "*Title column*". Change the contents in the title column cell by means of the property: "*Cell*" > "*Appearance*" > "*Cell title*". When the title is changed, the source 1246 is automatically set to "*User-defined*". If the associated cell is linked with a variable (e.g. DisplayVar), some properties of the variable such as the name, can be used as the title (see "*Source* 1246").
Alignment of the texts	The alignment of the texts (right/left/centered) can be set by means of various properties: • Column header (title) and the column cells: Property: "*Column*" > "*Appearance*" > "*Value alignment*" • Title column and its cells: Property: "*Column*" > "*Appearance*" > "*Title alignment*" • Table caption (title): "*Table*" > "*General*" > "*Title*". You can place the title using the position selection.

Entry-editors for the cells

To enter texts and numbers, you can also use other input techniques. For each cell, you can define an "*Editor*". Property: "*Column/Cell*" > "*Behaviour*" > "*Editor*".

In some cases, the cell's zones 1530 determine what selection possibilities exist and the scope of the editors.

The table's Editors

Editor	Description
Text	Entry of numbers and texts. When zones are used, no selection box is offered.
Combo Box	Entry of numbers and texts using a selection box. The zones determine the selection options available.
Spinner	91314 Entry of numbers. Techniques for making changes include the use of arrow keys in the Editor and the mouse wheel. In these ways, targeted increase/decrease of individual digits can be achieved.
Button	Changes the value while the mouse button is pressed.
Switch	Changes the value upon each click of the mouse button. Display of the value.
Check Box	Changes the value upon each click of the mouse button. Display of a checkbox.
Slider	Entry of a value using a slide control. The range boundaries are defined by the cell -range.

Zones in the table

On this topic please observe the description of the "*Zones-dialog* 1252". There are some additional settings which apply to the table and are described here.

Using the zones you can specify whether texts or numbers are to be set or displayed.

Option	Description
Text as set value	Activate this mode in order to write texts to text variables. Variables as "*Set values*" are not supported in this mode. There are no ranges.
Numerical zones	The variable's value is displayed numerically. The cell displays various colors depending on the zone-boundaries and the current value. If "*Text as set value*" is activated, the "*Set value*" can also be a text. In this case, the display output and the set value are the same.
Textual zones	The variable's value is displayed as a text. In this mode, there is a distinction between "*Set value*" and the displayed text ("*Text*"). In accordance with the zone boundaries and the current value, the "*Text*" of the active zone and the respective color are displayed in the cell.

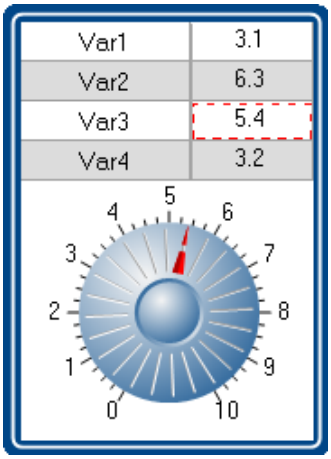
Accept only Zone-values as entries

"*Accept only Zone-values as entries*": While this option is activated, no entry can be made which does not match the "*Set value*". The way this is done is different from Editor to Editor:

Editor	Deactivated	Activated
Slider	Using the slider control, any value between the lowest and the highest "*Set value*" can be chosen.	Using the slider control, only the "*Set values*" can be chosen. The slider skips between positions as you move the mouse. If the variable's value is provided from outside, the value display and the slider control conform to each other.
Combo Box	Using the list, you can select the "*Set values*". You can also enter a different value in the input box.	Using the list, you can select the "*Set values*". There is no option for manual entry.
Button	The value toggles between the first and the second zones.	The value toggles between the first and the second zones.
Switch	The value toggles between the first and the second zones.	The value toggles between the first and the second zones.
Check Box	The value toggles between the first and the second zones.	The value toggles between the first and the second zones.
Spinner	You can enter a value into the input box.	You can enter a value into the input box. If the value does not match any "*Set value*", the system rejects it.
Text	You can enter a value into the input box.	You can enter a value into the input box. If the value does not match any "*Set value*", the system rejects it.
Auto	See Combo Box	See Combo Box

Embedding a Widget

Drag another Widget from the tool window to below the bottom cell of a column. The Widget is displayed as embedded there.



In this way, you can expand each column with its own Widget. Not all Widgets offer this capability.

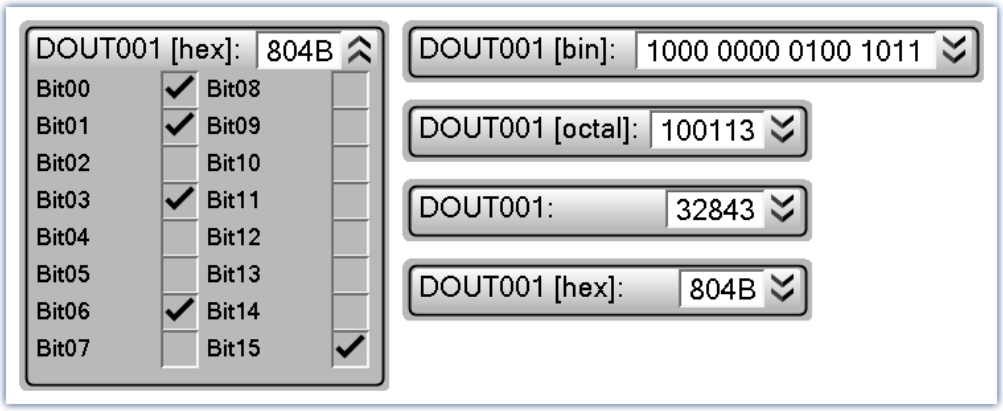
The Widget is linked with the selected cell of the column, which is highlighted by a **red frame**. The Widget displays the value of the selected cell. Additionally, the Widget can also be used as the entry for this cell.

13.7.7 DIO

The DIO-Widget is primarily designed for the DIO-ports and similar data types. But it can also be used with other data types, within limitations 1534.

Visual display is provided of a port's (a variable's) individual bits. Additionally, the complete value can be outputted in various formats: binary, octal, decimal or hexadecimal.

Below some examples:



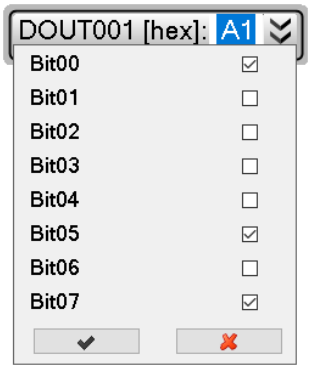
DIO-Widget: Examples of the various display varieties

When a variable is linked with the Widget, the display is automatically modified according to the respective data type. This means: an 8-bit port displays 8 bits and a 16-bit port accordingly displays 16 bits.

The Widget is subdivided into two regions. The **port-display** (top) and the **bit-display** (bottom). The bit-display can be expanded/collapsed by means of the Drop-Down button (at right). Using the option "*Style*", it is possible to hide the port-display, so that **only the bit** are still displayed.

Using the option "*Format*" you can define the **formatting of the port-display**: binary, octal, decimal or hexadecimal.

Changing the values

Type	Description
Changing **bits individually**	Place a check in the box next to the bit. The change is applied immediately in the variable
Changing the port or changing **bits together**	 *Port-input with bit-list* Select the port-value. The port can now be specified by user entry. Additionally, a list appears offering all bits which can be selected individually. The change is only applied once the input is confirmed.

Additional display options

Actions	Description
Formatting the port-display	The formatting of the port-display is defined, by means of the option "*Format*": binary, octal, decimal or hexadecimal.
Hiding bits	By means of the context menu: "*Hide selected bits*" or "*Arrange bits*" Notes: Hidden bits no longer affect the port-value. Example: The second bit (Bit01) in a 4-bit-port is hidden. The bits have the following values: **Bit** / **Value** Bit00 / 0 Bit01 (hidden) / 1 Bit02 / 0 Bit03 / 1 Since Bit01 is hidden, the following port-value is displayed: 4(hex) (although the complete port has a value of A(hex) - 100(bin) instead of 1010(bin). Even for the input, the bit has no significance. When a value of 2(hex) is entered, then Bit02 is set to 1. The others receive the value 0, except for the hidden bit. That obtains the value: 0110(bin).
Moving bits	By means of Drag&Drop or via the context menu: "*Arrange bits*"
Renaming bits	By double-clicking on the bit or via the context menu: "*Rename bit*"
Dedicated icons	By means of the options "*On*"/"*Off*", it is possible to specify dedicated icons for the checkboxes.

Bit-collection

Displaying multiple individual-bit-variables together (e.g. multiple virtual bits)

If no complete port is available but only individual bits, these can still be displayed together. Each variable is moved to the Widget via Drag&Drop, where the mouse position determines the variable's location. The variable can **replace an existing bit** or be **inserted between two bits**.

Note: In this mode, the port-value usually has no significance and can be hidden using the option "*Style*".

Display of other data types

<table>
<tr><th>Type</th><th>Description</th></tr>
<tr><td>32-bit Integer pv-variable</td><td>The display is limited to 32 bits (0-31). Thus all numbers can be displayed which the data type makes possible. Even negative numbers can be processed, since the integer-pv-variable is a signed integer.
<table>
<tr><th>Decimal</th><th>Hexadecimal</th></tr>
<tr><td>2147483647</td><td>7FFF FFFF</td></tr>
<tr><td>1</td><td>0000 0001</td></tr>
<tr><td>0</td><td>0000 0000</td></tr>
<tr><td>-1</td><td>FFFF FFFF</td></tr>
<tr><td>-2</td><td>FFFF FFFE</td></tr>
<tr><td>-2147483648</td><td>8000 0000</td></tr>
</table></td></tr>
<tr><td>Float-variable

e.g. display variable or float pv-variable</td><td>The display is limited to 23 bits (0-22). Negative numbers can not be precessed directly. Numbers which are too large can cause unexpected results.
<table>
<tr><th>Decimal</th><th>Hexadecimal</th></tr>
<tr><td>8388607</td><td>7F FFFF</td></tr>
<tr><td>1</td><td>00 0001</td></tr>
<tr><td>0</td><td>00 0000</td></tr>
<tr><td>-1</td><td>7F FFFF <- overflow</td></tr>
<tr><td>-2</td><td>7F FFFE <- overflow</td></tr>
<tr><td>8388608</td><td>00 0000 <- overflow</td></tr>
</table></td></tr>
</table>

13.8 Pages

13.8.1 Insert - Dialog / Report

The following ways to add a page are available:

- via the page title's "*context menu* 1231"
- via the menu ribbon
- via the tool window "*Page templates* 1229" by means of Drag&Drop or double-clicking

Ribbon	View
Panel-Design > Insert Page ( )	Complete
Panel-Design > Standard Dialog ( )	Complete

Menu item	Description
Insert Page	Selection window for creating a new Dialog- or Report-page is opened.
Standard Dialog	Creates a new Standard dialog page.

Note Replacing a page

When you use Drag&Drop to move a template from the tool window "*Page templates* 1229" to an empty page, it is replaced.

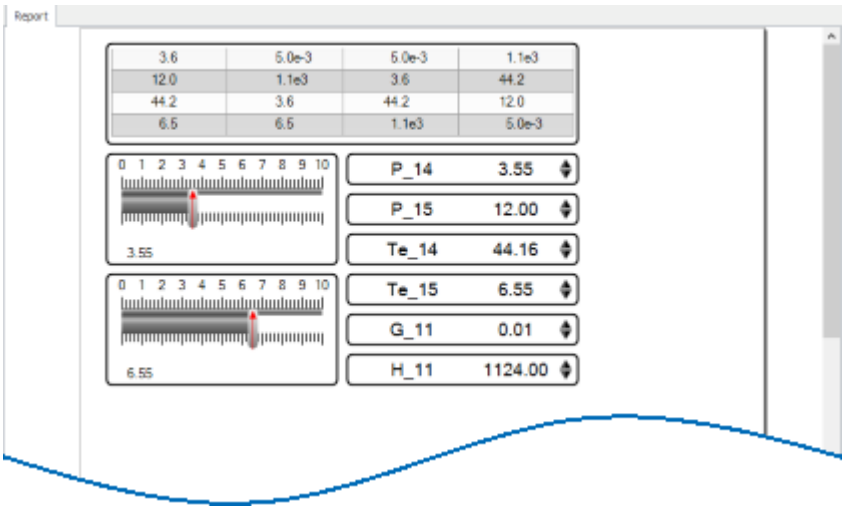
A page can be generated as a "*Dialog-page*" or as a "*Report-page*".

Report-page

The Report-page is optimized for printout. On the screen, one of its distinguishing features is that "margins" are displayed. By default, the pages are displayed in a fixed size.

Change the size and other page settings using the dialog: "*Page setup*" ("*Page Layout for Printing*" via the context menu 1231 of the Panel page's tab).

Menu item	Description
Page Layout for Printing	Opens the dialog: "*Page Setup*" Here, you can change the setup and printout settings for Report-pages (including paper size, source, orientation, margin).



Example of a Report-page

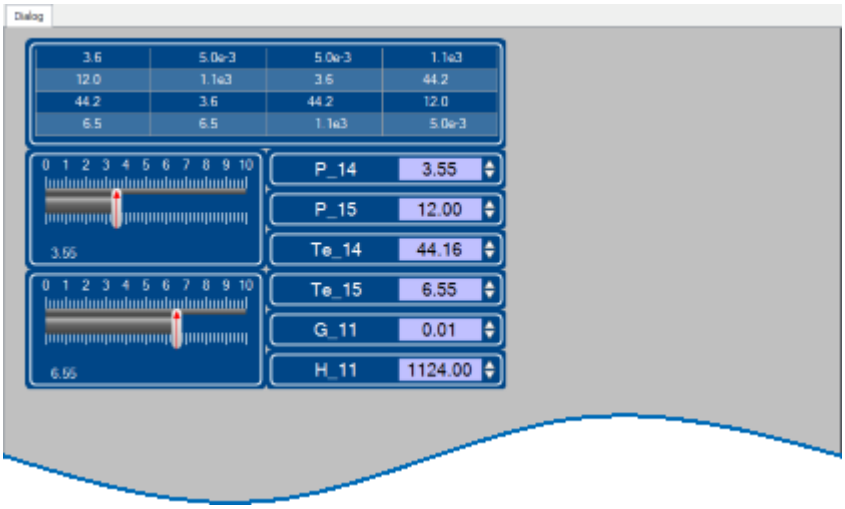
To **print out the page in the page format set**, select the menu item: "*Print* 1206" or the command: "*Print Panel page*". By means of **PDF-export** (in the menu ribbon or the corresponding command), you can generate a corresponding PDF in that format.

Detailed information is presented in the chapter: "*Page printout or print to PDF* 1548".

Dialog-page

The dialog-page is optimized for screen display. By default, the pages are **generated in maximum size**.

Change the size via the Page Properties 1539.

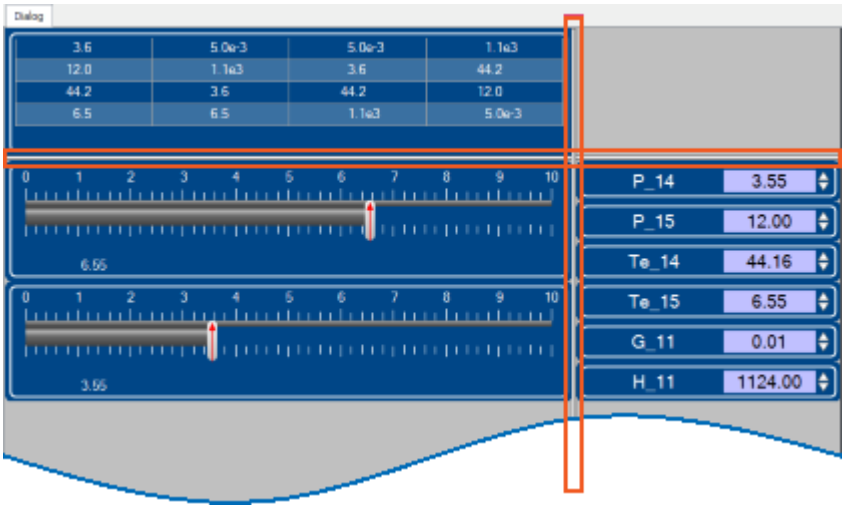


Example of a Dialog-page

After the program is started, the Panel contains a default Dialog-page.

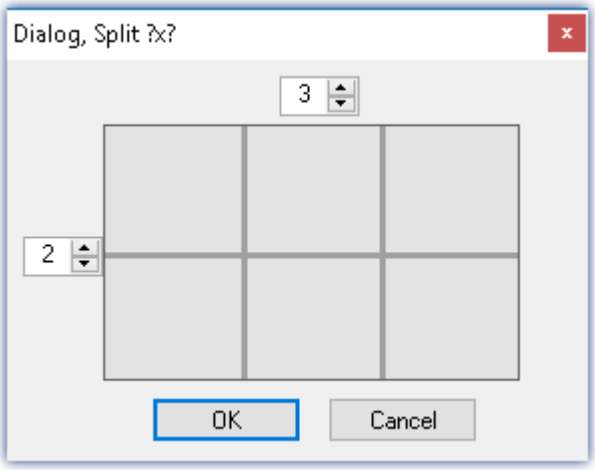
Divided dialogs, e.g. "Dialog 2x2"

These dialogs are partitioned into regions by movable sliders. The size of the docked 1244 Widgets adjusts to the position of the sliders.



Example of a partitioned Dialog-page
Sliders are framed in red

Using the choice "*Dialog, ?x?*", you can personally define the partitioning:

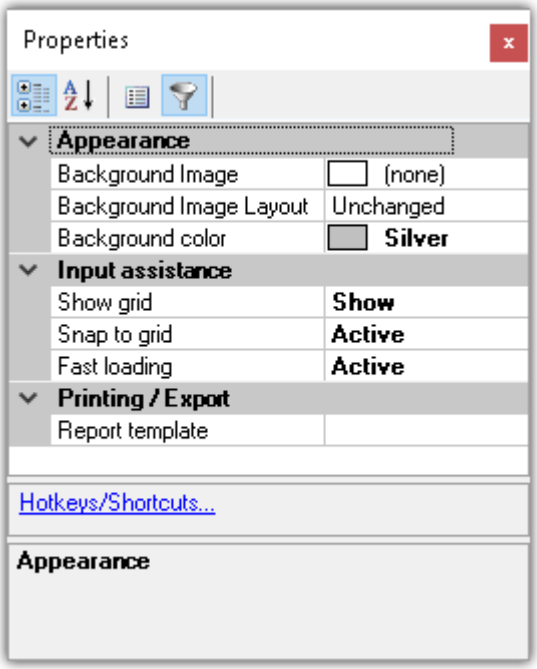


Partitioned format

13.8.2 Properties - Pages

With this tool window, you can set the page's properties. The Panel must be in Design Mode in order that the properties can be seen/edited.

Right-click on an empty area of the page and select from the context menu the command ***Properties*** (to see all the properties, click on the symbol for "All properties"):



General

Property	Description
Name	Name of the page. Each page has a unique name.
Title	The title displayed in the page's tab. Default: <empty>; in that case, the page name is used. If the title is changed, the page name is no longer used.
Multilingual title	The title can be pre-defined for multiple languages 1555.

Data linkage

Property	Description
Script name	Panel-scripts are associated with Panel pages. This can be defined directly by means of the script-properties. If no Panel page is selected, the script can be associated with the page by means of this property (Script name).

Layout

Property	Description
Page size	Size of the Panel page (<Width>; <Height>) The page size adjusts itself to the setting for the Zoom 1544 and to the font size (dpi) in Windows. Example: Suppose the current size is "1200; 500". After zooming, or changing the Windows font size to 125 %, for instance, the new value is "1500; 625". **Advantage:** When a larger text scaling value is set, the texts and widgets on the Panel page are thus larger and more legible. **Disadvantage:** When the text scaling is changed, the Panel page becomes larger/smaller and may no longer fit in the display area. If appropriate, have the page size adjusted automatically 1541, or use the Zoom-function to compensate for the font size.
Width	Width of the Panel page
Height	Height of the Panel page
Page	Pages can be shown or hidden (not hidden in Design mode).

Appearance

Property	Description
Background image	An image can be displayed on the page. In order to select an image, click in the cell and next confirm by clicking on the button `[...]` on the right edge. In order to delete an image, delete the text "`[Bitmap]`".

Property	Description
Background image layout	Placement of the background image in reference to the page size: • Unchanged: The image is placed in the upper left corner in original-size. • Tile: multiple copies of the image are displayed side-by-side. The entire page is filled up. • Centered: The image is displayed centered and in original size. • Stretch: The image is displayed centered over the entire page (stretched) • Zoom: The image is displayed centered in optimum size (zoomed). The aspect ratio remains intact.
Background color	The page's displayed background color.
Background plain colored	Display of the background color: • Solid: The background color is displayed over the entire page. • Color gradient: The background color's shade becomes increasingly pale in the downward direction. The page's upper edge conforms to the background color. This progression is not visible for all colors.
Skin	You can choose among multiple pre-defined color schemes (Skin). The color scheme influences the foreground and background colors of the Widgets and pages.

Printing / Export

Property	Description
Report template	Define a Report-page as a print template. The print template must contain the Widget "*Print Preview*", which is filled in upon executing Print/Export. See: "*Page printout or print to PDF* 1549".

Input assistance

Property	Description
Grid size	The distance between the grid points.
Show grid	Hides/shows grid (see "Aligning to grid 1543")
Snap to grid	Activates/deactivates the alignment of the Widgets to the grid. • Activated: New Widgets or changes to the Widgets (size and position) are aligned automatically to the grid. • Deactivated: Widgets can be positioned freely.
Fast loading	• Activated: Going to pages for which the option "Fast loading" is activated happens quickly since the page's setup is retained in memory (default selection). • Deactivated: Going to pages for which the option "Fast loading" is deactivated happens more slowly since the page's setup needs to be loaded first. This option is recommended for elaborate pages in order to conserve memory resources.

13.8.3 Adapt/automatically adapting the size of pages and Widgets

You can adjust the size of the pages and Widget automatically.

- To do this, open the context menu of the Panel page
- Select *Adapt Page Size*:

Adapt Panel page to window size (or for all pages)	The Panel page takes up the space available.
Adapt Panel page to window size (keep aspect ratio) (or for all pages)	The Panel page retains the aspect ratio, but adjusts its height/width to the available space.
Adapt Widgets to the size of the Panel page (or for all pages)	The Panel page retains its page size. The Widgets take up the space on the page currently available.

Note

Notes on automatically aligning to the grid

- If the Widgets are automatically oriented to the **grid**, this is taken into account when adjusting the size.
- Among other effects, this can lead to the current aspect ratio of the Widgets to no longer persist after the adjustment.
- On every page, there can be a resultant discrepancy of ±1 grid increment. If this is not desired, first deactivate the grid orientation 1543.

all pages

The functions can be applied either to the current page or to all loaded pages simultaneously.

The following pages are not affected by the functions applied to all pages:

- all pages in which the property "fast loading" was deactivated, unless they are already open currently.

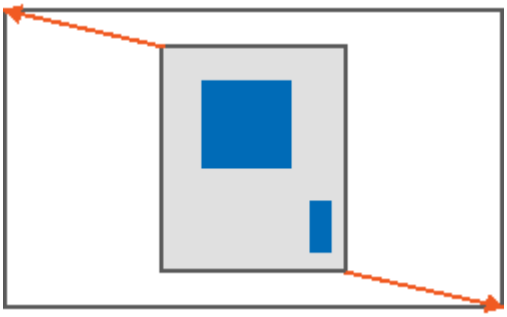
Use of the Zoom function

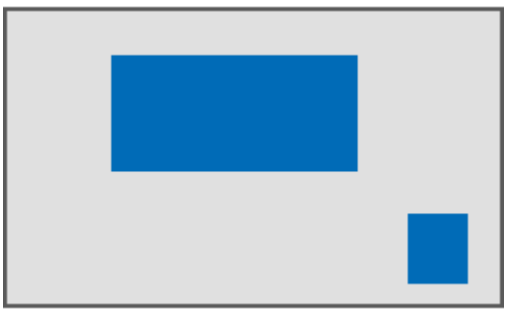
The Zoom-factor is deactivated whenever the Panel page is adjusted to the window size.

Adapt Panel page to window size

The Panel page takes up the space available. This page size remains constant, even if the program window size is changed.

- The aspect ratio of the individual Widgets does not remain intact (observe the note above about the Grid)



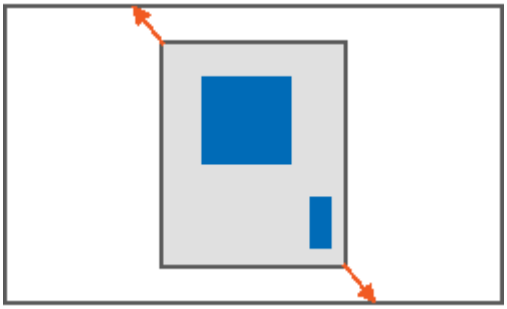


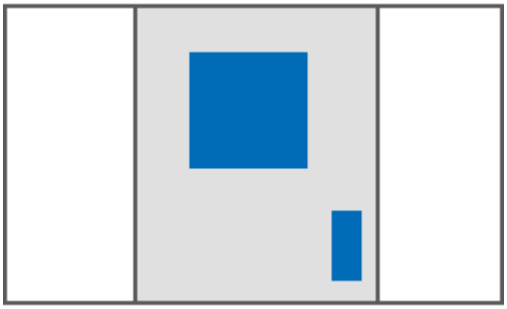
Result

Adapt Panel page to window size (keep aspect ratio)

The Panel page retains the aspect ratio, but adjusts its height/width to the available space.

- The aspect ratio of the individual Widgets remains intact (observe the note above about the Grid).



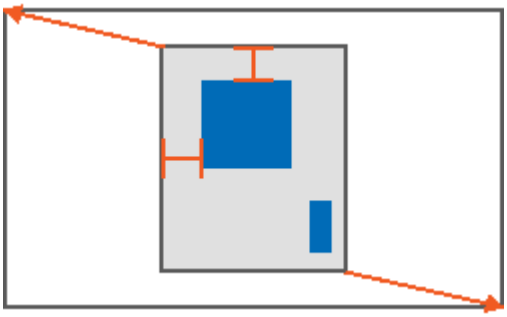


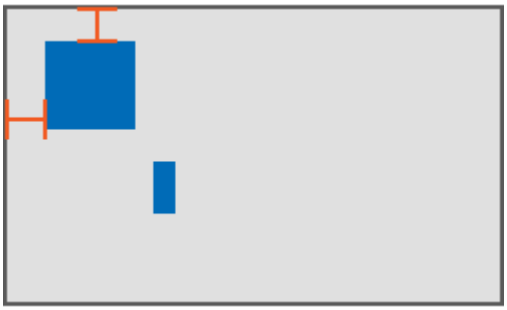
Result

Adapt Panel page to window size (keep Widget size)

The Panel page adapts to the size of the currently available space by only adjusting the page size. This is like when you adjust either the right or the bottom page margin.

- The sizes of the individual Widgets thus remain intact.



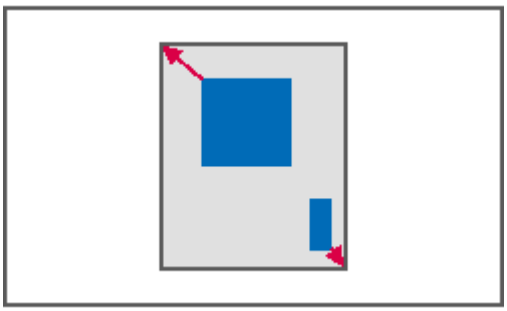


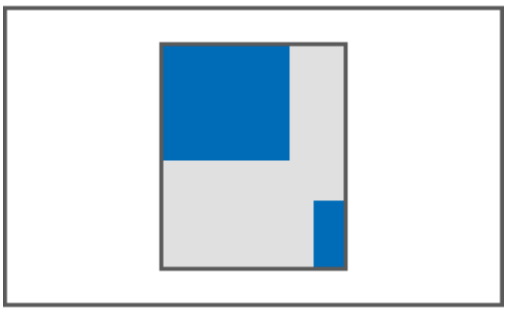
Result

Adapt Widgets to the size of the Panel page

The Panel page retains its page size. The Widgets take up the space on the page currently available.

- The aspect ratio of the individual Widgets is not retained (observe the note above about the Grid).





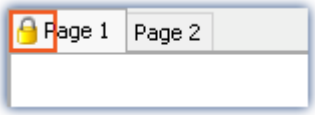
Result

13.8.4 Locking/unlocking a page

You are able to completely protect a page from being changed (also in "Design Mode"). Lock/Unlock the page via the associated menu item or via the context men of the Page-tab.

Ribbon	View
Panel-Design > Lock/Unlock Page ( )	Complete

The associated symbol appears next to the page-name.



The locking/unlocking **applies only to the current page**.

In contrast to the deactivated "*Design Mode* ", not even Widgets can be placed on a locked page by means of Drag&Drop.

Note No protection against deletion of the page

Locked pages can also be deleted. All that is protected against changes is the content.

13.8.5 Aligning to grid

To simplify the positioning and moving of Widgets on the page, the grid can be used. The Widgets can be aligned to the grid. The distance between grid points can be adjusted in the page's Properties (*Page Properties > Grid size*).

To activate/deactivate the grid or to align all Widgets to the grid, open the Panel page's context menu .

- Select *Grid* ( ):

Action	Description
Show Grid	Hides/shows grid (also adjustable via the page's settings)
Snap to Grid	Activates/deactivates the alignment of the Widgets to the grid (also possible via the page's settings). • Activated: New Widgets or changes to the Widgets (size and position) are aligned automatically to the grid. • Deactivated: Widgets can be positioned freely.
Realign to Grid	All Widgets are aligned to the grid (even when *Align at Grid* is not activated)

13.8.6 Zooming Panel pages

For better visibility of Panel pages, for instance, you can zoom in on the display.

- To do this, open the context menu of the Panel page's tab.
- Select *Zoom* ( ) or *Zoom (all Pages)* ( ):

Action	Description
50%	Panel page is zoomed to 50%. Image ratio: 1:2
100%	Panel page is displayed in original size. Image ratio: 1:1
>100%	Panel page is zoomed to 200% or 400%. Image ratio: 2:1 or 4:1
Zoom	Arbitrary zoom-factor

Note all pages

The zoom can be applied either to the current page (*Zoom*), or to all loaded pages (*Zoom (all Pages)*) simultaneously.

The following pages are not zoomed with the function "*Zoom (all Pages)*":

- all pages in which the property "Fast loading" was deactivated, unless they are already open currently.

13.8.7 Fullscreen

Panel pages can be displayed on a monitor in a full-screen mode. The Fullscreen-mode offers certain advantages:

- **larger display area**, since the menu and tool windows are covered over
- **more protection** against unintended changes, since the fullscreen mode must be exited explicitly

Folgende Vollbild-Arten bietet imc STUDIO

- Display of a page on a monitor (no "genuine" fullscreen, without protection)
- Panel in fullscreen mode, so that the imc STUDIO user interface is no longer accessible (imc STUDIO Professional and higher)

Display of a page on a monitor (no "genuine" fullscreen, without protection)

Panel pages can be displayed in **fullscreen on individual monitors**. The imc STUDIO user interface can still be operated. However, the Panel page is in the foreground (so if there is only one monitor, it covers the user interface). Thus, you could **observe the Sequencer on the main monitor**, while the **second monitor shows measurement readings** on a Panel page in fullscreen mode.

When a Panel page is displayed on a **second monitor**, this page layout is **saved in the experiment** and restored upon subsequent loading. If the monitor is no longer present upon loading, the page appears again, embedded in the Panel.

- To do this, open the context menu of the Panel page tab.
- Select "*Show Page on Monitor*" ( ):

Action	Description
Page on monitor	• embedded: The page is displayed in the plug-in's Panel main window (default selection) • <Monitor name>: The page is displayed in fullscreen mode on the monitor selected. Independent of the opened main window.

Uses	Description
Automatic assignment by command: Browse in workspace	When the command "*Browse in workspace* 1750" is executed, imc STUDIO jumps to the selected page. When a **Panel page is defined as a target**, then it can also be **displayed in fullscreen mode** on any desired monitor. In this case, imc STUDIO does not change the main window.

Note — No comprehensive protection

This mode **offers no protection against changes** to the imc STUDIO user interface. The button for exiting the fullscreen can be hidden. However, it is possible to exit Panel by typing the key combination `alt+F4`. Or when using multiple screens, the program can be relocated to the second monitor.

Hiding the Exit-button in this fullscreen view

The button is hidden when the following conditions apply:

- The user who is logged on does not possess the right to close the fullscreen view.
- The Panel is already in fullscreen view (see the possibility 2).
- No tabs are shown in the fullscreen mode.

Panel in fullscreen mode (imc STUDIO Professional and higher)

The Panel is displayed in fullscreen mode. The imc STUDIO **user interface is hidden** and can thus no longer be operated or accessed (excepting the Panel pages).

Ribbon	View
Start > Panel Fullscreen ( )	Standard
Panel-Control > Panel Fullscreen ( )	Complete
Panel-Design > Panel Fullscreen ( )	Complete

Note

Protection against changes

This mode **offers protection from changes** to the imc STUDIO user interface. The button for exiting the fullscreen mode can be hidden if one possesses the pertinent access rights. Without such rights, it is not possible to exit the fullscreen mode.

By means of the User administration, you can **prohibit exiting Fullscreen mode**.

Right 154: Panel fullscreen: Button "Exit fullscreen mode"

Navigation through the pages

In this case, show the tabs for the individual pages. Or use buttons and action, in order to navigate to the Panel pages. E.g., you can skip between the individual pages using the commando: "*Browse in workspace* 1750 ".

Moving the fullscreen view to a different monitor:

The fullscreen can only be moved to a different monitor via Windows-shortcuts. Use the keyboard combination `Windows + Shift + left/right`.

Uses	Description
Automatic launch in Fullscreen mode	Using the command-line parameters 35, you can start an experiment directly in Fullscreen mode. Thus, you can **start the desired experiment immediately**. And you can **prevent unauthorized modifications**.
Automatic Fullscreen mode by means of the command: Browse in Workspace	When the command "*Browse in Workspace* 1750" is executed, imc STUDIO jumps to the selected page. If the **Panel is defined as the target**, it can be **displayed in fullscreen mode**.
Combination of the Fullscreen variants	If the Panel is in Fullscreen mode, you can additionally display more Panel pages on other monitors.

Title Bar - Overview of functions

The fullscreen window has its own title bar (menu). A **variety of functions** can be called from this menu (most of the functions are only available in the actual fullscreen mode).

	The menu includes various buttons. It can be **expanded and moved**.
	By means of this button, you can expand the menu to reach **additional functions**.
	Using this button, you can **move the menu**, in order to access **elements located behind** it. When you move the mouse pointer over this button, you obtain a special cursor (◀•▶). While holding down the mouse button, you can change the menu's position within the top edge.
	By means of this button, you can exit **Fullscreen mode**. When the menu is expanded, this function is located at this button: Exit Fullscreen Mode.
	This button **minimizes the program**.
	Thus button **closes the program**.
Show Tabs	These checkboxes control whether the tabs for the Panel-pages are hidden or shown.
Data Browser	This button Show Data Browser causes the free-floating Data Browser 1214 to be shown.

Additional functions can be accessed via the context menu or via the button in the expanded menu.

Menu item	Description
Login 151	Logs in a user
Logout 151	Logs a user out
Design Mode	• Activated: the Panel page can be edited, the Widgets can not be operated. • Deactivated: The Panel page can not be edited. The Widgets can be operated.
Info	Opens the dialog with version information
Product Configurator	Opens the dialog for product configuration

Note — Notes on modifying the fullscreen size

The full screen has **mode available space**, for which reason you can **increase the page size**. If you already have finished pages and you wish to **magnify these pages**, then in Fullscreen mode use the function: "Adapt Panel page to window size".

You may need to first deactivate the grid alignment, since there may otherwise be unintended displacements. For more information, please see the chapter: "*Adapt/automatically adapting the size of pages and Widgets* 1541".

13.8.8 Page printout or print to PDF

Panel pages can be printout out as hard copy or to a PDF-file. imc STUDIO offers a variety of possibilities for this purpose: via the menu ribbon or via the commands ("*Print Panel page* 1735" / "*Export Panel page* 1735").

Ribbon	View
Panel-Control/Navigation > Print/Print Preview (/)	Complete
Edit > Print/Print Preview (/)	Complete
Home > Print ( )	Standard

Ribbon	View
Panel-Control/Navigation > As PDF (Export) ( )	Complete
Home > As PDF (Export) ( )	Standard

The respective menu item allows you to **print/export only one page at a time**.

If you wish to keep **multiple pages** together, then it is best to use **the respective command**. You can, for example, place this on the Panel page and execute it from there. You can hide its button when printing the page out.

Note Font in the printout

"Font in the printout" is a topic which we can only improve step-by-step. imc STUDIO passes the information about your page to the default printer set up for your system. This then produces either the PDF or the printout. For this reason, it is hard for us to influence the results, which strongly depend on the printer driver.

If you experience problems, please send our tech support 8 the **exact identification** of your system's **default printer**. Switching the default printer temporarily may be helpful.

PDF with vector-elements

In response to the menu item "*As PDF (export)*", a PDF is generated on which graphics are displayed. If the graphics quality is not sufficient, it is possible in some cases to generate vector elements in the PDF by means of a PDF-printer. This, too, depends on the printer driver and not necessarily every PDF-printer is able to do it. Additionally, not all Widget support this function. Some will still embed the image as a graphic.

Therefore, please test this technique with your printer driver before choosing it.

Report-pages - Adjusting the size

The Report-page is optimized for printout. You can make changes to the size and other page settings by means of the dialog: "*Page Setup*" ("*Page Layout for Printing*" via the context menu 1231 over the tab for the respective Panel page)

Menu item	Description
Page Layout for Printing	Opens the dialog: "*Page Setup*" Here, you can change the setup and printout settings for Report-pages (including paper size, source, orientation, margin).

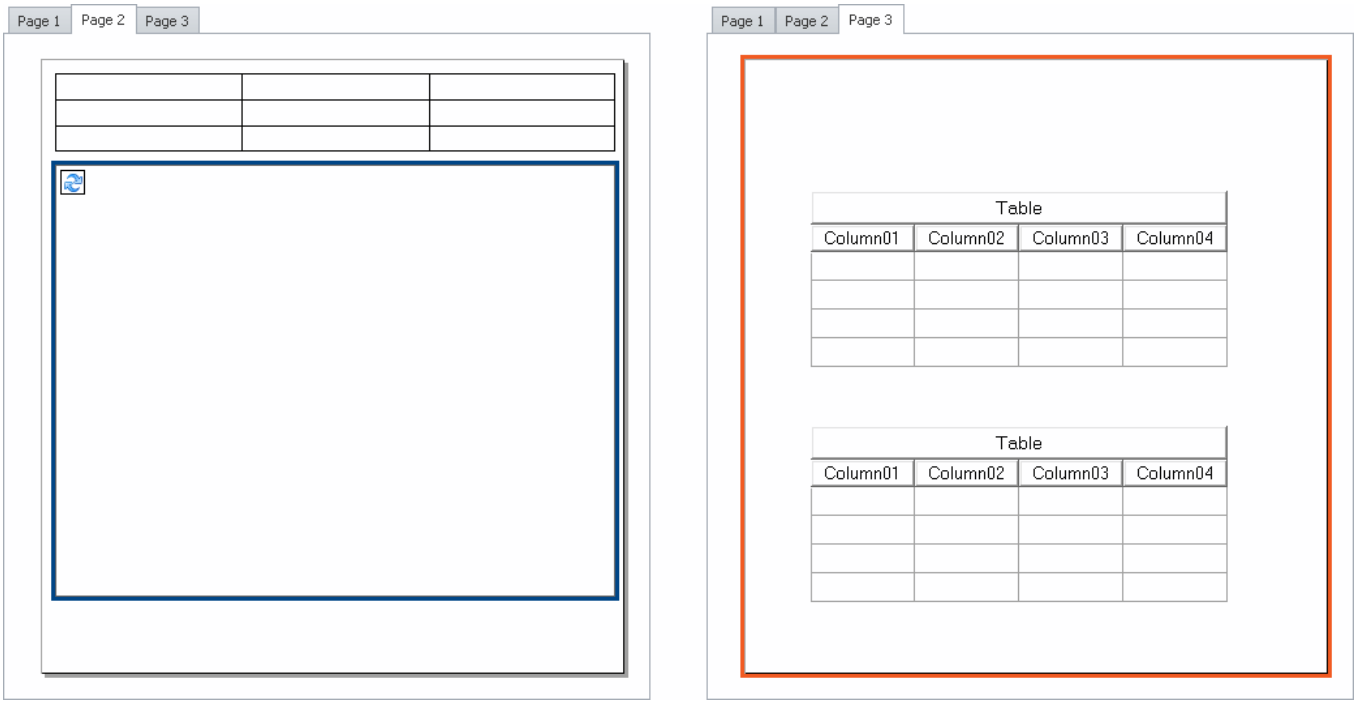
Report template - Header

For purposes of export or printing a dialog page as a report, it is possible to define another report page as the template. This "template page" can be set up as a kind of "page header" and be filled with multiple report pages upon printout. E.g. texts and logos which are to appear on every page.

On this report page the Widget: "*Print preview*" appears.

In the properties of the dialog page, select the report page under "*Report template*".

If the dialog page is now printed, the Widget on the report page is filled with the page to be printed.



Example: When it is printed out, the Panel page on the right side will appear inside of the frame appearing within the page shown on the left side.

13.9 Variable linkage

To link a Widget with a variable, there are two methods:

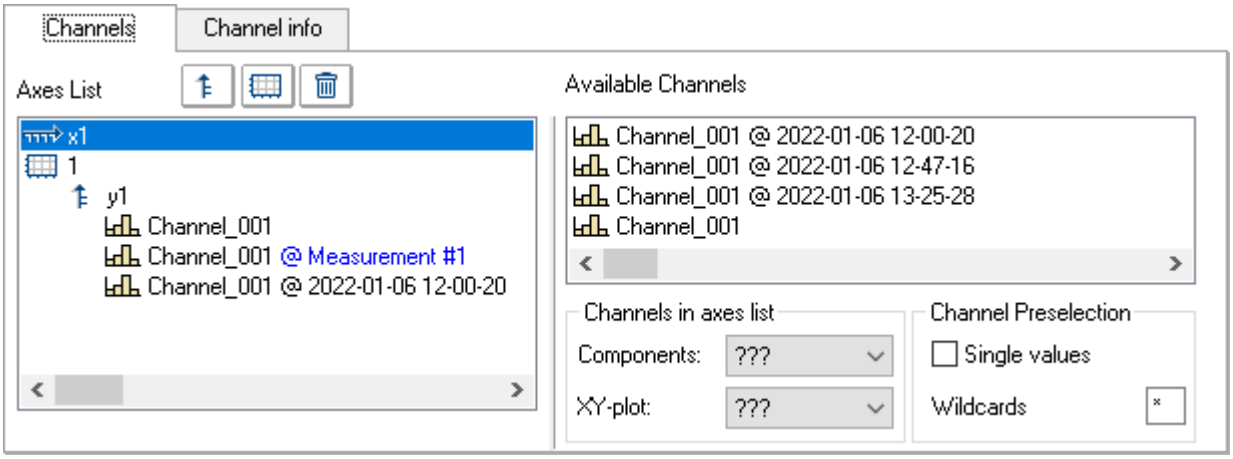
Variable linkage by Drag&Drop 1551	**Move the variable** using Drag&Drop from the Data Browser 1214 **to the Widget** or the Panel page.
Variable linkage by Widget Properties 1551	Open the "**Properties** 1245" window of the Widget and there click on "***Variable***".

Even **without existing variables**, it is possible to set up a linkage (see: "*Variable linkage to not yet existing variables* 1553").

Directly connecting with a measurement, or changing measurements at the click of the mouse (measurement number)

You can link a variable either by its **fixed name** or its **symbolic name** with a Widget.

Fixed name	**<VariableName>@<Measurement Name>** e.g. **Channel_001** In this way, the Widget always indicates the current measurement. or **Channel_001@2022-01-06 12-00-20** in this way, the Widget is always linked with this concrete measurement
Symbolic name	**<VariableName>@Measurement#<Measurement number>** e.g. **Channel_001@Measurement#1** Contains the name of the variable followed by a measurement number. The measurement number can be assigned variably with the Data Browser. This enables you to **compare measurements**. After measurement, you can select the saved measurement (it is then assigned the the number "1"). Subsequently, **all Widgets** show the variables belonging to the saved measurement.



Properties of the curve window. Example: three times the Channel_001.
1. Current measurement
2. Selected measurement (Measuremnt number 1)
3. Saved measurement with the defined name

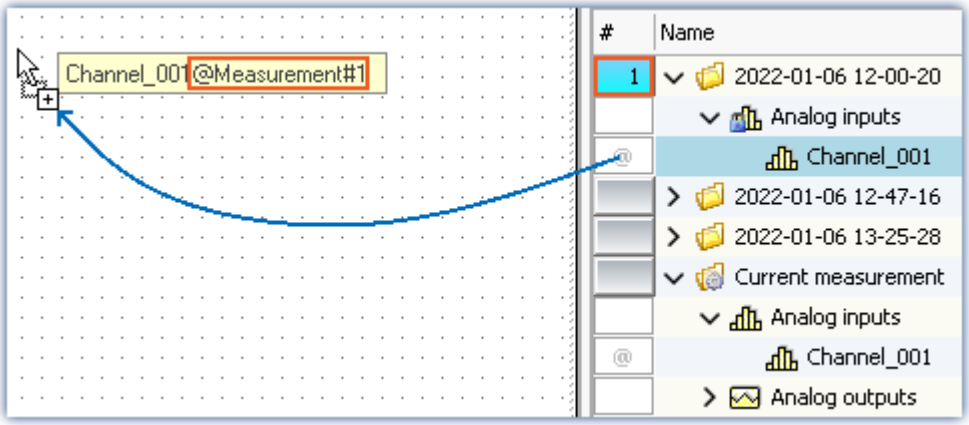
Browsing data

Once you have linked a Widget with a **symbolic name**, you can view the various measurements by simply clicking in the Data Browser (see: "*assigning a measurement number* 1220").

Variable linkage by Drag&Drop

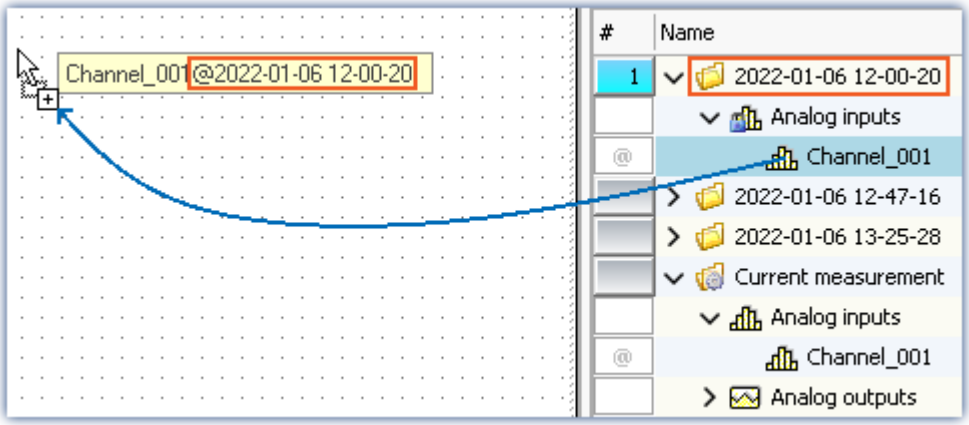
If you have opened a measurement in the Data Browser, the **"@" Symbol** (@) appears in the **Numbering column**.

- To link a variable with a Widget **via a measurement number** (symbolic name), **use Drag&Drop** to move the variable **from the "@"-symbol** to the page.



Variable linkage with symbolic name.
Example: "Channel_001@Measurement#1"

- To link a variable with a Widget by its **fixed name**, move the variable to the page **from the name column using Drag&Drop**.



Variable linkage with fixed name.
Example: "Channel_001@2022-01-06 12-00-20"

- Release the mouse button over the Panel page and select the Widget with which the variable is to be displayed,
- or release the mouse button over a Widget. The Widget is linked to the variable.

Variable linkage by Widget Properties

Note

Only for Widgets having the property "*Variable*". For the curve window, please see the separate documentation 1265.

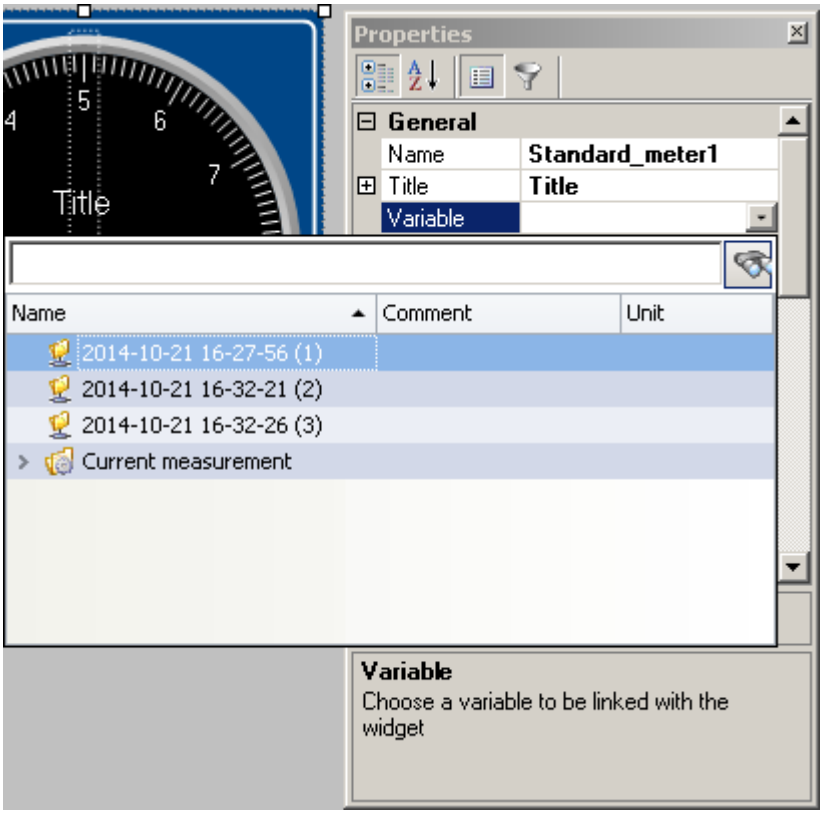
You can establish the link with the variable by means of the Widgets's properties.

- Open the properties of the Widget
- There, click on the entry ***Variable***

You can use the dropdown list or enter a name for the variable.

Variable linkage by dropdown list

- Click on the dropdown button ( )
- A Data Browser opens

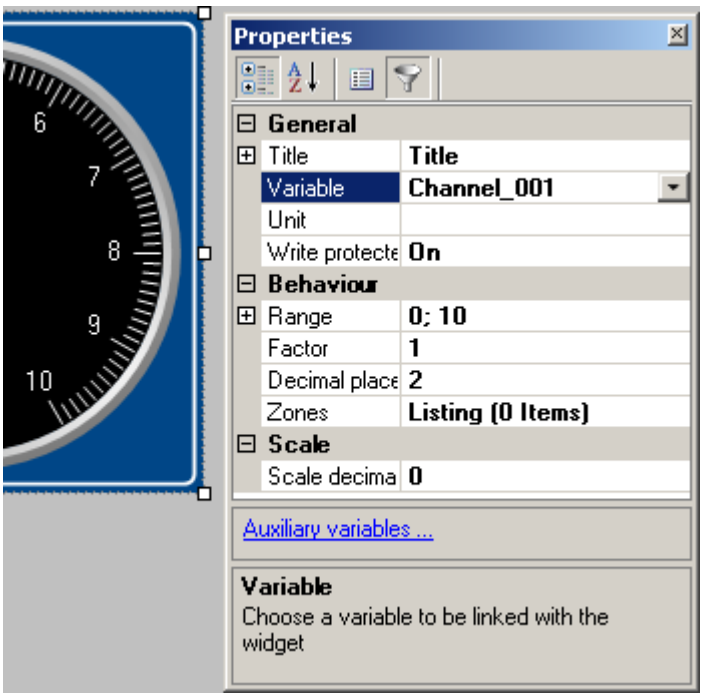


- Select the desired variable

The Widget is now linked with the variable.

Entry of the variable's name

- Enter the variable's name in the text box.



The Widget is now linked with the variable.

By means of the property "*Variable*", you can also set up a link using a **fixed name** or a **symbolic name**:

Fixed name Current measurement	e.g. **Channel_001**
Fixed name	e.g. **Channel_001@2022-01-06 12-00-20**
Symbolic name Measurement#x	e.g. **Channel_001@Measurement#1**

Variable linkage to not yet existing variables

You can even design a Panel page before any variables exist.

Using the property "*Variable*", you can establish a future variable linkage. *

- Enter the variable's name in the text box

The Widget is linked with the non-existing variable.

Since the variable doesn't exist, a yellow warning triangle appears on the Widget.

As soon as the variable is created, the Widget displays the variable's value and the warning triangle disappears.

* (Only for Widgets having the property "*Variable*". See Variable linkage by Widget Properties 1551)

13.10 Information and tips

13.10.1 Tips for the curve vindow

Curve window - Multiple-triggered signals

Signals which cause multiple trigger releases generate "*Events*" which can accumulate to a data volume which overloads the system.

Tip	Description
Last Event	Use the curve window menu item "*Configuration*" > "*Events, Segments, Periods*". In the dialog which appears, select the page Events. Enter how many of the most recent events you really wish to have displayed. In that case, the curve window plot will always be drawn at the same speed.

Curve window Updating

Many curve updates per second during a running measurement make it easy to view the data plotting.

Tip	Description
Fast interface	The Ethernet connection needs to be as free of interference as possible.
Small segment	The curve window requires longer to plot the data if the **number of data displayed** is large or if the **curve window itself is simply very large**. If the **number of channels displayed** is reduced, the speed is increased. The time needed for drawing the plot also depends on your graphics card's power.
Fewer channels transferred to the PC	If data from fewer channels are transferred to the PC, the performance improves. The other channels are saved to the internal device hard drive, for instance.
Monitor channels	For viewing purposes, channels whose data points have already been averaged or simply more slowly sampled channels (monitor channels) can be used. Data storage can then be performed on the fast-scanning channels.

13.10.2 Jiggling display and heavy burdening of the processor

Virus scanner

The presence of many channels can burden the PC if a virus protection program scans the data captured. We strongly recommend removing imc STUDIO from the virus scan.
See: Recommended virus scanner settings 41

Refresh rate of Widgets

Placement of many Widgets burdens the PC's capacity. You may need to adjust the refresh rates of some Widgets if, for instance, the curve in the curve window shakes.
See "*Properties - Widget* 1246"

13.10.3 Multi-lingual text input

In many cases you can pre-define titles and zone-texts for a variety of languages. Depending on what the operating system's language is, the respective language variant is displayed. If any text is not defined for any language, the respective English text is displayed (fallback-language). The English text is always automatically filled first.

Example

You define the title of the Panel page in German, English and French. If you start the experiment on a German system, the German text is displayed; on an English-language system the English text, etc.

When you start the experiment on a system not set to any of the three languages, the English text is displayed.

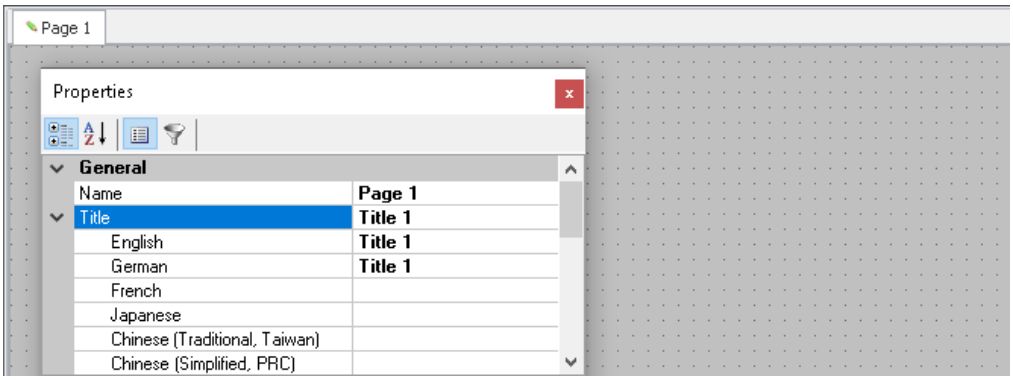
Option: Deactivating multi-language text input

You can, if desired, activate multi-language input: "*Panel*" > "*General Options*".

Option	Description
Multi-lingual text input	Enables textual input for computers with other language settings. For example, in this way, Panel pages can be pre-configured for different language settings. If the language set is configured for the respective text, it is displayed. Otherwise, the default language: "English" is displayed.

Defining titles and automatic adaptation

The various languages are usually only shown once the **Properties-filter** is deactivated. Display all of the properties ( ). Expand the Title-property. In the main branch, you see the **currently displayed title**. Below that are all available **languages and the content**.



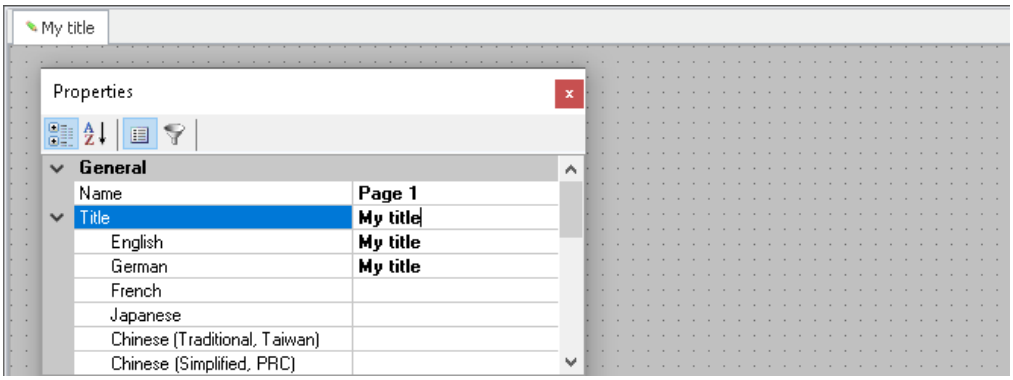
Example - Title of the Panel page

In most cases, the operating system's language and English are defined, unless English is already the operating system's language.

For the following example cases, the operating system language is (1): "*English*" (2): "*German*".

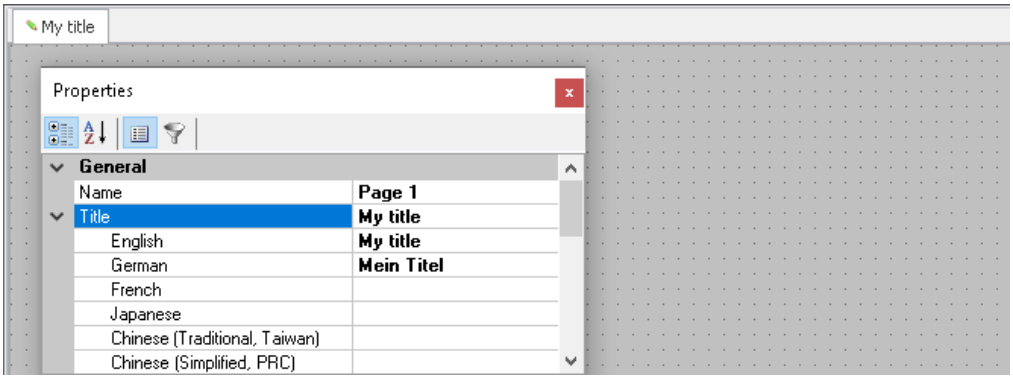
Case 1: You change the title in the main branch (all existing titles are the same):

The following is implemented: (1) the English entry is set to the new title | (2) English and German entries are set to the new title.



Case 2: You change the title in one language (e.g. German):

The following is implemented: German systems now display the new title.



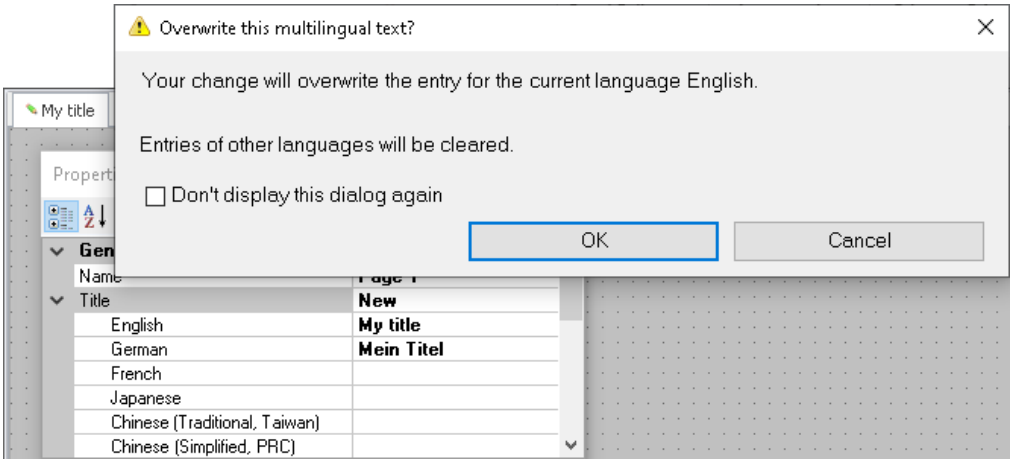
Fall 3: You change the title in the main branch while the titles in the other languages are all different:

The following is implemented: An error message is posted.

(1): "*Your change will overwrite the entry for the current language English. Entries of other languages will be deleted.*"

(2): "*Your change will overwrite the entry for the current language German and the entry of the default language English. Entries of other languages will be deleted.*"

The languages stated are assigned the new title; all others are deleted.



14 Automation - Real time test automation

This chapter describes the use and operation of "imc STUDIO Automation".

Automation enables **implementation of test stand automation** meeting real-time capability requirements. This implementation is achieved by uniting imc Online FAMOS with PC-based design- and visualization components. Internally, the visualized elements are automatically converted to imc Online FAMOS-code. The code is not accessible for viewing.

Automation is an aid in **defining and implementing state-based processes**. Toward this end, **logical states** are named and **associated with an action** by means of Drag&Drop. Examples include setting a digital output, browsing to a certain Panel page, running a signal plot output etc.. You generate states which then govern how your system runs.

A **variety of control structures** (For, Do, While, If) are available for your use. For the purpose of evaluating measured data, you can insert imc FAMOS sequences (see "*System requirements*"). In order to be able to **observe results at the PC**, or to be able to intervene in the procedure, you can use the Panel.

The Automation runs independently of the PC, like any normal measurement. **Connection** between the PC and the imc device **is not required**. The connection is only needed for configuration purposes. Interfaces with the PC as well as imc FAMOS-sequences always require a connection with the PC.

System requirements

Measurement device requirements
The same hardware prerequisites apply as for imc STUDIO Setup.
An extra device option is required: imc Online FAMOS Professional

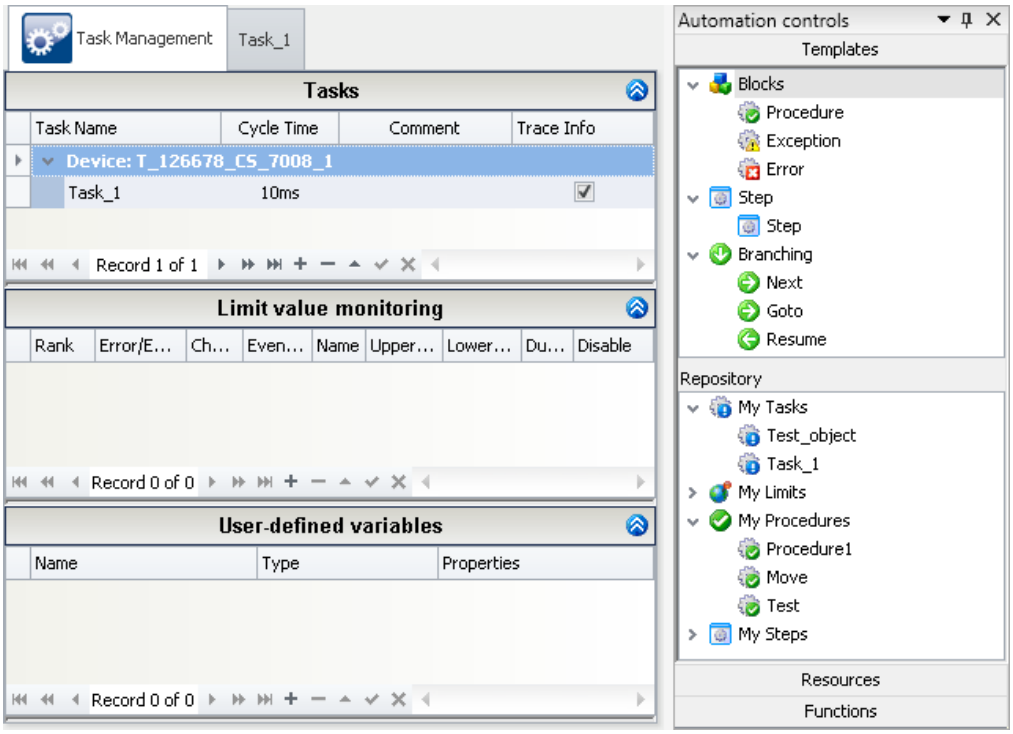
Optional products
imc FAMOS: The product is sold separately. The prerequisite is installation of an imc FAMOS version of 2021 or higher, as a **Runtime**, **Professional** or **Enterprise** Edition (see TD: "*Additional imc Software Products*").

Chapter overview

Synopsis	Section
What should I do first? How does the task begin? What components are there?	Getting started 1559
Creating a "*Task*". What is a cycle time? Monitoring of signals. What kinds of variable types can be created?	Task Management 1573
What can this be used to make? The various kinds of "*Blocks*" available and their purposes.	Task Editor 1584
Creating the "*Steps*". Things to keep in mind. What elements can be incorporated into the "*Blocks*"? How the procedure responds to parallel structures/tracks?	Block Editor 1586
The properties of the elements. Blocks, steps, states, loops and branching. Interaction with the PC: commands and imc FAMOS-analysis.	Templates - Elements for the Editors 1588
Functions for special applications. Signals, tolerances, timers, ...	Functions 1611
Various sample applications for practice purposes.	Tutorial 1628

14.1 Getting started

Structure of the user interface



Components of the Automation

In the picture at left, you see the "*Task Management* 1573" in which the **tasks** (flow controls) **are created**. In the tool window "*Automation controls* 1569" at right, process elements, functions and access to the input/output parameters (resources) are available.

Definition of terms

Term	Description
Tasks 1573	A task refers to the **complete processing** of a variety of commands in real time. Within a task, it is exactly **defined what actions are to be performed**. You may generate up to **five Tasks** 1573 **for a device**.
Initialization 1590	Required in order to bring the testing system into a **defined initial position upon starting**.
Terminate 1590	Used to bring the testing system into a **defined end position** and to conclude the measurement.
Procedures 1590	Procedures define the **actual course of the automation run**. A procedure may contain a variety of elements. In contrast to the initialization and the termination, there can be **multiple procedures**. At the end of one procedure it is possible to use a "branching" to switch over to a different procedure, or to repeat the same procedure.
Exception 1591	An exception is used in order to, for instance, **remedy minor errors** occurring during the automation **without canceling the automation**. If an exception occurs, the automation is interrupted and "exception handling" is performed in order to resolve the error. Afterward, the automation can be resumed.

Term	Description
Error 1591	An error is a **major error**, causing **canceling of the automation**. The "error handling" is used in order to bring the testing system into a **defined end position** and to stop the measurement. The **termination is not performed**.

14.1.1 How to create a operating plan?

In order to create a complete routine for an assignment, it is necessary to first **compose the outline of an operating plan**. Only then is it advisable to focus on individual items of the plan and to specify more exact conditions and steps.

For this purpose, the Automation uses a structure which allows the **larger blocks** of steps to be planned **first** and for the **details** to be designed later **step-by-step**.

Creating a plan

First create the complete tasks using the blocks and then proceed in more detail step-by-step.

Structure	Description
Task 1573	A Task represents a complete routine. Within a task it is possible to create multiple "*blocks*".
Block 1589	A Block represents a portion of a routine. A block comprises multiple "*steps*".
Step 1595	A Step represents one concrete action to be taken. This is filled with the actual action elements: "*states* 1596", "*loops* 1599" and "*branches* 1603".

Example with multiple blocks and steps

In the image above, **three blocks** are seen, which are **to be executed in sequence**.

- Within the first block: "*Move Up*", a test object is to be moved to the correct position.
- Within the second block: "*Test*", the various tests are to be performed.
- Within the third block: "*Move down*" a test object is to be moved to the initial position.

Next, the individual steps are defined. And these can be filled with the corresponding action elements. E.g., in the step "*Movement to perform*", the exact states are specified, which move the object from its present position to the correct height:

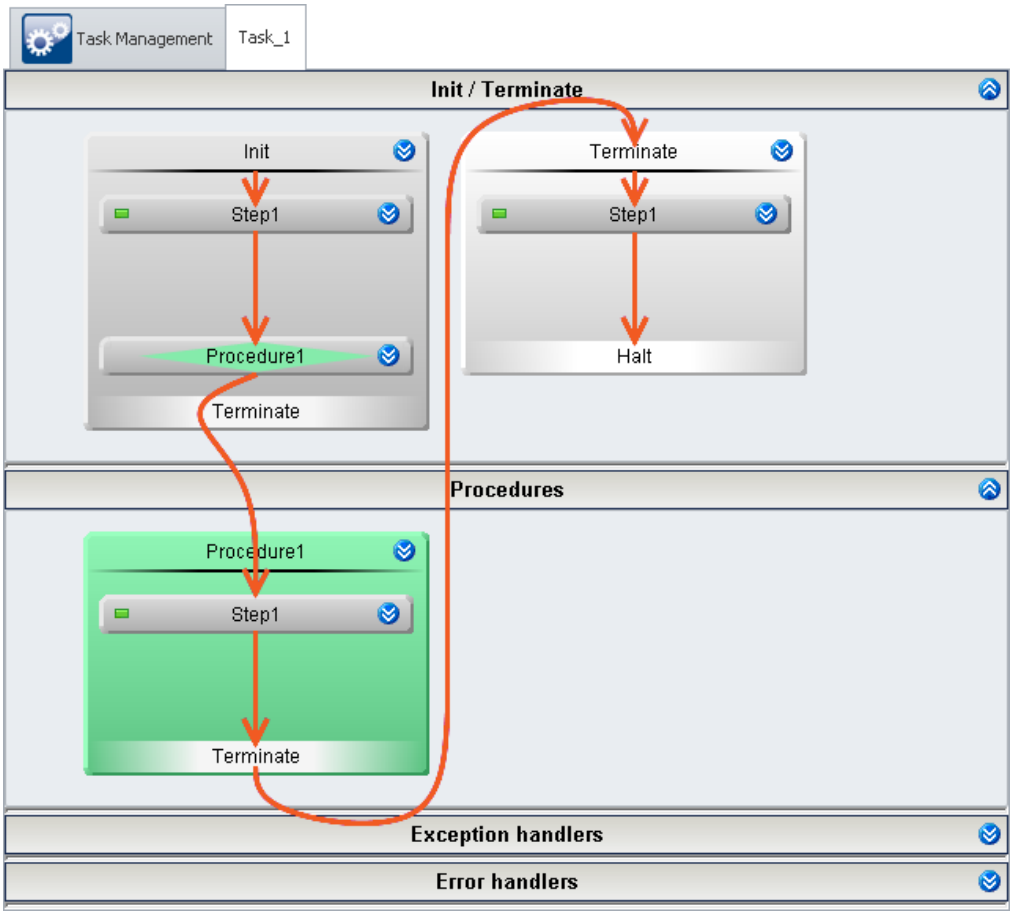
- Start movement
- End movement once height reached

14.1.2 How does the measurement work?

A task **begins at the start of the measurement**.

After the start, the "***Initialization (Init)*** 1590" **is called**. You must specify how matters proceed next. By default, the initialization contains a "*Next branching* 1592". There, the task **skips back to the first "*procedure*** 1590**". At the end** of the task, the block "***Terminate*** 1590" is performed (as long as no error has occurred). This concludes the task with "*Halt*".

Below is a simple example:



Minimal task without any funtion

The relationship between the measurement and the "task"

A task is always associated with a running measurement.

The **task is concluded** when the routine arrives at a "*Halt*" (e.g. at the end of the "*Terminate*"-block). The conclusion of a task has no effect on the measurement's state; the measurement continues. Using the "*Automation Task* 1625" Widget, you can repeat concluded tasks without needing to re-start the measurement. Or stop them, if desired.

If the **measurement ends** before the task is finished, the "*Terminate*"-block is called automatically (if the task is not in an "*Error handling*" stage). The task continues to run until the "*Terminate-block*" or the "*Error handling*" has been concluded. One way to observe this is in the "*Trace Info* 1624".

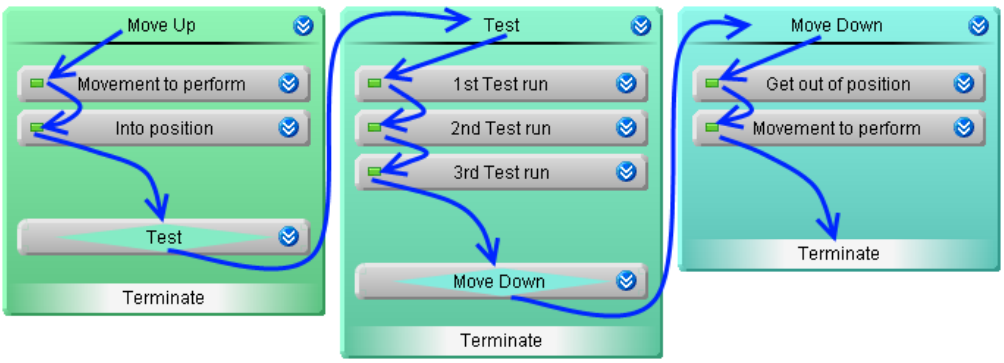
If the user attempts to re-start the measurement while the task is in this state, or if imc STUDIO is disconnected from the device while the task is running, a dialog appears, which offers the choice of ending the task.

Action	Description
Stop	Ends the task without running any further actions. No conclusion phase is run. Even any running error handling processes are interrupted.

Use of multiple blocks and steps

- The task jumps from block to block.
- Once a block has been processed, the next one is started.
- If a block starts, the first step within this block starts and is processed.
- Once a step has been processed, the next one is started.
- This continues until the block has no further steps.

Below, an example with three steps is shown:

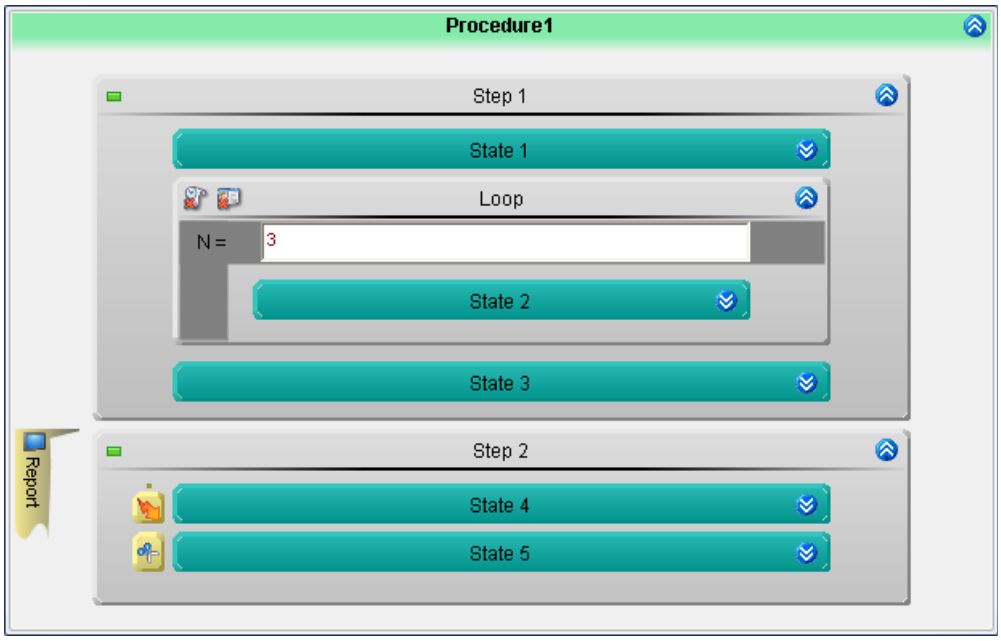


After the block "*Move Down*", the block "*Terminate*" starts.

Jumping from state to state

If a step starts, its contents are carried out. Here, the task comes to its **actual actions**: To the "**States**". The measurement device only knows this layer. The actual procedure jumps from state to state.

Example



It is assumed that all states in the example have the cancel condition {immediately}.

- If the block "*Procedure 1*" is started, the state: "*State 1*" is carried out (see the section: "State 1596").
- After conclusion of the cycle time set, the state: "*State 2*" is carried out.
- In accord with the For-loop, this is carried out three times in succession. Evaluation of the **For-loop** does **not affect the speed**.
- After "*State 2*" comes "*State 3*", then "*State 4*" and "*State 5*". The fact that these are in **different steps** does **not affect the speed**.

Thus, the block "*Procedure 1*" is processed within seven cycle times. Subsequently, the system jumps to the next block's state.

Note

Evaluation of the For-loop and of the jump to the next step or block does not affect the speed.

14.2 Operation

The plug-in Automation offers a variety of operation possibilities.

Opening and closing windows

In the plug-in ***Automation***, it is common for windows to be used which must be displayed either open or in minimized ("closed") mode.

There are different ways to open or close windows.

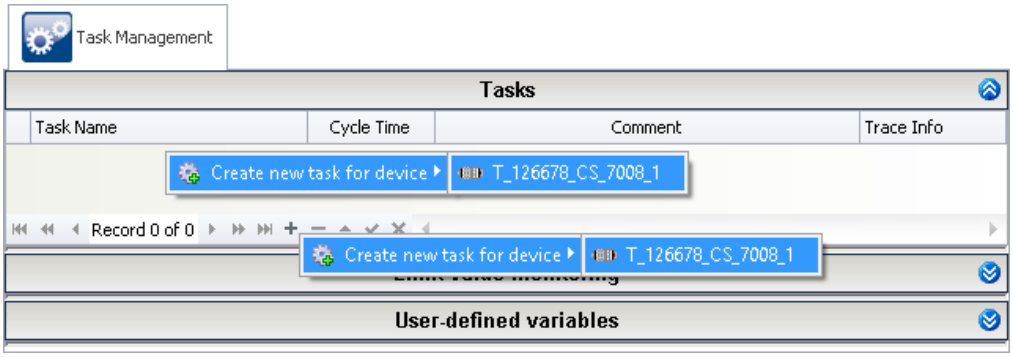
- Double-click on the area title
- Click on the double-arrow (or), or:
- select either the item "*Open*" or "*Close*" from the context menu.

14.2.1 Operation in the Task management

Creating elements

To add an element at the desired position, there are two possibilities:

- Click on the plus-symbol ( ) and follow the menu guidance, or
- open the context menu over the empty areas and follow the menu guidance.



Example: Creating a task by means of the context menu or "+" symbol

Deleting elements

To delete an element, there are two possibilities:

- Select it and left-click the mouse on the minus-symbol ( ), or
- open the context menu over the empty areas and follow the menu guidance.

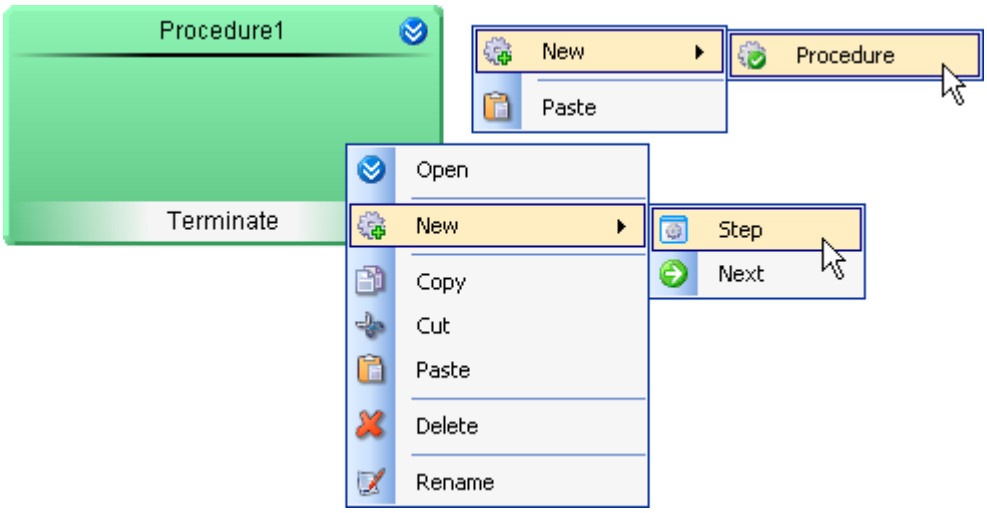
Exporting an element to a different device

You can export tasks from one device to another.

- Right-click the mouse over the task, select "*Export selected task to device*".
- Select the desired device.

14.2.2 Operation in the Task and Block editor

Creating elements



New element in an example with the Task Editor

There are two ways to add an element at the desired position:

- by means of Drag&Drop
- by a context menu command

Creating elements by means of Drag&Drop

- Use Drag&Drop to move the desired element from the tool window **Automation Elements** to the desired position.
- Release the mouse button.

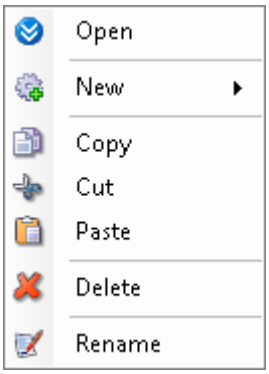
The element is created at the selected location.

Creating elements by context menu

- Open the context menu of the desired free space.
- Select *New*.
- Select the desired element.

The element is created at the selected location.

Copying, cutting, pasting and deleting elements



Context menu of a block

To perform these actions, there are several possibilities:

- via the Edit menu
- via the context menu
- via shortcut (Ctrl + c/x/v and Del)

Element added by means of the context menu appear at the position where the menu was opened. Elements added by means of the *Edit* menu appear at the back of the stack.

Added elements are renamed to be identifiable as a copy.

Renaming an element

The name of most elements can be chosen freely (with the constraint that every block name must be unique).

- To do this, open the element's context menu and select *Rename*.
- Or select the element with the left mouse button and then click again with the left button on the element.

Stretching elements

Some elements in the Task Editor can change in size and thus also change their association with other elements. This is a way of synchronizing "tracks" 1587 or extending imc FAMOS-analyses across multiple element.

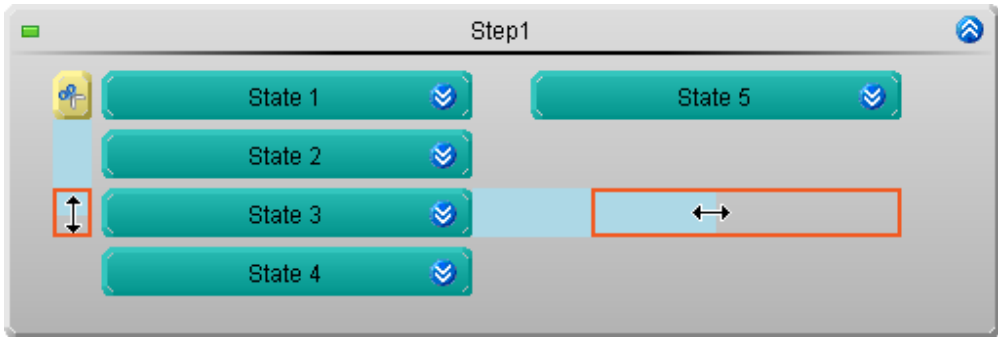
To change the stretching, proceed as follows:

- move the mouse over the element's edge. An arrow symbol appears
- click on the left mouse button and hold it down
- drag the mouse to the desired position and release the mouse button

Example

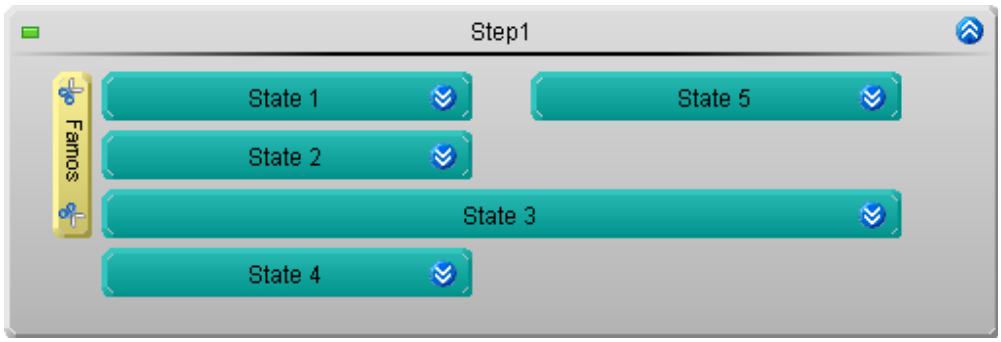
"*State 3*" is intended to synchronize across two tracks, and a imc FAMOS evaluation is intended to "cut" three states.

- Grab the edges of the element and release then within the marked space.



Example: Changing the extent of the elements

The elements are extended.



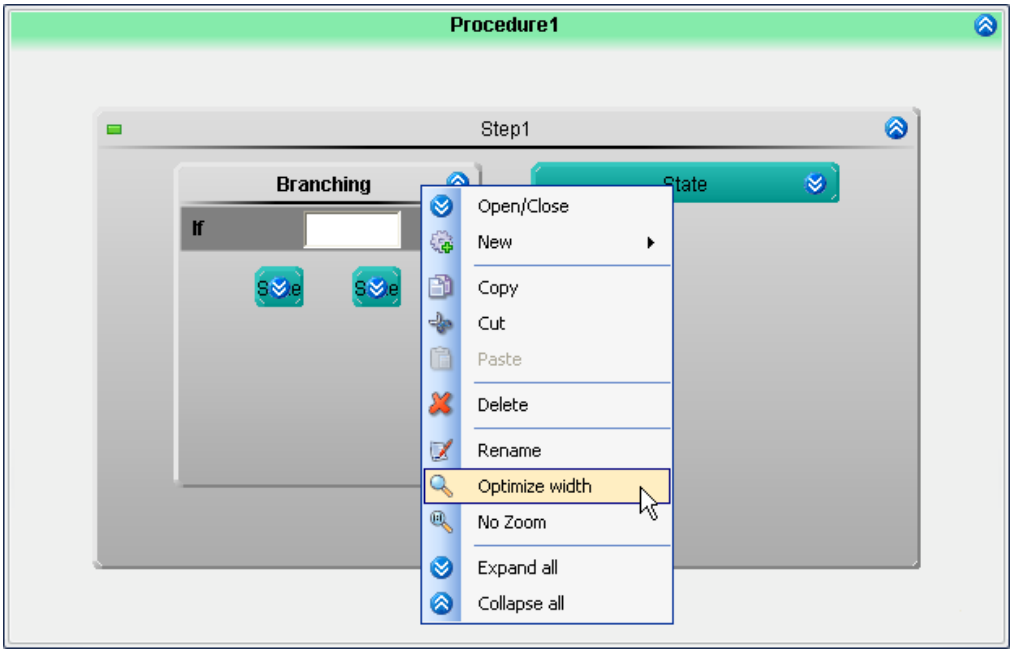
Example: The elements were extended

14.2.3 Zoom - Adapting an element's size

When elements are used in parallel or if multiple nested loops are used, the elements become to small. in consequence, it is no longer possible to read the texts and no more settings can be made in the elements.

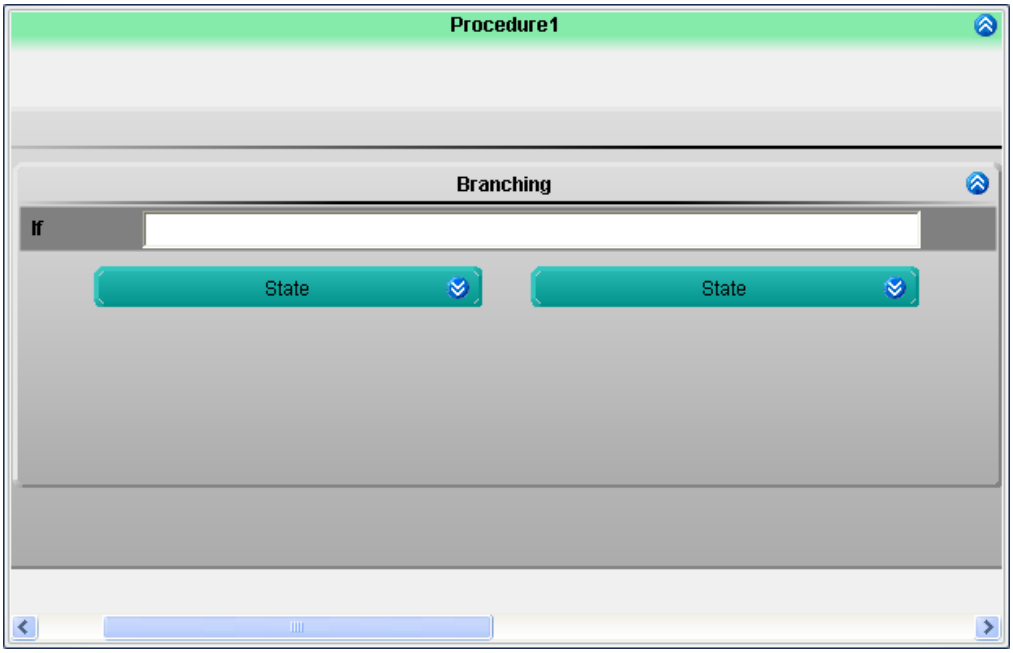
For this reason, in order to keep a clear overview, it helps to assign descriptive names to the nested loops, which reflect what they do.

In order to be able to edit the elements, you must **zoom it out**.



Toward this end, there are multiple possibilities:

- open the element's context menu and select the item "*Optimize width*", or
- open the procedure's context menu and select one of the specified Zoom sizes.



Example:
Zooming in a branch

Reversing the zoom

To reverse the zoom, there are two ways:

- open the element's context menu and select the item "*No Zoom*", or
- open the procedure's context menu and select "*Zoom*" > "*Adapt to page size*"

14.3 Tool window Automation elements

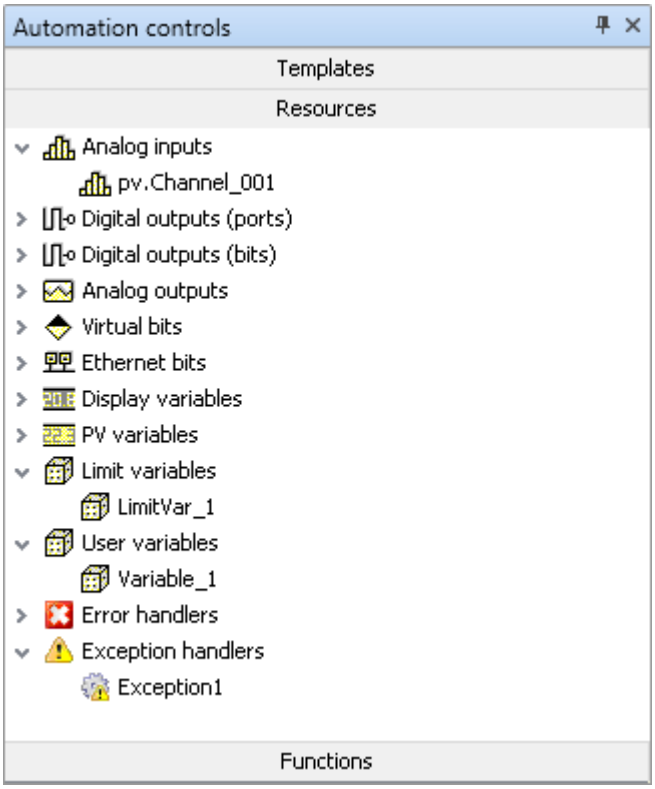
The component Automation includes the tool window "*Automation Elements*".

The tool window is divided into four fields:

Fields	Description
Templates 1588	This region contains the element which you can insert into the current main window.
Repository 1571	You can copy such elements as "*Steps*" and "*Procedures*", "*Limit value monitoring*" and whole "*Tasks*" to the repository, or use ones which are already there.
Resources 1570	This region contains the variables which can be used in a Automation-task.
Functions 1611	You can use functions in "*States*", "*Loops*" and "*If-branches*". This region contains functions which allow special applications in a Automation-task, as well as imc Online FAMOS functions which can be used in the control command "`OnSyncTask`".

14.3.1 Resources - Variables

This region contains variables which can be used in a Automation-task.



Resources

Operation

Inserting a variable into the active element (e.g. into a state description):

- double-click on the variable (the input box must be selected), or
- using drag&drop, or
- enter the variable (keyboard combination <CTRL>+<SPACE> helps in making the entry).

Variables

All single-value device variables are available. Channel data streams can not be subjected to calculations in a task. Instead, use the channels' pv-variables. Also CAN-bits and DI-bits (in the mode: "*Sampling*") can be used, for example.

A task's user-defined variables only appear in the list if the associated task is opened/selected. Variables belonging to other tasks can not be used. The only exception are the pv-variables; these can be used across the boundaries of different tasks.

Exception handling and error handling, as well as value limit monitoring processes, have their own dedicated variables. With those of value limit monitoring as with user-defined variables, they can be used as pv-variables in other tasks.

14.3.2 Repository

Such elements as "*States*" "*Steps*" and "*Procedures*", "*Limit value monitoring*" and whole "*Tasks*" can be copied to the Repository, or retrieved from there for use.

Note States in the repository

It is possible to record states in the repository. Events and their associated commands are included along with the state.
imc FAMOS-data cutting is discarded, since it is not necessarily associated with only one state.

Note Saving the repository

The elements in the repository are saved in an external file.
`%PROGRAMDATA%\imc\Common\Repository\imcCommonUserRepository.asr`
As soon as a new element is introduced into the repository, the file is updated. Therefore, any changes are applied without having to save.

- See also: "*What is saved where?* 162"

Tip: There is no way to export the repository. However, you can use the file to make the repository available to other PCs. The file only contains the Automation-repository. Replace the file on the target PC. However, there is no way to combine the contents.

Operation

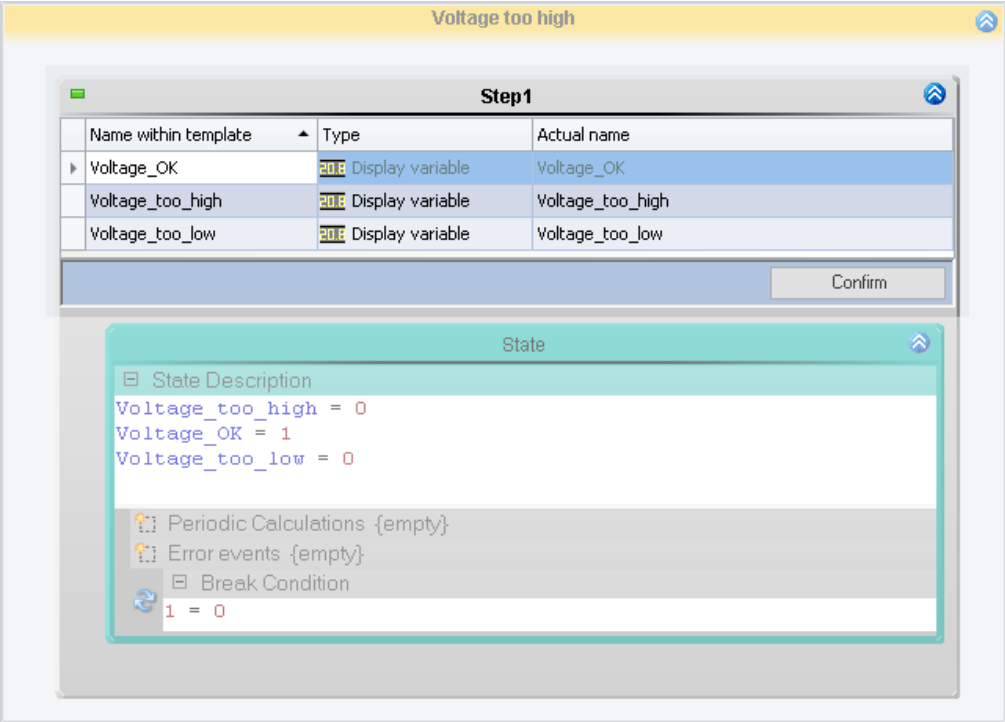
Copying an element into the repository or retrieving one from there for use:

- by means of drag&drop.

Unconfirmed template variables

In some cases you can automatically rename variables.

When you use a "*Step*" from the template, then at the upper edge of the step there is an additional window (you may need to open the step). Here, you can set new names for the variables ("*Actual Name*"). When you click on the "*Confirm*"-button, all variables in the step are renamed accordingly.

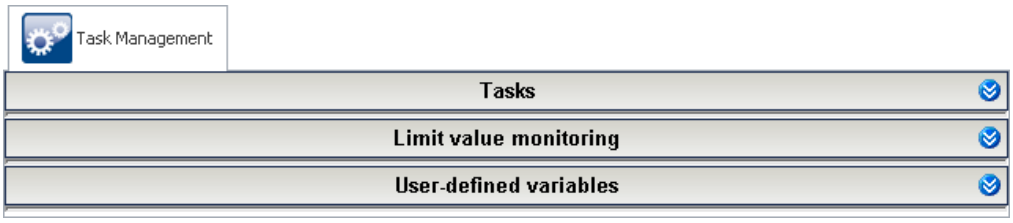


When a "*Task*" from the template is used, an additional grouping appears above the grouping "*Init/Terminate*", where you can change the names.

Note Replace variables in imc FAMOS sequences

imc FAMOS Sequences (data cutting): **The In/Out variables** can also be replaced. The sequence and the imc FAMOS-variables are not taken into account.

14.4 Task management



Task Management (Fields minimized)

"*Task Management*" consists of three fields:

Fields	Description
Tasks 1573	Here you **define** the **tasks**.
Limit value monitoring 1575	Here you can define **events for every task**, **catch any exceptions or errors** that occur, and have them handled in a controlled manner.
User-defined variables 1580	Here you define the **variables for every task**, which are used in the Task Editor 1584, among other places.

Each task has its own combination of "*Limit value monitoring*" and "*User-defined variables*".

For each task, a new tab appears next to the "*Task Management*": (here) "Task_1".

Tab-list

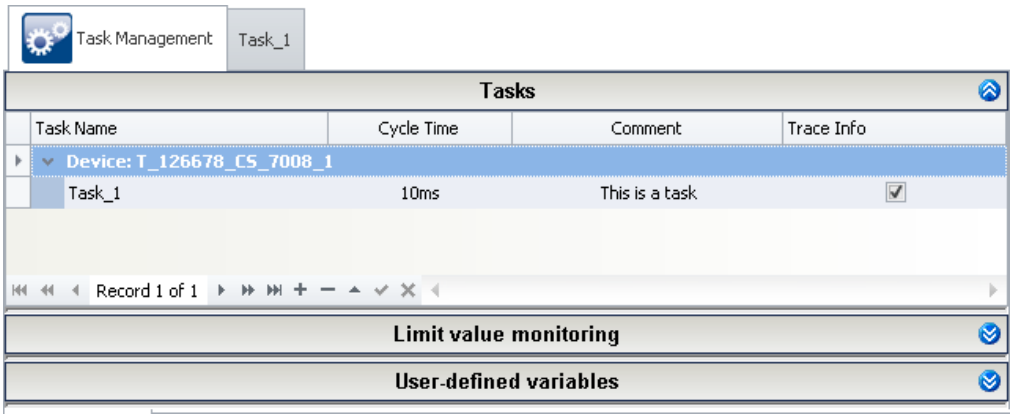
When you **click** on this, **you come to the "Task Editor 1584"** with which you can **create the flow of the selected task**.

14.4.1 Tasks

A Task is a **complete routine of various commands** performed in real-time. Functions to be performed in the task are defined exactly. For this purpose, various elements are available to the task: Limit value monitoring 1575, User-defined variables 1580, Blocks 1584.

For each device, up to five tasks can be created.

- Tasks are created for a particular device. In order to be able to create a task, a device must be selected.
- Tasks run on the device independently in real-time as an imc Online FAMOS program, in parallel with and mutually independent of each other.
- Tasks can be created for and performed by one or multiple imc measurement systems independently.



Example of a new task

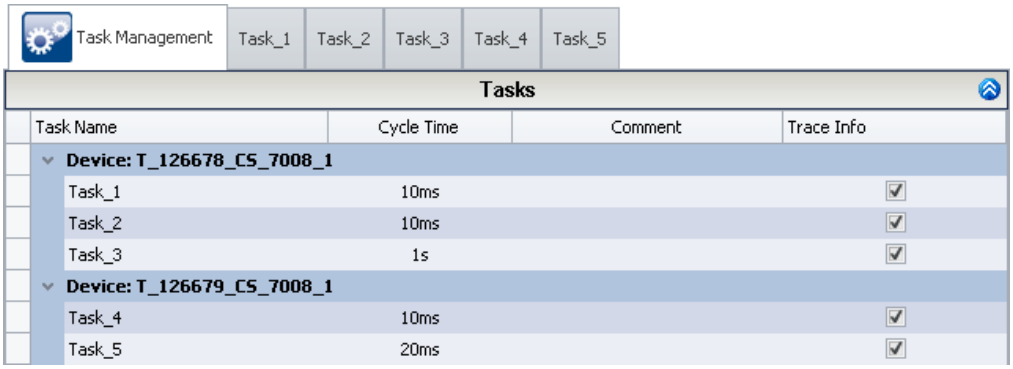
Parameter	Description
Device	Device to which the task is assigned. The tasks are grouped according to the device to which they belong.
Task Name	Name of the task. The name must be unique.
Cycle Time	(processing speeds) Cycle time for the task. Selectable from 100 µs to 1 s. Default value: 10 ms.
Comment	User-defined comment
Trace Info 1624	Activation/deactivation of monitoring of the tasks' current status.

For each task, a new tab appears next to the "*Task Management*": (here) "Task_1". When you **click** on this, **you come to the "Task Editor** 1584**"** with which you can **create the flow of the selected task**.

Cycle time and parallel tasks

Tasks are processed in real time. For this purpose, **assign the task a cycle time** (input range from 100µs to 1s). The cycle time represents the **duration of a state**, and thus reflects the task's speed. For more information, see the section "*State* 1596".

Up to five tasks per device can be set up. These are performed in parallel. For each device, **you can assign a different cycle time to each task**.



Synchronous task

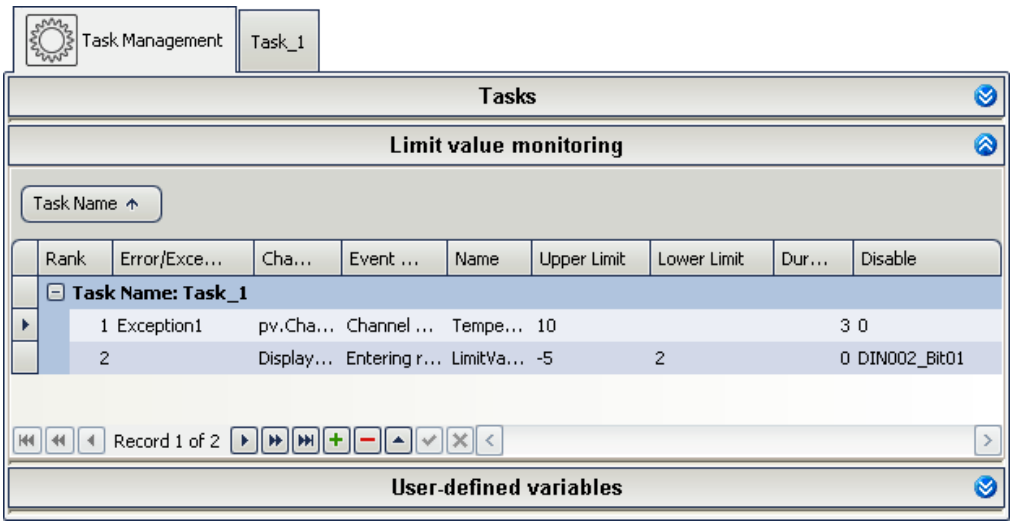
Trace Info

While the Automation is running, you can **observe** which "*block*" and which "*Step* 1595" **are currently being processed**. For more information, see the section "*Trace Info* 1624".

14.4.2 Limit value monitoring

In the field Limit value monitoring you can **define events**. Events can be assigned directly to an "*Exception handler* 1591" or an "*Error handler* 1591". If one of these events occurs while the task is running, the **procedure which is under way is automatically interrupted** and the appropriate error handling is performed.

Minor errors can be dealt with in a controlled way by means of an exception handler. In the case of a serious error, the running task can be concluded in a controlled way by means of error handling. More information on the error treatment procedures is presented in the section "*Blocks: Init, Terminate, Procedures and Handlers* 1591 ".



New limit value monitoring

Parameter	Description
Task-Name	Task to which the limit value monitoring is assigned.
Rank	Priority of the limit value monitoring. If multiple events occur, the limit value monitoring having the lowest number is run.
Error/Exception	Name of the exception handler 1591 or error handler 1591 to be performed, if the event occurs.
Channel	Channel to be monitored.
Event Type	Event type to be monitored
Name	Name of the variable set by the limit value monitoring. Value of the variable: • 0: if not all conditions are met • 1: if all conditions are met
Upper Limit	For the respecting event type the upper limit is entered here. Either a constant or a variable can be used.
Lower Limit	For the respecting event type the lower limit is entered here. Either a constant or a variable can be used.
Duration	This time amount determines how long the condition formulated must be met before triggering the Limit value monitoring routine. Stated in s. (See "*Sample calculation* 1579")
Disable	Here you can deactivate the "*Limit value monitoring*".

Note Clock rate of the monitoring

Every limit value monitoring procedure is assigned to a task and is evaluated at its clock rate.
The monitoring always occurs at the beginning of a cycle time. When a violation of the value limit is detected, the routine immediately branches off to the linked exception-/error-block. (See "*Sample calculation* 1579")

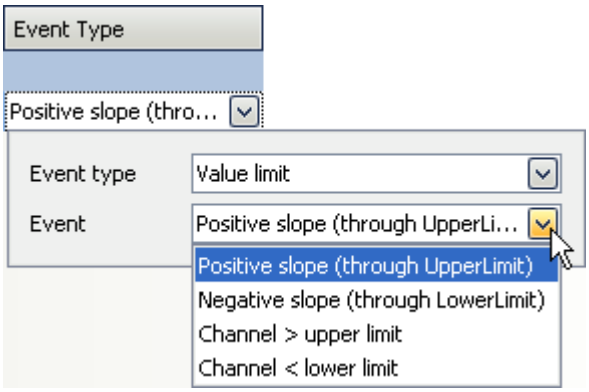
Events

For the limit value monitoring, an event must be selected. The event is one of the conditions which must be met in order for one of the handling processes to start.

To set the event, there are two selection boxes:

- Event type
- Event

As the **event type** can select between "*Value limit*", "*Range*" and "*Bit*". The event can then be set accordingly.



Possible events

Event type: Value limit

Event	Description
Positive slope (through upper limit)	If the monitored channel passes through the **upper limit** from below upward, this condition is met.
Negative slope (through lower limit)	If the monitored channel passes through the **lower limit** from below downward, this condition is met.
Channel > upper limit	As long as the monitored channel's signal is above the **upper limit**, this condition is met.
Channel < lower limit	As long as the monitored channel's signal is below the **lower limit**, this condition is met.

Event type: Range

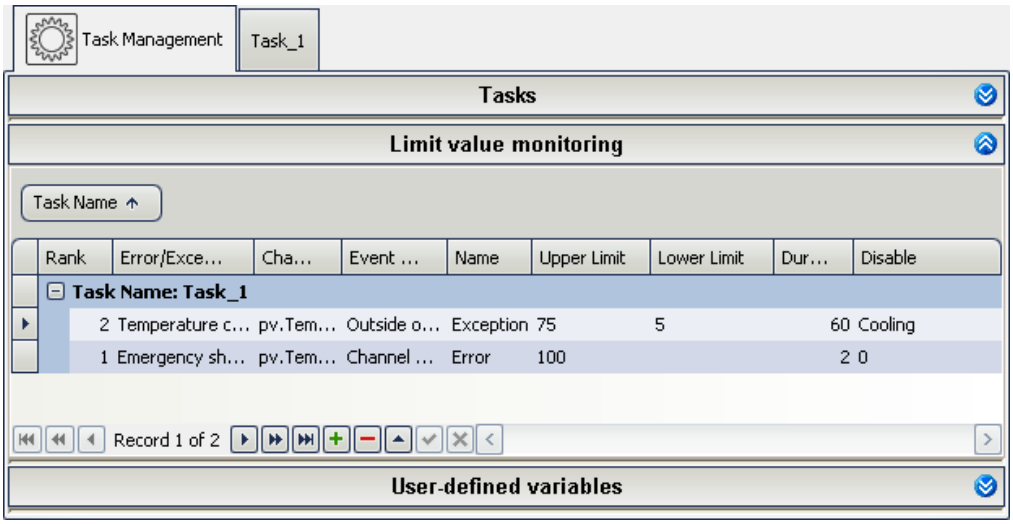
Event	Description
Entering range	If the monitored channel enters a range from either above or below, the condition is met.
Exiting range	If the monitored channel exits a range from either above or below, the condition is met.
Inside of range	As long as the monitored channel is within the range, the condition is met.
Outside of range	As long as the monitored channel is outside the range, the condition is met.

Event type: Bit

Event	Description
Transition from 0 to 1	If the monitored channel transitions from 0 to 1, the condition is met.
Transition from 1 to 0	If the monitored channel transitions from 1 to 0, the condition is met.
Signal = 1	As long as the monitored channel's value is 1, the condition is met.
Signal = 0	As long as the monitored channel's value is 0, the condition is met.

Limits

If you have specified an associated event type, you must assign a value for the "*upper limit*" and/or "*lower limit*". This can be accomplished by means of either a fixed number or a variable (also possible: "*user-defined variables* 1580").



Example of temperature monitoring

Activation/Deactivation

You can activate and deactivate Limit value monitoring functions. To do this, select a variable in the cell "*Disable*" (also possible: "*user-defined variables* 1580").

- If this variable has a value **other than "0"**, the **Limit value monitoring is deactivated**.
- If its value **is "0"**, the **Limit value monitoring is activated**.
- If the "0" is selected in the cell instead of a variable, the Limit value monitoring is always activated.

Name: Variable for the limit value monitoring

The value **limit monitoring's name** becomes a **variable**, which can also be processed in the tasks. Thus, for example, it is possible to check whether all of a value limit monitoring's conditions are still met.

Value of the variable:

0: if not all conditions are met

1: if all conditions are met

Note Making monitored variables globally available

Normally, it is only possible to access the variable within the task. By defining the variable as a pv-variable, you can also access it in other tasks or in imc Online FAMOS. Additionally, you also display the value on the Panel.

Set the string "pv." at the beginning of the name. If you have already used the variable in the blocks, you must rename it in all of its occurrences there ("*Find and replace* 1622").

The conditions

If all of a value limit monitoring's conditions are met

- its variable is set to 1 and
- a handling process is started if one was selected.

Note

- If no handling process was selected, it is only possible to detect that all conditions are met by the variable. The task continues to work without interruption.
- If a handling process starts, the running procedure is interrupted, for example.

The following three conditions must be met:

- the "***event***" must occur,
- the "***minimum duration***" must be reached and
- the value limit monitoring may **not be deactivated**.

Order and priority of the limit value monitoring

The order specifies the relative priorities for monitoring functions. The following rules apply:

- Error handling always has higher priority than exception handling,
- lower numbers have higher priority.

Example Priority

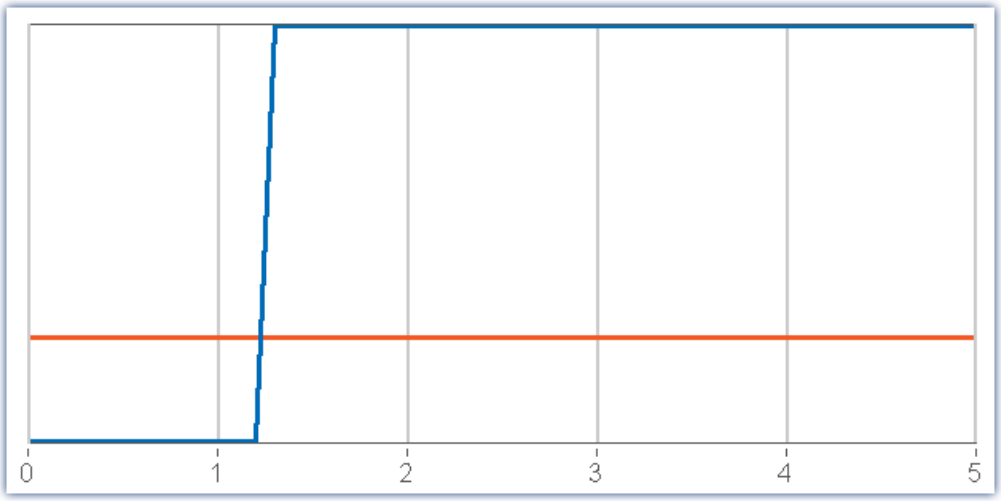
1. The conditions for multiple monitoring functions of the *same type* are fulfilled:
   - the handling with the *lowest number* is started
2. The conditions of multiple monitoring functions of *different types* are fulfilled:
   - the error handler with the *lowest number* is started
3. If an error handling function is already running and the conditions of a limit value monitoring function of *higher priority* are fulfilled:
   - the running error handling function is interrupted
   - the limit value monitoring function with higher priority is started

Note — Occurrence of the value limit monitoring's event while the Terminate block is in process

If all of a value limit monitoring's conditions for starting an error handling are met during the Terminate block, the subsequent error handling is performed.
This does not apply for an exception handling. The termination has higher priority 1592 than the exception handling and is carried out the conclusion.

Sample calculation - At which cycle-steps is handling started?

In this example, a signal is monitored. Whenever a threshold has been violated, a handling procedure is to be started.



blue: monitored channel signal
red: threshold
grid: cycle steps

Question: At what point in time is the limit violation captured and the exception handling started?

At the point in time #1 the value is below the threshold. The limit violation is only registered as of the point in time #2. The handling process begins immediately. At time point #3, the second state of the handling process is already being performed.

Question: As of what time does the clock start if a minimum duration is specified? And when does the handling process start?

The time is counted as of time point #2. The number of cycle steps is determined as follows:

Cycle-steps = Duration / SyncTime.

Sample data: Minimum duration: 2.5s; with a cycle time = 1s

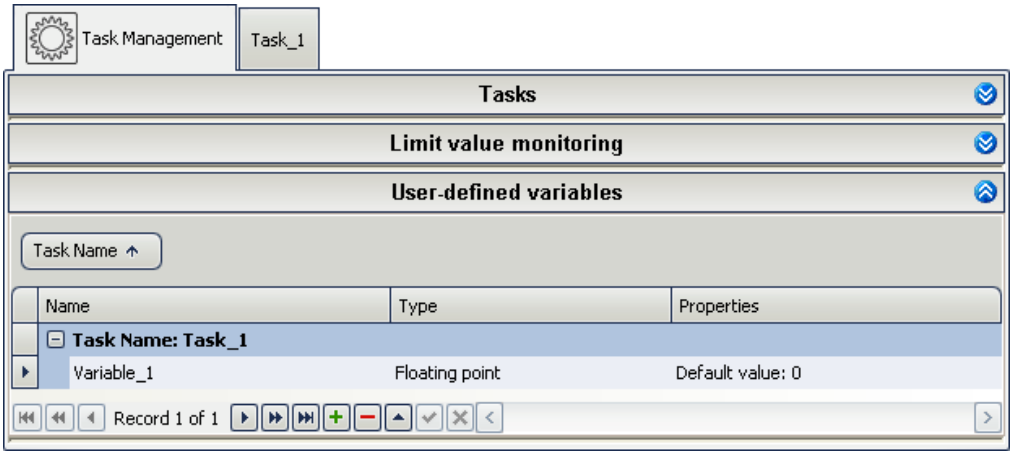
Thus in this example:

Cycle-steps= 2.5 (rounded to 2).

The handling process is thus launched at time point #4.

14.4.3 User-defined variables

In the field User-defined variables, you can **declare variables for the automation for each task**. Each task has its own set of user-defined variables. The variables can be defined in such a way that they can only be edited or accessed by their respective associated task. Or alternatively, (as a pv-variable) by all tasks.



New user-defined variable

Parameter	Description
Task-Name	Task to which the user-defined variable is assigned.
Name	Name of the variable
Type	The variable's data type. More information presented below.
Default Value	Here you can assign an initial value to the variables. The recorded value is written to the variable upon reconfiguration 195 (or whenever the "Prepare" procedure is performed to update the configuration). This is the same as what you see happening in imc Online FAMOS in the block: `OnInitAll`. The exception here are persistent variables; on this topic, see the description of "*Default values and persistent values* 1582".
Persistent	The variable can save the current value in the device so that it is available for use upon the next activation of the device. As soon as "persistent" is set, the variable is converted to a pv-variable.

Variable-type

Variable-type	Description
Floating point	Single value as Float Syntax example: `Variable = 12.345` Possible numbers: ±1.0E35 (exponent: 8 bits, mantissa: 23 bits)
Integer	Single value as 32-bit Integer Syntax example: `Variable = 12`

Variable-type	Description
Virtual channel from single value	This lets you generate a channel which is sequentially filled with single values. The imc Online FAMOS-function "`SingleValueChannel`" generates the same variable type. In the curve window, the variable is displayed as a virtual channel. However, in the Automation it is treated as a single value. Whenever a value is assigned to the variable, this value is entered as a value in the virtual channel. By default, the distance between points is given by xd="1" (Auto). You can modify this in the settings for the variable. Syntax example: `Variable = 12.345` The following settings choices are available: • **Y-unit:** unit of the values entered. Auto = no unit • **X-unit:** unit of the actual "*time axis*". Auto = no unit • **X Offset:** Offset of the first point. Auto = 0 • **X Delta:** Distance from the next point. Auto = 1 More information is presented in the description of the function "`SingleValueChannel`"
	The circular buffer memory set also applies to this variable type. With the preset "*1 min*" and a value of "*X Delta*" of "*1*", exactly 60 values are displayed.
	This variable is missing any reference to time. For this reason, displaying it in a curve window along with other channels will not be meaningful in most cases. Conversely, the variable can very well be displayed in a table, since the individual values are often significant.
	Use this channel type as a results-variable, for instance. Whenever the result of a calculation is generated, it can be appended to the variable.
Vector from datapool	The vector must be present in the Data Browser. When starting the measurement, the vector's content is written to the device. Thus it is available for the task with the current values. When the values of a variable are changed, the new values are used without any further action upon next starting the measurement. The changes are recognized and "*Prepare*" is performed. It is possible to access the elements of the vector with the help of variable. The first element is addressed with the value "1". The second with "2", etc.

	In the following example, the value is written to the vector by means of the "pointer"-variable: "DisplayVar_01":
	
	"Vector" is a user-defined variable of the type: "Vector from datapool".
	`; writing to a vector at a variable position`
	`Vector[DisplayVar_01] = DisplayVar_02+10`
	
	In the following example, the value is read from the vector by means of the "pointer"-variable: "DisplayVar_01":
	`; reading from the vector at a variable position`
	`DisplayVar_03 = Vector[DisplayVar_01]`

Variable-type	Description
	There are a variety of ways you can create the variable in the Data Browser. E.g. as a user-defined variable in the Data Browser 1149 or by means of imc FAMOS 1708. You can also create and fill the variable by means of the commands: "*Load/Refill variable* 1760", for instance.
Static vector	This lets you generate a channel whose elements can be changed during the measurement. The imc Online FAMOS-function "`VectorStatic`" generates the same variable type. All elements are initialized at "0". It is possible to access the elements of the vector with the help of variable. The first element is addressed with the value "1". The second with "2", etc. Example: See "*Vector from datapool*". Accessing the individual elements works in the same way. You can make the following settings: • **Elements:** Count of elements in the vector. Once the measurement has been started, the count can no longer be changed. • **Y-unit:** Unit of the values entered. Auto = no unit • **X-unit:** Unit of the actual "*Time axis*". Auto = no unit • **X Offset:** Offset of the first point. Auto = 0 • **X Delta:** Distance to the next point. Auto = 1 More information is presented in the description of the function "`VectorStatic` 1034" Note the vector's refresh rate on the Panel. With the default setting, the vector is only updated every 10 seconds. If appropriate, modify the setting if higher-speed refreshing is needed: Setup-page "*Analog Channels*"; "*Histogram / Rainflow*" > "*Histogram display update interval* 372".

Process vector variables

You can assign a prefix "pv" to the variable created. For instance, "pv.Variable_1". By this means, this variable can also be used in other plug-ins, such as in the Panel. For more information on process vector variables, see the section "*Process Vector Variables* 1141".

Default values and persistent values

Default value of non-pv-variables:

The default value is "0" and can be modified in accordance with the respective variable type. The value is written to the variable when the imc Online FAMOS-block: `OnInitAll` is executed (e.g. when reconfiguring).

Default value for pv-variables:

The default value is "Keep current value". This can be modified in accordance with the respective variable type.

- Keep current value: The variable has no default value. It is not modified upon reconfiguration and keeps the value it already has.
  When the device is deactivated and reactivated, the variable retains its current value which is displayed for the variable in the Data Browser.
- a number: matches the behavior as for non-pv-variables.

Default value for persistent pv-variables:

The default-value is "Keep current value". It can be modified in accordance with the respective variable type.

- Keep current value: matches the behavior of non-persistent variables. Persistent has no effect.
- a number: matches the behavior of non-pv-variables only upon the first reconfiguration.
  In this case the variable is "persistent" (see also the description of the imc Online FAMOS-syntax "`restore`").
  The value of the variable is saved in the device upon deactivation and before the "Prepare" procedure.
  When "Prepare" is first performed on the experiment after the device has been started, the value is retrieved and written to the variable again (immediately following the `OnInitAll`-block). The assignment is performed after a plausibility check of whether the values can be assigned to the existing pv-variables (e.g. whether they belong to the experiment and to the Automation-task).
  The default value entered has no longer has any significance for the rest of the process. It is set only in case no value is saved in the device.

What is saved are the current experiment's values. Values from previous experiments can not be restored.

Examples:

<table>
<tr><th colspan="3">Parameter</th><th colspan="3">Action</th></tr>
<tr><th>Persistent</th><th>Process-vector variable</th><th>Default value</th><th>Reconfiguration</th><th>Measurement start (after reconfiguration)</th><th>Switch device off and back on
Start measurement</th></tr>
<tr><td>no</td><td>no</td><td>5</td><td>5</td><td rowspan="8">Keeps the current value</td><td>5</td></tr>
<tr><td>no</td><td>no</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>no</td><td>yes</td><td>5</td><td>5</td><td>5</td></tr>
<tr><td>no</td><td>yes</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>no</td><td>yes</td><td>Keep current value</td><td rowspan="2">Keeps the current value</td><td rowspan="2">Keeps the current value</td></tr>
<tr><td>yes</td><td>yes</td><td>Keep current value</td></tr>
<tr><td>yes</td><td>yes</td><td>5</td><td>Keeps the current value
Except upon first reconfiguration: 5</td><td rowspan="2">Loads the value present in the device when the device was deactivated</td></tr>
<tr><td>yes</td><td>yes</td><td>0</td><td>Keeps the current value
Except upon first reconfiguration: 0</td></tr>
</table>

Deleting saved values

For certain applications, the persistent values must be deleted in time for the next start. To delete the values, click on the button "*Delete*" which is in the column "*Process vector backup file*" on the Setup page: "*Devices*". You may need to first show the column 248.

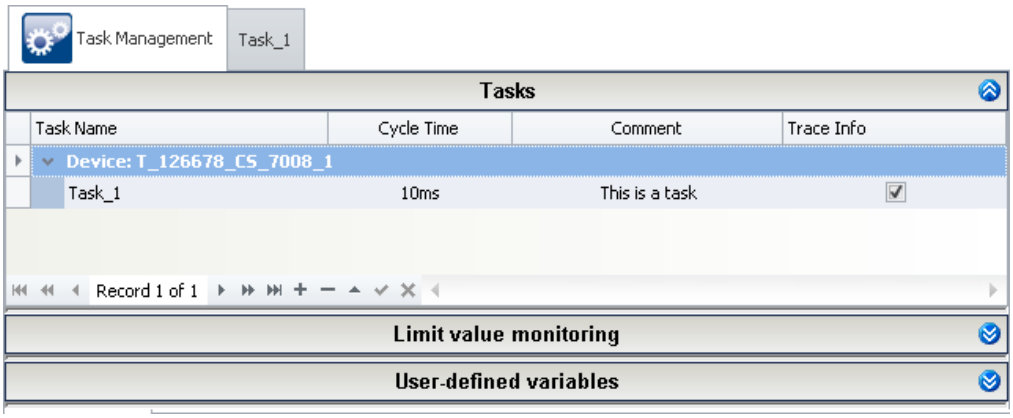
When the function is selected, all persistent values are deleted upon the next start and the device is prepared with the "*Default values*".

14.5 Editors

Two levels of editing help with the implementation.

Editor	Description
Task Editor 1584	Here you create the **outlines of your tasks' flow**.
Block Editor 1586	Here you set the **exact procedures of your tasks' individual Steps** 1595.

Only after you have added a task you can open the Task Editor and the Block Editor.

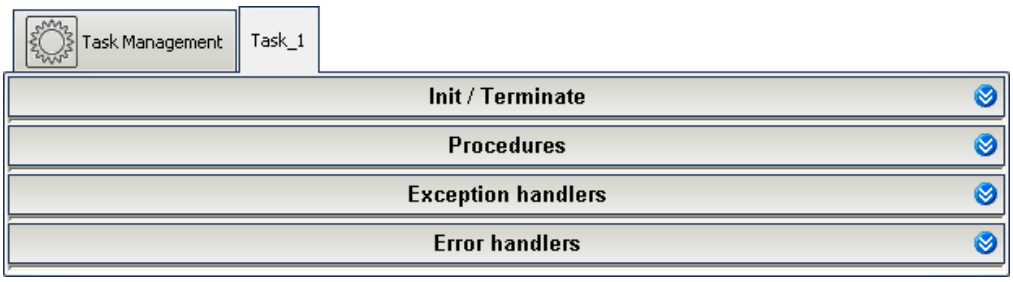


Tast Management is open

In the image above, you see the "*Task Management*" with the task: "*Task_1*".

By clicking on the corresponding tabs, you can switch between the "*Task Management*" and "*Task_1*" (which opens an editor). If you have opened an editor (e.g. "*Task_1*"), you can switch between the editors "*Task Editor*" or "*Block Editor*".

14.5.1 Task editor - Block overview

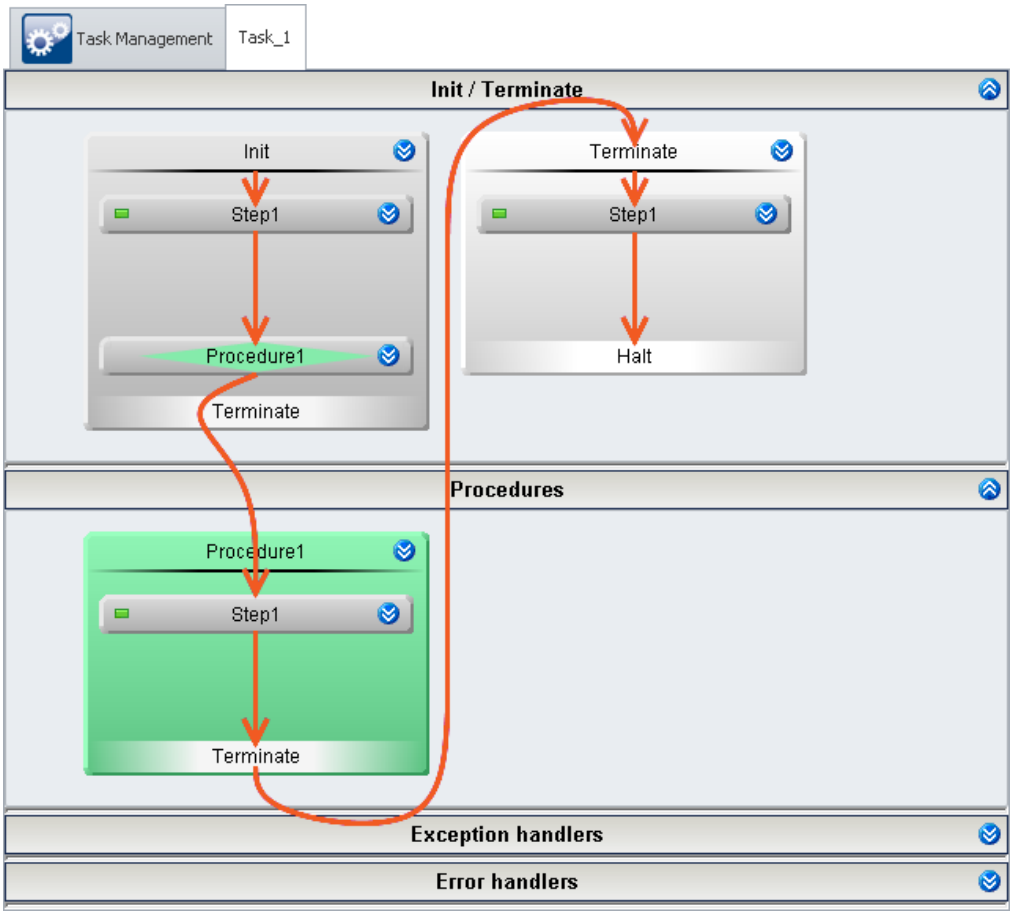


Task Editor

The "*Task Editor*" normally consists of four sectors:

Fields	Description
Init / Terminate 1590	Here, the beginning and end of the Task are defined.
Procedures 1590	Here is where the actual flow is created.
Exception handlers 1591	Here, possible exceptions are defined in order to be able to remedy them.
Error handlers 1591	Here, processes are created which enable controlled closure of the test in case of major errors.
Unconfirmed template variables	This sector only appears when you import a task from the repository. See: "*Repository* 1572".

Each of these fields contains individual blocks. A task consists of at least three blocks: an "*Init*", a "*Procedure*" and a "*Terminate*". A block **combines multiple steps**, which in turn are a combination of multiple other elements.

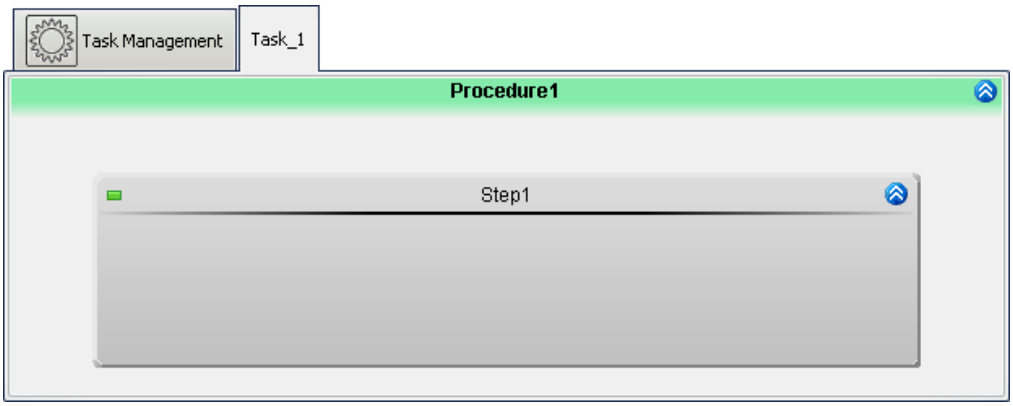


Minimaler Task without function

You can add **any amount of blocks** consisting of "*procedures*", "*exception handlers*" and "*error handlers*".

14.5.2 Block editor

In the "*Block Editor*", **the detailed behavior** in the individual "*Steps* 1595" of a block are specified.



Block Editor with a procedure step created

In the Block Editor, all of the opened block's "*steps* 1595" are displayed. Here, steps can be created or existing steps can be edited. In a step any desired amount of the following elements can be combined in order to provide a better overview:

Elements	Description
States 1596	The element State puts the measurement system in a certain "state" or "condition. For this purpose, a variety of "functions" are available.
Loops 1599	Within a loop it is possible to perform various elements repeatedly.
If-branches 1603	The If-branch element enables other elements to be performed or skipped in dependence upon a condition.

Interaction between the devices and the PC can be set up by means of the following elements:

Elements	Description
General Event 1604	Some elements can be linked with a command sequence, which is executed upon the start or end of the element (called an event).
imc FAMOS Automation (data cutting) 1607	From within imc STUDIO it is possible to transfer variable currently being measured to imc FAMOS and to process them with a imc FAMOS sequence there. The results of the imc FAMOS sequence can be transferred back to imc STUDIO and displayed in a Panel page, for instance.
Page through Panel 1610	Steps can be linked with a Panel page, which is displayed with the step starts.

Execution

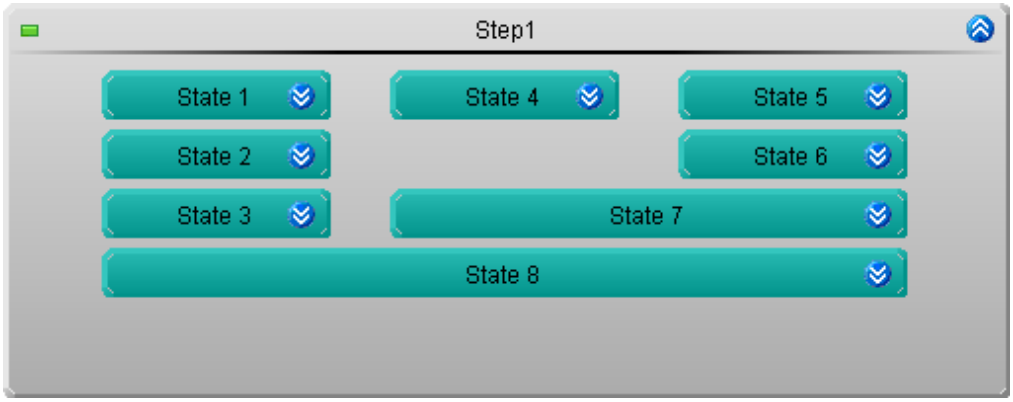
- When a block is executed, all active steps which it contains are executed from top to bottom in succession.
- When a step is executed, all its elements are executed from top to bottom in succession.

Reference

More information on execution of the block is presented in the section: "*How does it work?* 1561".
Or, in reference to the individual elements, in the section: "*Templates* 1588".

Example of a procedure

14.5.2.1 Track: Parallel elements



Tracks: Parallel arrangements of multiple states

Within one step, it is possible **to process multiple elements in parallel**.

- This results in so-called "*tracks*"

In order to create a track, an additional element must be set up to the right or left of an element.

Tracks can be composed of states, loops and If-branches.

Synchronization

Parallel tracks do not need to have the same processing duration.

In order to achieve synchronization between tracks, they are concluded with the same element.

Example featuring states:

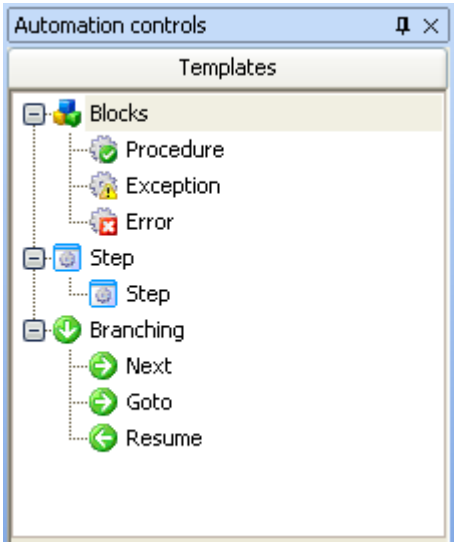
- "*State 7*" synchronizes Tracks 2 and 3
- "*State 7*" is only started, when "*State 4*" and "*State 6*" have finished.

This also works with loops and branchings.

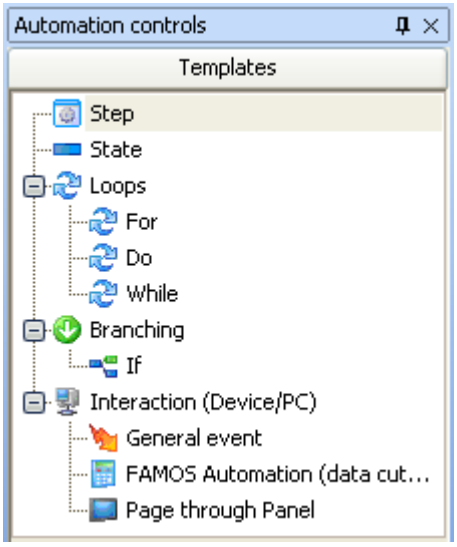
Reference	Stretching elements

Descriptions of how to operate the elements are provided in the section: "*Operation in the Task and Block Editor* 1567"

14.6 Templates - Elements for the editors



Templates (Task Editor)



Templates (Block Editor)

This area presents the elements available for use in the current main window. The elements can be positioned where desired by means of the mouse and the Drag&Drop technique. Or, by using the context menu (see section "*Operation in the Task and Block Editor* 1565").

The templates adapt to the main window, i.e. in the "*Task Editor*" and "*Block Editor*" only such templates are visible which can be used.

A variety of elements are available:

Task Editor templates	Description
Blocks 1589	A block combines multiple steps, which in turn are a combination of multiple other elements.
Steps 1595	A step comprises multiple states, loops and If-branches.
Branchings 1592	By means of a branching, it is possible to switch from a block to a procedure. This makes it possible to jump from one task to another.

Block Editor templates	Description
Steps 1595	A step comprises multiple states, loops and If-branches.
States 1596	The element State puts the measurement system in a certain "state" or "condition. For this purpose, a variety of "functions" are available.
Loops 1599	Within a loop it is possible to perform various elements repeatedly.
If-branches 1603	The If-branch element enables other elements to be performed or skipped in dependence upon a condition.
Interaction (Device/PC)	• Some elements can be linked with a command sequence, which is executed upon the start or end of the element (called an event). • From within imc STUDIO it is possible to transfer variable currently being measured to imc FAMOS and to process them with a imc FAMOS sequence there. The results of the imc FAMOS sequence can be transferred back to imc STUDIO and displayed in a Panel page, for instance. • Steps can be linked with a Panel page, which is displayed with the step starts.

14.6.1 Blocks: Init, Terminate, Procedures and Handlers

Five different blocks are available (see also the detailed material under "*Definition of terms* 1559"):

Block	Description
Init 1590	Here, the beginning of the Task is defined.
Terminate 1590	Here, the end of the Task is defined.
Procedures 1590	Here is where the actual flow is created.
Exception handlers 1591	Here, possible exceptions are defined in order to be able to remedy them.
Error handlers 1591	Here, processes are created which enable controlled closure of the test in case of major errors.

You may add **any desired amount** of blocks of "*Procedures*", "*Exception handlers*" and "*Error handlers*".

Structure of a block

Each block has **a name** and a **fixed End-branch**. In each **block**, it is possible to add a variety of elements.

Execution

- **After** the task **starts**, "***initialization***" is called. You specify the subsequent process.
- When the block starts, it is executed from top to bottom.
- By default, the initialization contains a Next-branching. There, the task jumps to the first Procedure.
- **At the end** of the task, the **block "*Terminate*"** is run (as long as no error has occurred). This **concludes the task with "*Halt*"**.

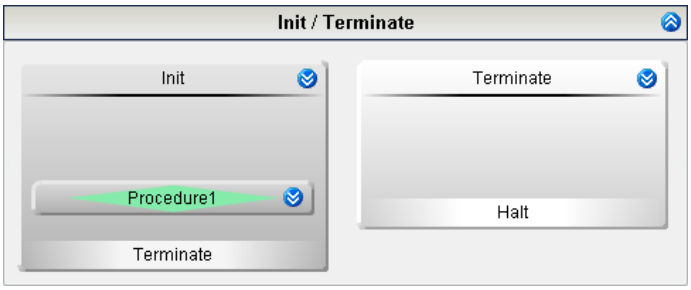
If the measurement concludes with the stop button ( ), then the block ***Terminate*** is called automatically, as long as no handling is running.

The Terminate-branch is not an obligatory branch. You can insert a branch beforehand to any desired procedure (not in Terminate or Error Handling).

Init / Terminate

The field **contains the beginning** ("*Init*") and **the end of the Automation** ("*Terminate*"), if no error occurs.

There is only one Init block and one Terminate block. The names can not be edited.



Initialization and termination

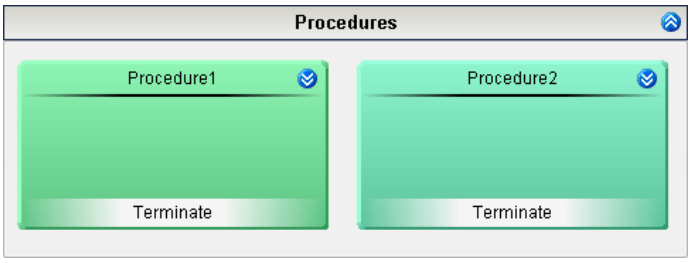
With the help of the "*Init*", the measurement system is placed in a **defined initial state**.

"*Terminate*" brings the measurement system to a **defined conclusive state**.

Block	Elements	End of the block
Initialisierung	**Steps** and **Next**-branchings	• **Next**-branching: The task skips to the procedure defined in the branch. • **No branching**: the Terminate block is started.
Abschluss	**Steps**	At the end of Terminate, the task is concluded with "Halt". Skipping back to a procedure is not possible.

Procedures

The field Procedures **contains the actual algorithm**.



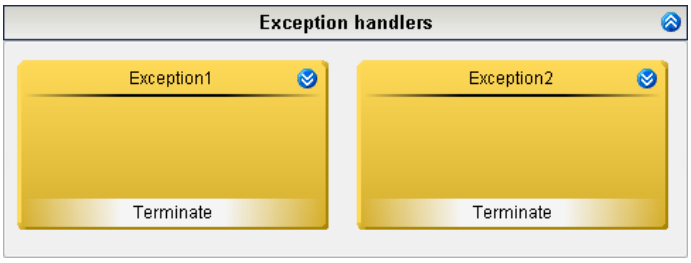
Procedures

Block	Elements	End of the block
Procedures	**Steps** and **Next**-branchings	• **Next**-branching: The task skips to the procedure defined in the branch. • **No branching**: the Terminate block is started.

Exception handler and Error handler

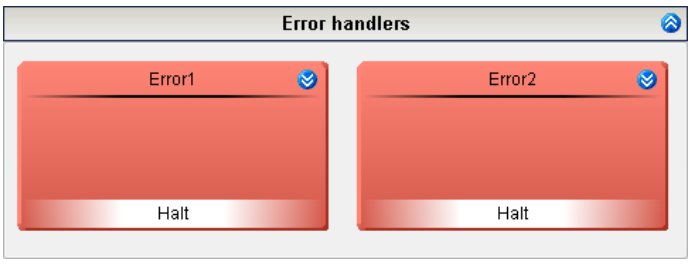
Handlers are blocks for **treating any disturbances or errors**. If a disturbance occurs, the **current "*procedure*" is stopped** and **the handler is run**.

There are two types of handlers:



Exception handlers

A **minor error** can be **remedied in a controlled manner using an exception handler**.



Error handlers

In the case of **major errors**, you can **end** the running task **in a controlled manner** with an error handler.

Block	Elements	End of the block
Exception handler	**Steps** and **Resume**-branchings and **Go to**-branchings	• **Resume**-branching: the Task resumes from the point at which the first exception handler was started. The state is then re-implemented completely (also the "State description"). • **Go to**-branching: the Task skips to the step specified in the branch. • **No branch**: the Terminate block is started.
Error handler	**Steps**	At the end of an error handler, the "**Halt**"-state is set and the task is concluded.

Starting a handler

It is possible to start a handler routine either from within a "*limit value monitoring*" process or from a "*state*".

Defined in	Description
Defined in the Limit value monitoring:	This **applies to the entire task**. **It is started if all conditions** of a limit value monitoring routine. The handling which is defined in the "*Exception/Error*" box of the Limit value monitoring is started. See "*Limit value monitoring* 1575"
Defined in a State:	This **only applies to the state**. The handling routine is started if a disturbance occurs during the state. The **procedure** running **is interrupted** in such a case. See "*State* 1596"

Priorities of the blocks - Branching across block types

The routine is always running within one of the blocks. The priority is given as follows:

1) Error handlers
2) Terminate
3) Exception handlers
4) Procedure/Init

The following branching situations are possible:

If the program run is at the stage of "*Error handling*" (1), then once this stage is concluded, the routine stops. The "*Termination*" (2) is no longer performed.

If the program run is at the stage of "*Termination*" (2), then upon conclusion of the Termination, the program is finished. It is possible to branch off to an "*Error handler*" (1), but not to an "*Exception handler*" (3).

If the program run is at the stage of an "*Exception handling*" (3) process, it is possible for it to skip back to any "*Procedure*" (4).

14.6.2 Branching (Task editor)

Next-, Goto- and Resume- branches can only be created in the Task Editor. By means of a branching, it is possible **to switch from a block to a procedure**. This makes it possible to jump from one task to another.

Structure

In the **respective blocks**, **only certain branchings** are permitted.

Branches are set up on the bottom edge of the block. They form the conclusion of the block and they determine what procedure follows next.

There are three types of branches

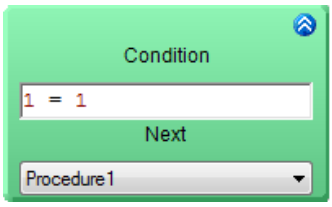
Branch	Location of use	Condition
Next	Init and procedures	If the condition **is met**, the procedure selected is performed. If it **is not met**, the next item in this block is run. This can either be a further branch or the conclusion of the block.
Resume	Exception handlers	If the condition **is met**, then the Task is resumed **from the point at which the first exception handler** was started. The **state is then re-implemented completely** (also the "*State description*"). If it **is not met**, the block's next item is run. This can either be another branch or the block's termination.
Goto	Exception handlers	If the condition **is met**, then the Task is resumed **at the selected step** in the selected procedure. If it **is not met**, then the next item in the block is run. This can either be another branch or the block's termination.

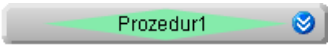
Conditions

Each of these branches **is linked to a condition**. This condition is an "If"-query. If it is **fulfilled**, the **branch is run**.

In the condition's box, any single-value variables and values can be used, **plus functions**. If the condition is intended to be met always, it is possible to use a "1" or "1 = 1".

Branching: Next



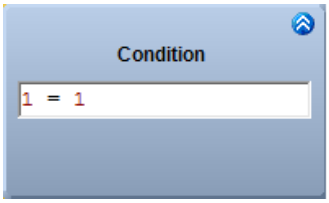


This branch is used in the Init and in procedures.

The branch's **color** adapts to the target.

Next-branch open and closed

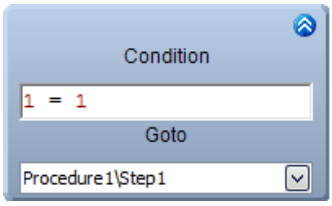
Branching: Resume

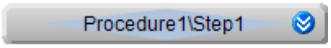


This branch is used in exception handlers.

Resume-branch open and closed

Branching: Goto



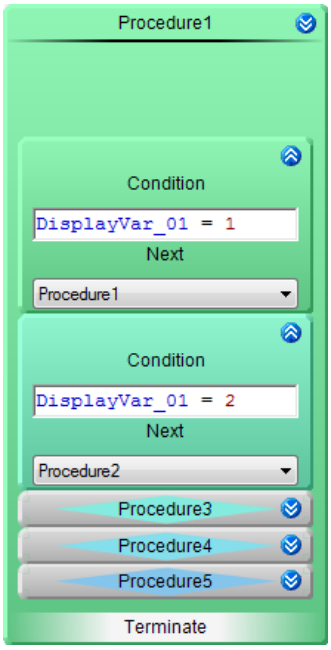


This branch is used in exception handlers.

The branch's **color** adapts to the target.

Goto-branch open and closed

Multiple branches



Multiple branches in succession

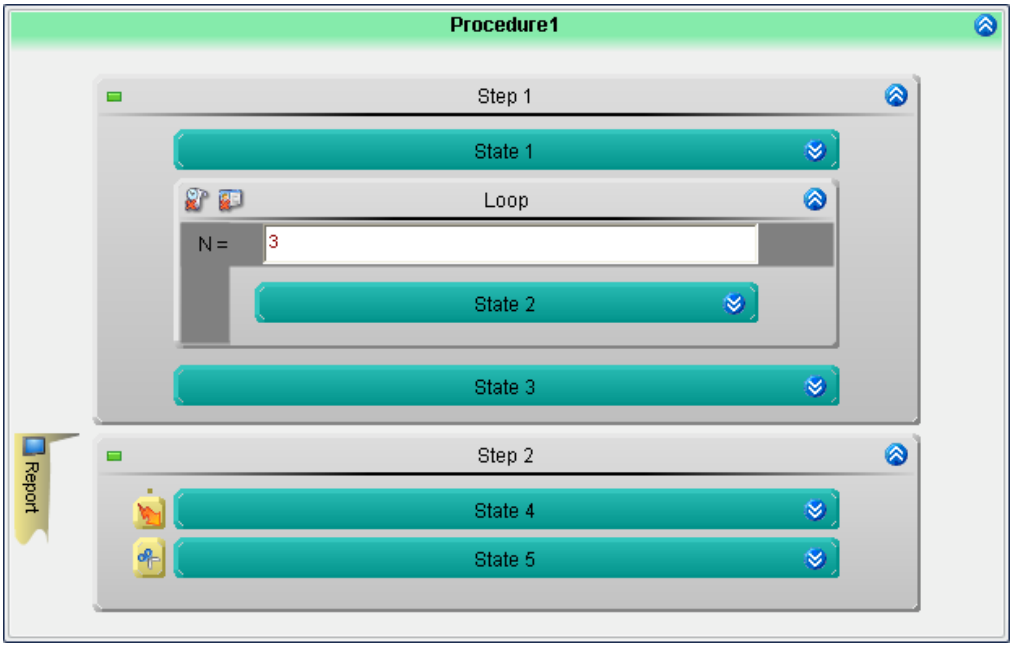
Multiple branches can be placed in succession. If the conditions of the first branch are met, then the next one is tested.

14.6.3 Steps

Empty step

Each block can contain a number of steps. Steps can be created in the Task Editor or in the Block Editor.

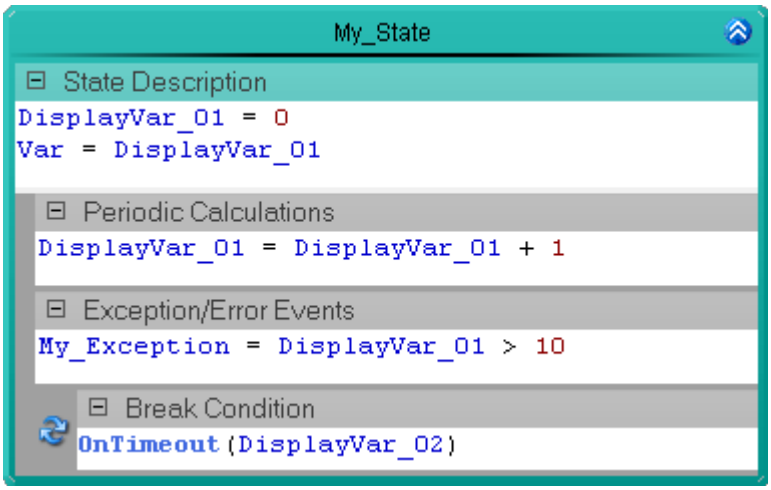
A step comprises multiple states, loops and If-branches.



Example: Two steps
A step can contain various elements

14.6.4 State

The element "*State*" puts the measurement system in a certain "state" or "condition. For this purpose, a variety of "*functions* 1611" are available.



State

Positioning

A state can be located within a step, loop or If-branch.

Structure

A state consists of multiple input boxes:

Input box	Description
State description	In executing the state, the measurement system is brought into a defined state. Here, it is possible to perform a calculation.
Cyclical calculations	Similar to state descriptions, a calculation can be performed here
Exception and error handling 1598	This input box serves the purpose of monitoring measurements. If a condition defined here occurs, the respective handling begins.
Break condition 1598	The "*cyclical calculations*" and "*exception and error handling*" are performed until the "*break condition*" is met or an error occurs.

Execution

Once the state is started, all commands are executed from top to bottom.

- **All commands in the state description** are executed
- **All commands of the cyclical calculation** are executed
- A **check is performed** of whether a **condition in the exception- or error handling** is true. If this is the case, execution of the state is interrupted and the handling starts.
- If there was no error, the system **checks** whether the **break condition** is true. If it is, the state is exited, and in it is false, the program **skips back to the cyclical calculation**.

Timing

One iteration of the state lasts one cycle.

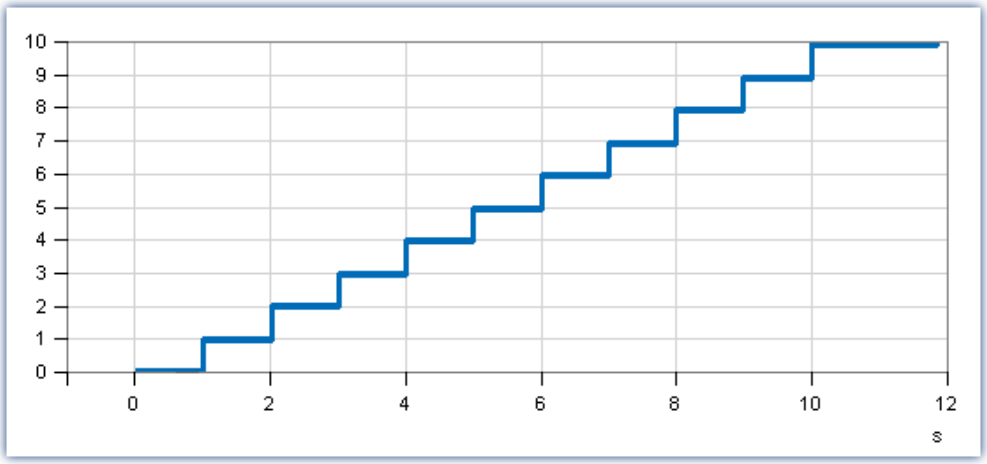
- If the **break condition is true**, then in consequence the **next state starts after exactly one cycle**.
- If the **break condition is not true**, the state is processed again (**Cyclical calculations** without the State description) within one cycle.

Example A counter

- Cycle time: 1s
- State description: DisplayVar_01 = 0
  (setting the Display variable to 0 when starting the state is needed for initializing the counter. In the counter, the value is immediately afterwards increased in the cyclical calculations.)
- Cyclical calculations: DisplayVar_01 = DisplayVar_01 + 1
- Break condition: DisplayVar_01 = 10

Assuming that when the state starts, t = 0s

State	Description
t = 0s	The state is started: • State description: DisplayVar_01 is set to 0 • Cyclical calculation DisplayVar_01 is set from 0 to 1 • Break condition is not true (1 =! 10)
t = 1s	After exactly one second, the cyclical calculation is performed again • Cyclical calculation DisplayVar_01 is set from 1 to 2 • Break condition is not true (2 =! 10)
...	This continues until DisplayVar_01 = 10
t = 9s	• Cyclical calculation DisplayVar_01 is set from 9 to 10 • Break condition is true (10 = 10) • The state is concluded
t = 10s	The next state starts



Example: Counter
Plot of the variable: DisplayVar_01

Exception handling and error handling

This input box is for monitoring purposes. If the condition defined here occurs within this state, an (exception/error) handler is started 1591. The entry must follow this pattern:

<**Name** of the handler to be started > = <Condition>

e.g.

```
My_Exception = DisplayVar_01 > 10
```

or

```
My_Error = Virt_Bit01 = 1
```

Multiple monitoring mechanisms can be entered, e.g.:

```
Exception1 = Cycles_1 % 5 = 0 or Calibration_Act = 1
Exception2 = Cycles_1 % 100 = 0
```

The following branching types are possible:

States in "*Initialization*" and "*Procedures*" may branch to "*Exceptions*" and "*Errors*".

States in "*Exceptions*" and "*Termination*" may branch to "*Errors*".

States in "*Errors*" may not branch.

When the handler is concluded

If the handler concludes with a "*Resume-branching*", processing of the routine continues from the point at which the first exception handler had been started. The state is re-run again in its entirety (including the "*State description*").

Additional possibilities are presented in the section: "*Blocks: Init, Terminate, Procedures and Handlers* 1591"

Break condition

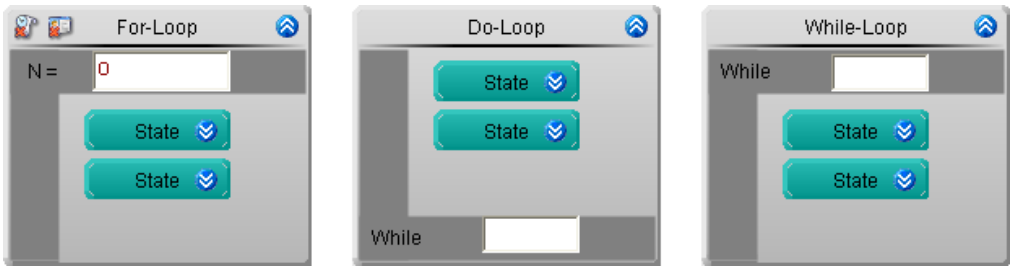
If any break condition is defined, processing of the routine remains in the same state until the condition is met or a handler is started. If the break condition is not true, then upon the next cycle, the "*Cyclical calculation*" is executed again, but not the "*State description*".

Examples are presented with the documentation for the functions:

- OnRampEnd 1615
- OnTimeout 1617
- OnStable 1621

14.6.5 Loops

Within a loop it is possible to perform various elements repeatedly. As the repetition's condition, there are a number of commands available.



Loops

There are three loop types in the Block Editor:

Loop types	Description
For-Loop	Counting loop (for additional properties, see section: "*For-loop* 1600") With the For-loop, the header specifies how often the loop is to be performed. The specified value exactly corresponds to the number of runs. Here, it is not necessary to set up a counter by hand.
Do-Loop	Post-test loop The Do-loop is run at least once. The test of the condition is performed only at the end of the loop. The loop is performed as long as the condition is true.
While-Loop	Pre-test loop The While-loop checks the specified condition before entering the loop. Then the loop is performed as long as the condition remains true. By specifying the condition 1=1, i is easy to create a permanent loop.

Positioning

A loop can be located within steps, loops, and If-branches. In loops, it is possible to insert all elements which could also be inserted into steps.

Timing

The **test** of the condition does **not use up any cycle time**. I.e., after the last state in the loop, either the loop's first state or the first state outside of the loop is started next.

Condition

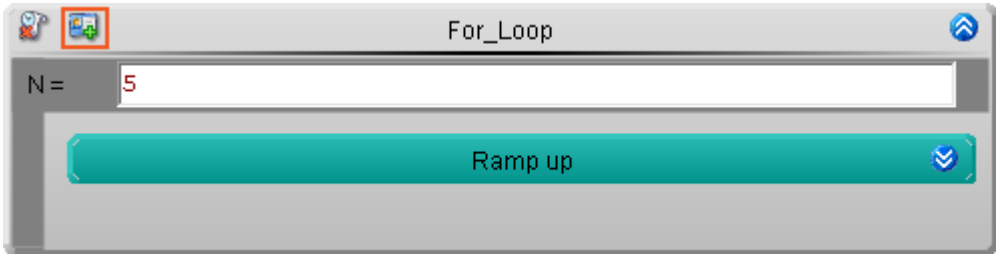
A test box is available for the loops, in which a condition can be entered. This box can contain numbers, variables or other commands. E.g., for the For-loop it is possible either to enter a 10, or a variable can be used which represents a number. Note that the variable's values could change, for which reason the loop might end prematurely.

14.6.5.1 Properties of the for-loop

Loop counters as variables - Named/Unnamed loop counters

The element **For-loop** can write the current counter reading to a variable. Thus, it is possible for other elements to access the current counter.

To write the current counter reading to a variable, click on the button ( ).



Loop counter with name

The counter reading is written to a variable.

A variable with the name of the For-loop is created in the group: "*User-defined variable*" in the "*Resources* 1570".

No variable is created.

Note Unique association with the task

The variable created is only visible within the task.

Process vector variables

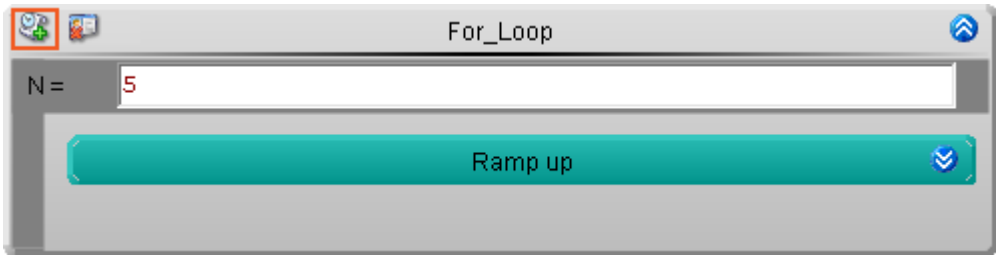
You can assign a prefix "pv" to the variable created. For instance, "pv.Loop".

- It is possible **to access** the variable **globally throughout the task**.
- It can be displayed on a Panel page.

Persistent/Non-persistent loop counter

If the element **For-loop** is run through multiple times in succession, the loop counter always starts counting from "1". In some cases, however, it is important for the loop counter to save the last value and to start counting from that value upon the next run.

To save the loop counter's last value, click on the button *Persistent* ( )



Persistent loop counter

The loop counter is persistent. Upon the next iteration of the loop, counting starts from the last value.

The counter always starts at "1".

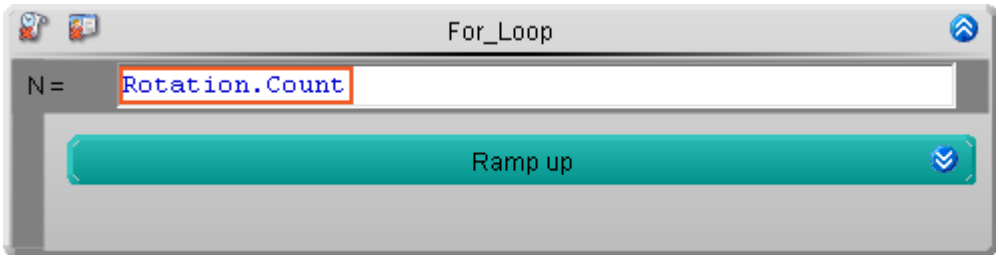
Note If both properties are activated

If both properties are **active simultaneously**, the variable becomes a **process vector variable**.

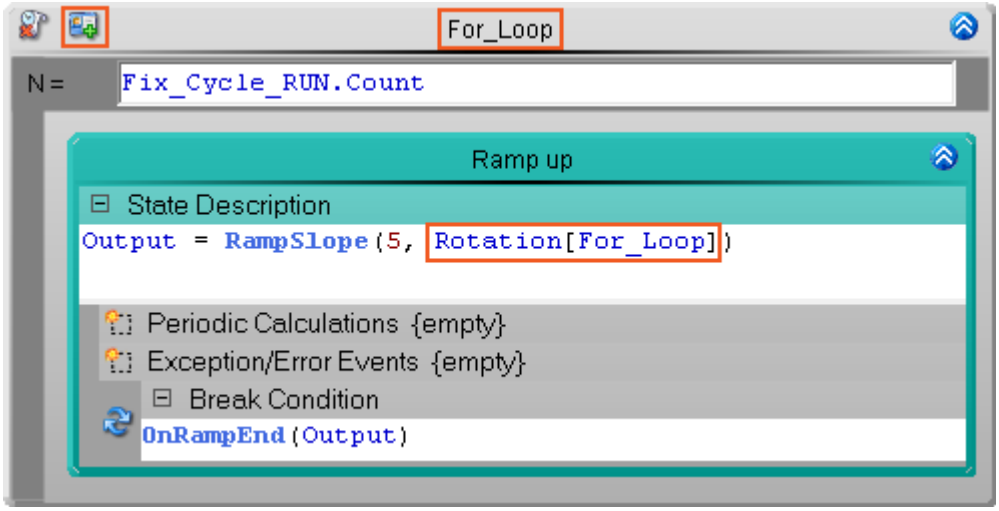
- It is possible **to access** the variable **globally throughout the task**.
- It can be displayed on a Panel page.

For-loop through the count of elements in a variable

"*VarName.Count*" returns the number of elements a variable contains.



If you additionally activate the For-loop's loop counter, then in this way it is possible to access all of a variable's elements in succession.



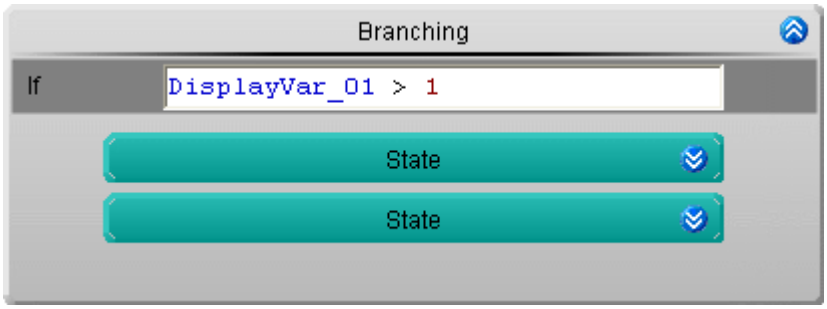
Example: For all points belonging to the variable "Rotation", a ramp is successively generated.

Reference Tutorial

A detailed example is presented in the tutorial: "*Actuator-control - Ramps* 1657"

14.6.6 Branching (Block editor)

The If-branch element enables other elements to be performed or skipped in dependence upon a condition. A variety of commands are available for the condition.



If-branch

Positioning

An If-branch can be located within a step, loop, or other If-branch. It is possible to insert If-branches in any elements into which it is also possible to insert steps.

Execution

With a If-branch, a condition is specified in the header. Before the If-branch starts, the specified condition is checked.

- If the condition is true, the contents are carried out.
- If the condition is not true, the contents are skipped.

Timing

The **test** of the condition does **not use up any cycle time**. I.e., after the last state in the loop, either the loop's first state or the first state outside of the loop is started next.

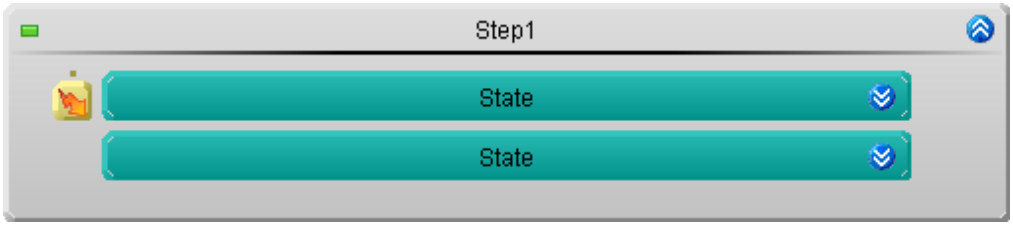
Condition

A test box is available for the loops, in which a condition can be entered. This box can contain numbers, variables or other commands. E.g., as the condition, one could specify Display_Var_01>1. Only if the value of Display_Var_01 is greater than 1, the content of the If-branch is executed.

14.6.7 Interaction (device/PC)

14.6.7.1 General event

Some elements can be linked with a "*command sequence*", which is executed upon the start or end of the element (called an "*event*").



Example: Starts a command sequence after the state is concluded

Note

Immediate trigger

The PC interaction can only be performed if the device has at least one active channel with a "immediate trigger" (e.g. the "*BaseTrigger*"). "Immediate triggers" are triggers which fire upon starting the measurement, and have no other associated sources.

Block "*Terminate*"

The "Interactions (device/PC)" in the block "*Terminate*" are no longer performed if the block "*Terminate*" is performed due to the end of the measurement.

Positioning

The element "*General event*" can be located to the **left beside states, loops and If-branches**. **Only one device/PC interaction** can be to the left of these elements.

Parallel General events are possible by means of separate "*tracks* 1587".

Execution

When the specified event occurs (beginning or end), the command sequence is started.

In the event of an error:

When an **error is reported**, the sequence is ended and the subsequent commands are not executed (see "*Stopping due to an "Error"* 1689"). It is not possible to deactivate "*Stop on error*" for the Automation.

The error has no effect on the task. While processing synchronized events, the task does **not stop** if an error occurs, but continues instead.

Timing

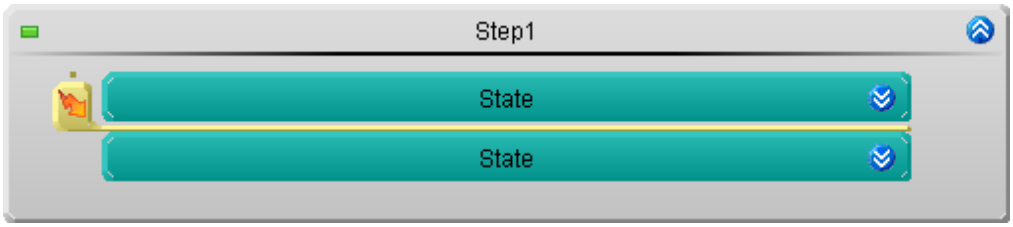
The element "*General event*" **does not use up any of the devices cycle time**. The command sequence runs on the PC. Thus, the command sequence starts and **the task proceeds without restrictions**, although the command sequence is still running.

Synchronization

However, in some cases it is necessary for the **command sequence to be concluded before the next state** can be performed. In order to prevent the next state from starting before the command sequence is concluded, there is one possibility:

- open the element's context menu and select "*Synchronize*"

Now the next state only starts once the command sequence is finished. This is indicated graphically by a **synchronization line below the element**.



Example: A synchronized command sequence starts once the state is finished

Note

The duration of the command sequence depends on the computer. In this case, it is not possible to say when the next state starts.

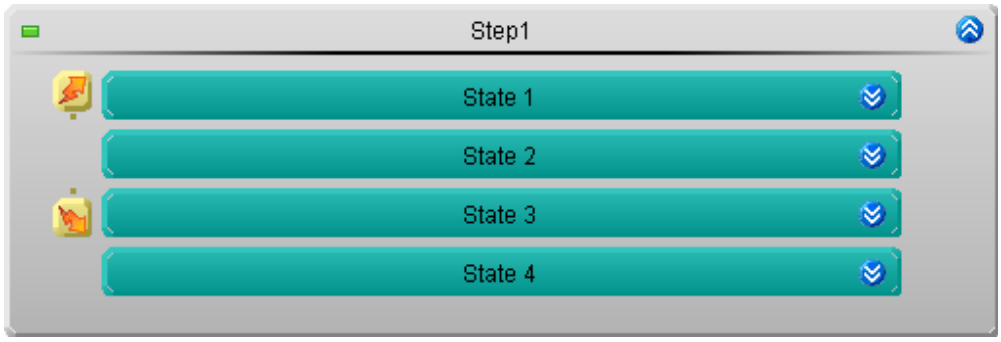
Execution at the beginning or at the end

The command sequence can start either when the element starts or when it is concluded.

To set when the command sequence is to start, open the context menu of the "*General event*" element and select the corresponding entry from the drop-down list:

General event	Description
Before	Execute at beginning
After	Execute at end

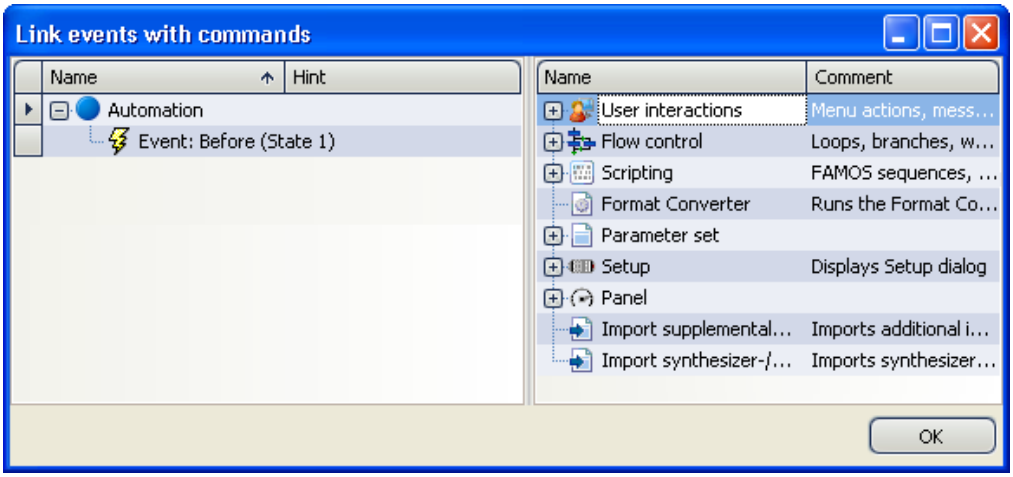
Example Execute



- When starting "*State 1*", a command sequence is started
- When starting "*State 3*", a command sequence is started

Opening the Event Dialog and making settings

To open the Event Dialog, double-click the left mouse button over the element.



Example: Event Dialog
Event "Before" of the element State 1

Reference — Command Reference and Command Sequence

- The workings of the individual commands is described in the section "*Command Reference* 1706" beschrieben.
- A **sequence of commands** can be set up in a variety of imc STUDIO plug-ins and is described separately (see chapter "*Sequencer, events and commands* 1681").

14.6.7.2 imc FAMOS automation (data cutting)

From within imc STUDIO it is possible to transfer variable currently being measured to imc FAMOS and to process them with a imc FAMOS sequence there. The results of the imc FAMOS sequence can be transferred back to imc STUDIO and displayed in a Panel page, for instance.

Note — **Prerequisite**

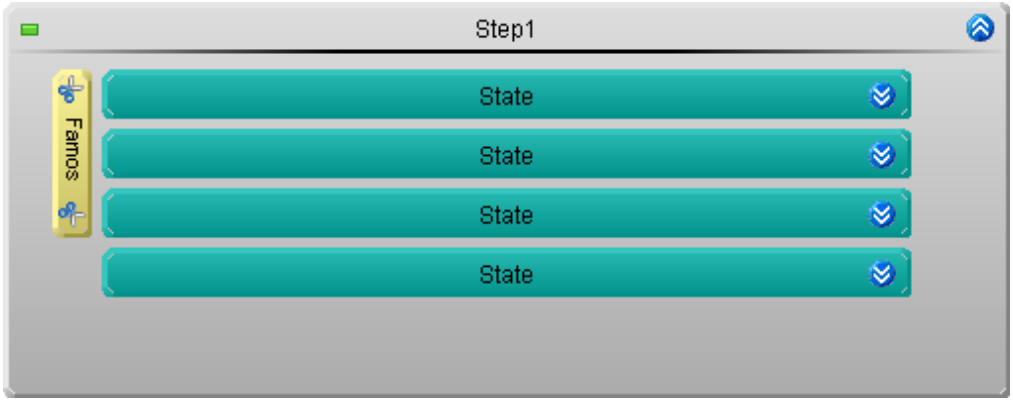
The prerequisite is installation of imc FAMOS on the same (see "*Technical data sheet*" > "*Additional imc Software Products*").

Reference — **Functions**

Information on imc FAMOS' scope of functions is presented in the separate imc FAMOS manual

Reference — **imc FAMOS Dialog**

More information on the imc FAMOS dialog is provided in the section:
"*Data analysis and signal processing*" > "*imc FAMOS dialog* 1088".



Analysis with imc FAMOS

Note

Immediate trigger

The PC interaction can only be performed if the device has at least one active channel with a "immediate trigger" (e.g. the "*BaseTrigger*"). "Immediate triggers" are triggers which fire upon starting the measurement, and have no other associated sources.

Block "*Terminate*"

The "Interactions (device/PC)" in the block "*Terminate*" are no longer performed if the block "*Terminate*" is performed due to the end of the measurement.

Positioning

The element imc FAMOS Automation can be located to the **left beside states, loops and If-branches**. **Only one device/PC interaction** can be to the left of these elements.

imc FAMOS Automation can perform analysis over the space of one element's time frame, or across multiple elements.

Parallel analyses by means of separate "*tracks* 1587" are possible.

Execution

"*imc FAMOS Automation*" starts the automation **after** the last element, with which "*imc FAMOS Automation*" is connected, has finished. Only then are all the data available.

What data are transferred?

In contrast to the other plug-ins, **not all data measured ate transferred to imc FAMOS**.

The **data measured which accrued within the connected elements** are transferred to imc FAMOS. The data are cut out: referred to below as "***Cutting***". For this reason, **no source** can be selected.

Example

imc FAMOS Automation is waiting for two states. Channel_001 is transferred to imc FAMOS.

- *State 1* lasts 5 seconds
- *State 2* lasts 2 seconds
- Once *State 2* has been concluded, **the last 7 seconds** before Channel_001 are cut and transferred to imc FAMOS for analysis.

Note

Data types

Only single values and equidistantly sampled channels can be transferred; these include the "*Analog channels*", for example.

What are the consequences when a channel contains no data?

If the point in time arrives when the data cut is to be concluded and there are no data at the channel in the pertinent time frame, a warning is posted: "*Waiting for imc FAMOS data from channel '...'; please check your configuration.*".

Timing

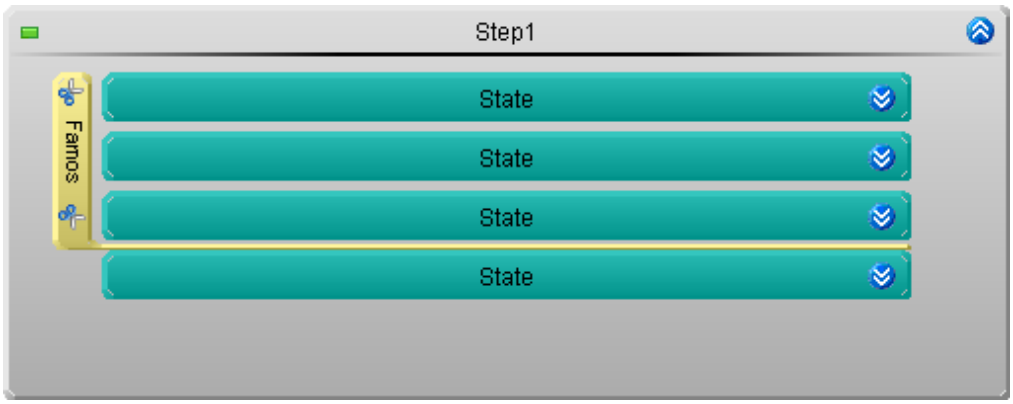
The element "*imc FAMOS Automation*" **does not take up any of the device's cycle times**. The analysis is performed by the PC. Thus, the analysis starts and **the next state is carried out immediately** although the analysis is still running. (See also the notes in the section: "*Information and Tips* 1097")

Synchronization

However, in some cases it is necessary for the **results to be available before the next state** can be performed. In order to prevent the next state from starting before the imc FAMOS analysis has finished, there are two possibilities:

- open the element's context menu and select "*Synchronization*", or
- open the imc FAMOS dialog and select "*Synchronized event*"

Now the next state only starts once the imc FAMOS analysis is finished. This is indicated graphically by a **synchronization line below the element**.



Synchronized analysis with imc FAMOS

Note

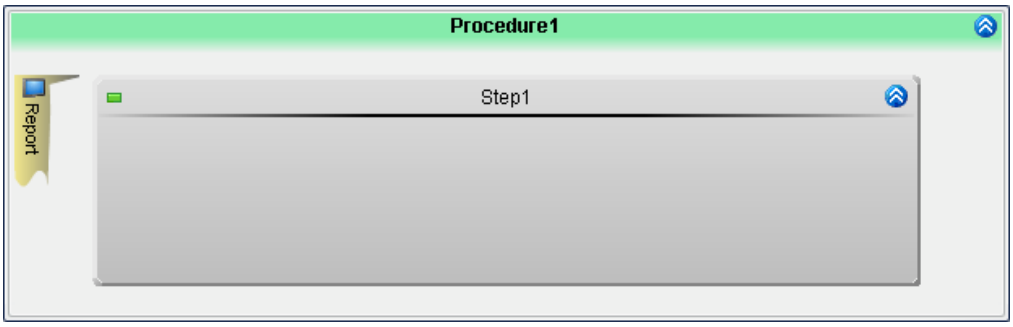
The duration of the analysis depends on the sequence and the computer. In this case, it is not possible to say when the next state will start.

Opening the imc FAMOS dialog and making settings

To open the imc FAMOS Dialog 1088, double-click the left mouse button over the element.

14.6.7.3 Page through Panel

Steps can be linked with a Panel page, which is displayed with the step starts.



Example: The view switches to the Panel-page "Report" when the step starts

Note

Immediate trigger

The PC interaction can only be performed if the device has at least one active channel with a "immediate trigger" (e.g. the "*BaseTrigger*"). "Immediate triggers" are triggers which fire upon starting the measurement, and have no other associated sources.

Block "*Terminate*"

The "Interactions (device/PC)" in the block "*Terminate*" are no longer performed if the block "*Terminate*" is performed due to the end of the measurement.

Positioning

The element "*Page through Panel*" can be positioned to the **left of a step**. **Only one device/PC interaction** can be to the left of these elements.

Execution

When starting the step, the view switches to the Panel page set.

Timing

The element "*Page through Panel*" **does not use up any of the device's cycle times**. The view switching is on the PC. Thus, the page is turned and **the task continues without any limitations**.

Selecting a Panel page

If the element is added, it is not linked with any Panel page. For linking the element with a Panel page, there are multiple possibilities:

- open the element's context menu and select an already existing page by selecting the item "*Page through Panel*", or
- open the element's context menu and select "*Create a new Panel page*". A "*standard dialog*" is created in the Panel.
- double-click on the element. A "*standard dialog*" is created in the Panel.

The element is now linked with the Panel page.

By means of the context menu, you can now

- select another Panel page, or
- open the selected Panel page

Reference	Panel

The description of the Panel pages is presented in the description of the plug-in "*Panel* 1205".

14.7 Functions

You can use functions in "*States*", "*Loops*" and "*If-branches*".

This includes functions which enable special applications to be performed in a Automation-task, as well as imc Online FAMOS functions which can be used in the control command "`OnSyncTask`".

Operation

Adding a function into the active element (e.g. to a state description):

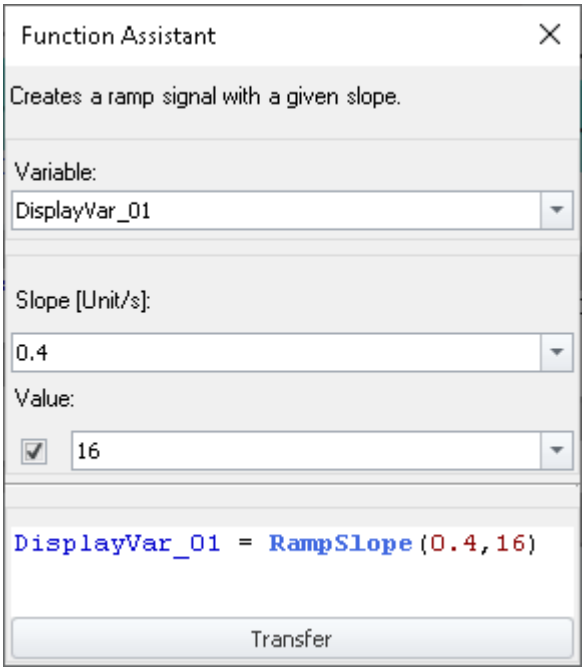
- Double-click on the function (the input box must be selected), or
- use drag&drop, or
- use the "*Functions Assistant*" (the input box must be selected), or
- enter the function (the keyboard combination <CTRL>+<SPACE> helps in making the entry).

Functions Assistant

The Functions Assistant helps you make entries.

- Select the target location in which to insert the function
- Select the function and click on "*Assistant*"
- In the Assistant, configure the function and subsequently transfer it by means of "*Transfer*"

The function is inserted at the location selected.



Functions Assistant
In the bottom line, the result is presented

Function description

The next sections provide descriptions of the groups: "*Signals*", "*Timers*" and "*Tolerances*". The explanations of the other functions are provided in the Help-window below the Assistant, or in the functions reference of the imc Online FAMOS-functions.

Function group	Description
Signals	You can generate ramp signals which depend on the signal slope or the target time.
Timer	Use timers, for example, in order to prolong certain states.
Tolerances	You can test whether signals have adhered to limits or have achieved a stable level.

14.7.1 Signals

Note Notes on using ramps

A ramp function only needs to be launched once; the task itself does not need to remain in the associated state until the ramp is finished.

Instead the **ramp continues running until it reaches its final value** or it is stopped/interrupted. **The ramp function is not automatically stopped**, either by the end of the task or the **end of the measurement**. To be on the safe side, ensure that all ramps are finished at the latest in the Terminate-block, in case these functions affect any external elements.

Internally, the ramp is calculated in a "`OnSyncTask`" at a cycle time of 1 ms. The individual steps of the ramp are derived from the results. Please note that only five tasks can be created in the Automation. If there are already five tasks present, the cycle time of one of those tasks is applied as the ramp's cycle time. In order to preserve the precision level, the next shorter cycle time available is selected. But if the cycle times of all five tasks are longer, the next longer one available is applied, which results in larger ramp steps.

The slope or time periods can be calculated as follows:

RampTime

Slope = (Final value - Initial value) / Time
Example: Slope = (20 V - 10 V) / 100 s = 0.1 V/s

Slope per cycle step = (Final value - Initial value) * cycle time / Time
Example: Slope per cycle step = (20 V - 10 V) * 1 ms / 100 s = 0.0001 V

RampSlope

Slope per cycle step = Slope * Cycle time
Example: Slope per cycle step = 0.1 V/s * 1ms = 0.0001 V

Time = (Final value - Initial value) / Slope
Example: Time = (20 V - 10 V) / 0.1 V/s = 100s

Reference Tutorial

You will find a detailed example of the use of the ramp function in the tutorial: "*Actuator-control - Ramps* 1657".

RampSlope: Generating a ramp of defined slope

Variable = RampSlope (Slope, Final value)

Variable: Variable whose values are to be outputted in a ramp shape **Slope**: the ramp's slope [unit/s]	Optional parameter: **Final value**: Final value which is held

Generates a ramp from the variable's current value and having a defined "*Slope*". The ramp signal shape is imposed on the variable assigned.

If no final value is defined for the ramp, the value increases until the ramp is stopped ("`StopSignal`") or paused ("`PauseSignal`").

The function "`OnRampEnd`" is used to query whether the end of the ramp has been reached yet.

RampTime: Generating a ramp which reaches its end after a defined amount of time

Variable = RampTime (Final value, Time [s])

Variable: Variable whose values are to be outputted in a ramp shape	**Final value**: Final value of signal ramp **Time**: Time in seconds

Generates a ramp from the variable's current value up to the specified "*Final value*". The slope is calculated such that the final value is reached after elapse of the parameter "*Time*". Then the "*Final value*" is held.

The ramp signal shape is imposed on the variable assigned.

The function "`OnRampEnd`" is used to query whether the end of the ramp has been reached yet.

Note Ramp with high slope

Be aware that ramps generated by this function can have varying slopes.
If the initial and the final values lie far apart, the ramp's slope may become very high. If the signal is outputted by a DAC, please make sure that any mechanical equipment which moves in response is able to handle this slope rate.

OnRampEnd: Has the end of the ramp been reached?

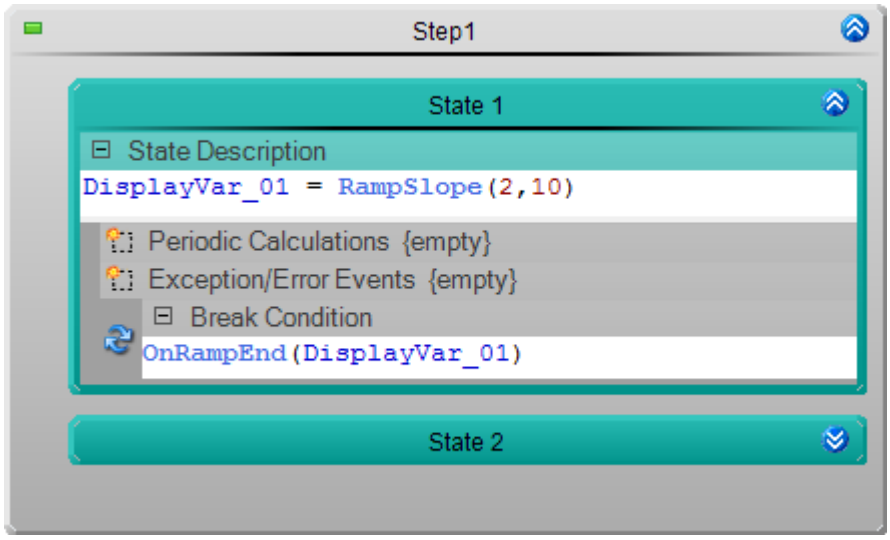
Result = OnRampEnd (Variable)

Result: 1, if the "*Final value*" is already reached, else 0	**Variable**: Variable assigned as the signal

The function tests whether a ramp has reached its "*Final value*". The ramp is previously started with either of the functions "`RampTime`" or "`RampSlope`".

Example Holding a state until the ramp is concluded

The state is only ended once the ramp has concluded.



The ramp is started with "*State 1*": Slope 2; Final value 10. According to the break condition, the state is only ended when the function "`OnRampEnd`" returns a 1, which means when the ramp has concluded.

PauseSignal: Pausing the signal

Variable = PauseSignal (Value)

Variable: Variable assigned as the signal	Optional parameter: **Value**: Instead of the last value, this value is held

Interrupts the process of generating a ramp/signal in the variable. The last value is held, or optionally a substitute value can be outputted.

By means of the function "`ResumeSignal`", it is possible to resume generating the signal.

ResumeSignal: Resuming the signal

Variable = ResumeSignal ()

Variable: Variable assigned as the signal

Resumes the process of generating a ramp/signal in the variable, which had been interrupted by the function "`PauseSignal`".

StopSignal: Stopping the signal

Variable = StopSignal (Value)

Variable: Variable assigned as the signal	Optional parameter: **Value**: Instead of the last value, this value is held

Stops the process of generating a ramp/signal in the variable. The last value is held, or optionally a substitute value can be outputted.

14.7.2 Timer

InitTimout: Starting a local timer

InitTimout (Duration [s])

Duration: A local timer clocks the duration specified, in seconds.

The function "`OnTimeout`" is used to query the timer's state.

Example illustrating a "*State*":

In the state description, you start the timer with: `InitTimout (DisplayVar_01)`

In the break condition, you wait until the timer is finished by using: "`OnTimeout`"

Recommendation: To start the timer, use the function "`OnTimeout`".

OnTimeout: Starting and/or checking a local timer

Result = OnTimeout (Duration [s])

Result: 1, if clocked duration elapsed, else 0	Optional parameter: **Duration**: Local timer clocks the duration specified, in seconds.

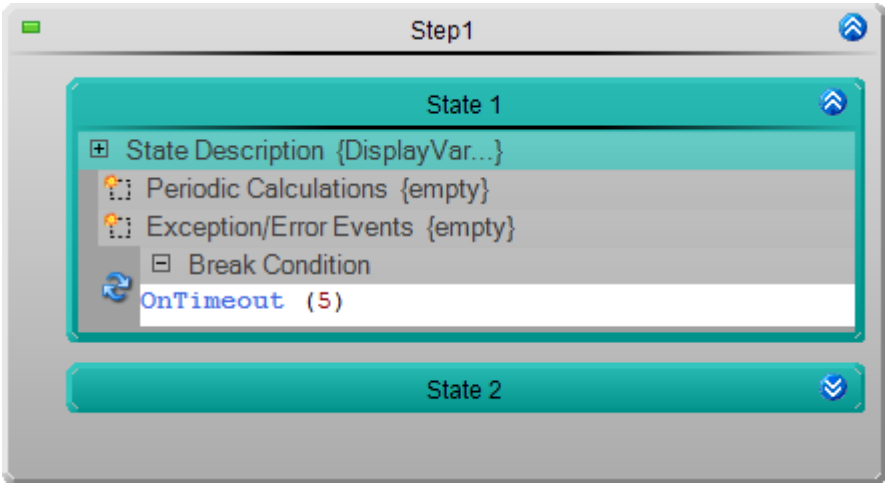
The function tests whether the local timer has already clocked the specified duration.

The timer had previously been started using the function "`InitTimeout`".

Recommendation: Optionally, a "*Duration*" can be defined. Thus the function itself starts a local timer. The function "`InitTimeout`" is not required.

Example — Retaining a state until a defined time has elapsed

The state is only ended after 5 seconds.



InitTimer: Starting a global timer

Result = InitTimer ("Name", Duration [s])

Result: 1, if clocked duration elapsed; else 0	**Duration**: Clocked duration in seconds

A global timer is started. It can be addressed by means of its "*Name*".

Once the timer's clocked duration is elapsed, it can be restarted by means of the function "`RestartTimer`".

OnTimer: Testing the global timer

Result = OnTimer ("Name")

Result: 1, if clocked duration elapsed; else 0	**Name**: Designation of the timer

The function tests whether the duration which the global timer "*Name*" clocks has already lapsed.

The timer is previously started by the function "`InitTimer`".

RestartTimer: Restarting the global timer

RestartTimer ("Name", Duration [s])

Name: Designation of the timer	Optional parameter: **Duration**: Global timer clocks the new duration specified, in seconds.

The global timer "*Name*" is re-started. Optionally, a new duration can also be specified.

The timer must previously have already been run at least once by means of the function "`InitTimer`".

StopTimer: Stopping the global timer

StopTimer ("Name")

Name: Designation of the timer

The global timer "*Name*" is stopped.

14.7.3 Tolerances

Note Tolerances

Tolerances can only be checked if the functions are called at regular intervals; e.g in a break condition. A one-time call of the function returns a "0". The function must then be called at least a second time after elapse of the minimum dwell time. In order to be able to use the function effectively, it should be called very often within the dwell time.

Only the current value is evaluated. If the input variable violates the condition between the test steps, this is not captured. Optimally, the test would be repeated in accordance with the sampling rate of the input variable.

The maximum dwell time (*t2*) should not lie below the test's repetition interval.

OnAboveLevel: Is the boundary value exceeded?

Result = OnAboveLevel (Channel, Level, t1, t2, t3)

Result: 1, if the condition is met; else 0 **Channel**: Input-variable **Level**: Boundary value for the test	Optional parameters: **t1**: The minimum dwell time [s] above the boundary to meet the condition **t2**: The maximum dwell time [s] below the boundary which the test against the condition tolerates **t3**: Duration in [s] before the function begins testing against the condition

The function checks whether the current value of "*Channel*" exceeds the "*Level*".

OnBelowLevel: Has the boundary value been violated?

Result = OnBelowLevel (Channel, Level, t1, t2, t3)

Result: 1, if the condition is met; else 0 **Channel**: Input-variable **Level**: Boundary value for the test	Optional parameters: **t1**: The minimum dwell time [s] below the boundary to meet the condition **t2**: The maximum dwell time [s] above the boundary which the test against the condition tolerates **t3**: Duration in [s] before the function begins testing against the condition

The function checks whether the current value of "*Channel*" falls below the "*Level*".

OnInsideRange: Does the value lie within a range?

Result = OnInsideRange (Channel, Reference, Delta1, Delta2, t1, t2, t3)

Result: 1, if the condition is met; else 0 **Channel**: Input-variable **Reference**: Reference value within the boundaries **Delta1**: Defines the upper boundary: Positive distance from the reference value	Optional parameters: **Delta2**: defines the lower boundary: Negative distance from the reference value **t1**: minimum dwell time [s] within the boundaries to meet the condition **t2**: maximum dwell time [s] outside of the boundaries which the test against the condition tolerates **t3**: duration in [s] before the function begins testing against the condition

The function checks whether the current value of "*Channel*" lies within the boundaries.

The boundaries are defined as follows:

upper boundary = Reference + Delta1

lower boundary = Reference - Delta2

If "*Delta2*" is not defined, "*Reference*" is the lower boundary.

OnOutsideRange: Does the value lie outside of a range?

Result = OnOutsideRange (Channel, Reference, Delta1, Delta2, t1, t2, t3)

Result: 1, if the condition is met; else 0 **Channel**: Input-variable **Reference**: Reference value within the boundaries **Delta1**: Defines the upper boundary: Positive distance from the reference value	Optional parameters: **Delta2**: defines the lower boundary: Negative distance from the reference value **t1**: minimum dwell time [s] outside of the boundaries to meet the condition **t2**: maximum dwell time [s] within the boundaries which the test against the condition tolerates **t3**: duration in [s] before the function begins testing against the condition

The function checks whether the current value of "*Channel*" falls outside of the boundaries.

The boundaries are defined as follows:

upper boundary = Reference + Delta1

lower boundary = Reference - Delta2

If "*Delta2*" is not defined, "*Reference*" is the lower boundary.

OnStable: Has the value stabilized?

Result = OnStable (Channel, Delta, t1, t2, t3)

Result: 1, if the condition is met; else 0	Optional parameters:
Channel: Input-variable	**t2**: maximum dwell time [s] outside of the boundaries which the test against the condition tolerates
Delta: defines the upper and lower boundaries	**t3**: duration in [s] before the function begins testing against the condition
t1: The minimum dwell time [s] within the boundaries to meet the condition	

The function checks whether the current value of "*Channel*" lies within the boundaries.

The boundaries are defined as follows:

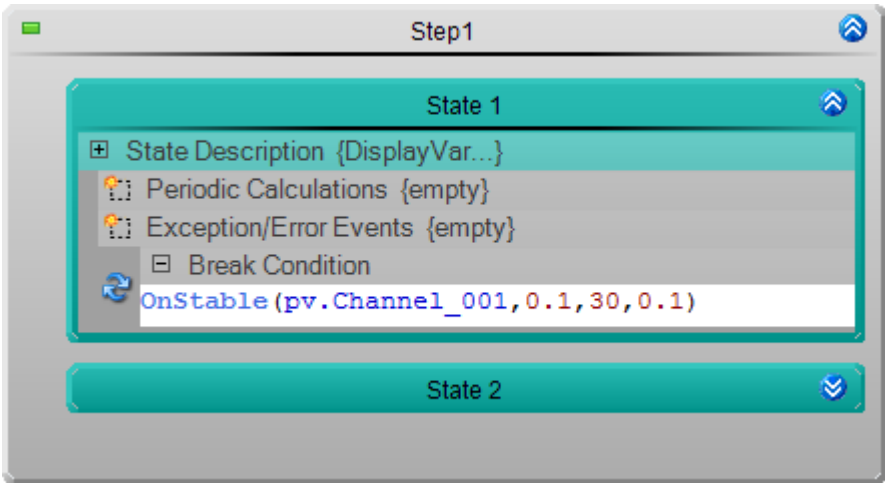
upper boundary = Channel + Delta
lower boundary = Channel - Delta

The boundaries are determined upon the first call of the function.

If a boundary value violation is detected, new boundaries are determined. If a time value is specified for "*t3*", the function waits again before performing the test.

Example — Holding a state until the input signal stabilizes

The state only ends once the input signal has stopped fluctuating for a relatively long time period.



When the function is run the first time, the boundaries are determined. The current value of the channel ± a Delta value of "0.1".

14.8 Information and tips

14.8.1 Source Code - find and replace

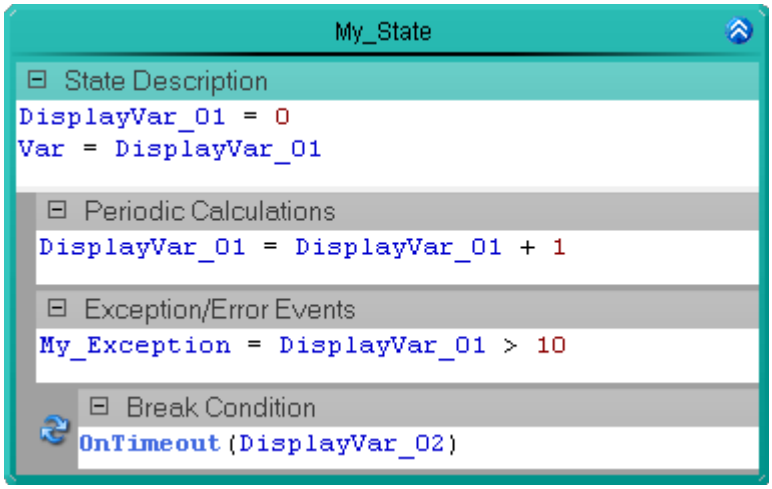
The Automation has a text search enabling texts to be found and replaced. The source code is searched within all of the elements: e.g. the description of a state or a loop's condition.

To find a text, open the dialog "*Find and Replace*".

Menu item	View
Edit > Find and Replace ( )	Complete

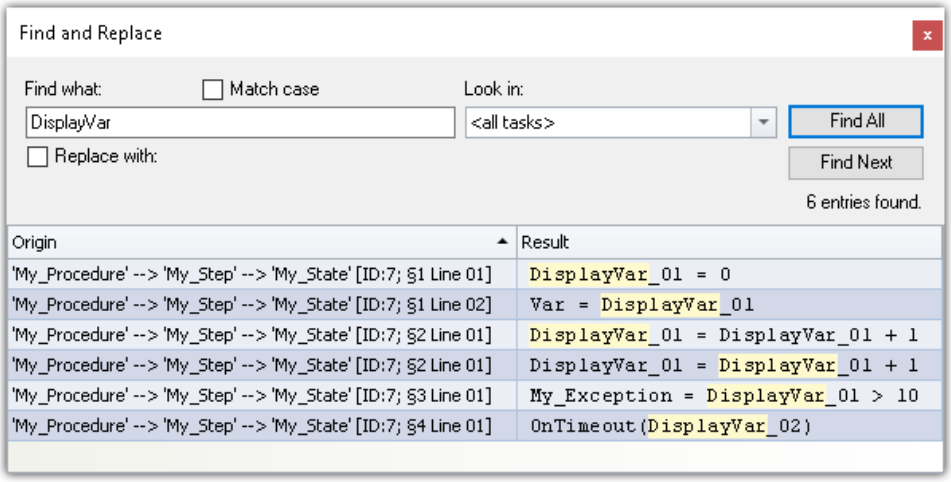
Example

The state "*My_State*" is located within "*My_Step*" and "*My_Procedure*".



State

The search is for the text: "*DisplayVar*". This text exists six times within the state.



Example of a text search

Note — Find/Replace variables in imc FAMOS sequences

imc FAMOS Sequences (data cutting): The **In/Out variables** can also be detected or replaced. The sequence and the imc FAMOS-variables are not taken into account.

Search

To find a text:

- enter a text to find in the box under "*Find what*".
- click the button "*Find all*".

Search option	Description
Match case	Takes the exact spelling in regard to capitalization into account in the search.
Task-limited search	Either all tasks or individual tasks can be searched. Select the appropriate entry in the selection list below "**Find in:**" <all tasks> — Search is performed in all tasks Name of task — Search is performed in selected task

Replacing text

The text search enables simultaneous replacement of the words.

To replace texts

- Click in the checkbox "*Replace with*"
- Enter a text in the box below "*Replace with*"
- Click on the button "*Find all*".
- Select any entry found and click on the button "*Replace*", or click on the button "*Replace all*".

Navigate - Find next

To navigate to the entries found, there are several possibilities:

- Double-click the mouse button over an entry
- Click on the button "*Find next*". In this case, the system navigates to the next entry.

The view switches to the position of the selected entry and the entry is selected.

Column: Origin

In the column Origin, the location where the entry was found is indicated.

Example: The text was found in the text box: Cyclical calculation of a state.

- 'Procedure-Name' --> 'Step-Name' --> 'State-Name' [ID: 11; §2 Line 07]

In this case, the first three names specify the element's position.

The content in brackets [] states the position within the element:

ID	Internal number of the element (depends on the setting)
$	Cell position: • 1: state description • 2: cyclical calculation • 3: exception- and error handling • 4: break condition
Line	line within the cell

14.8.2 Trace info

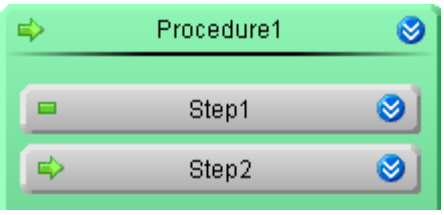
The trace info **displays during a task's execution** which "*procedure*" and which "*step*" is **currently being processed**. To activate/deactivate the trace info for the task, click on the corresponding checkbox in the Task Management's area Tasks. By default, this option is active.

While the task is running, an arrow symbol ( ) appears at left next to the name at the current position.

There are always two arrow-symbols:

- one at the current procedure and
- one at the current step in the procedure.

If a **step** is not currently running, it has a bar symbol ( )



Arrow symbol for the Trace Info

Displaying the current status on the Panel page

You can extract a task's current state from a variable.

Variable: <Task-Name>.Trace

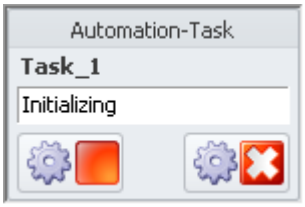
This returns a multi-line text indicating where the task is currently located.

Structure:

Task-name:
 Procedure-name
  Step-name
  State-name

14.8.3 Widget - Automation task

Using the Widget "*Automation Task*" gives you the ability to intervene in a running program. To do this, link the Widget with the task (variable: "*<Task-Name>*").



The Widget represents the current state; e.g. "*Initializing*", "*Running*", "*Terminating*", "*Done*", ...

Additionally, you can use buttons to start, close or halt the task:

Action	Description
Start	Re-starts the task once it has finished.
Finish (Close)	Interrupts the current procedure/exception handling. The block Terminate is run unless an error handler is currently running.
Stop (Halt)	Ends execution of the task without executing any more actions. No Terminate-phase is run. Any running error handlers are also interrupted.

14.8.4 Report of Automation-configuration

You can **create a report** of the **current Automation-configuration**. The report is opened in the Browser and can be **saved or printed** there. Start the report via the menu ribbon if Automation is open.

Ribbon	View
Edit > Print ( )	Complete
Edit > Print Preview ( )	Complete
Home > Print ( )	Standard

There are three varieties available extending from a broadly outlined structure to the contents with the source text of all elements. Two examples are presented below.

Init / Terminate

Block	**Init**	
	Step	Step 1
	Next	Terminate

Block	**Terminate**	
	Next	Halt

Procedures

Block	**Prozedure 1**	
	Step	Step 2
	Step	Step 3
	Next	Terminate

User-defined variables

Name	Type	Default Value	Persitent
Variable_1	Float	NaN	false
pv.Variable_2	Float	NaN	true

Limit value monitoring

Name	Processing Order	Channel	Exception	Event Type	Lower Limit	Upper Limit	Duration	Disable
Limit_1	1	pv.Channel_001	true	EXCESS_LIMIT		12	0	0

Example of a report
Overview of structure, without contents

Init

Step	**Step 1**
State	State 1 [Event: ID 9] Event: After (State 1)
Once	DisplayVar_01 = 1
State	State 2 [Begin: FAMOS Sequence ID 16] [End: FAMOS Sequence ID 16]
Once	DisplayVar_01 = 2 DisplayVar_02 = 0
Always	DisplayVar_02 = DisplayVar_02+1
Break	DisplayVar_02 >10
Next	Terminate

Name	FAMOS Sequence ID	FAMOS Sequence Filename
Famos	16	

Name Famos

FAMOS Sequence ID 16

FAMOS Sequence Filename

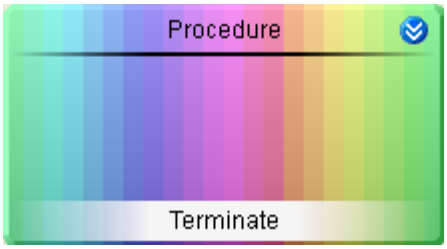
To FAMOS		From FAMOS	
Name	**Name in FAMOS**	**Name**	**Name in FAMOS**
Channel_001	Channel_001	MeanValue	MeanValue

Source Code
MeanValue = mean (Channel_001)

Example of a report (showing small excerpt)
A more exact listing of all elements and their contents

14.8.5 Procedures colors

Each procedure is distinguished by its name and a color. When a new procedure is created, it is assigned a new color. There are 22 pre-defined colors.



The 22 colors for the procedures

To change the colors later, proceed as follows:

- Open the procedure
- Open the context menu of the procedure
- Select the item "*Select procedure's color*"

Subsequently, the color selection window appears. Select the desired color and click on the "*OK*"-button.

14.9 Tutorial

Below we present some examples of the plug-in **Automation**. The examples assume that the plug-ins **Setup** 186 and **Panel** 1205 are installed.

A prerequisite for operation of the plug-in **Automation** is an imc measurement device with imc Online FAMOS Professional installed.

- Blinking LED using timer 1628
- Blinking LED using Cycle Time 1636
- Traffic light 1642
- Limit value monitoring 1648
- Actuator-control - Ramps 1657

14.9.1 Blinking LED using timer

Assignment:

Set up two LEDs on the Panel page and have them flash in alternation. The flash duration should be adjustable by means of a potentiometer.

Create for this purpose a **Task** and use an appropriate **timer function** from among the **automation controls functions**.

Learning goals:

- Creation of a **task**
- Use of **timer functions**
- Use of **status descriptions** and **break conditions**

Elements used:

- Init / Terminate
- Procedures
- Steps
- States (State description and Break conditions)
- Next branching
- While loops

Additional plug-ins used:

- Setup
- Panel

Procedure:

Start imc STUDIO as described in the chapter Start 31.

14.9.1.1 Setup - Settings

To begin, the measurement device is selected and configured.

- Open the **device configuration page**: "*Setup*".
- Go to the **tab**: "*Devices*" .

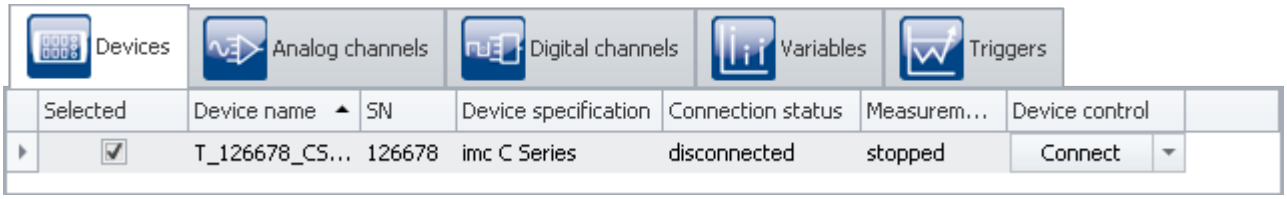
Here you will find any devices which are known to the system due to having been used before. Following the first installation of imc STUDIO, the list is empty.

Select your device:

- If the device desired **does not appear in the list**, perform a device search 241 ( ). Subsequently, the list will display all devices found.

Ribbon	View
Home > Search for devices ( )	all
Setup-Control > Search for devices ( )	Complete

- Click on the Checkbox-symbol ( ) in order **to select the device**.



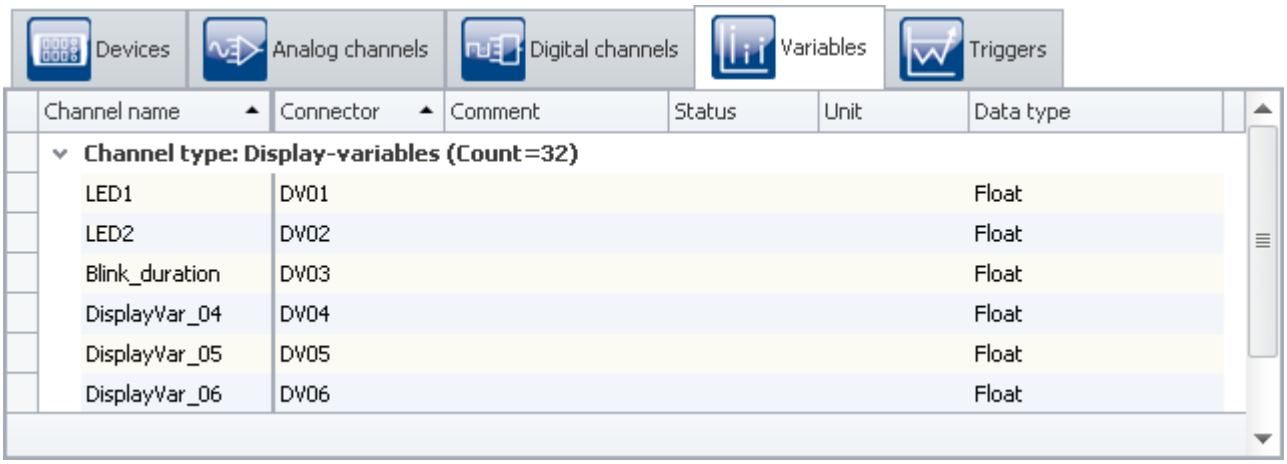
Selecting a device

The device is now known and selected, and is available each time imc STUDIO is started.

Configuring the variables

Now we assign appropriate names to the variables used.

- Go to the **tab**: "*Variables*" .
- Change the name of the **Display variables** 01, 02 and 03 to *LED1*, *LED2* and *Blink_duration.*



Assigning a name

Note

In order to prevent the **automation** from ending prematurely, make sure that the first channel's (***Channel_001***) **measurement duration** is set to *undefined* (=infinite).

- To **apply the changes**, click on the button "*Process configuration* 241" ( ).

Ribbon	View
Home > Process (✔)	all
Setup-Control > Process (✔)	Complete

By this means, the device settings are available to other components.

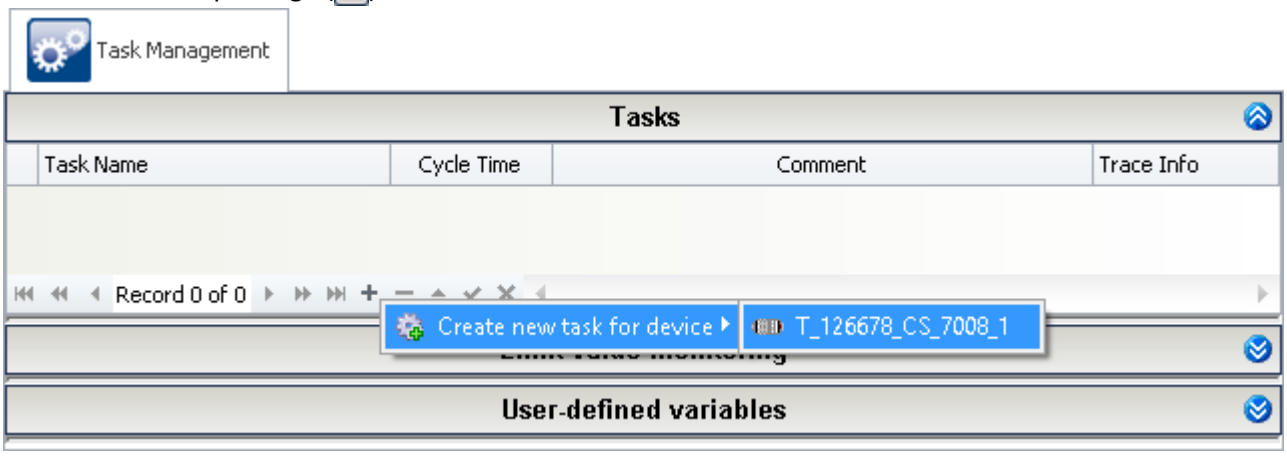
14.9.1.2 Automation - Settings

Now it is possible to set up the **Task**.

- Open the page: "*Automation*".

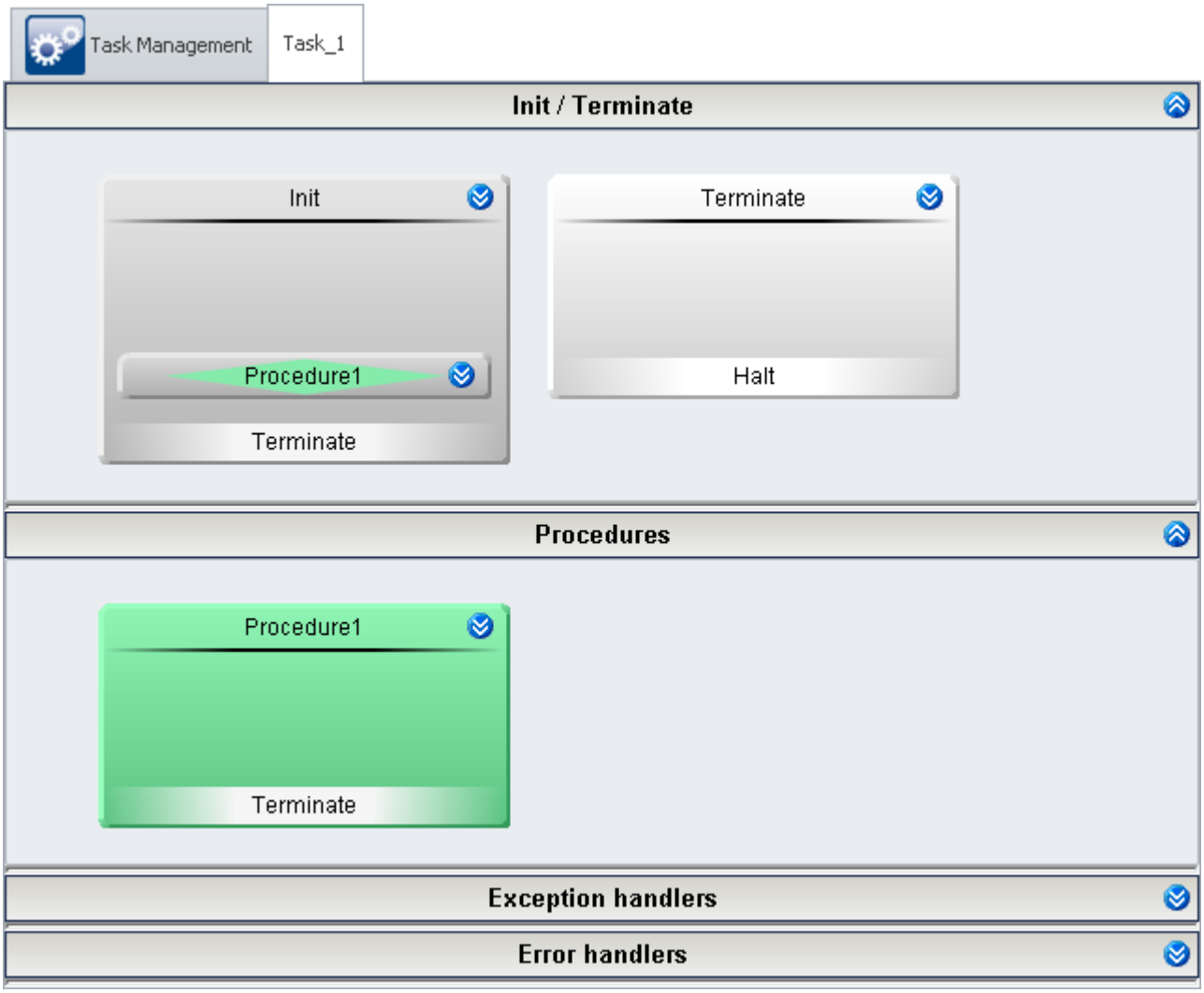
You are now in the **Task Management** interface. Proceed with creating a **Task**.

- Click on the plus-sign (+) and select the **Measurement device**.



Task Management: Creating a task for the measurement device

Along with the **Task Management**, a new tab for the **Task** created appears. This takes you to the **Task Editor**. Open it next.



The Task Editor for "Task_1"

Init

In **Init**, create the initial values for the *LEDs* and the *Blink duration*.

Create a **Step**:

- Drag from the **Templates** a **Step** in the **Initialization** or open the context menu of the **Initialization** and select *New, Step*.

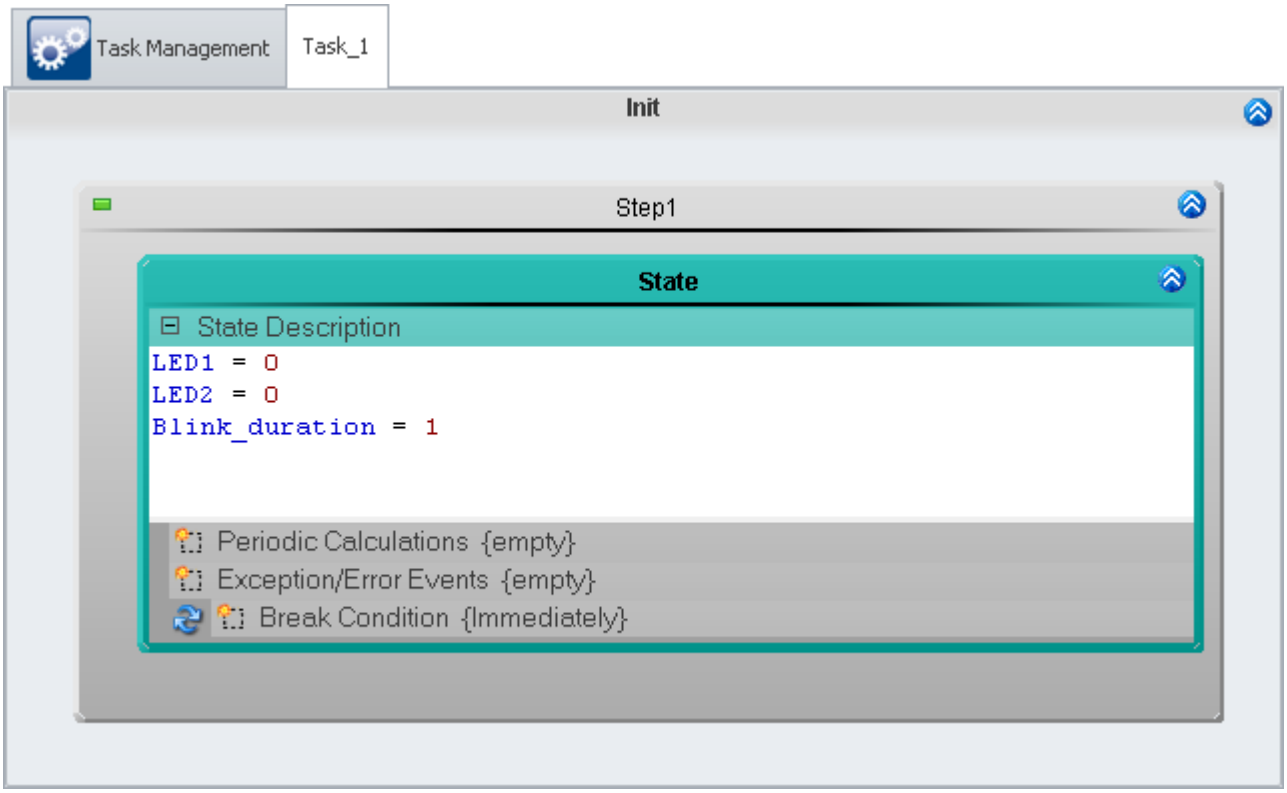
Open this **Step** by clicking on the double arrow ( ). You arrive at the **Block Editor**.

In the **Step**, add a **State**.

- Drag a **State** from the **Templates** to the **Step** or open the **Step's** context menu and select *New, State*.
- Open this **State** by clicking on the double-arrow ( ).

In the **State description** enter the initial values.

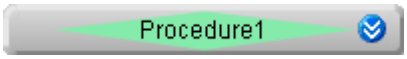
```
LED1 = 0
LED2 = 0
Blink_duration = 1
```



Setting the initial values

Close the **Block Editor** by clicking on the double-arrow ( ) in the Init dialog.

Once having concluded the **Init(ialization)**, next run *Procedure1*. In order to make this skip, a **Next-branch** is used, which is already provided.

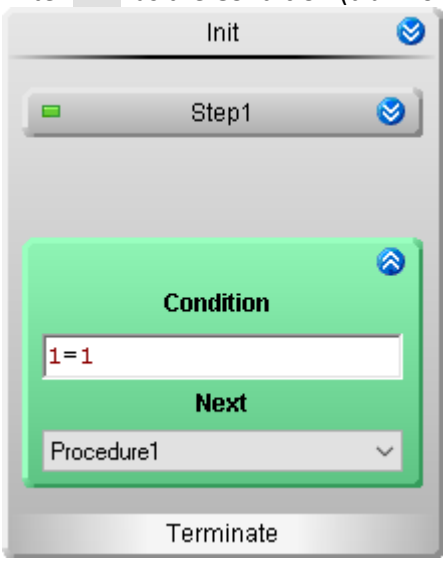


Branching to Procedure1

Open the **Branch** by clicking on the double-arrow ( ).

In the **Branch's** upper input box, the **Condition** for skipping (e.g.: `1=1` is true, in which case the skip is always made) is stated. In the lower box, the target procedure is stated.

- Enter `1=1` as the **Condition** (*blank* or just `1` is are also interpreted as true).



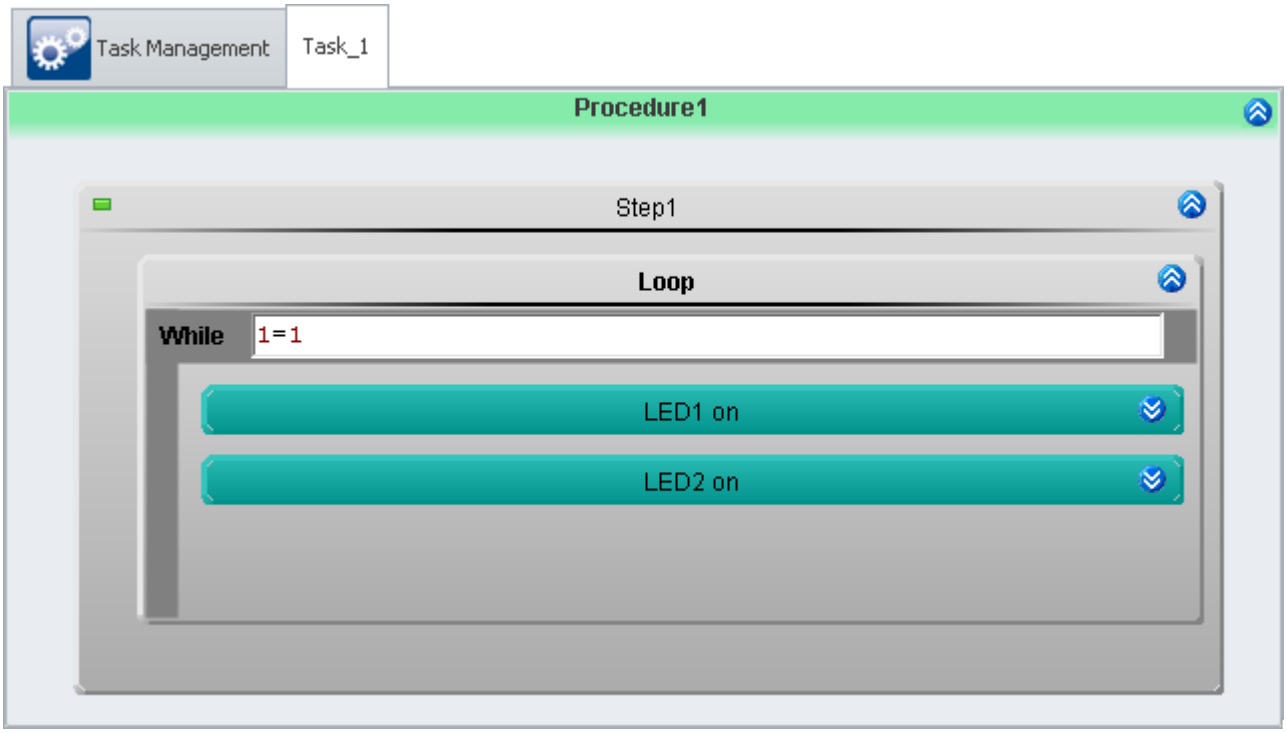
Skip from block "Init"
to the block "Procedure1"

Procedure

Open *Procedure1* by clicking on the double-arrow ( ). This takes you to the **Block Editor** of this **Procedure**.

Here is where the actual routine is programmed.

- Add a **Step**
- Open the **Step**
- Inside this step, add a **While-loop**
- Open the **loop**
- Set the loop's **condition** to `1=1` (this means the condition is always fulfilled and thus that the loop's instructions are always carried out).
- Add two **states** to the loop
- The first **state** is called *LED1 on* and the second is *LED2 on*. To do this, open the **state's** context menu and select *Rename*.



Sequence, while loop and two states

Open the **State** *LED1 on*. In this **State**, what should happen is:

- the *LED1* should go on and
- the *LED2* should go off.

The **State** should be exited once the *Blink duration* set has elapsed.

Enter in the box **State description** the following instructions:

```
LED1 = 1
LED2 = 0
```

The box **Break condition** contains the condition for when the **State** is to be exited. Here, enter the following condition:

```
OnTimeout(Blink_duration)
```

The function `OnTimeout` starts a local timer. The timer has as its parameter the **Running time**. If the time has elapsed, the function returns `1`, otherwise `0`. The **Condition** is thus true when the timer period elapses and the **State** is exited.

Open the **State** *LED2 on*. In this **State**, what should happen is:

- *LED1* should go off and
- *LED2* should go on.

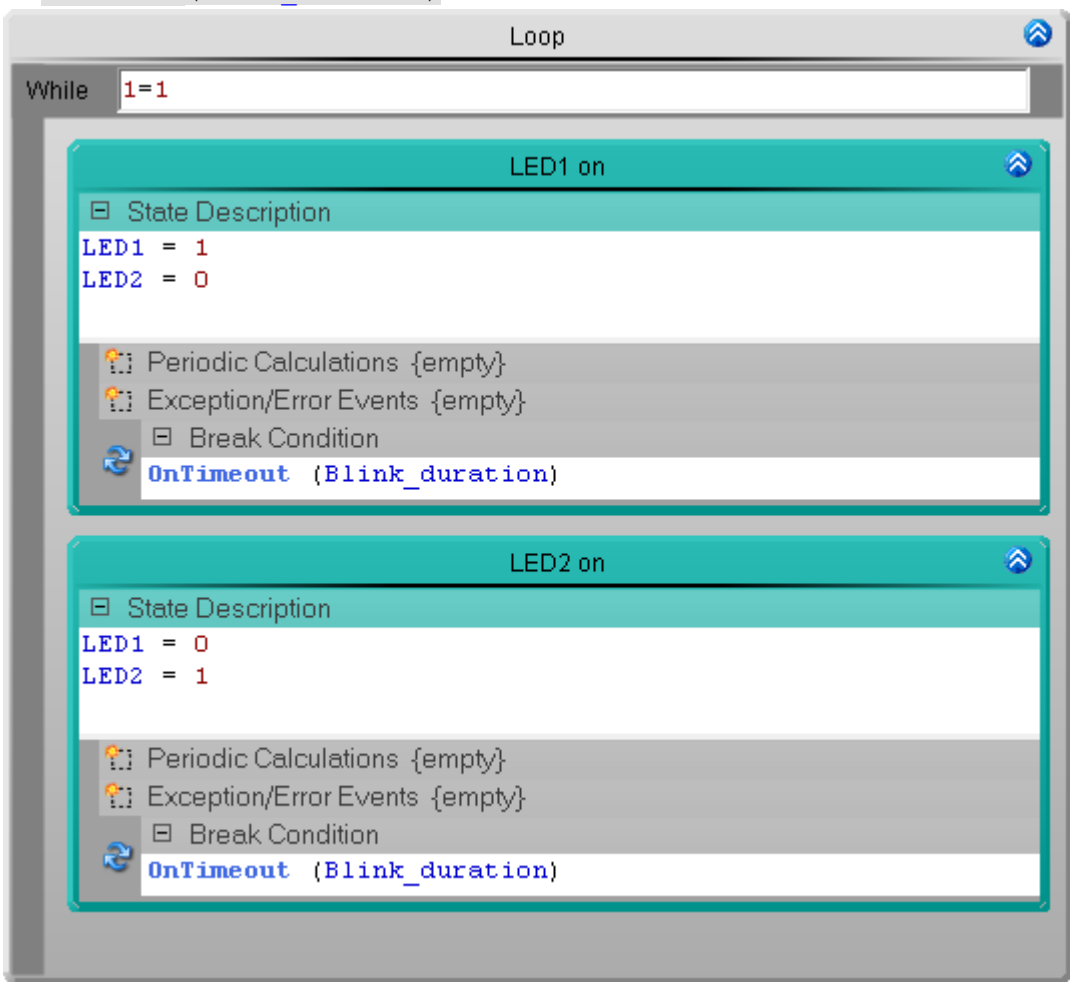
The **State** should be exited once the *Blink duration* set has elapsed.

Enter in the box **State description** the following instructions:

```
LED1 = 0
LED2 = 1
```

The **Break condition** is the same as for the **State** *LED1 on*.

```
OnTimeout(Blink_duration)
```



Source code in the states

The **State** *LED2 on* is below the **State** *LED1 on*. The **States** are processed in the order from top to bottom (unless otherwise programmed, e.g. in a **Branch**).

Check and Apply

Once the entire **Task** has been set up, the syntax of the program should be checked.

- Check and apply the changes, click on the button "*Process configuration* 241" (✔).

14.9.1.3 Panel - Settings / Start measurement

Now the Panel page can be set up.

- Open the page: "*Panel*".
- Use the Drag&Drop technique to move the **Display variables** *LED1* and *LED2* to the Panel page and display them as LEDs.
- Represent the **Display variable** *Blink_duration* as a potentiometer.
- In the potentiometer's **Properties**, an appropriate **Range** must be set, e.g. 0.5-5
- The scale **Dividors** must be set next. In this example, there are 9 scale divisions.
- For all controls, display the **Title**.
- Start the measurement 241 ( )

Ribbon	View
Home > Start ( )	all
Setup-Control > Start ( )	Complete

Test the blinking behavior for various settings of the *Blink duration*.

Panel- Blinking LED using timer (example)

14.9.2 Blinking LED using Cycle Time

Assignment:

Set up the flashing LED from the previous example Blinking LED using timer 1628 using **Cycle Time** and **User-defined variables**.

To do this, create a **Task** with an appropriate cycle time and **User-defined variables** and create **counters** which work with the **Cycle Time**.

Learning goals:

- Creation of a **task**
- Use of **Cycle Time**
- Use of **User-defined variables**
- Use of **Periodic calculations**

Elements used:

- Init / Terminate
- Procedures
- Steps
- States (State description, Periodic calculations and Break conditions)
- Next branching
- User-defined variables

Additionally used plug-ins:

- Setup
- Panel

Procedure:

Begin by starting **imc STUDIO**.

14.9.2.1 Setup - Settings

First, the measurement device is parameterized and an appropriate name is assigned to the variables used.

- Open the plug-in **Setup**.

Proceed as described in the previous example Blinking LED using timer 1628.

14.9.2.2 Automation - Settings

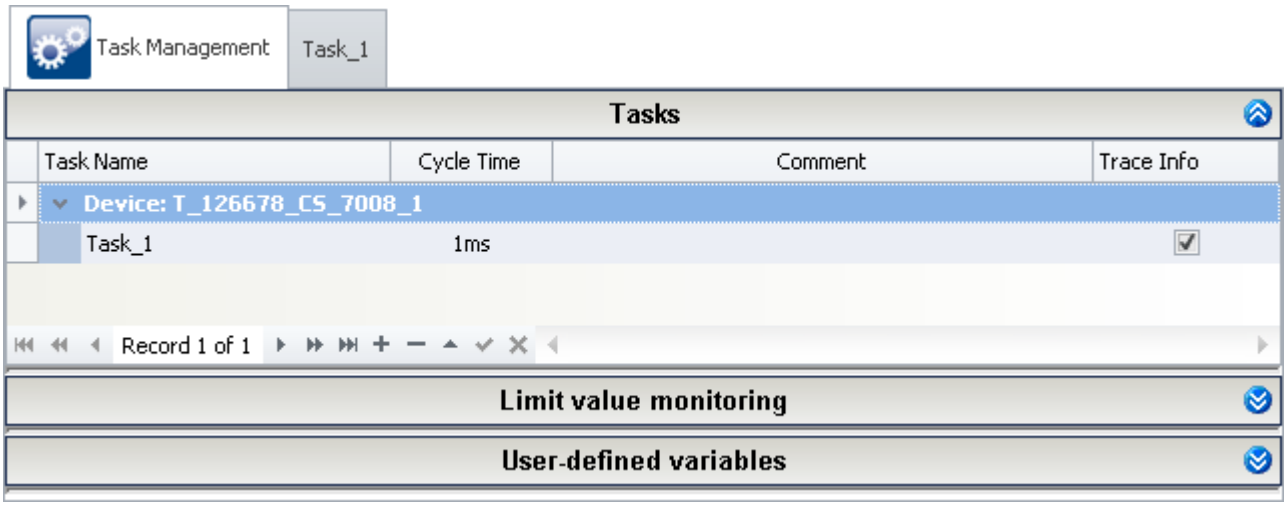
Now it is possible to set up the **Task**.

- Open the page: "*Automation*".

You are now in the **Task Management** interface. Proceed with creating a **Task**.

Tasks are performed in real-time. For this purpose, assign the **task** a **Cycle Time** (adjustment range: *100µs* through *1s*). The **Cycle Time** stands for the duration of a **state**.

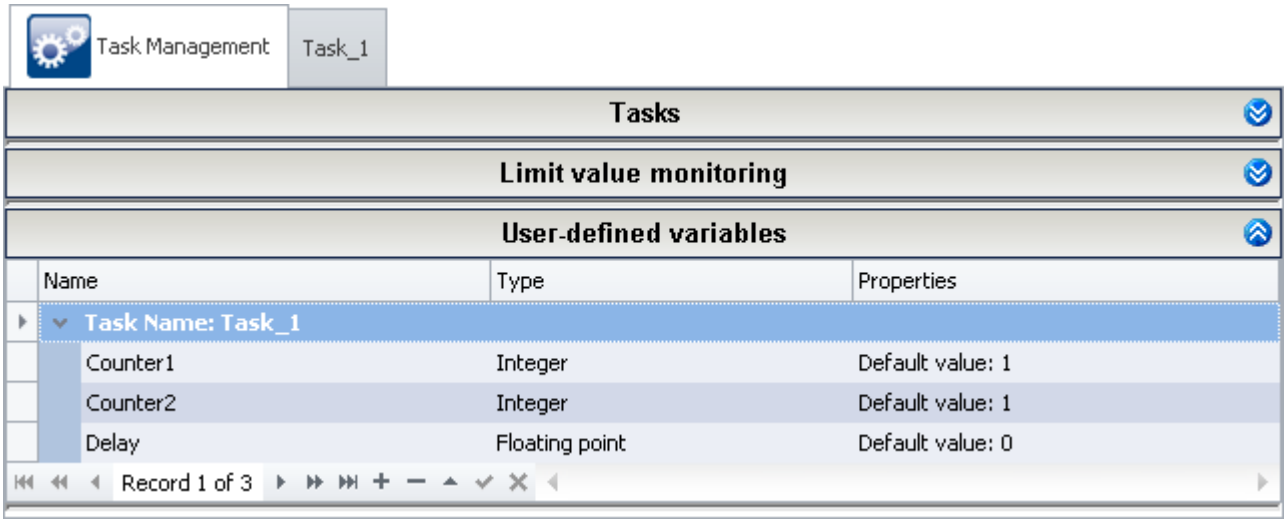
- Enter 1ms as the **Cycle Time**.



Task with a 1 ms Cycle Time

- Create three **User-defined variables**.

Name	Type	Properties
Counter1	Integer	Default value: 1
Counter2	Integer	Default value: 1
Delay	Floating point	Default value: 0



User-defined variables

Now go to the **Task Editor** (*Task_1*).

Init

In the **initialization**, set the initial values for the *LEDs* and the *Blink duration*.

Open this block by clicking on the double-arrow ( ). This takes you to the **Block Editor**.

- There, create a **step**
- In this **step**, add a **state**.
- Add the initial values in the **state description**.

```
LED1 = 0
LED2 = 0
Blink_duration = 1
```

Close the **Block Editor** by clicking the initialization's double-arrow ( ).

Open the existing **Next-branch** by clicking on the double-arrow ( ).

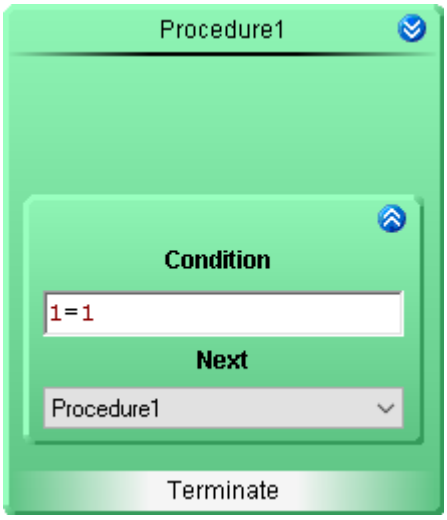
- Enter `1=1` for the **condition**.

Once the **initialization** has been run, *Procedure1* is performed automatically.

Procedure

In the example Blinking LED using timer 1628, a continual repetition is carried out by means of a **While-loop**. This can also be achieved by a **Next-branch**.

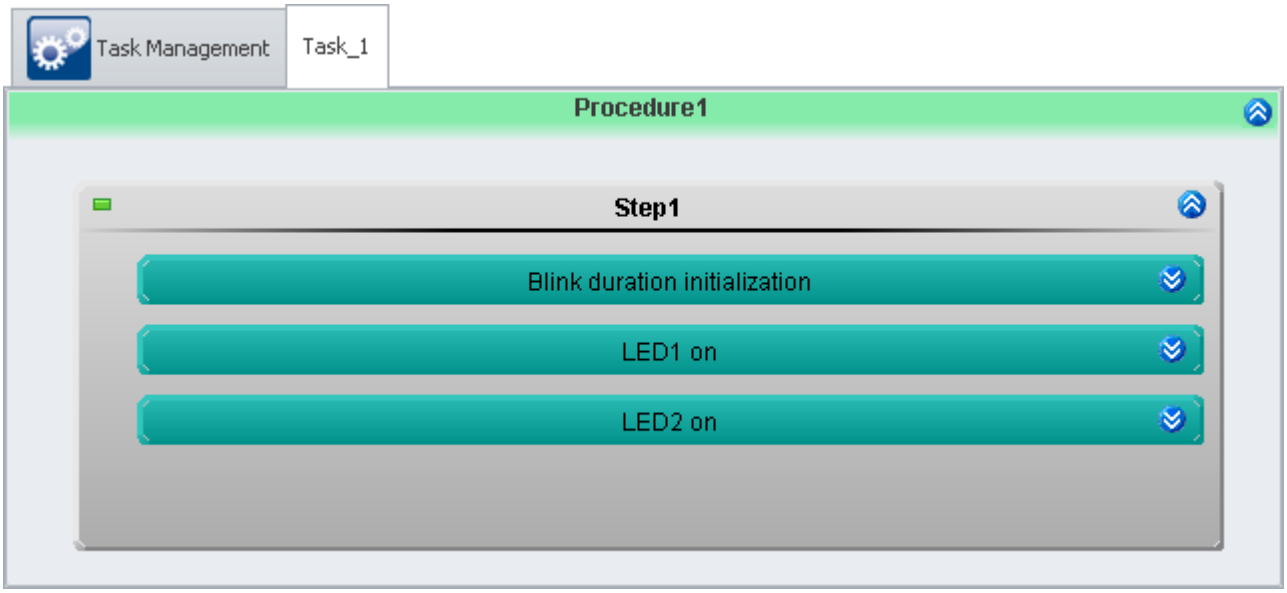
- Add a **Next-branch** into *Procedure1.*
- Open the **Next-branch**
- Set the **condition** to `1=1`
- The procedure is is own target (**Next**).



Now the **procedure** calls itself again when it finishes. But using the **Next-branch**, it is also possible to call any other **procedure**.

Open *Procedure1* by clicking on the double-arrow ( ). This takes you to the **Block Editor** of this **procedure**.

- Add a **step**
- Open the **step**
- Add three **steps**: *Blink duration initialization, LED1 on* and *LED2 on*.



Procedure1 with one step and three states

Each **state** is processed with a *1ms* cycle time.

The **states** *LED1 on* and *LED2 on* should end only after the *Blink duration* set. For this purpose, the following boxes are used:

- **Cyclical Calculation**: Incrementation of a counter
- **Break Condition**: `Counter >= Delay` (= Blink_duration/Cycle Time)

The instructions in the box **Periodic Calculations** are repeated at the period duration set in the **Task Management** under **Cycle Time**. Afterwards, the **break condition** continually re-assessed. If it is met, then the **state** is exited.

Now create the source code for the **states**:

Blink duration initialization

State Description

```
Delay = Blink_duration * 1000
```

Periodic Calculations {empty}

Exception/Error Events {empty}

Break Condition {Immediately}

Initializing the Blinkduration - calculating the delay

LED1 on

State Description

```
LED1 = 1
LED2 = 0
Counter2 = 1
```

Periodic Calculations

```
Counter1 = Counter1 + 1
```

Exception/Error Events {empty}

Break Condition

```
Counter1 >= Delay
```

Source code for LED1 on

LED2 on

State Description

```
LED1 = 0
LED2 = 1
Counter1 = 1
```

Periodic Calculations

```
Counter2 = Counter2 + 1
```

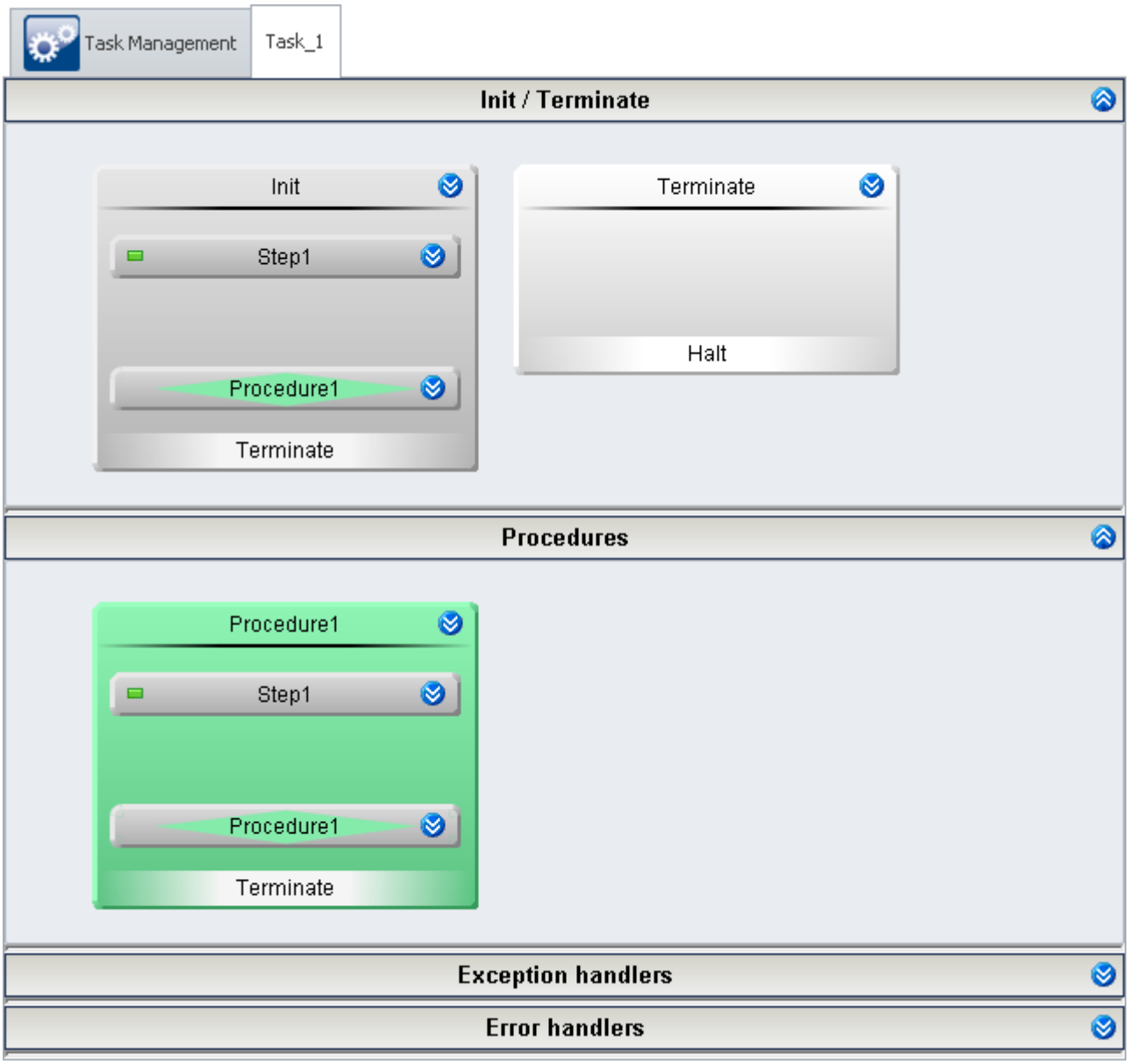
Exception/Error Events {empty}

Break Condition

```
Counter2 >= Delay
```

Source code for LED2 on

Every **state** is now described exactly and the **procedure** is thus completed. When you exit the **procedure**, you return to the **Task Editor** and see the outline of the procedure.



Check and Apply

Once the entire **Task** has been set up, the syntax of the program should be checked.

- Check and apply the changes, click on the button "*Process configuration* 241" (✔).

14.9.2.3 Panel - Settings / Start measurement

Now the Panel page can be set up.

- Open the page: "*Panel*".

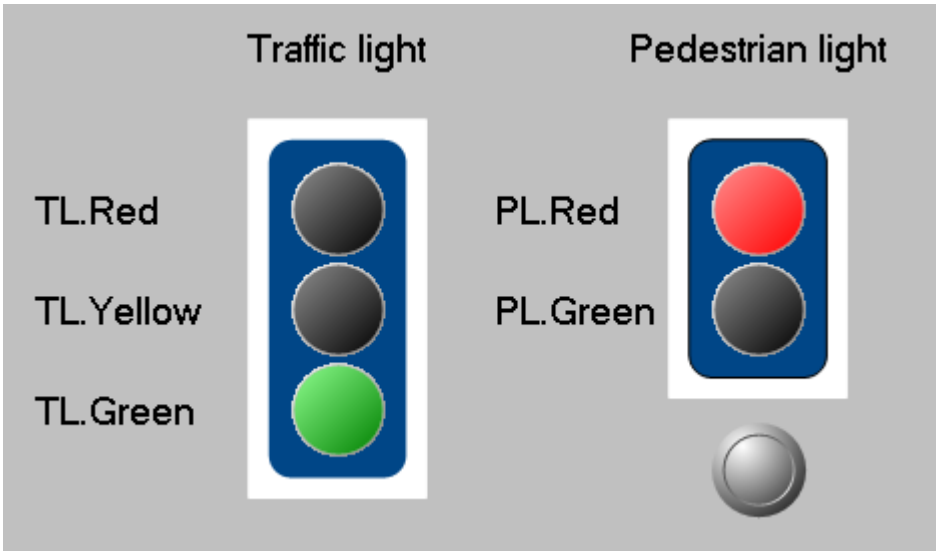
Proceed as described in the previous example Blinking LED using timer 1628.

14.9.3 Traffic light

Assignment:

Make a pedestrian intersection crossing light.

A stoplight for vehicles (TL = traffic light) has the colors red, yellow and green. But the light for pedestrians (PL = pedestrian light) is either red or green and has a button for requesting.



Panel - Traffic light (example)

Learning goals:

- Creation of a **task**
- Use of **timer functions**
- Use of **State description** and **Break conditions**

Elements used:

- Init / Terminate
- Procedures
- Steps
- States (State description and Break conditions)
- Next branching

Additionally used plug-ins:

- Setup
- Panel

Procedure:

Begin by starting **imc STUDIO**.

14.9.3.1 Setup - Settings

To begin, the measurement device is selected and configured.

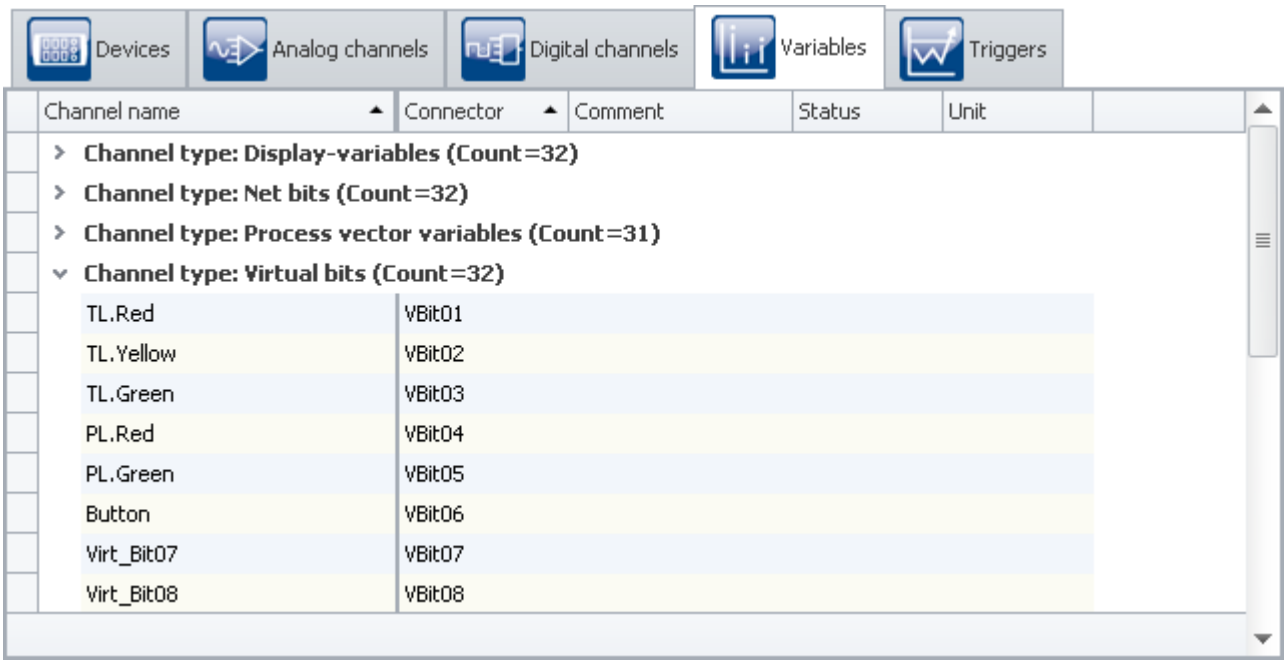
- Open the **device configuration page**: "*Setup*".
- Go to the **tab**: "*Devices*" ..

Select your device.

Configuring the variables

Now we assign appropriate names to the variables used.

- Go to the **tab**: "*Variables*" .
- Change the name of six **Virtual bits**:
  TL.Red, TL.Yellow, TL.Green,
  PL.Red, PL.Green,
  Button.



Assigning names

Note

In order to prevent the **automation** from ending prematurely, make sure that the first channel's (***Channel_001***) **measurement duration** is set to *undefined* (=infinite).

- To **apply the changes**, click on the button "*Process configuration* 241" ( ).

14.9.3.2 Automation - Settings

Now it is possible to set up the **Task**.

- Open the page: "*Automation*".

You are now in the **Task Management** interface. Proceed with creating a **Task**.

The following table presents the **states** and their dwell times, which the stoplight is to run through after the button is pressed.

	TL.Red	TL.Yellow	TL.Green	PL.Red	PL.Green	Dwell time in sec.
A	0	0	1	1	0	until button pressed
B	0	1	0	1	0	3
C	1	0	0	1	0	3
D	1	0	0	0	1	5
E	1	0	0	1	0	3
F	1	1	0	1	0	1
back to A	0	0	1	1	0	until button pressed

Init

In the **Init** block, set the initial values.

Open this block by clicking on the double-arrow ( ). This takes you to the **Block Editor**.

- There, create a **step**
- In this **step**, add a **state**.
- Add the initial values in the **state description**.

```
TL.RED = 0
TL.YELLOW = 0
TL.GREN = 1
PL.RED = 1
PL.GREEN = 0
Button = 0
```

Close the **Block Editor** by clicking on the double-arrow ( ) for the initialization.

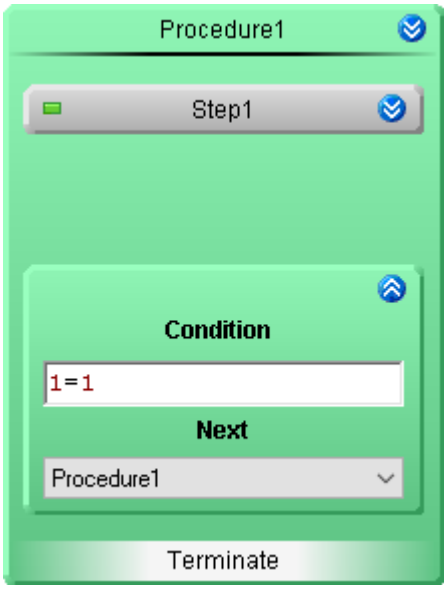
Open the existing **Next-branch** by clicking on the double-arrow ( ).

- Enter `1=1` for the **condition**.

Once **Initi** has been run, *Procedure1* is performed automatically.

Procedure

- Add a **Next-branch** into *Procedure1.*
- Open the **Next-branch**
- Set the **condition** to `1=1`
- The procedure is is own target (**Next**).



Procedure1

Open *Procedure1* by clicking on the double-arrow ( ). This takes you to the **Block Editor** of this **procedure**.

- Open **Step1**
- Create six **states**: ***A*** through ***F***.

Individual states of the stoplight control

- In the individual **states** you determine which LED are to blink and which ones are not.
- Also supply an appropriate B**reak condition** (press button or `OnTimeout`)

As an example: the **states** *A* and *B*.

```
A
State Description
TL.Red = 0
TL.Yellow = 0
TL.Green = 1
PL.Red = 1
PL.Green = 0

Periodic Calculations {empty}
Exception/Error Events {empty}
Break Condition
Button = 1
```

```
B
State Description
TL.Red = 0
TL.Yellow = 1
TL.Green = 0
PL.Red = 1
PL.Green = 0
Button = 0

Periodic Calculations {empty}
Exception/Error Events {empty}
Break Condition
OnTimeout(3)
```

Source code for the first two states

Note

By deleting the **virtual bit**: *Button* in **state** *B*, it is specified to behave like a **pushbutton**. However, in the Panel you can also assign the **switch** to have *Switching behavior*, then you don't need to reset *Button* here. This happens upon releasing the **pushbutton**.

Check and Apply

Once the entire **Task** has been set up, the syntax of the program should be checked.

- Check and apply the changes, click on the button "*Process configuration* 241" (✔).

14.9.3.3 Panel - Settings / Start measurement

Now the Panel page can be set up.

- Open the page: "*Panel*".
- Use the Drag&Drop technique to move the stoplight's virtual bits to the Panel page and display them as LEDs.
- Drag the virtual bit: ***Button*** from the Data Browser to the Panel page and display it as a switch or as a pushbutton (depending on the widgets that have been installed with your product).

You can also set the **Switching behavior** of the two models as desired. To do this, open the **properties** of the respective **Widgets** and change the property **Switching behavior**.

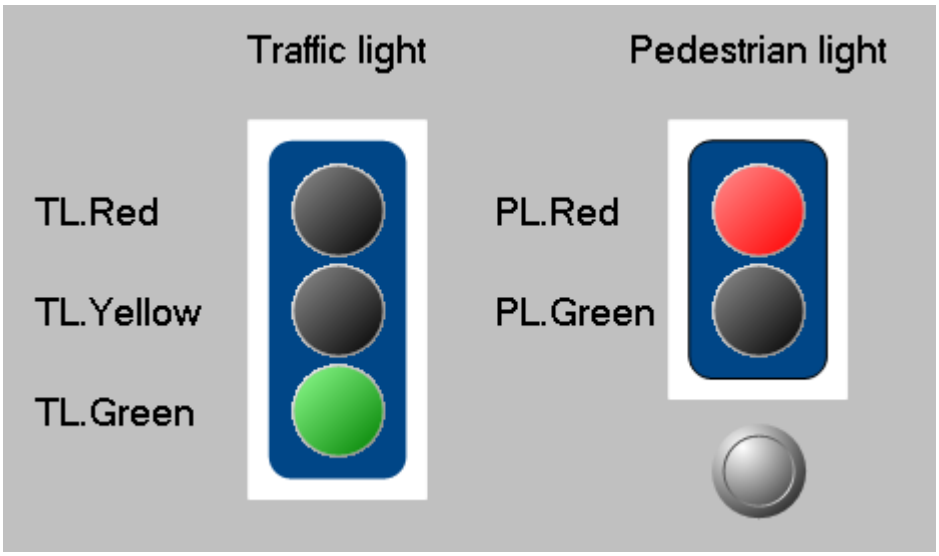
- If you have not reset *Button* with the **state** *B* in the **task**, then select the *Push-button behavior*
- If you reset it by program control, the switch does not need *Push-button behavior*

Adjust the zone colors of the LEDs:

- 0->0.5: Black
- 0.5->1: Red, Yellow or Green

You can also insert a background image and adapt the LED Color Control to it.

- Start the measurement 241 ( )



Panel - Traffic light (example)

14.9.4 Limit value monitoring

Assignment:

Set up **Limit value monitoring** for a voltage value.

The voltage value is to be set using a **potentiometer**. By means of two **input boxes**, an upper and a lower value limit can be set. Three **LEDs** indicates whether the current voltage value is too high, within the limits, or too low.

Learning goals:

- Creating a **task**
- Setting up **Limit value monitoring**
- Use of **timer functions**
- Use of **State description** and **Break conditions**
- Use of the **Repository**

Elements used:

- Init / Terminate
- Procedure
- Exception
- Step
- State (State description and Break conditions)
- Resume branching

Additionally used plug-ins:

- Setup
- Panel

Procedure:

Begin by starting **imc STUDIO**.

14.9.4.1 Setup - Settings

To begin, the measurement device is selected and configured.

- Open the **device configuration page**: "*Setup*".
- Go to the **tab**: "*Devices*" .

Select your device.

Configuring the variables

Now we assign appropriate names to the variables used.

- Go to the **tab**: "*Variables*" .
- Change the name of six **Display variables**: *Voltage*, *Upper_limit*, *Lower_limit*, *Voltage_too_high*, *Voltage_too_low*, and *Voltage_OK*

Note

In order to **prevent** the Automation from **ending prematurely**, make sure that the first channel's (***Channel_001***) **measurement duration** is set to *undefined* (=infinite).

- To **apply the changes**, click on the button "*Process configuration* 241" ( ).

14.9.4.2 Automation - Settings

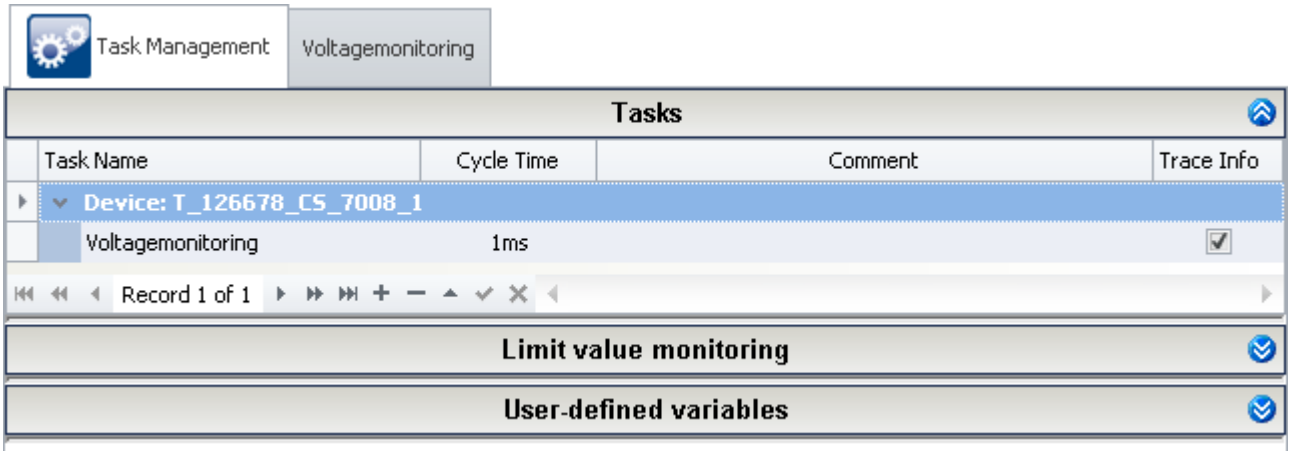
Now it is possible to set up the **Task**.

- Open the page: "*Automation*".

You are now in the **Task Management** interface.

Create a **task** with the name *Voltagemonitoring*

- Change the **Cycle time** to "1 ms", so that the voltage monitoring can respond more quickly.



Task for a voltage monitoring function

Open the **task**.

Procedure

- Change the name from *Procedure1* to *Monitoring*

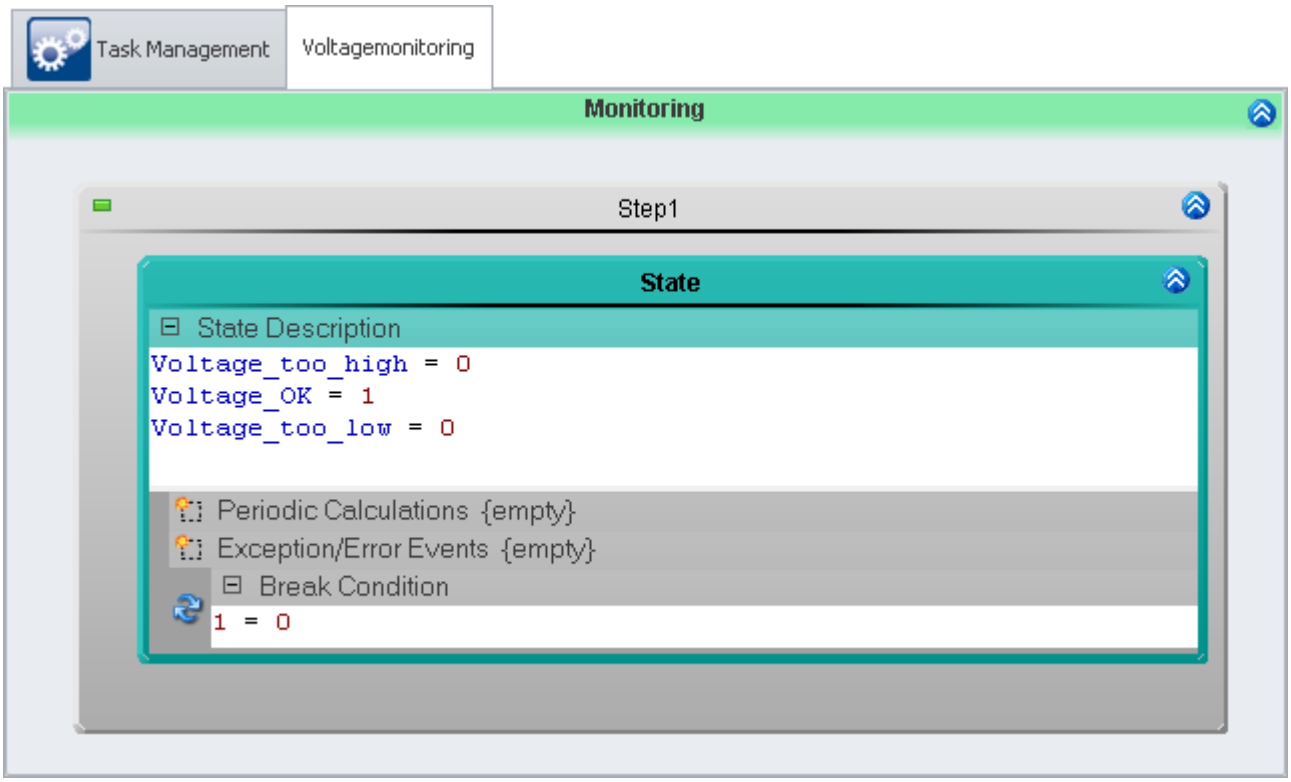
Open this **block** by clicking on the double-arrow ( ). This takes you to the **Block Editor**.

- There, create a **step**
- In this **step**, add a **state**.

In the **State description**, enter the following instructions:

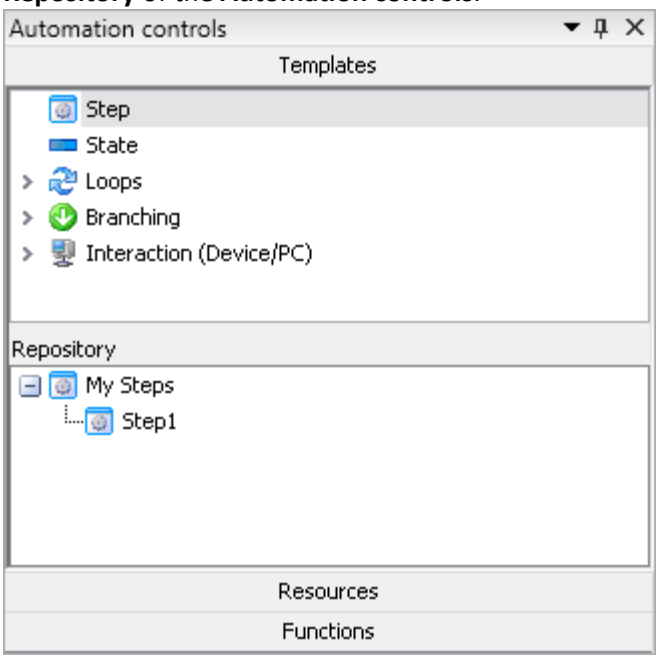
```
Voltage_too_high = 0
Voltage_OK = 1
Voltage_too_low = 0
```

This should always be the current **state**. For this reason, enter `1=0` for the **Break condition**. The **state** is only exited when a handler is started.



Source code: Voltage within the specified interval

Minimize the *Step1* by clicking on the step's double-arrow ( ). Copy it by means of Drag&Drop into the **Repository** of the **Automation controls**.



Repository

Close the **block**.

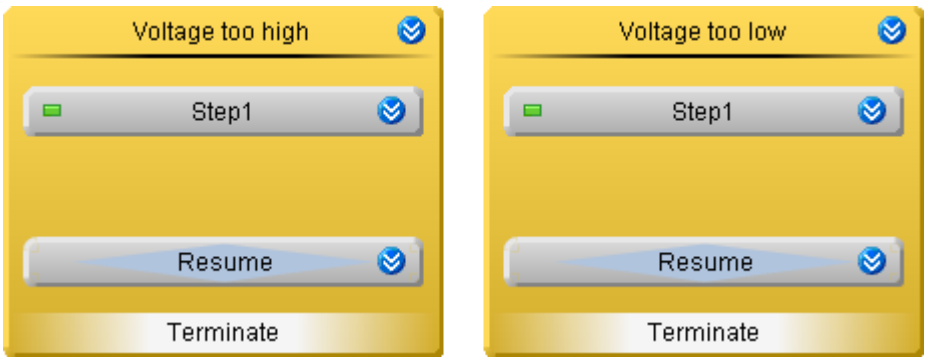
Exception handling

Create two **exception blocks** in the **Exception handling**.

- *Voltage too high*
- *Voltage too low*

These can be used in the **Limit value monitoring**.

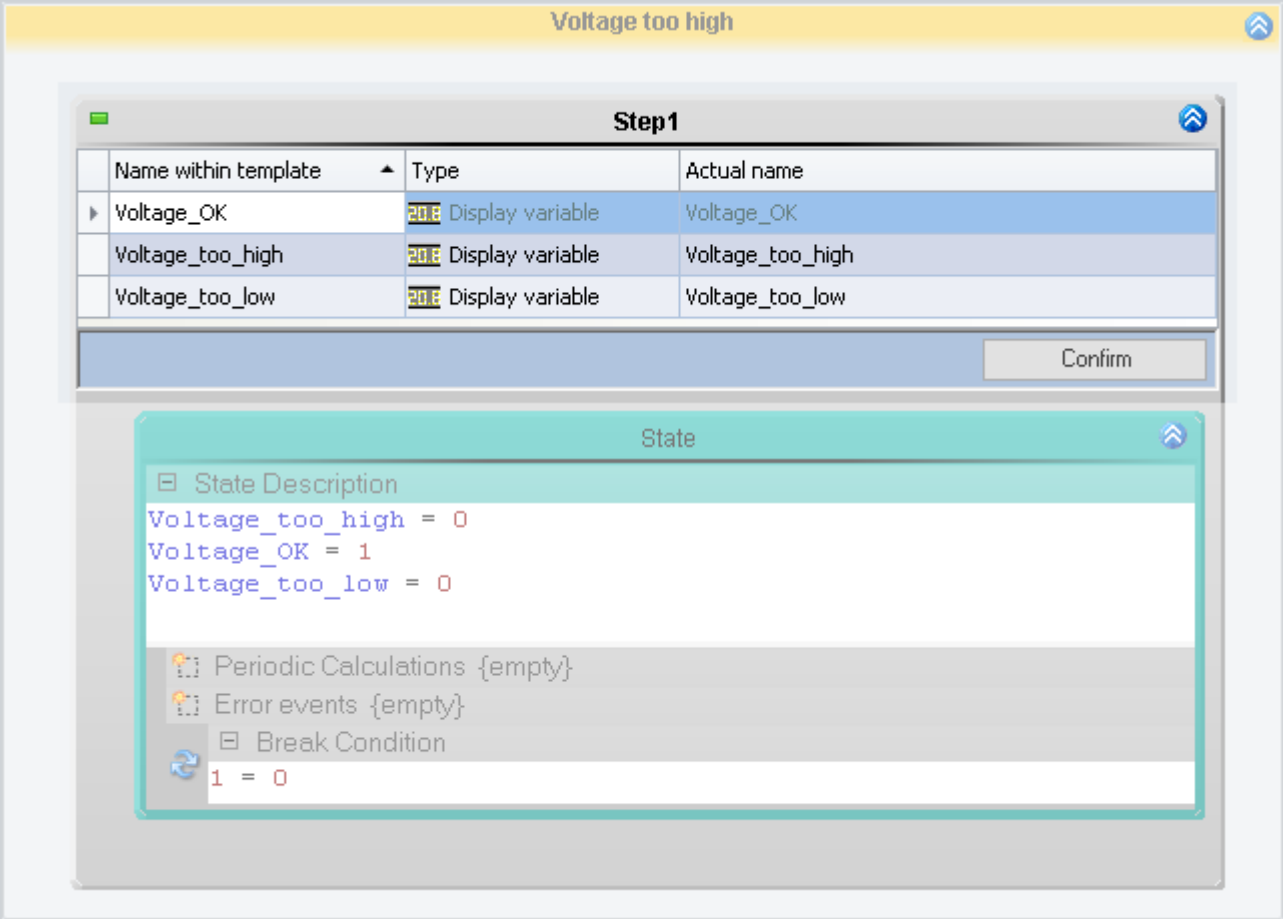
- Use Drag&Drop to copy the previously added **step** *Step1* to both **blocks**
- Add to each **block** one **Resume-branching** having the **condition**: `1=1`



Exception handling

Voltage too high

Open the **block** *Voltage too high* and in it open *Step1*. A **confirmation prompt** appears.



Confirmation prompt

- The variables used are the same as before. Click on *Confirm*.

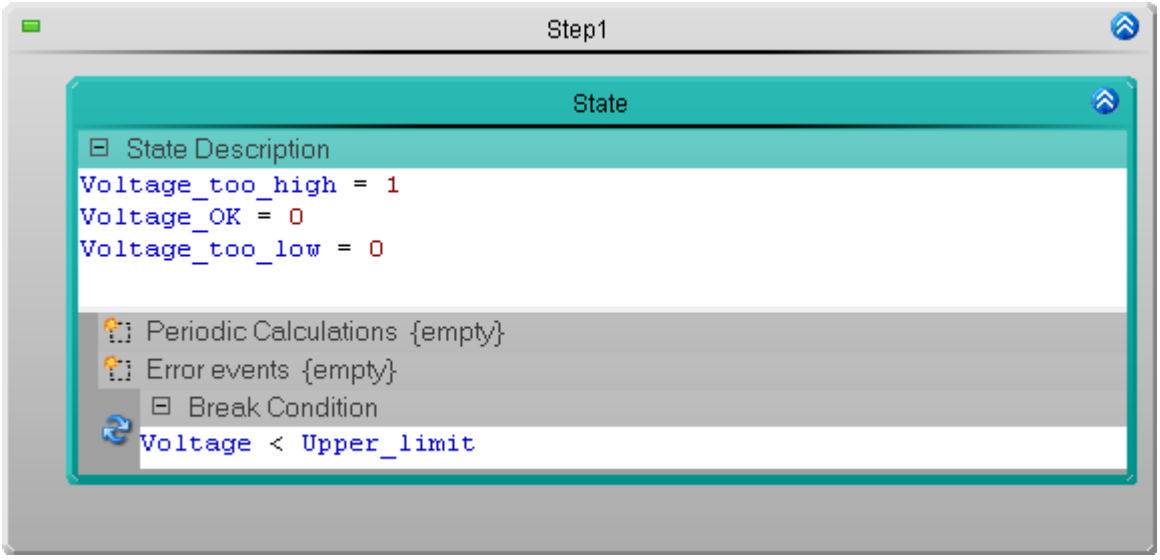
Change the values in the **State description**:

```
Voltage_too_high = 1
Voltage_OK = 0
Voltage_too_low = 0
```

Replace the existing **Break Conditions**:

```
Voltage < Upper_limit
```

The **state** is exited once the *Voltage value* is back below the *upper limit*.



Source code: Voltage over the specified interval

Close the **block**.

Voltage too low

Open the **block** *Voltage too low* and in it *Step1*. Another **confirmation prompt** appears.

- The variables used are the same here, too. Click on *Confirm*.

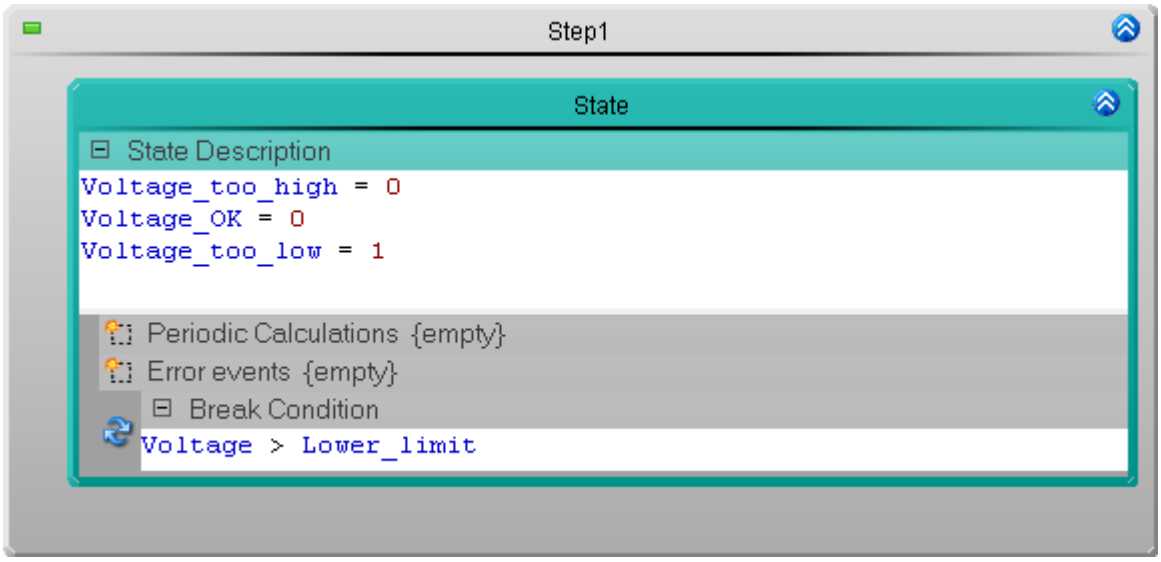
Change the values in the **State description** as follows:

```
Voltage_too_high = 0
Voltage_OK = 0
Voltage_too_low = 1
```

Replace the existing **Break Condition** as follows:

```
Voltage > Lower_limit
```

The **state** is exited once the *Voltage value* is back above the *lower limit*.



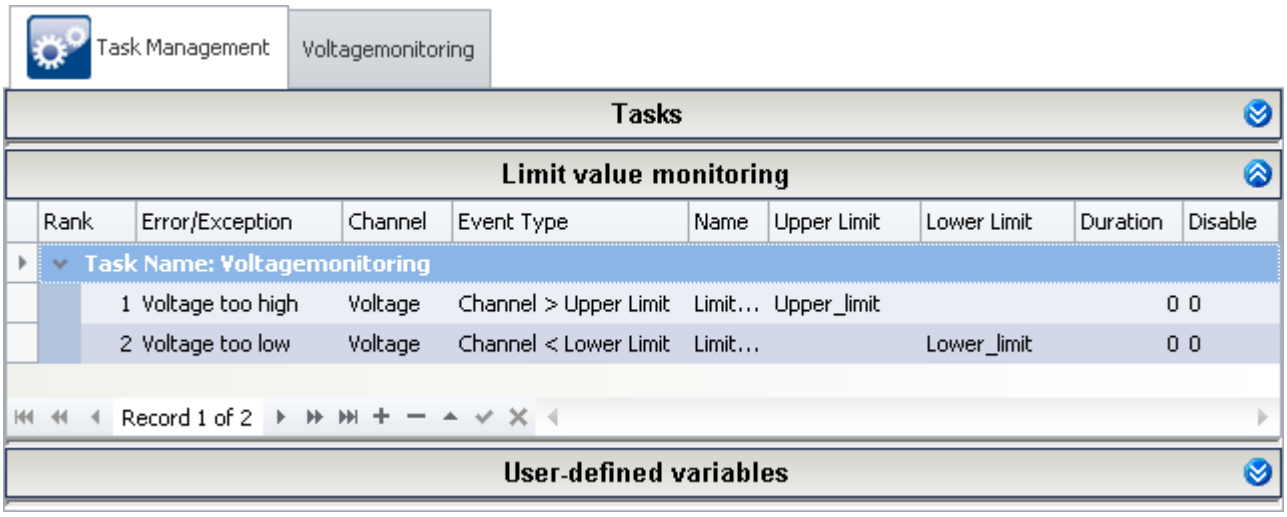
Source code: Voltage below the specified interval

Limit value monitoring

Go to the **Task Management**.

Create two **Limit value monitoring** functions and parameterize them:

	Error/Exception	Channel	Event Type	Upper Limit	Lower Limit
1	Voltage too high	Voltage	Channel > Upper Limit	Upper_limit	
2	Voltage too low	Voltage	Channel < Lower Limit		Lower_limit



Setting the value limit monitoring

If one of the two monitored events occurs, the corresponding **handling** is performed automatically.

Check and Apply

Once the entire **Task** has been set up, the syntax of the program should be checked.

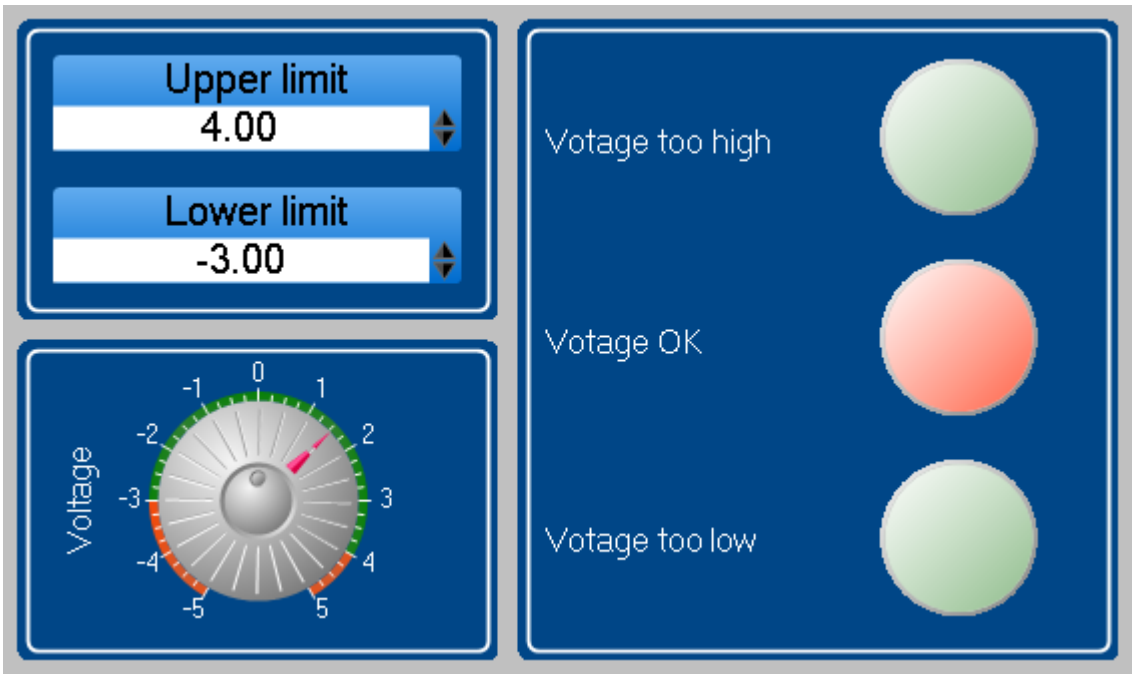
- Check and apply the changes, click on the button "*Process configuration* 241" (✔).

14.9.4.3 Panel - Settings / Start measurement

Now the Panel page can be set up.

- Open the page: "*Panel*".

Drag the **Display variables** used to the Panel page and display them with the appropriate elements.



Panel - Limit value monitoring (example)

- Set the limits.
- Start the measurement 241 ( )

Test the monitoring function by adjusting the *voltage* to higher than one of the *limits* by means of the **potentiometer**.

14.9.5 Actuator-control - Ramps

Assignment:

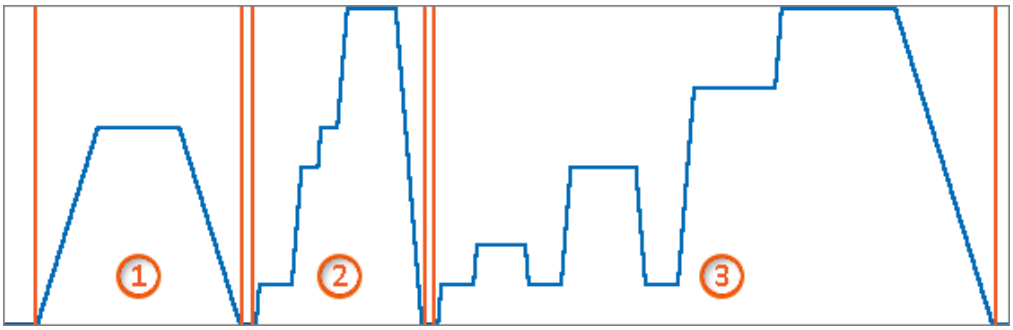
An actuator is assigned an RPM-value. In order to make this possible without any additional auxiliary equipment, the DAC-output is simulated by the device.

The output is represented by a display variable. This display variable generates a virtual channel.

Note: You could also connect an actuator to the DAC and measure and display its revolutions.

The output signal consists of three cycles which continually repeat.

1. Run-in cycle
2. Variable cycle
3. Fixed cycle



The cycles run in succession.

Enhancements:

- Individual steps can be switched off
- Cycles can be switched off
- Iteration counter

Learning goals:

The individual cycles differ in their respective implementation and operation.

The first cycle is for introducing the principle of ramps. The second is a simple progression from the first variant, in which the ability to modify individual steps is provided.

The third variant is a simplification. It can include any arbitrary number of steps, but these must be stated before beginning measurement.

Additional prerequisite:

In order to create a dat-file, imc FAMOS must be available during the segment: "*Fixed cycle*". The other segments can be performed without imc FAMOS.

14.9.5.1 Preparing the task

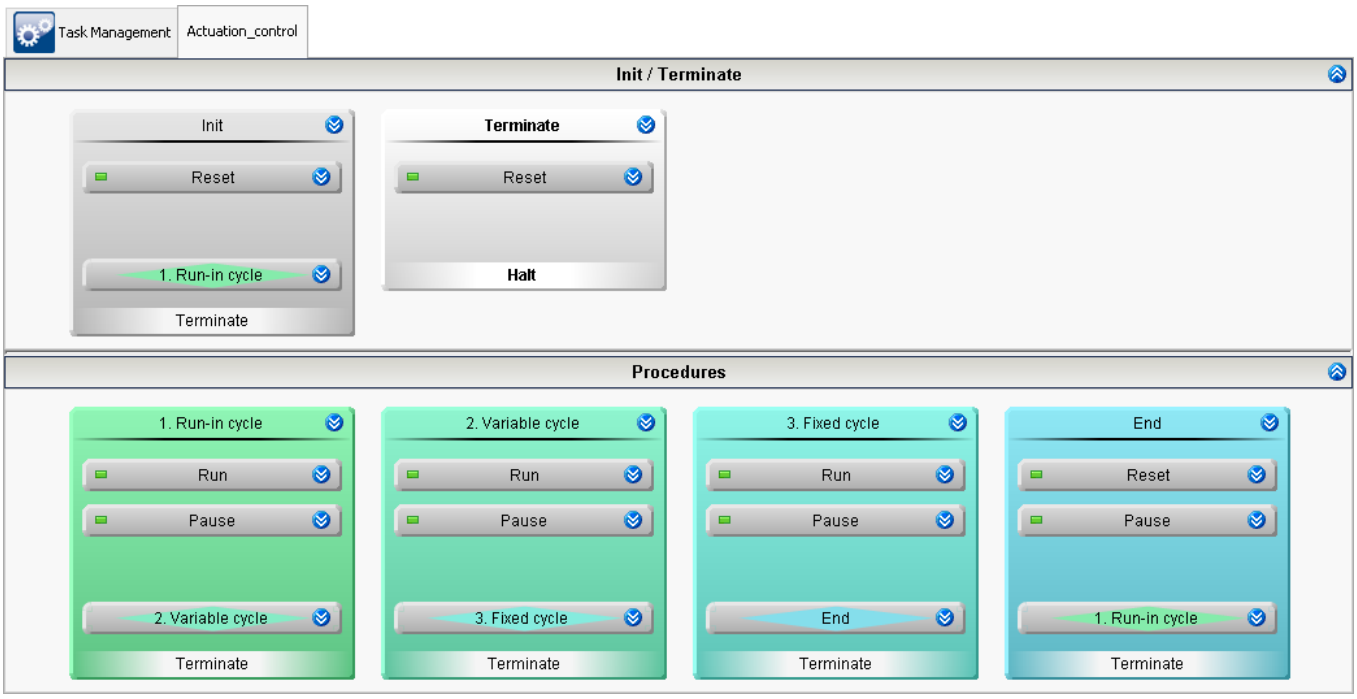
Begin by describing the routine. Only then is it time to fill the individual elements with actions. In the example, only one single procedure is used for each cycle. This should make things more clear.

- Save the experiment: e.g. under "*Actuation control 1*"
- Select a device
- Create a Automation-task:
  - Name: Actuation_control
  - Cycle time: 10 ms

Create three additional procedures in addition to the first one. The four procedures are assigned the names:

- 1. Run-in cycle
- 2. Variable cycle
- 3. Fixed cycle
- End

Create the individual steps and branches to the next procedure as illustrated by the example:



Overview of all steps

All branch operations are always performed. Consequently, no conditions are needed.

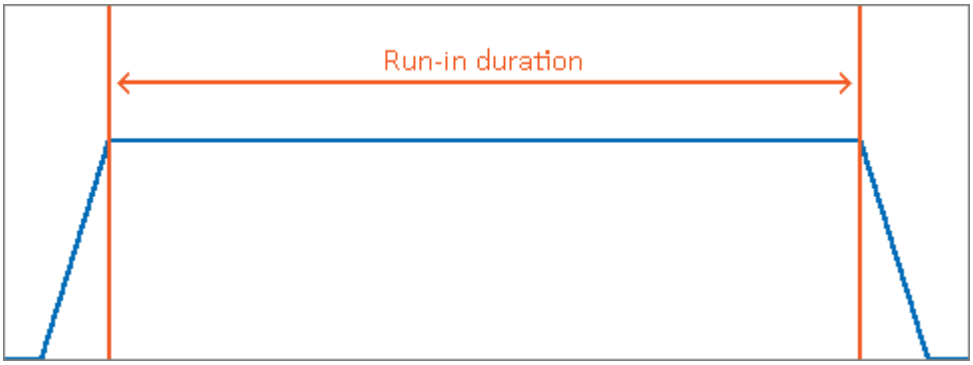
In the Reset-steps, the system is placed in an idle state and the variable values are reset. This is good to do at every possible stage of the routine, in case some unforeseen situation raises an error or exception.

The exception/error-handling is not a part of this example, but should always be considered as a possibility.

14.9.5.2 Run-in cycle

The steps are now filled in one-by-one. Start with the procedure "*1. Run-in cycle*".

This cycle is practically a warmup phase. The actuator slowly ramps up, stays running at high value and then returns to its rest state.



Plot of Run-in cycle

During the time in which the rotation speed is held steady, it is possible to measure and evaluate characteristic parameters. In the example, this state is called "*Capture parameters*". For this purpose, a duration time is defined. By contrast, no time frames for the acceleration and braking are defined, but a slope is instead. Thus the duration depends on the initial and the target rotation speeds.

Notes Ramp-functions

There are two ramp-functions in the Automation: "*RampSlope*" and "*RampTime*". In this example, "*RampSlope*" is used. With "*RampTime*", the time is specified and the slope is adjusted accordingly.

Variables used:

Variable	Description
Output	This variable is the actuator's output variable. It could be outputted by the DAC by means of imc Online FAMOS. For purposes of the simulation, a virtual channel is generated from it, in order to be able to view the results in a curve window.
pv.Run_in_Cycle_Ramp	specified slope (in rpm/s)
pv.Run_in_Cycle_Spd	final speed (in rpm)
pv.Run_in_Cycle_Dur	duration until braking

Output:

- Change the name of one of the display variables to "*Output*".

Virtual channel:

- Open imc Online FAMOS.
- Activate the control commands
- Use the following source code:

```
OnTriggerMeasure(BaseTrigger)
    DAC_Output = CreateVChannel ( Channel_001, Output )
End
```

You thus obtain a simulated channel which would represent the DAC's output signal.

In the Automation, create three variables for the Task:

Variable	Properties
pv.Run_in_Cycle_Ramp	Integer - Default value: 120; persistent
pv.Run_in_Cycle_Spd	Integer - Default value: 5; persistent
pv.Run_in_Cycle_Dur	Integer - Default value: 7; persistent

Notes Clarification

pv.: generates a variable which can be viewed and assigned on the Panel. Thus the values can be modified at any time.

persistent: if the value is changed, it is still available at a later point in time, even if the device and the PC had been deactivated.

Modifying the procedure

Create three states in the step: "*Run*":

- State: Ramp up
- State: Capture parameters
- State: Ramp down

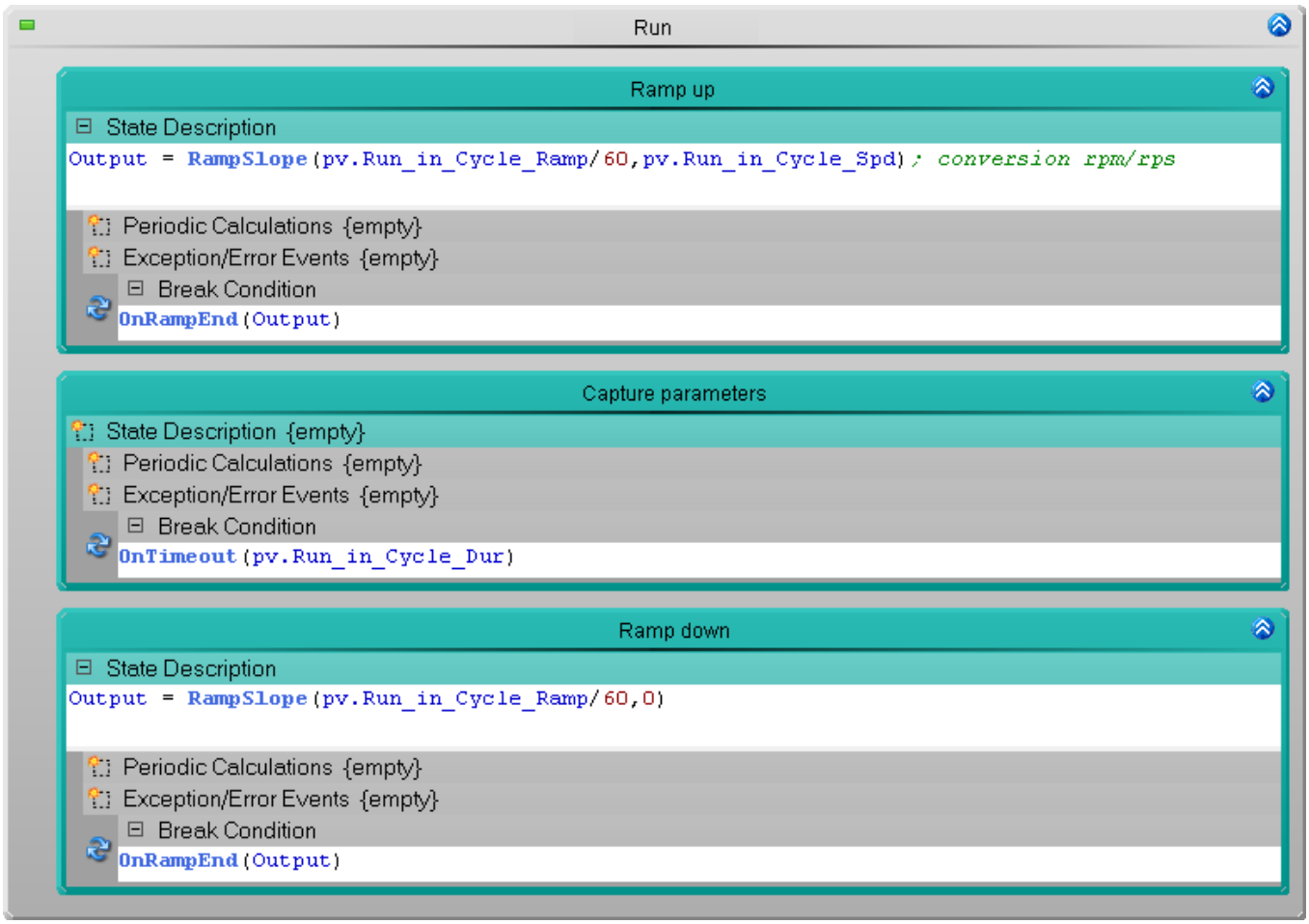
And one state in the step "*Pause*"

- State: Pause

States in the steps

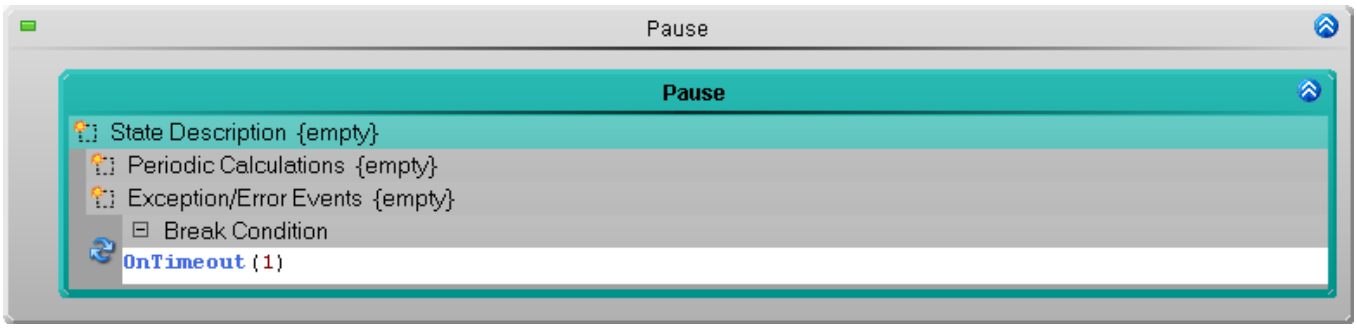
The routine for the first cycle is thus defined. Now fill the steps with the actions:

- State: Ramp up
  - State description: the output signal is continuously increased up to the specified final value
    `Output = RampSlope(pv.Run_in_Cycle_Ramp/60,pv.Run_in_Cycle_Spd) ; conversion rpm/rps`
  - Break condition: exit the state when the ramp has reached the end
    `OnRampEnd(Output)`
- State: Capture parameters
  - Break condition: exit the state when the specified time has elapsed
    `OnTimeout(pv.Run_in_Cycle_Dur)`
- State: Ramp down
  - State description: the output signal is continuously reduced down to "*0*"
    `Output = RampSlope(pv.Run_in_Cycle_Ramp/60,0)`
  - Break condition: exit the state when the ramp has reached "*0*"
    `OnRampEnd(Output)`



1st state: Ramp up
2nd state: Capture parameters (Wait)
3rd state: Ramp down

- State: Pause
  - Break condition: Wait for a specified time (1 s) until starting the next cycle
    `OnTimeout(1)`



Waiting for exactly one seond

Reset-functions:

As a precaution, the DAC output should be put in its rest state at the beginning and at the end. This will ensure that the cycle always starts from the same point of departure.

Example: If you stop the measurement while a signal is applied at the DAC, then the DAC should always be reset back to "*0*". Not abruptly, but at the rate of acceleration of the Run-in cycle, for example.

To do this, copy the state: "*Ramp down*" to the "*Repository*".

Add this state from the repository to all three "*Reset*"-steps, in the following blocks:

- Initialization
- Terminate
- End

This ensures that there is a rest state at the end of cycle, and at the start and end of any Task.

Measurement and display on a Panel page

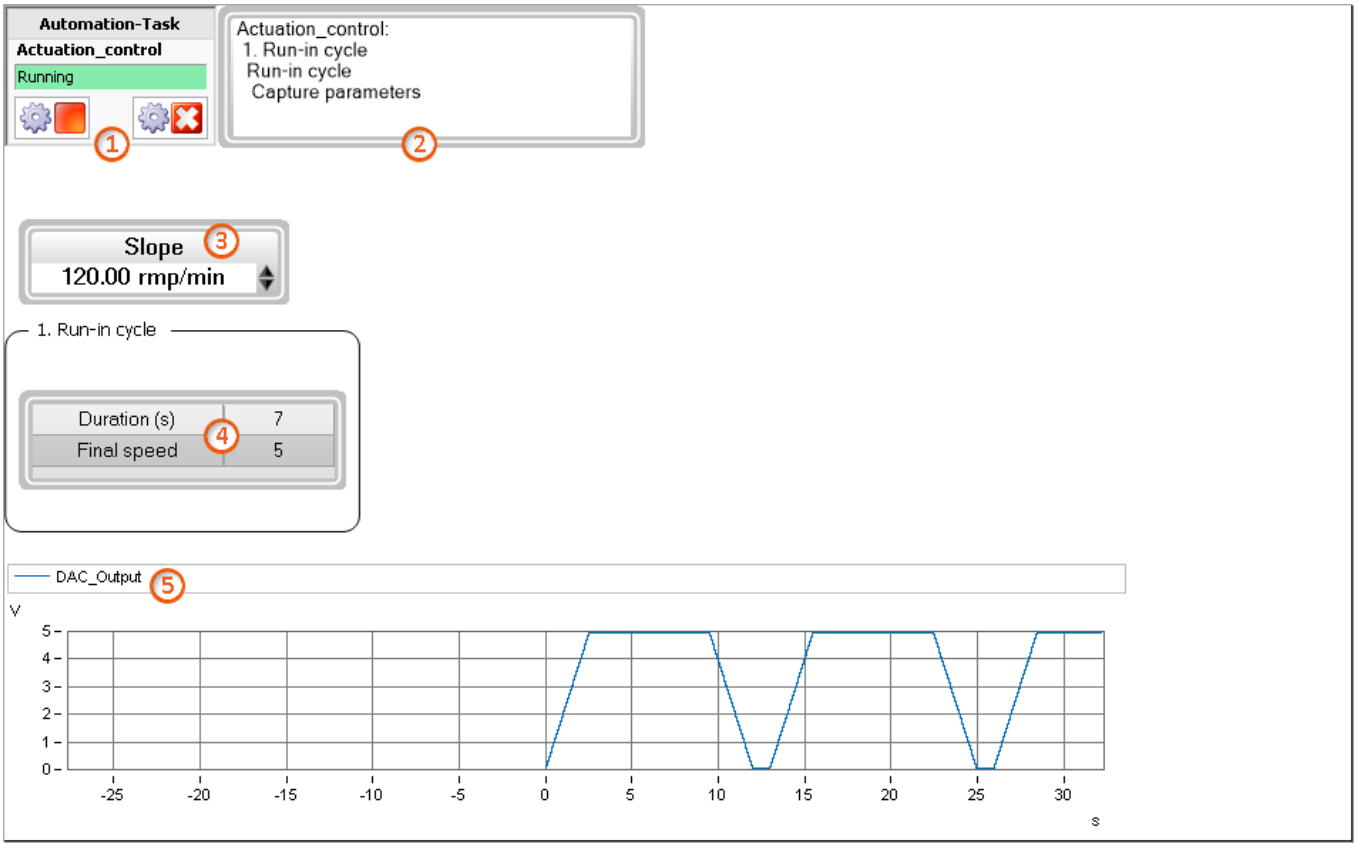
First deactivate the data saving of the channels: Channel_001 and DAC_Output. Saving of measured data is no longer required.

Perform "*Process Configuration*". This provides a check of whether the configuration is correct, and the new variables are made available to the Panel.

Creating and configuring the Widgets:

The specified Widgets are possible examples, but you could also use other ones. The Widgets from the example shown in the picture are configured in a way which presents the values clearly. The Widget-configuration is not described here, since its characteristics are not relevant to the procedure. Design the display appearance according to your own choice.

- 1: Widget: Automation-Task - linked with "*Actuation_control*"
- 2: Widget: Input, output > Text - linked with "*Actuation_control.Trace*"
- 3: Widget: Input, output > Numerical - linked with "*pv.Run_in_Cycle_Ramp*"
- 4: Widget: Table - linked with "*pv.Run_in_Cycle_Spd*" and "*pv.Run_in_Cycle_Dur*"

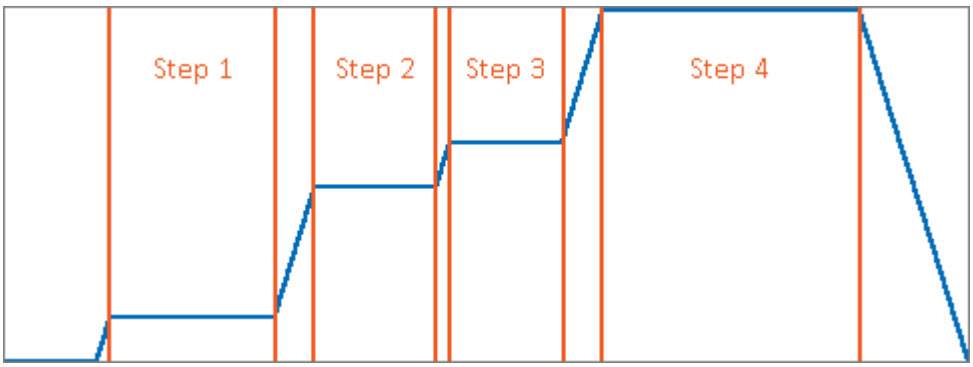


Display on the Panel page

Change the values of these variables in order to be able to observe their direct effect. Using the two upper Widgets, you can check on how the Automation-task is being run. If you stop measurement outside of the rest state, then it will be indicated there that the block: "*Terminate*" is running, in which the DAC is put in its rest state.

14.9.5.3 Variable cycle

In the same way, implement the second cycle next.



Plot of cycle: Variable

For each individual step, associated variables are needed. Create these nine variables for the Task in the Automation:

Variable	Properties
pv.Cycle_Ramp	Integer - Default value: 240; persistent
pv.Var_Cycle_Spd1	Integer - Default value: 1; persistent
pv.Var_Cycle_Dur1	Integer - Default value: 2; persistent
pv.Var_Cycle_Spd2	Integer - Default value: 4; persistent
pv.Var_Cycle_Dur2	Integer - Default value: 1; persistent
pv.Var_Cycle_Spd3	Integer - Default value: 5; persistent
pv.Var_Cycle_Dur3	Integer - Default value: 1; persistent
pv.Var_Cycle_Spd4	Integer - Default value: 8; persistent
pv.Var_Cycle_Dur4	Integer - Default value: 3; persistent

Furthermore, it should be possible to switch the individual steps on or off. For this purpose, create an additional variable for each individual step:

Variable	Properties
pv.Var_Cycle_RUN_1	Integer - Default value: 1; persistent
pv.Var_Cycle_RUN_2	Integer - Default value: 1; persistent
pv.Var_Cycle_RUN_3	Integer - Default value: 1; persistent
pv.Var_Cycle_RUN_4	Integer - Default value: 1; persistent

If the variable's value is "*1*", the individual step is performed; for "*0*" it is not.

Modifying the procedure

Next, configure the procedure: "*2. Variable cycle*". For this purpose, you can make use of the states already created. Copy the states "*Ramp up*" and "*Capture parameters*" from "*Run-in cycle*" to the repository (CTRL + Drag&Drop).

Copy the two states from the repository to the step "*Run*".

Modify the variables' names in the states:

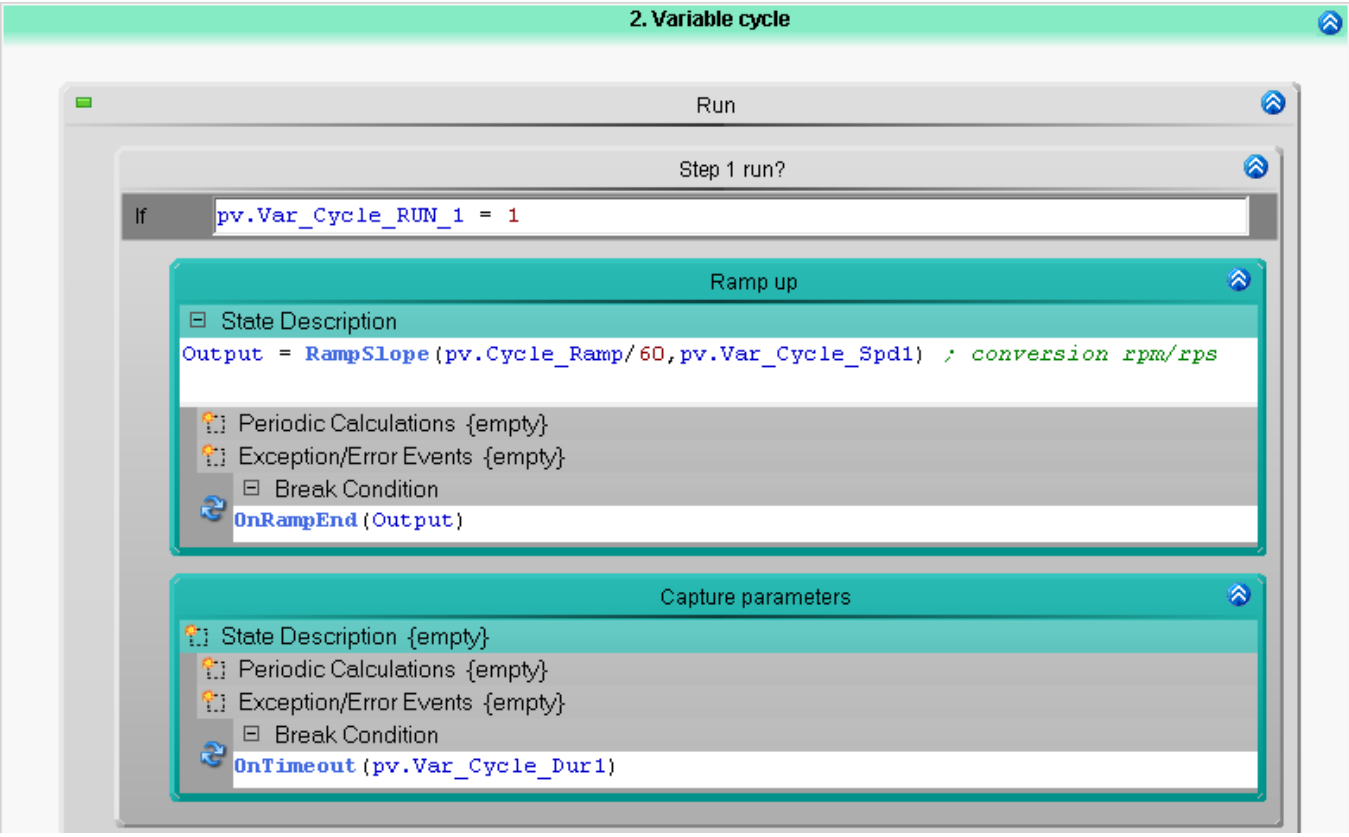
- State: Ramp up
  - State description:
    `Output = RampSlope(pv.Cycle_Ramp/60,pv.Var_Cycle_Spd1); conversion rpm/rps`
  - Break condition:
    `OnRampEnd(Output)`
- State: Capture parameters
  - Break condition:
    `OnTimeout(pv.Var_Cycle_Dur1)`

Next, this individual step should only be performed if the value of the variable "*pv.Var_Cycle_RUN_1*" is "*1*". Create an IF-condition. Assign an appropriate name to the condition: "*Step 1 run?*".

As the condition, enter the following:

```
pv.Var_Cycle_RUN_1 = 1
```

Drag the two steps to the If-condition.



Creating the first individual step:
1st State: Ramp up
2nd State: Capture parameters (Wait)

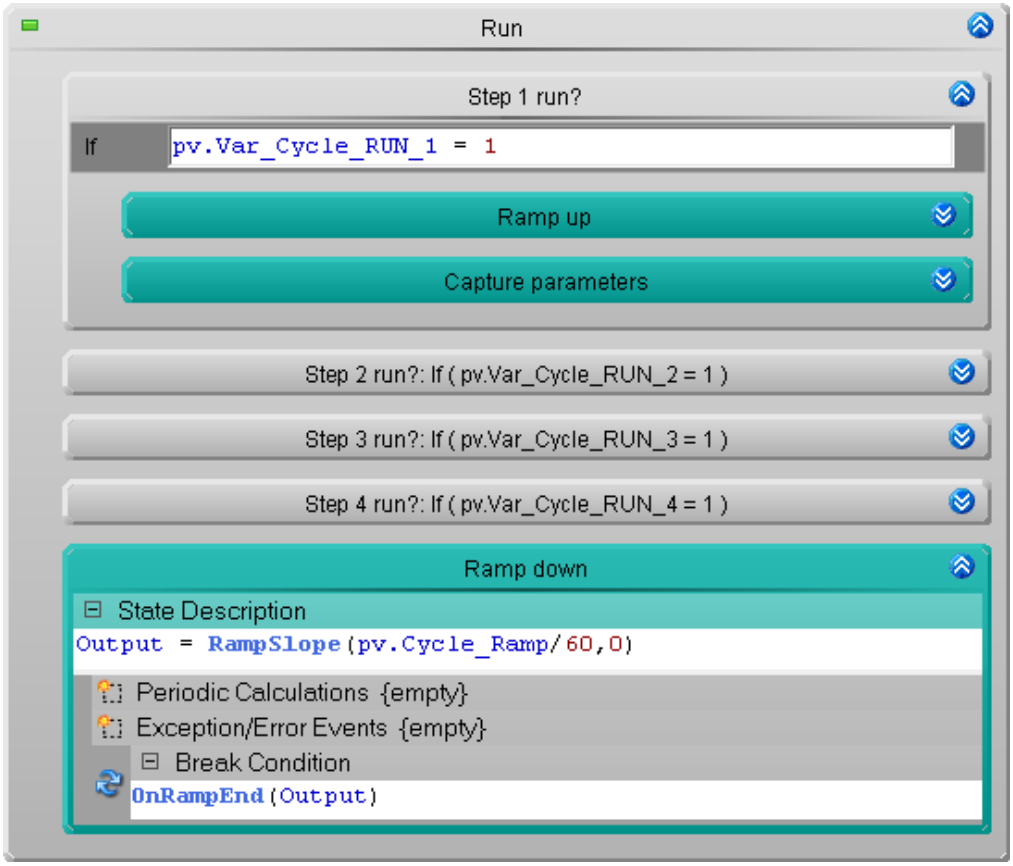
Thus the first individual step is finished. Now copy the IF-condition three times; e.g. by means of CTRL + Drag&Drop.

In the three newly created individual steps, please modify the respective variables and names of elements. For this purpose, only the respective **4 numbers** need to be modified; e.g. "*pv.Var_Cycle_RUN_1*" -> "*pv.Var_Cycle_RUN_2*".

After the last If-condition, add the state "*Ramp down*" to the step again from the repository. And modify the variable.

- State: Ramp down
  - State description:
    `Output = RampSlope(pv.Cycle_Ramp/60,0)`

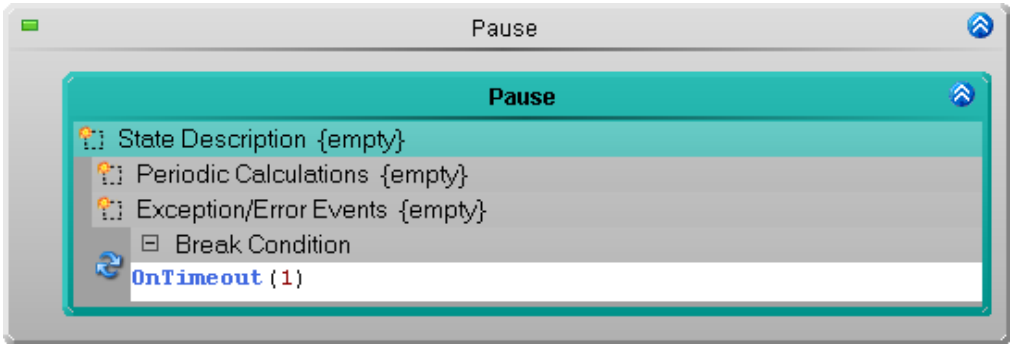
Thus, at the end of the cycle the rest state is again approached.



All four individual steps. At the conclusion, "Ramp down" again

Add the Pause-step accordingly to the Run-in cycle.

- State: Pause
  - Break condition:
    `OnTimeout(1)`



Waiting exactly one second

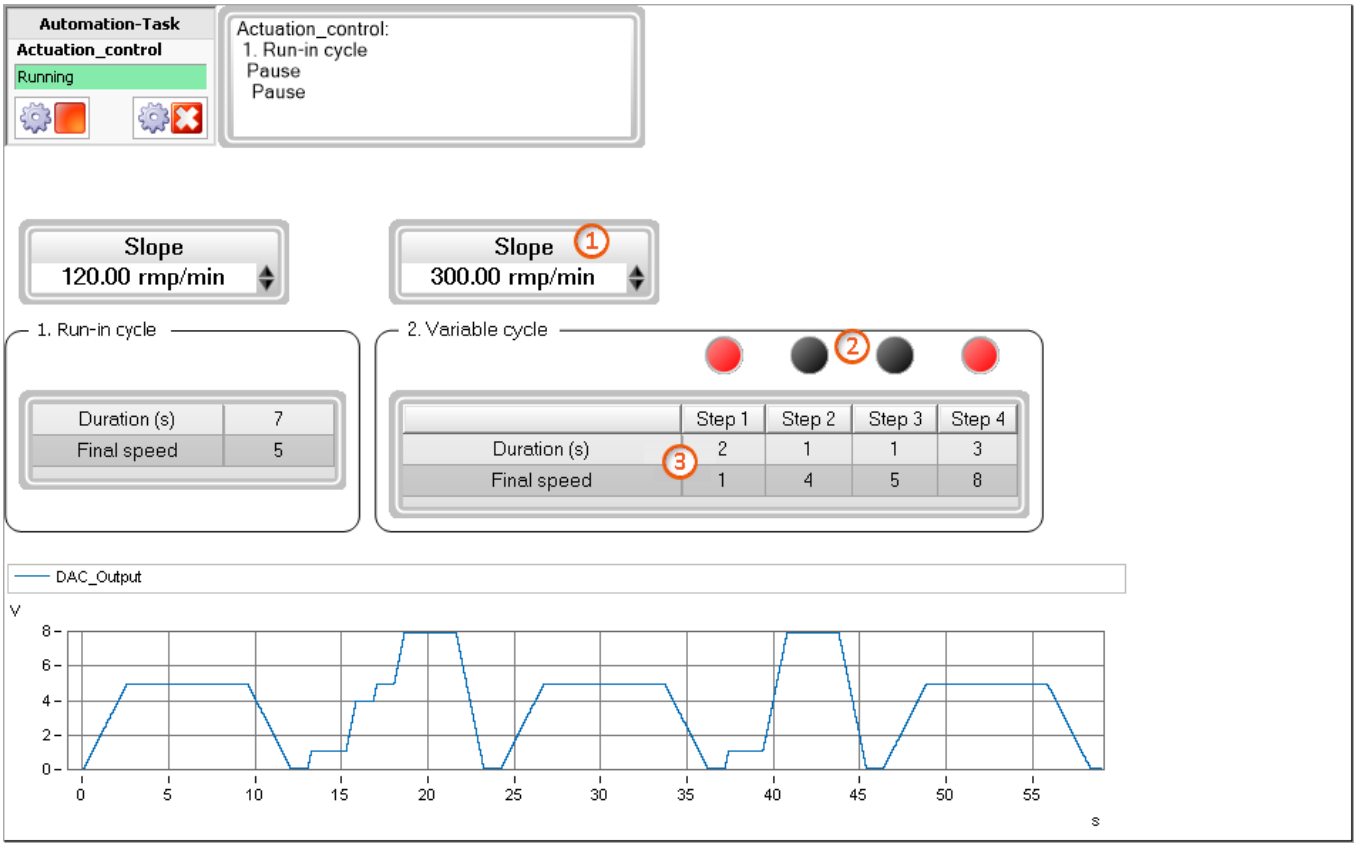
Measurement and display on a Panel page

Next, check the routine again and for this purpose modify the Panel page. But before that perform "*Process configuration*".

- 1: Widget: Input/output > Numerical - linked with "*pv.Cycle_Ramp*"
- 2: Widget: Switch/Pushbutton > 4 x LED-pushbutton (switching behavior: "*Switch*") - linked with "*pv.Var_Cycle_RUN_1*" through 4
- 3: Widget: Table - linked with "*pv.Var_Cycle_Spd1*" through 4 and "*pv.Var_Cycle_Dur1*" through 4

You can also use other buttons for the example. Configure the buttons with a color coding to indicate that for "*0*", the individual step is deactivated and for a "*1*", the individual step is active. In the image below, you can see, for example, that Steps 2 and 3 are deactivated.

In the curve window, you also see the result. In the first run, Cycle 2 has four steps. In the second run, only two steps. The individual steps can be modified during the measurement.



Display on the Panel page

14.9.5.4 Fixed cycle

Here, the second cycle is repeated with a different setup.

Disadvantage of the previous solution:

- For each individual step, 3 variables are needed
- Consequently for 20 individual steps, 60 variables would be needed

Solution:

- Use an array and a loop through the array's elements

In the Automation create three more variables for the Task:

Variable	Properties
Fix_Cycle_Spd	Vector from data pool
Fix_Cycle_Dur	Vector from data pool
Fix_Cycle_RUN	Vector from data pool

Notes Clarification

Vector from data pool: a channel in the data pool (visible in the Data Browser) is used

Advantages:

- very easy to manage
- no modifications necessary when changing the number of steps

Disadvantages:

- The last values are not saved without any additional work. They need to be loaded initially.
- The values must be present before the measurement starts; they can not be modified during the measurement.

The three variables are now created in the Data Browser. There are various ways to do this. In some cases, the routines are defined using an EXCEL-list. In this case, the easiest way would be to read the values from EXCEL by means of imc FAMOS and then to transfer them to imc STUDIO.

In the example here, we assume that the routine is saved in a dat-file.

To illustrate, use imc FAMOS to create three variables by means of the following source text:

```
Fix_Cycle_RUN = [1,1,1,1,1,1,1]
Fix_Cycle_Spd = [1,2,1,4,1,6,8]
Fix_Cycle_Dur = [2,3,2,3,2,4,5]
```

Save these three variable together in a dat-file. A suitable location to save them would be the Experiment folder. In that folder, create a subfolder named: "*Meta*".

"*..\Meta\Import_Data\Fix_Cycle.dat*"

On the Panel, set up a button.

- 1: Widget: Switch/Buttons > Push button - This runs the command "*Load variable*".

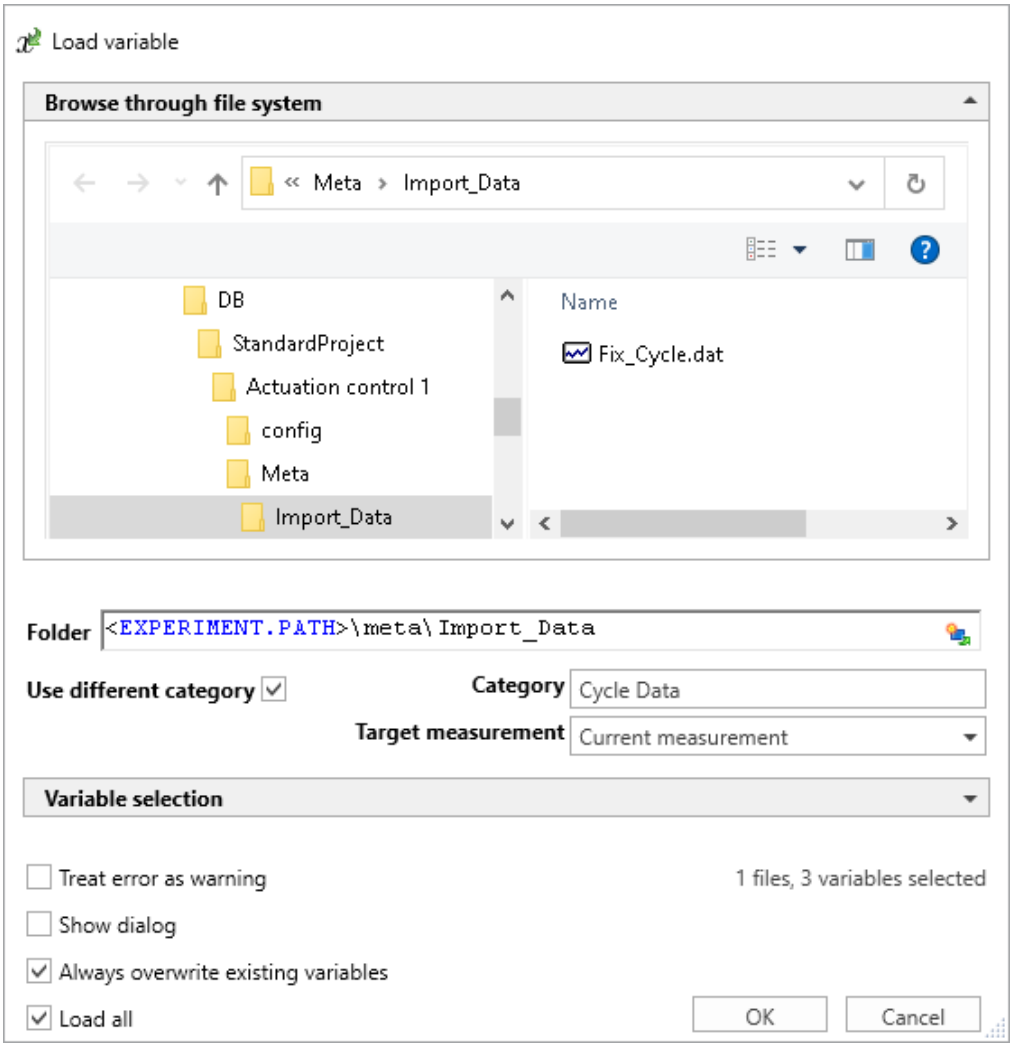
The command is configured to load the file which had just been created.

Under "*Folder*", enter the following:

<EXPERIMENT.PATH>\meta\Import_Data

The folder named "*Meta*" ensures that it will be ignored by imc STUDIO; e.g. for the display in the Data Browser, as well as in the case of interval saving with a specified count.

Thus the command doesn't depend on the experiment's name. You can save, export and import it under a new name. The respective associated experiment path and name will always be used.



Command: Load variable

When you check the box: "*Load all*", it doesn't matter what is in the file. If anything is added to it in future, that is automatically included when loading occurs.

Additionally, select overwriting without a prompt.

In order to achieve a good overview in the Data Browser, it is helpful to create a separate category. E.g. "*Cycle Data*"

Click on the button. The three variables appear in the Data Browser.

Back to the Automation

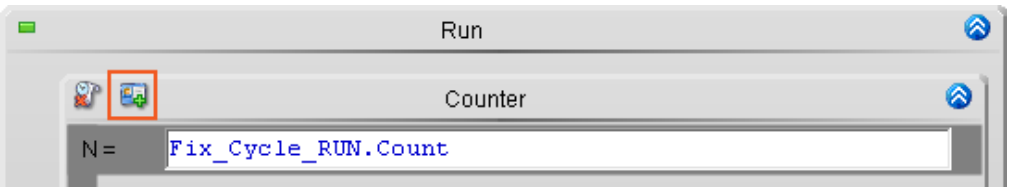
In the Run-step, create a For-loop

- Condition:
  `Fix_Cycle_RUN.Count`

".*Count*" returns the number of elements in the variable.

The For-loop can generate a variable having the current count of iterations.

- To do this, give the For-loop an appropriate name: "*Counter*"
- At the far left of the For-loop, click on the button on the right. This creates a variable having the name of the For-loop.



Activating the For-loop's counter variable

Copy an If-condition from the previous cycle, e.g. "*Step 1 run?*" and paste it into the For-loop.

Change the names and the variables:

- If-Name: "*Step run?*"
  - Condition:
    `Fix_Cycle_RUN[Counter] = 1`
- State: Ramp up
  - State description:
    `Output = RampSlope(pv.Cycle_Ramp/60, Fix_Cycle_Spd[Counter])`
  - Break condition:
    `OnRampEnd(Output)`
- State: Capture parameters
  - Break condition:
    `OnTimeout(Fix_Cycle_Dur[Counter])`

"*Counter*" counts the iterations of the For-loop.

"*[Counter]*" returns the array-element at the respective position.

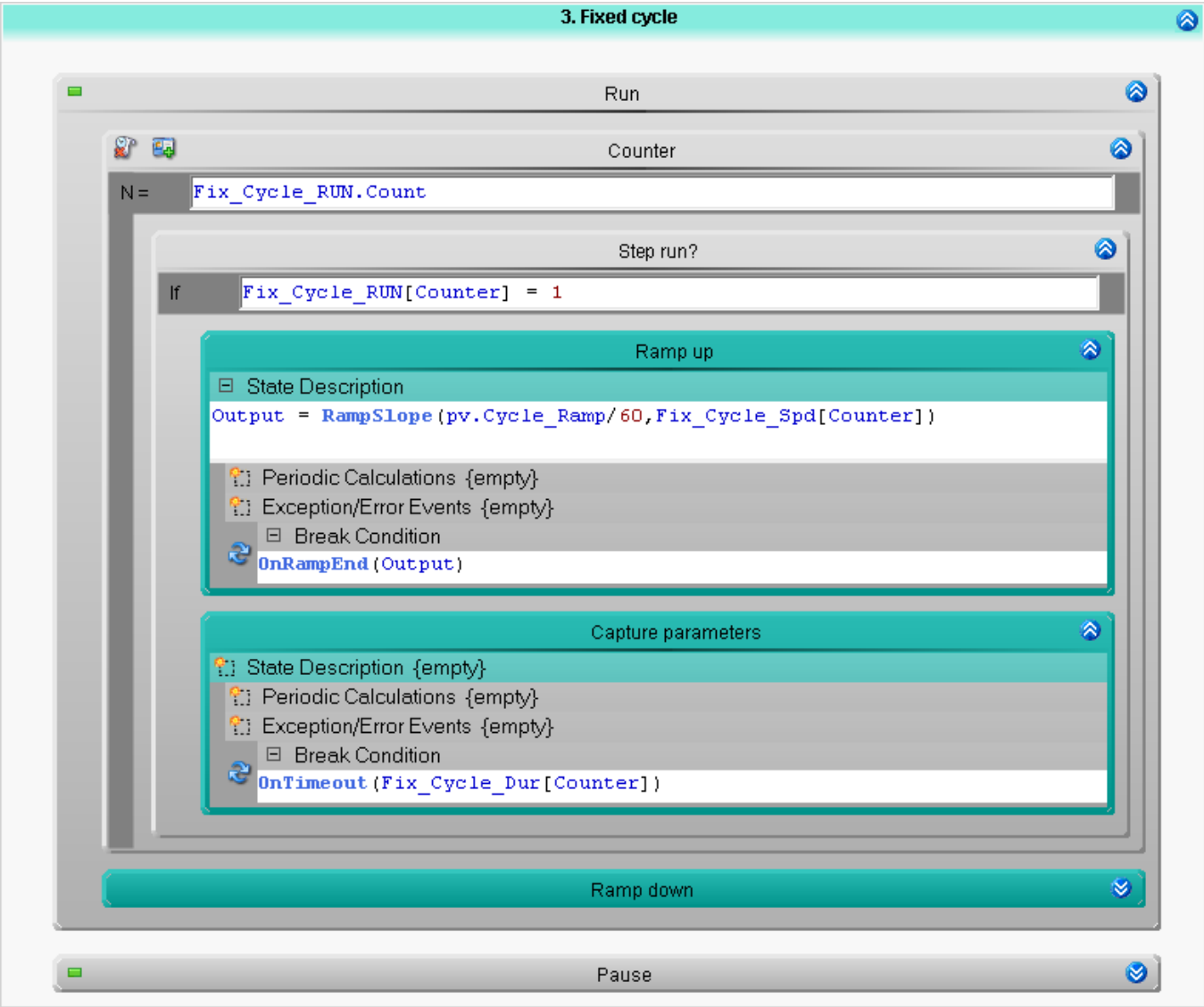
Below the For-loop, add the state "*Ramp down*" from the repository again. And modify the variable.

- State: Ramp down
  - State description:
    `Output = RampSlope(pv.Cycle_Ramp/60,0)`

Thus at the end of the cycle, the rest state is approached again.

Add the Pause-step like for the Run-in cycle.

- State: Pause
  - Break condition:
    `OnTimeout(1)`



The complete routine for the cycle

Measurement and display on a Panel page

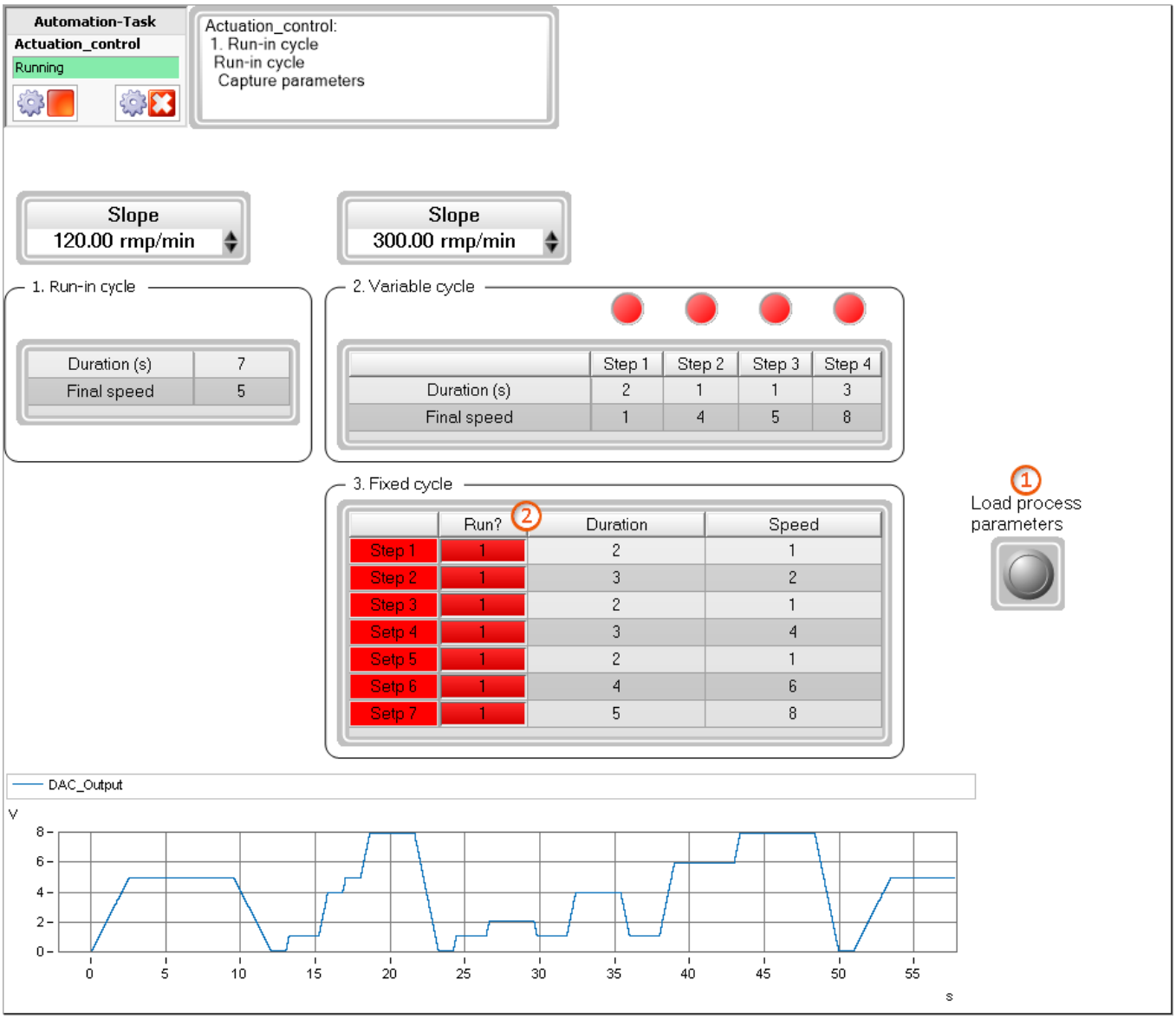
The routine is checked again and the Panel page is prepared for this purpose.

- 2: Widget: Table - linked with "*Fix_Cycle_RUN*", "*Fix_Cycle_Dur*" and "*Fix_Cycle_Spd*"

The settings for the table should be as follows:

- **Automatic table length**: active. In this way, the table automatically adjusts to the count of values in the dat-file.
- **Editor**: Fix_Cycle_RUN as Editor (column-property): "*Switch*".
  Tip: Use the Zones-editor to provide a color-coding to denote active and deactivated individual steps.
- **Direction**: "*Forwards*". The top element should be the first and not the last.
- **Allow a change of direction**: "*No*". This ensures that the direction for a column cannot be changed accidentally.

Start the measurement.



Display on the Panel page

Be aware that any changes to the values will have no direct effects on the procedure. When the values of a variable are changed, the new values are used without any further action **upon next starting the measurement**.

Note: You can also run the command "*Load variable*" automatically in response to an event. E.g. after loading the experiment.

14.9.5.5 Enhancement: Cycle-control

We want to make it possible to activate/deactivate individual cycles. Additionally, we will make a display of which cycle is currently running. To do this, create four additional variables for the Task:

Variable	Property
pv.Run_in_Cycle	Integer - Default value: 1; persistent
pv.Var_Cycle	Integer - Default value: 1; persistent
pv.Fix_Cycle	Integer - Default value: 1; persistent
pv.Status	Default value: 0

In the Run-step of each of the three cycle routines, place an If-condition and give it the name: "*Perform cycle?*"

As its condition, insert the corresponding one of the following:

```
pv.Run_in_Cycle = 1
pv.Var_Cycle = 1
pv.Fix_Cycle = 1
```

The content will now be performed when the variable's value is 1.

Insert a new state in the If-condition:

- State: Set status
  - State description:
    `pv.Status = 1`

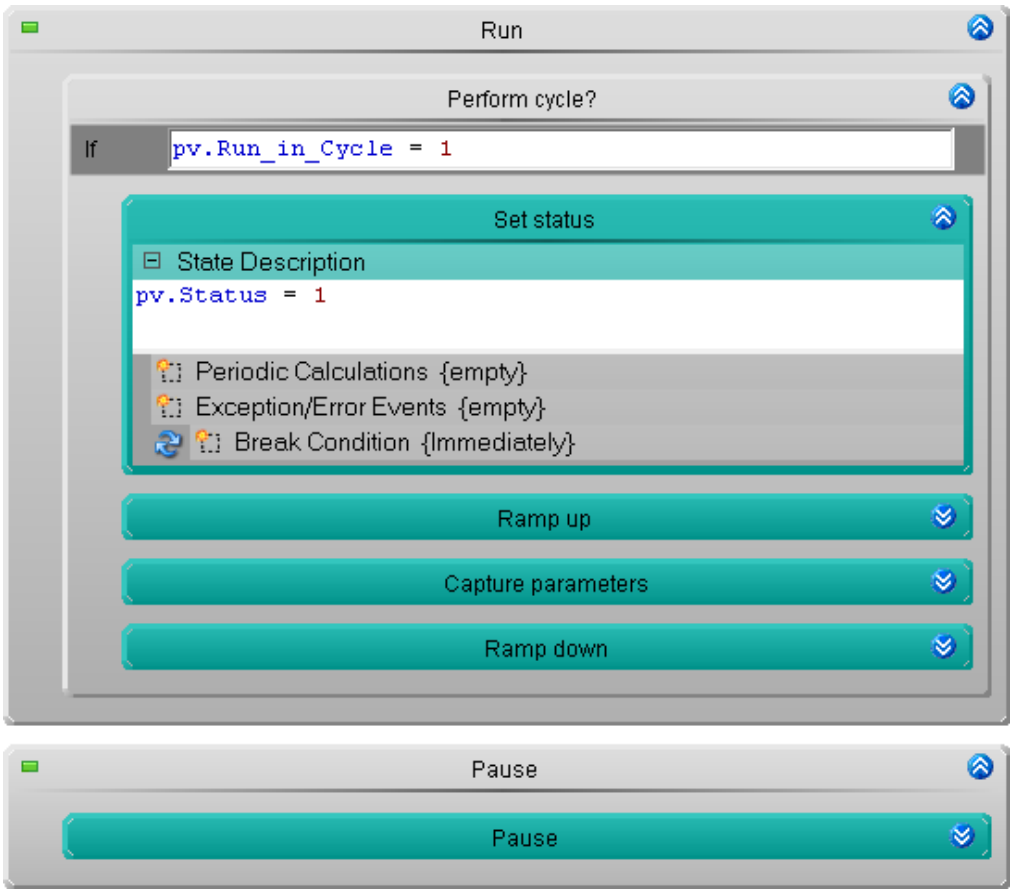
Set up this state in the If-condition of all three cycles.

For the individual states you can use the numbers 1, 2, 3, 4, 5, ... Then, in the display in a Widget: e.g. "*LED*" you would create individual zones for checking whether the value 4, for example, is currently applicable.

Tip: You can also make Widgets display individual bits of a variable. Then the zones remaining would be 0 and 1. In this example, that means: in the second cycle, the variable receives the value "*2*", and in the third the value "*4*". If there were any fourth cycle, the variable would receive the value "*8*" (2^n). In the display in a Widget, the appropriate bit of the variable can then be displayed.

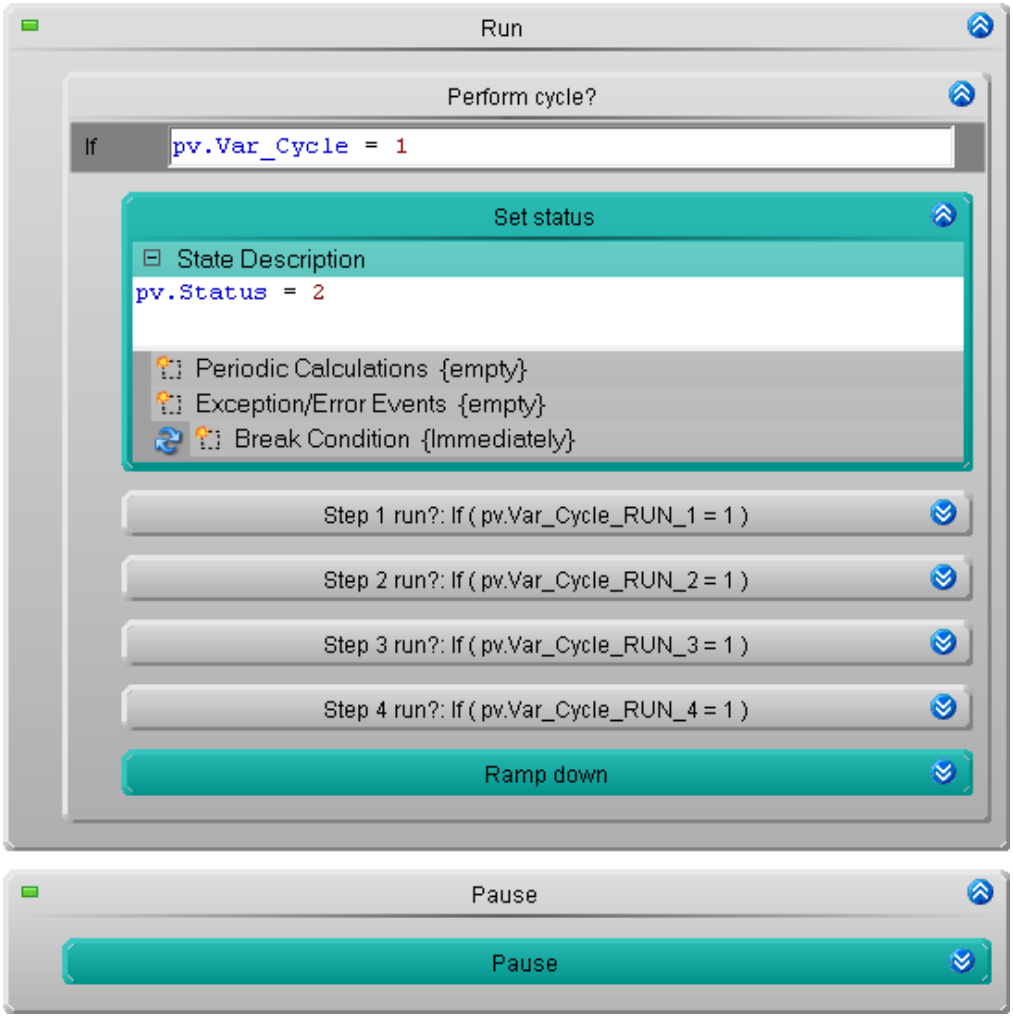
Set up this state also in the End-procedure, as well as in the blocks "*Initialization*" and "*Terminate*". There, "*pv.Status*" is set to "*0*".

Next, the states and loops/conditions of the respective cycles are move to under the new state "*Set status*" by means of Drag&Drop:



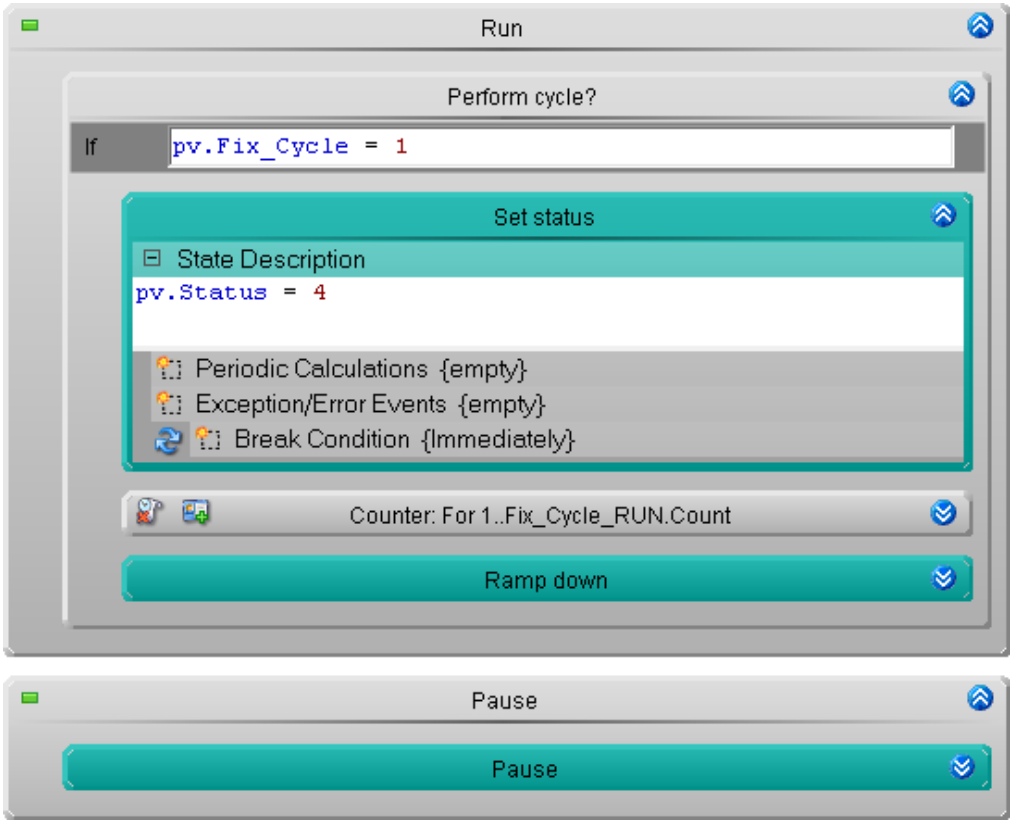
Run-in cycle in an If-condition

Also Cycle 2 accordingly:



Variable cycle in an If-condition

And also Cycle 3:



Fixed cycle in an If-condition

Measurement and display on a Panel page

The routine is checked again and the Panel page is modified. But before this, please perform "*process configuration*" to ensure that the new variables are available.

- 1: Widget: Switch/Pushbutton > 3 x Illuminated push-button (switching behavior: Switch) - linked with "*pv.Run_in_Cycle*", "*pv.Var_Cycle*", "*pv.Fix_Cycle*" - modify the colors again accordingly.
- 2: Widget: LED (Designer) > 3 x LED - linked with "*pv.Status*".

LED - properties:

- **Bit-Mask**: Set Bits 0, 1 or 2 according to the respective position.
- **Colors**:
  On: Green
  Off: Transparent

Place the LED Widgets in the background, so that it is displayed behind the cycles' display texts.

The three LEDs indicate which of the cycles is currently running. Using the switches on the right side, you can activate/deactivate the cycles.

Display on the Panel page

Start the measurement → All three cycles are performed. The LEDs indicate which state is running. You can now active/deactivate individual cycles.

14.9.5.6 Enhancement: One-time start phase

We will make an option to perform the first cycle only a single time after the start, but not again upon any repetitions. To do this, additional variables are needed: one to specify whether the cycle is to be performed multiple times, and one for determining whether the cycle had already been performed before.

Create two additional variables:

Variable	Properties
pv.Run_in_Include	Integer - Default value: 1; persistent. On the Panel, you specify whether the cycle is to be performed again
Run_in_Repeat	Integer - Default value: 0. Auxiliary variable indicating whether the first cycle had already been performed

There are a variety of ways to implement these goals. For instance, modify the IF-condition in the Run-In cycle's Run-step:

- Previously: Only run the content when the cycle is to be run.
  `pv.Run_in_Cycle = 1`
- New: Only run the content if the value of the variable "*Run_in_Repeat*" is still "*1*".
  `(pv.Run_in_Cycle = 1) AND (Run_in_Repeat = 1)`

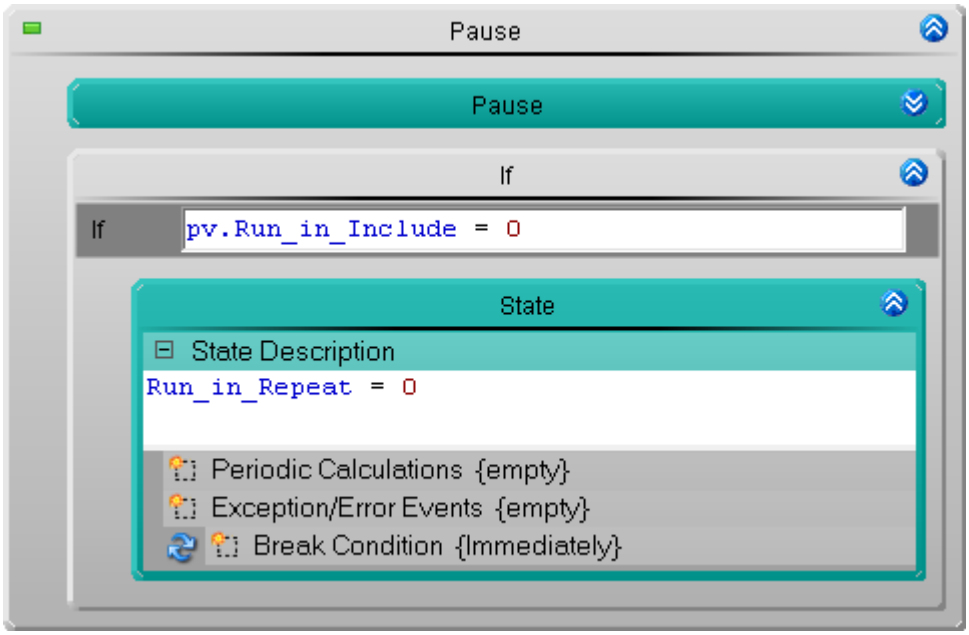
If the first cycle is not to be repeated, then after the first run, the variable "*Run_in_Repeat*" is set to "*0*".

To do this, add another IF-condition to the Pause-step, for example:

- Condition:
  `pv.Run_in_Include = 0`

Add a new state to the If-condition:

- State: Deactivate repetition
  - State description:
    `Run_in_Repeat = 0`



In order that the cycle be ready to be performed upon the next measurement start, the variable in the Initialization is reset. To achieve this, expand the state: "*Set status*" in the block "*Initialization*":

Initialization > Step: Reset

- State: Set status
  - Expand the existing description with:
    `Run_in_Repeat  = 1`

Measurement and display on a Panel page

Next, the routine is checked again and the Panel page is prepared for this.

Create a button linked with the variable "*pv.Run_in_Include*"

Upon each repetition, the Run-in cycle is performed. If the variable is set to "*0*" by means of the button, the Run-in cycle is only performed once more, but after that not any more.

14.9.5.7 Enhancement: Iteration counter with defined end

Now we intend to stop the routine after a specified amount of repetitions. For this purpose, additional variables are needed: The Maximum Iteration Count, and a counter.

Create two more variables:

Variable	Properties
pv.iteration_current	Integer - default value: 1: Iteration counter
pv.iteration_max	Integer - default value: 3; persistent: Maximum Iteration count

Upon starting the measurement, the counter starts from the beginning. To do this, expand the state: "*Set status*" in the block "*Initialization*":

Initialization > Step: Reset

- State: Set status
  - Expand the state description with:
    `pv.iteration_current = 1 ; iteration counter`

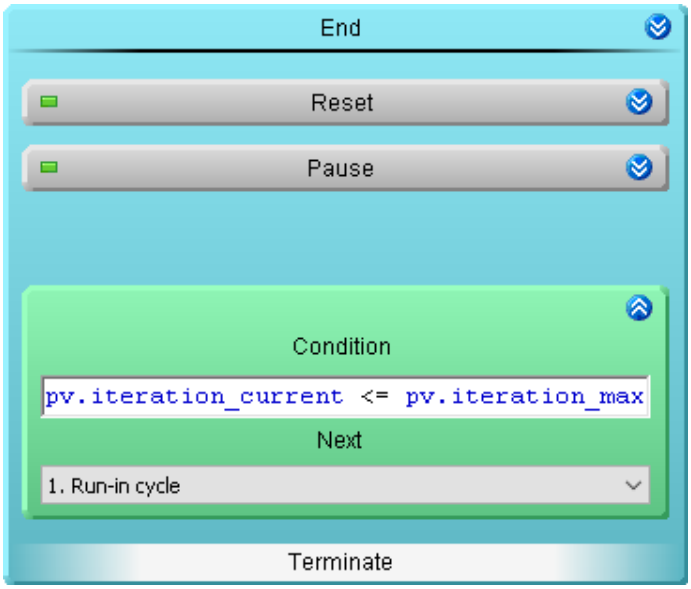
At the end of each run, the counter increments up. To do this, expand the state: "*Set status*" in the End-procedure:

Step: Reset

- State: Set status
  - Expand the existing state description with:
    `pv.iteration_current = pv.iteration_current + 1`

At the end of the End-procedure, the system checks whether an additional repetition may be performed. To do this, set a condition in the existing Next-branch in the End-procedure:

- Condition
  `pv.iteration_current <= pv.iteration_max`



Next - with a condition

In the Terminate-block, the counter is reset, since no more cycles are to be performed. To do this, expand the state: "*Set status*" in the Terminate-block:

Terminate > Step: Reset

- State: Set status
  - Expand the existing state description with:
    `pv.iteration_current = 0`

Measurement and display on a Panel page

Display the variable "*pv.iteration_max*" with the ability to edit, and the variable "*pv.iteration_current*" with write-protection.

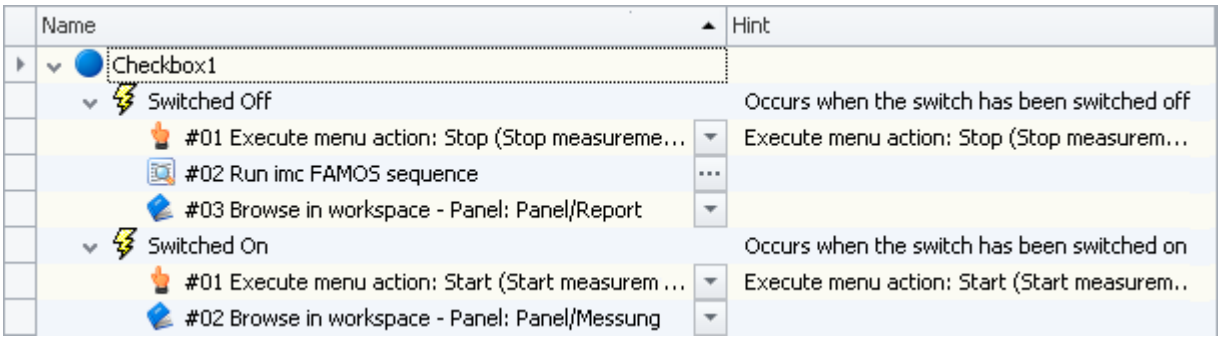
15 Sequencer, events and commands

Using imc STUDIO, you are able to create sequences, which carry out various commands in sequence either manually or automatically.

Sequences can be created in a variety of imc STUDIO components:

Component	Description
Sequencer	On the Sequencer-page, you can **create automated routines**. In the "*Sequence table* 1684", you can **use all commands**.
Panel Automation	You can **link events with commands** by means of the "*Event Dialog* 1686". The commands are executed when the event occurs. Here, you can use selected commands.
Events	On the Sequencer-page, you see an overview of all events 1694 and can configure these here. Additionally, you can create your own events 1696. Here, you can use selected commands.

The sequence



Sequence table: Example having multiple commands

A sequence consists of a table (referred to below as the "*Sequence table*"). The commands appear there in a tree diagram. The structure and complexity of the Sequence table depends on the component used.

Chapter overview

Synopsis	Section
Creation and configuration of a sequence of commands. Adding, moving and making settings for commands.	Creating a sequence of commands 1690
An overview of the Sequencer-page's user interface.	Sequencer page 1682 Sequence table 1684
What to pay attention to when starting/stopping the sequence.	Running and stopping 1688
Events relating to Panel-switches or Automation.	Event Dialog - Panel and Automation 1686
Creating one's own events.	User-defined events 1696
Description of all commands.	Command Reference 1706

15.1 Sequencer page

On the Sequencer page, you can **create automated routines**. The routines consist of "*commands*", such as "*Load*" and "*Start*" for experiments, "*configuration dialogs*" and "*imc FAMOS sequences*" to be carried out while Sequencer runs.

The sequence is created in a **table** 1684. The commands appear there in a tree diagram. The available commands can be found, among other places, in the "*Commands* 1683" tool window.

Further, you can **link the commands to "*events* 1694"**.

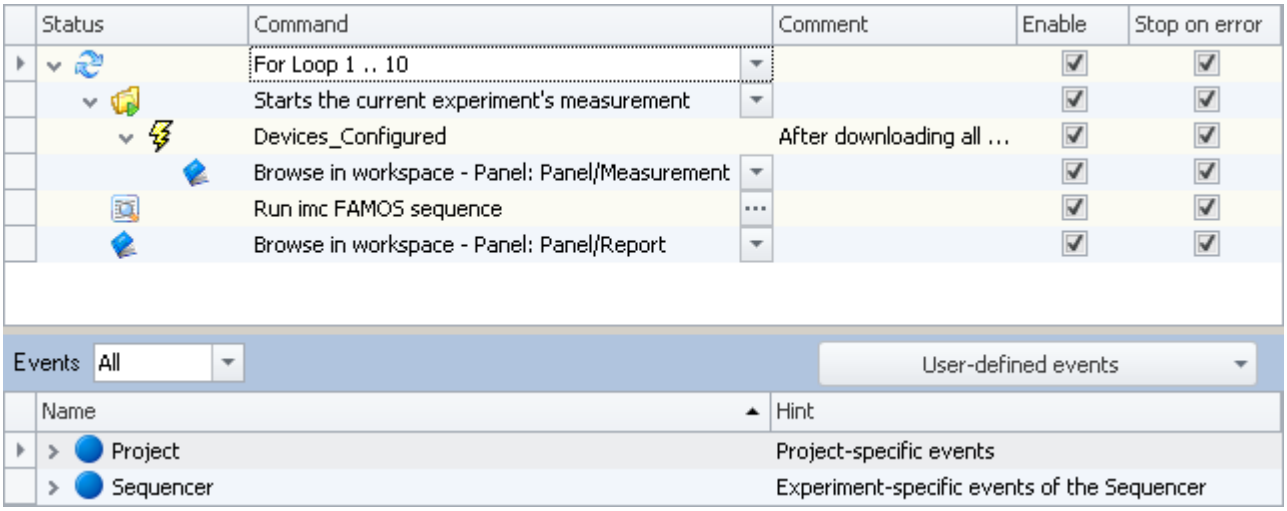
Reference Command Reference

How the individual commands operate is described in the chapter "*Command Reference* 1706".

Opening and configuring Sequencer

In order to open the Sequencer-page, click in the Navigator 139 ( ) on the title having its name.

The image below shows an example of a sequence table 1684 with multiple commands and the event handling 1694.



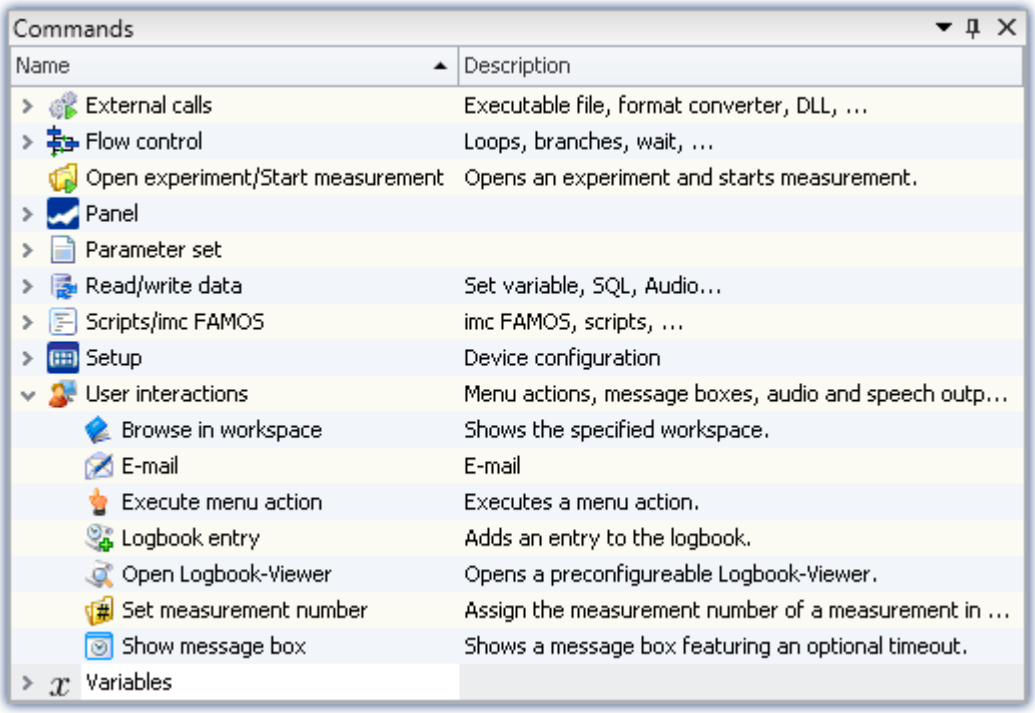
Sequence table with routine of the plug-in "Sequencer"'s commands

Note Saving the Sequence Table

The **content of the Sequence Table** is **saved** 110 **along with the experiment**.

15.2 Tool window commands

The component Sequencer has a tool window called "*Commands*". This comprises commands with which the sequence is created.



Commands tool window in Sequencer plug-in

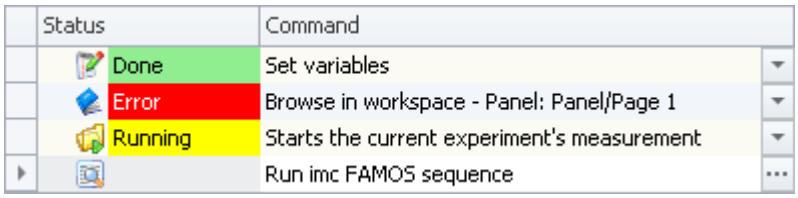
Reference Command Reference

How the individual commands operate is described in the chapter "*Command Reference* 1706".

15.3 Sequence table

The sequence table contains multiple columns.

Columns - Command sequence table

Column	Description
Status	In this column, the commands' status is indicated.  *Status in the sequence table* • **Empty:** The initial state before the sequence's first start. • **Running:** The command is still running and has not been concluded. • **Done:** The commando was concluded successfully. • **Error:** The command was concluded by an error.
Command	The command's name. The text entry can not be changed. The command's settings are accessible via the button at the cell's right edge.
Comment	Click in this cell and enter a comment.
Enable 1685	Indicates whether a command is run or ignored. By default, this checkbox has a mark in it, which means that the respective command will be performed during runtime. However, you can also deactivate a command by de-selecting this option.
Stop on error 1685	With this option, the sequence is halted 1689 when an error occurs while the command is running.

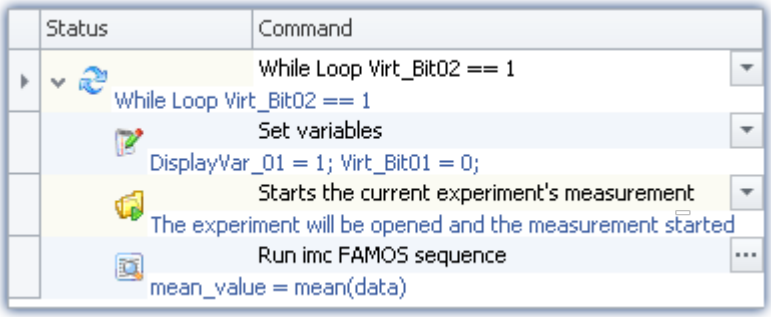
Reference More than one command sequence

With the help of the switch command you can simulate several command sequences. See the example: "*More than one command sequence* 1780"

Options - Display in the sequence table

Some commands offer additional information which you can have displayed in the Sequence table. (Applies only to the Sequencer main sequence, not to the events)

The following items belonging to the group "*Options*" ( ) are available via the table's context menu:

Option	Description
Show all events	All events which the current command ("*Open experiment/Start measurement* 1732") comprises are shown. Events used are shown even if the option is not activated. Just as there are events for each measurement, there are also events which can only be configured for a measurement.
Show property preview	The Sequence table sows important information from the commands.  *Properties preview*

Groups - Loop, IF, Switch

By placing an **command under a loop** (Loop) or a conditional branch (If, Switch), you create a so-called "*Group*". This means the sequence has groups and levels and branches, which can be represented by a tree diagram.

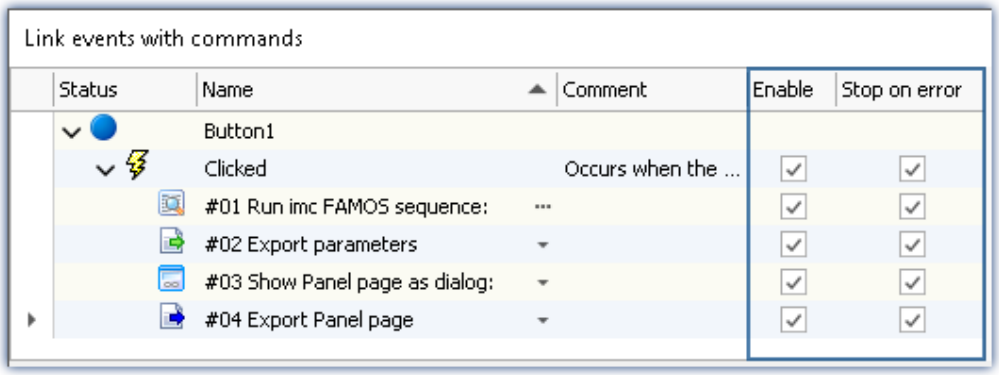
The commands within a group are executed when the group as a whole is activated.

By means of drag&drop, you can move or copy an entire group.

Groups can not be generated in conjunction with events.

"Enable" and "Stop on error"

In the sequences, you are able to activate/deactivate individual commands. Additionally, you are able to make a setting there which governs whether the sequence should be interrupted 1689 if an error occurs.

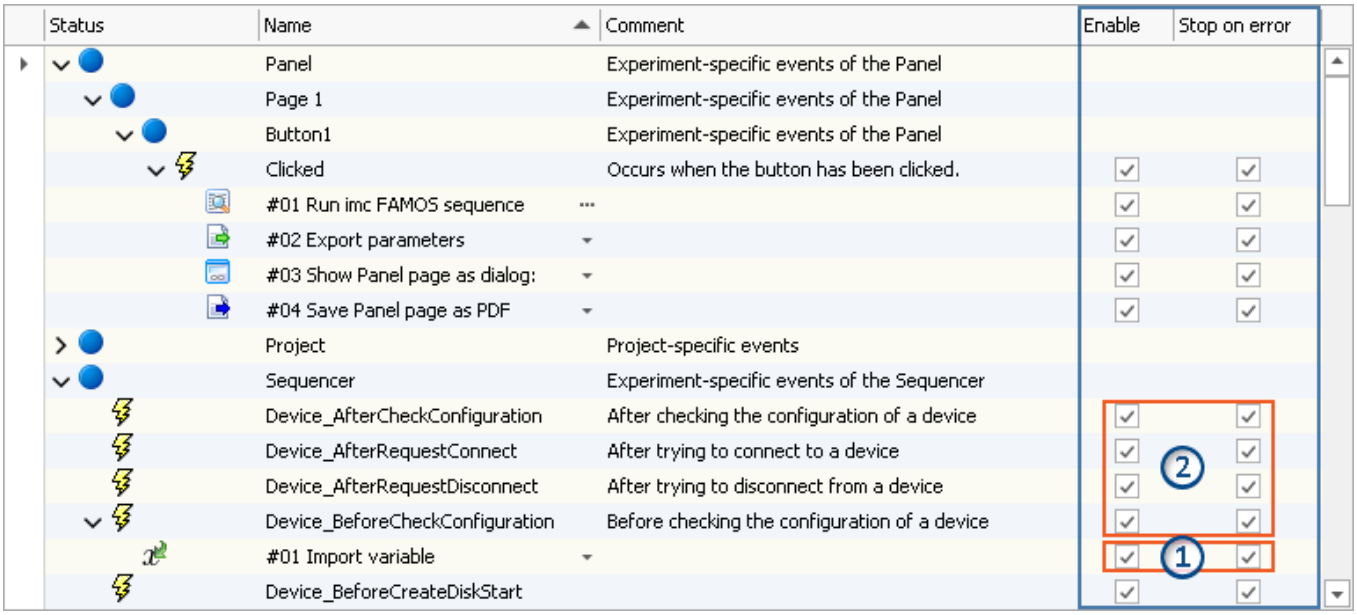


Commands associated with a button's event

You are able to deactivate complete events along with their commands, or only individual commands.

Example of "*Enabled*": A Timer-event is to start only once its configuration has been completed. You are able to deactivate it and to later re-activate it at the time desired.

Events and their commands



Commands in the Sequencer associated with the events
1: Applies to the respective command
2: Deactivation applies to all commands associated with the event

If the option is **deactivated for the event**, this applies to **all associated commands**. If the option is activated for the event, the setting is applied to the respective command.

15.4 Event Dialog - Panel and Automation

Certain **events** can be **linked with commands**. The commands are performed **upon occurrence of the event**.

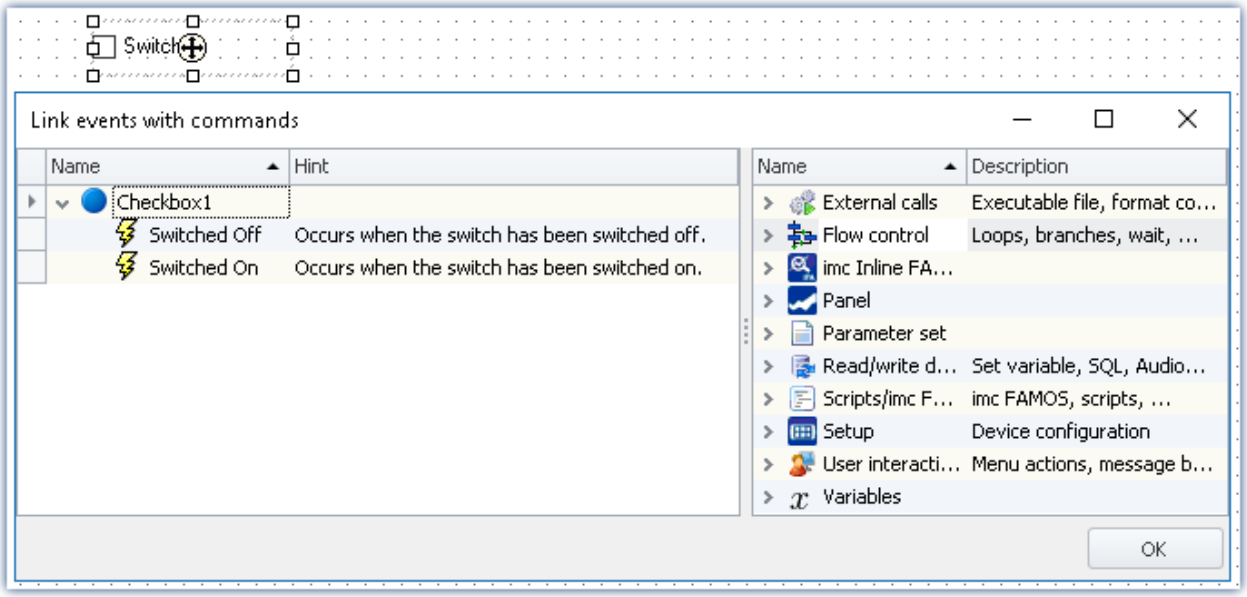
For configuring the event, the "*event dialog*".

Events can, for example, occur in the Panel when a switch is activated or in the Automation upon entering or exiting a signal state.

Example

As an example, the two events ( ) belonging to a "*Switch*" from the group "*Standard*":

Event	Description
Switched off	from On to Off
Switched on	from Off to On



"Event Dialog":
Events belonging to a switch from the group Standard

15.5 Running and stopping

The sequence is read, analyzed and implemented step by step. Analysis of the commands occurs at runtime. The top command in the sequence table is the first to be run.

Except in cases of intentional intervention, a **running sequence only stops** when the **last command** has been run.

Note Behavior upon stopping

If the sequence is stopped while a command is running, that command is halted and not executed all the way to its conclusion.
Special note concerning the command "*Open Experiment /Start Measurement* 1732": In this case, the measurement itself is additionally halted.

Running Sequencer-main sequence

Start/Stop the Sequencer via either the menu ribbon or the context menu 1692.

Menu item	View
Extra > Start ( )	Complete
Home > Start ( )	Complete

Targeted running of commands an events via the context menu

Action	Description
Start	Starts the sequence from the beginning of the table.
From selected command to end of group	Starts the sequence from the location where the context menu had been opened. Sequencer-Main sequence: The process flow remains in the current level/group 1685. This means, for example, that if the action is started from within a loop, the process flow ends at the end of the loop.
Only selected command	Only the command selected is run.

Stopping Sequencer-main sequence manually

You can stop the running Sequencer either via the context menu or using the menu item "*Stop*".

Menu item	View
Extra > Stop ( )	Complete
Home > Stop ( )	Complete

Menu item	Beschreibung
Abort	Stops the Sequencer and exits the current command.

This method **only stops the Sequencer**. Running commands belonging to events are not affected.

Stopping Sequencer-main sequence by command

You can stop the Sequencer by a command 1724.

This method **only stops the Sequencer**. Running commands belonging to events are not affected.

Stopping due to an "*Error*"

When there is an **error notification**, it usually means that something is interfering with the processing of the sequence. For this reason, **all sequences are immediately stopped** (including events). The current command posts "*Error*" in the status bar (see for instance "*Sequence table* 1684"). This is independent of the error notification's source. The notification may not have anything to do with the sequence.

Note

Deactivating "*Stop on error*"

If you wish that a **command is not stopped by an error notification**, then remove the check mark from the "*Stop on error*" box for the respective command. If an error is reported while the command is running, the sequence is not interrupted. This can be necessary in cases where, at certain points in the sequence, errors are possible but should not interrupt the sequence because they are handled in a different way.

Treat error as warning

Some commands offer the ability to **convert "personal" error messages to warnings**. E.g. the "*Load/Refill variable* 1763" command with the option "*Treat error as warning*".

The command can return an error if, for instance, a variable can not be loaded because it is already present. So if you wish to load multiple variables from a file, of which one of these might already be present, you can put a checkmark in this option's box. Then the message indicating that the variable can not be loaded will be outputted as a warning.

15.6 Creating a sequence of commands

The following examples are illustrated in the sequence table. However they also apply to the "*Event dialog* 1686".

To add a command to the sequence table, there are two options:

by Drag&Drop	• Use Drag&Drop to move the command from the "Commands" tool window to the desired position. • Release the mouse button.
by the context menu 1692	• Open the context menu of the desired position and select a command.

You can place a command at different positions.

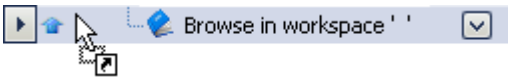
	Before the current position
	Below or within the current position: groups 1685 (Sequencer) or events 1694 (following also called group)
	After the current position

Examples of Drag&Drop

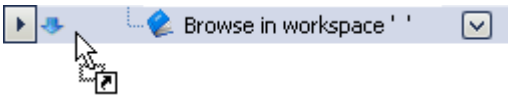
Use Drag&Drop to move the command from the "*Commands*" tool window to the desired position (in the "*event dialog*" from the right side of the dialog.).

Adding commands before or after existing commands

Commands can be added either **before** or **after** an existing command. To do this, move the mouse either a little upward/downward.



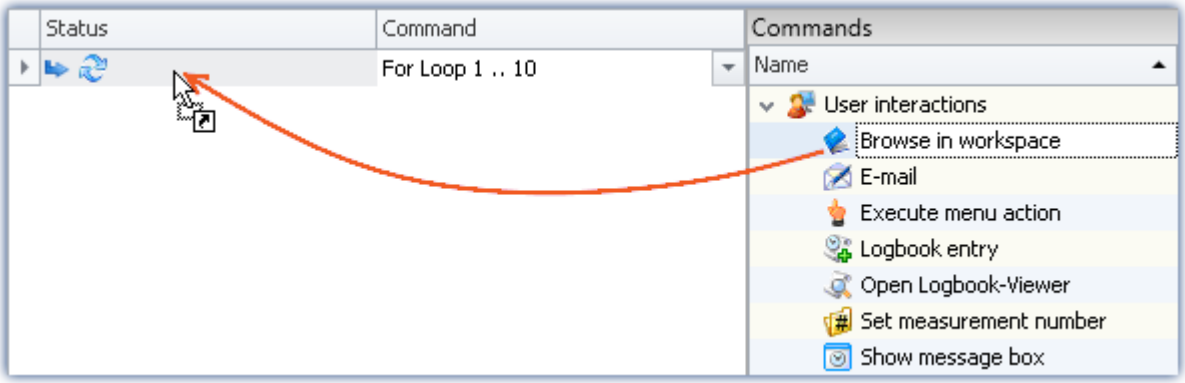
Inserting before the command: "Browse in workspace"



Inserting after the command: "Browse in workspace"

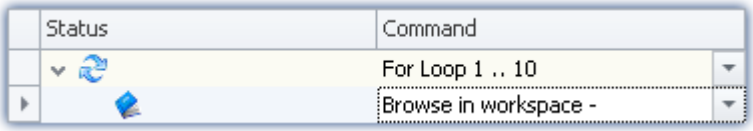
Adding commands to a group

Use Drag&Drop to place the command in a group, as seen below:

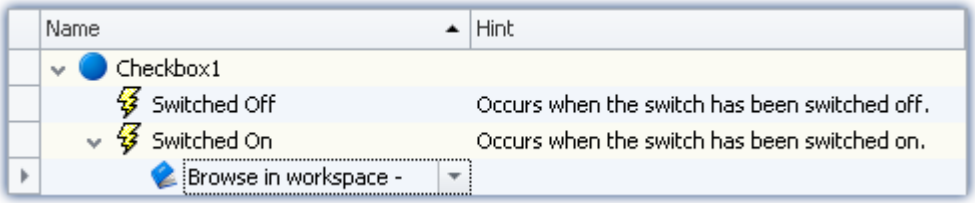


Example: Using Drag&Drop to add the command: "Browse in workspace" to the loop

To the left of the even the position is specified ( ). Release the mouse button. The command is added to the group:



Result: sequence table



Result: Event dialog

Deleting a command

To delete selected commands, highlight them and select the context menu item "*Delete*".

Action	Description
Delete	Deletes the selection
Delete all	Deletes all commands in the Sequence table or in the events.
	If all commands in the events are deleted (via the Sequencer), then the commands associated with the events of the Panel pages are also deleted.

Configuring a command

Most commands need to first be configured appropriately before they will work. E.g., for "*Load experiment*", it is necessary to define what is to be loaded. Or when using "*Export*", a setting must be made to specify what to export and to where, etc.

To make the configuration, select the command in the sequence table. **Click on the button** to the right of the command's name. In most case, a configuration dialog will open.

The **various configuration possibilities** are presented in the description of the commands: "*Command Reference* 1706".

15.7 Context menu

With the context menu, you can control all of the Sequencer's functions. In order to open the context menu, click on the right mouse button in the free area of the sequence table.

Menu item	Description
Add or New 1690 before/after/into	Sets up the selected command at the specified location.
Cut	Clips out the selection and moves them to the Clipboard.
Copy	Copies the selection to the Clipboard.
Paste	Inserts the content of the Clipboard at the selected position.
Delete	Deletes the selection
Delete all	Deletes all commands in the Sequence table or in the events. If all commands in the events are deleted (via the Sequencer), then the commands associated with the events of the Panel pages are also deleted.
Start 1688	
Start	Starts the sequence from the beginning of the table.
From selected command to end of group	Starts the sequence from the location where the context menu had been opened. Sequencer-Main sequence: The process flow remains in the current level/group 1685. This means, for example, that if the action is started from within a loop, the process flow ends at the end of the loop.
Only selected command	Only the command selected is run.
Abort	Stops the Sequencer and exits the current command.
Options 1685	Some commands offer additional information which you can show. (Does not apply to commands associated with events)
Show all events	All events which the current command ("*Open experiment/Start measurement* 1732") comprises are shown.
Show property preview	The Sequence table sows important information from the commands.

Menu item	Description
Expand/ Collapse	Expands/collapses either all groups 1685 or the selected group.
Element one level up	The selected commands are extracted from the current group 1685 (e.g. For-Loop, If, ...) and moved one level upward.
Move into for-loop	The selected commands are moved to a For-Loop (group 1685). You can modify the For-Loop at a later time (loop type: For/While, Start and End, ...).

15.8 Events

The bottom region of the Sequencer contains the event handling. Here, you can **link commands with events**. The linked commands are performed when the event occurs.

- Use Drag&Drop to move the command to the event. To do this, follow the same procedure as when creating a sequence 1690 in the Sequence table.

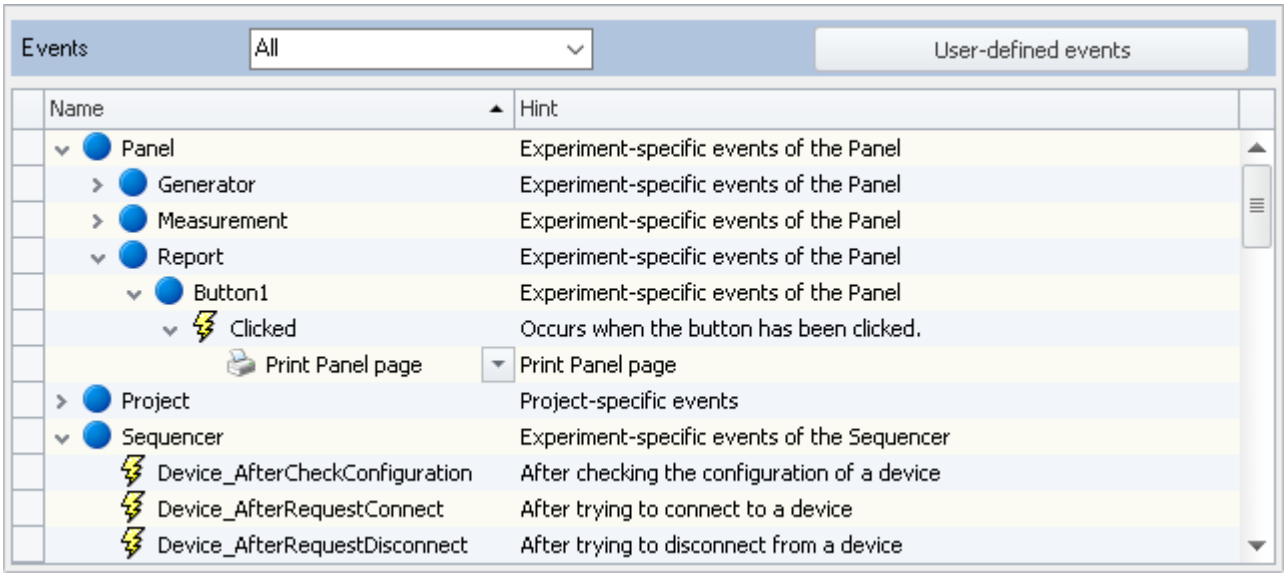
There are a variety of event types:

Type	Description
Panel	Events which are assigned to specific Widgets. E.g. a switch is actuated by a user.
User-defined events 1696	Events defined by the user such as "a variable exceeding a value limit".
(Standard) events	e.g. device actions such as "the measurement was stopped"

There are events specific to various scopes of applicability (with which they are respectively saved):

Scope	Description
Panel	Panel-events which are assigned to certain Widgets
Experiment	User-defined events which only apply within the current experiment
Sequencer	(Standard) events and user-defined events which only apply within the current experiment. These are also applicable when other experiments are loaded using the command: "Open experiment 1731".
Project	(Standard) events and user-defined events which apply to all of a project's experiments.
Command	(Standard) events which only apply within the current command ("*Open experiment/Start measurement* 1732").

The illustration below shows some events in the various applicability scopes. The event list consists of the **event/command name** and a **brief description** (column: "*Hint*").



Events on the Sequencer page

Most events affect device actions. In the column "Hint", there is an exact description of when which event occurs. For some actions, the particular moment in time is important. For instance, before the measurement is started, a Panel page is to be opened. Or after the measurement has concluded, the results and the report are to be exported.

Differing points in time

For the device actions, there is often a point in time "beforehand" and one "afterward". For instance:

Device_BeforeCheckConfiguration	Before preparation of a device configuration. The vent occurs before the preparation. The preparation is only performed once all of the event's commands have been concluded.
Device_AfterCheckConfiguration	After preparation of a device configuration.

Differing numbers of devices - Device_x and Devices_x

Some events are triggered for each device. Others are only triggered once all devices have concluded the action.

Devices_Started	After the measurement start for all devices. The event occurs once all devices have started the measurement.
Device_Started	After the measurement start for one device. The event occurs when one device has started measurement. This means that if you have two devices, the event occurs twice when measurement starts, as long as both devices actually start this measurement.

What is noteworthy here is that both events occur upon starting of a measurement. Thus with two devices, there are three events: two times "Device_Started" and one "Devices_Started".

Attempted and successful performance

For some cases there is a distinction between whether an action is attempted and whether it is started successfully.

Devices_BeforeRequestConnect	Before the attempt to connect with all devices.
Devices_Connected	Once connection with all devices has been established.

Note Avoiding time-sensitive combinations

Once events have been concluded, other actions are initiated. However, you need to plan for a certain delay time before these actions generate results.

Example: Once the measurement has been started, the configuration is to be saved along with the measured data as a parameter set.

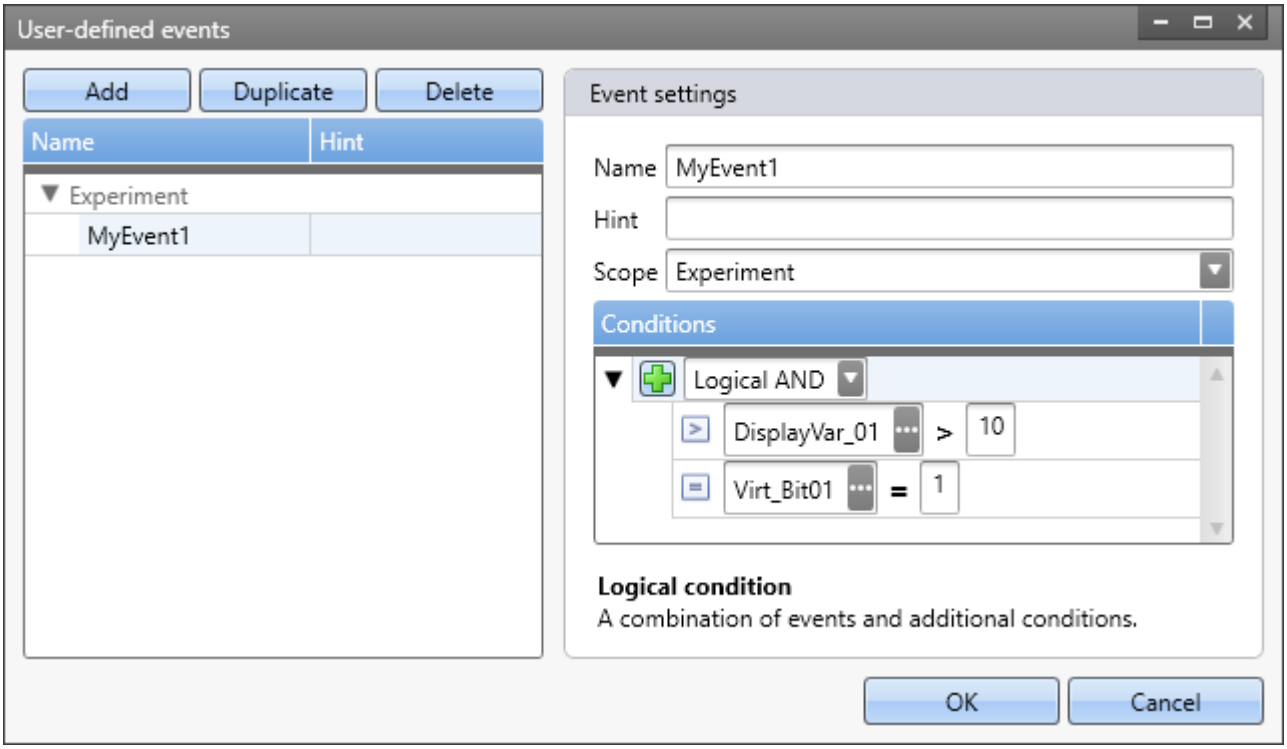
Possible implementation: Set up the command "*Export Parameter Set*" to be associated with the event "*Devices_Started*". Use "*<VARS["Channel_001"].PATH>\Parameter.csv*", for example, as the target folder.

Problem: Once the measurement starts, it takes a (very short) moment before the first measured data arrive at the PC. Next, the target folder in which the data of "*Channel_001*" is defined. The command: "*Export parameter set*" gets working faster and causes an error to occur in this case since the folder is not defined.

Solution: Avoid such time-sensitive combinations whenever possible. In this case, export could be performed at a different point in time, such as after conclusion of the measurement, for example. However, if it is not possible to change the procedure, insert the "Wait"-command (for 2 seconds, for example) at such locations.

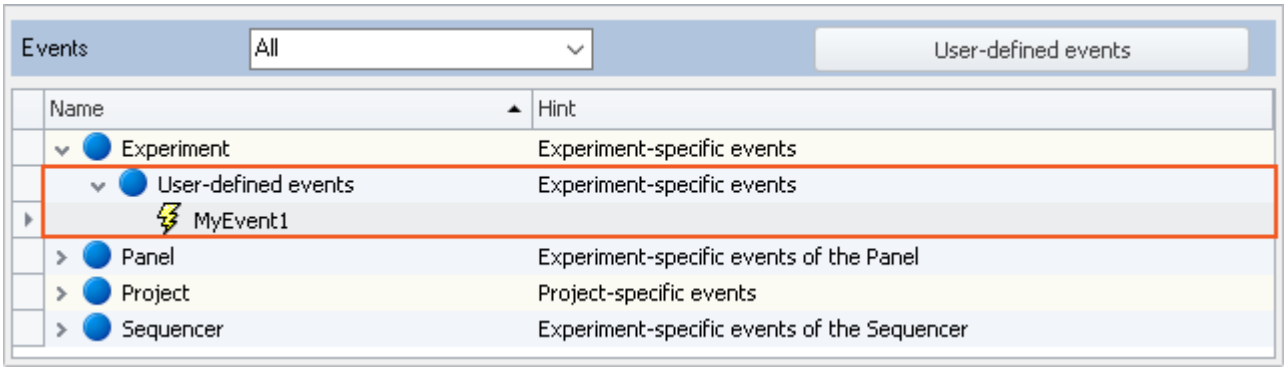
15.8.1 User-defined events

Using the button "*User-defined events*" you can create your own events. Here you can for instance, monitor variables and thus trigger actions in response.



Example: The event has occurred if:
- the Display variable is greater than 10
and
- the virtual bit equals 1

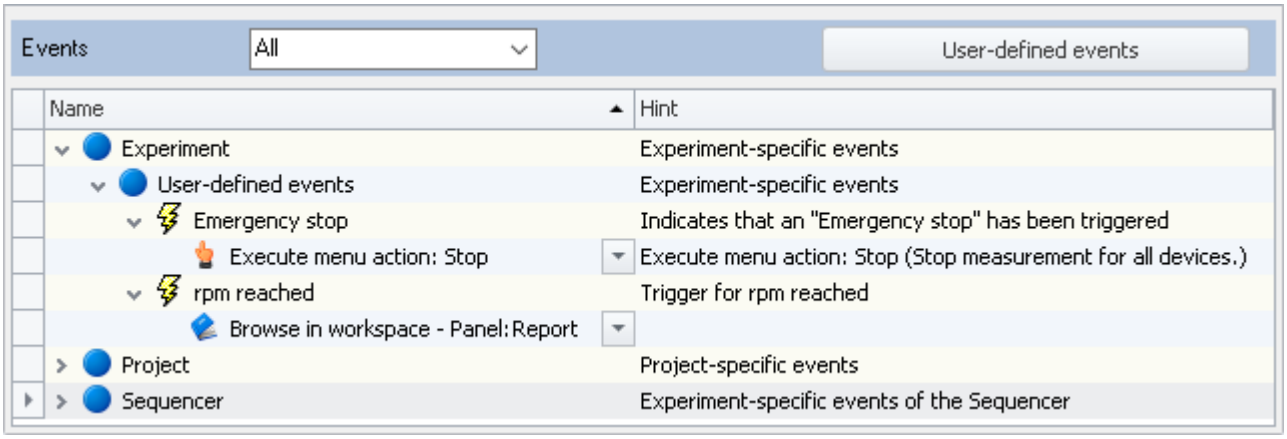
The event is found in the Events-list under the respective scope in the category "*User-defined events*".



New event, without command

Procedure illustrated by two examples

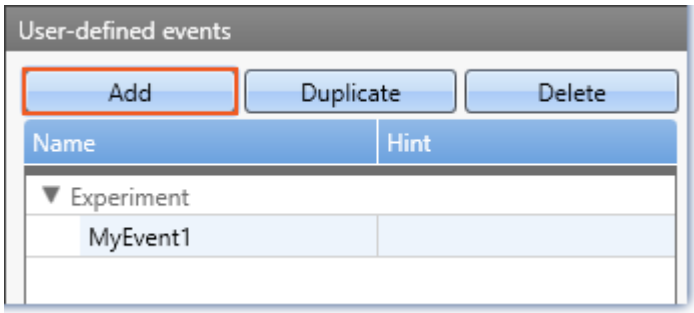
As the result of this example, you will have generated two events. Both events can cause commands to be performed.



User-defined events with commands

Example Monitoring variables

Click on the button "*User-defined events*" in order to open the dialog.
Create an event by means of the button "*Add*". An event appears in the list:

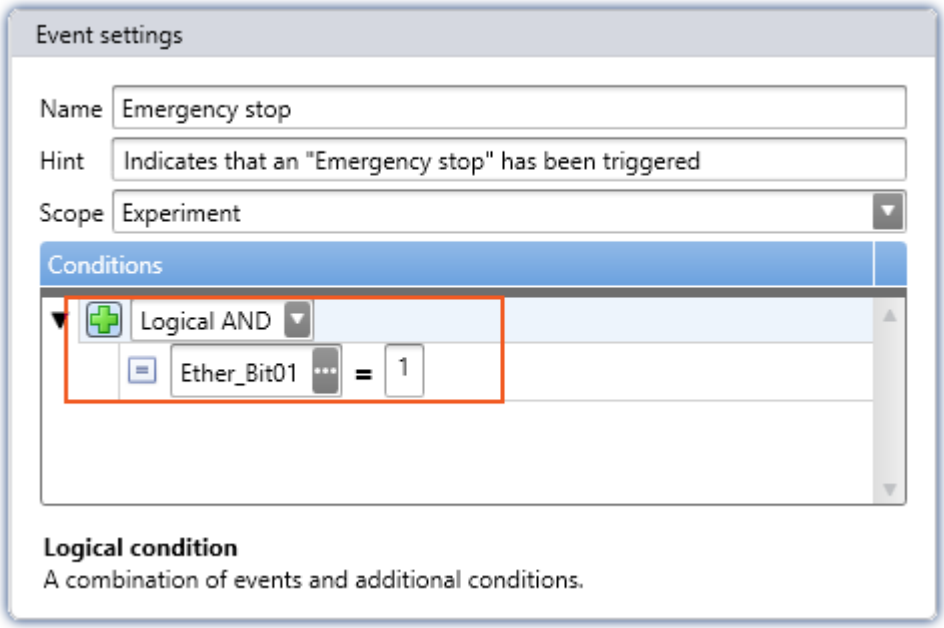


An event is created

Select the condition type via the button "+" ( ) at right.

E.g. "Equals"

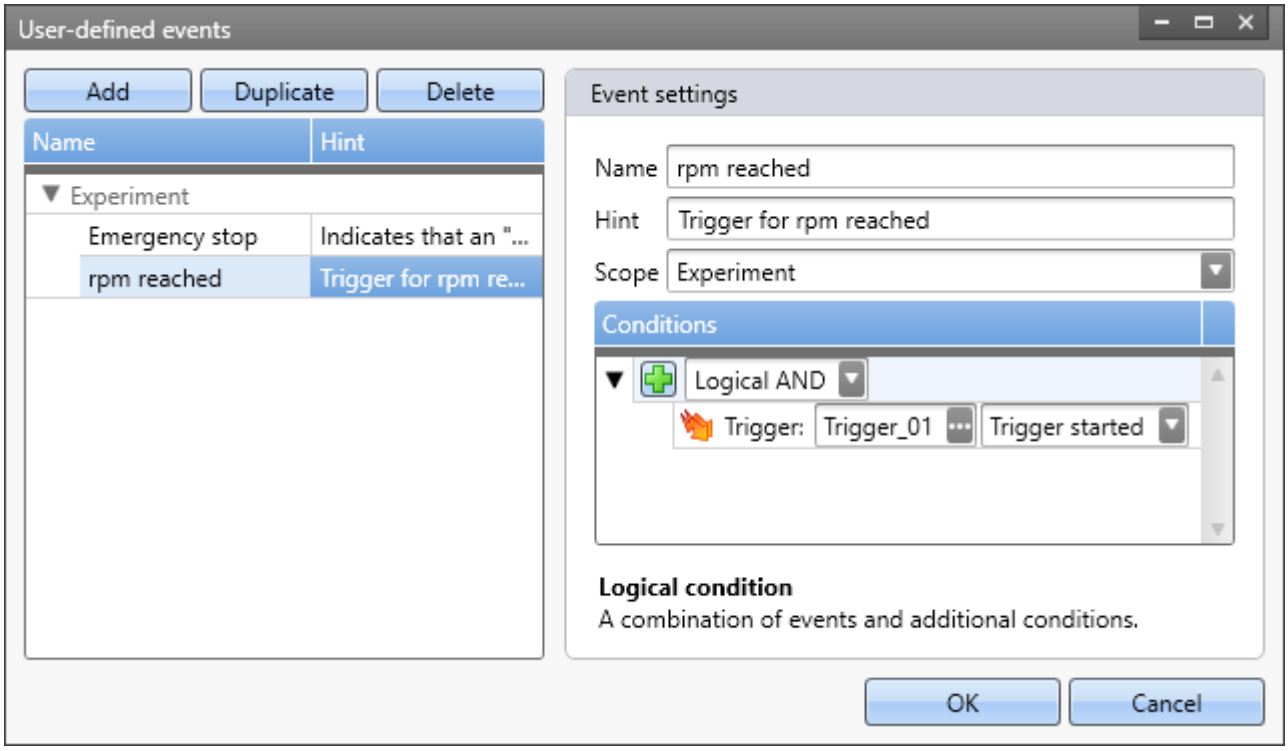
In the condition which has now been created, enter **the variable to monitor at left** and **the value at right**.
Modify the **name and the note (hint)**. These two texts are displayed in the Events-table and help keep a clear overview.



Event settings - Example of an event.
This event occurs if an "Emergency OFF" is displayed via the Ether_Bit01.

Example Trigger event

Create an event by means of the button "*Add*". A second event appears in the list.
Select a condition type by means of the "+"-button (+) at right.
E.g. "Trigger"
In the examination which has now been set up, enter **the trigger to be monitored at left** and **the trigger state at right**.
Modify the **name and the note (hint)**.



In the dialog you will find at left two configured events. After confirming by clicking on the "OK" button, both events can be found in the table.
Next, link commands with the two events.

Condition types

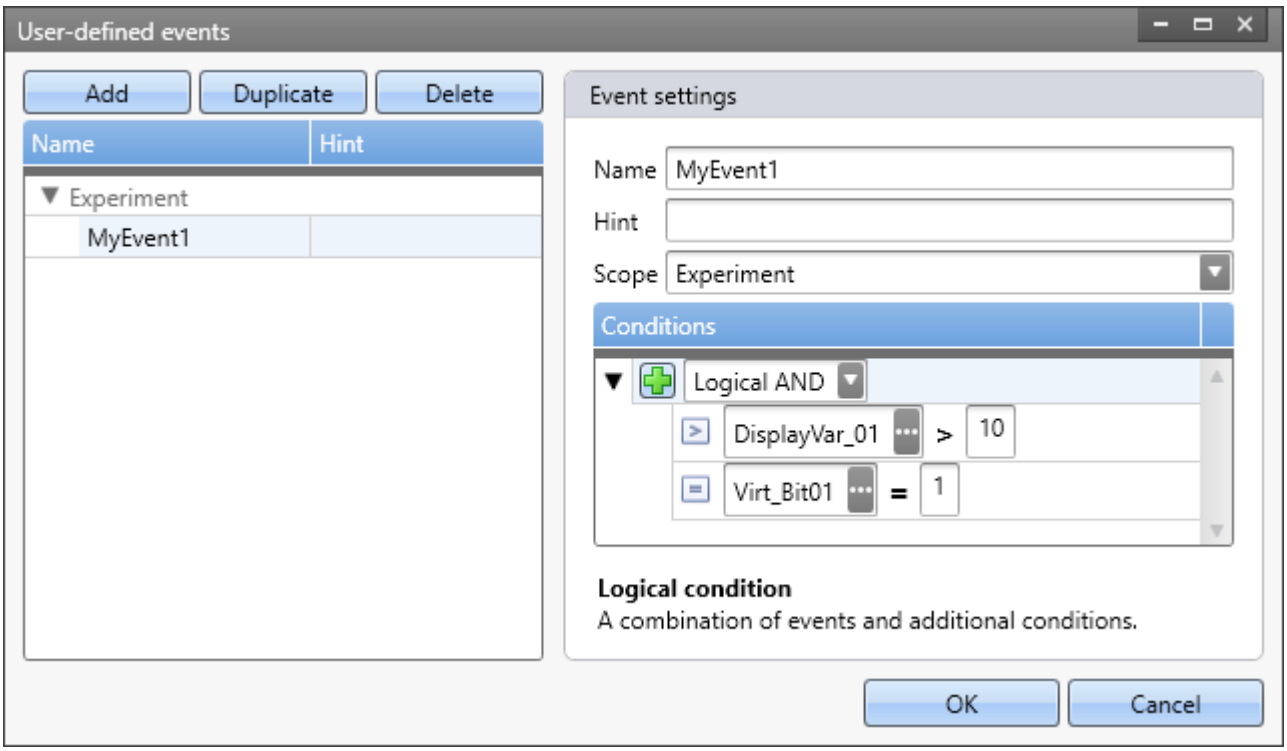
Select a condition type via the "+"-button (+) at right.

Here you find a variety of conditions. Among others, value comparisons such as "*greater*", "*lesser*", "..." which need no explanation at this moment. Below, additional special conditions are presented:

Condition	Description
Logical condition	Inserts a subordinate logical operation. This may in turn contain its own conditions.
User-defined event	This condition is met when another user-defined event occurs.
Trigger	Reacts to trigger states; the available options are "Trigger started", "Trigger stopped" and "Trigger changed"
Timer	Enables the commands to be run cyclically - *Attention: starts immediately upon setting the event (when the OK-button is clicked).*
Value change	This condition is met when the variable's value changes.

Linking with conditions - Logical condition

It is possible to link multiple conditions with an event.



Example: The event has occurred if:
- the Display variable is greater than 10
and
- the virtual bit equals 1

Operator	Description
Logical and (AND)	When all conditions are simultaneously "*true*", the result is "*true*"
Logical or (OR)	When one of the conditions is "*true*", the result is "*true*".
Exactly one condition	When exactly one condition is "*true*" (and no more), the result is "*true*".
Logical NAND	Unless all conditions are simultaneously "*true*", the result is "*true*". Or stated another way: If all conditions are simultaneously "*true*", the result is "*false*".
Logical NOR	Unless one of the conditions is "*true*", the result is "*true*". Or stated another way: If one of the conditions is "*true*", the result is "*false*".

Example

Condition: Within a range, but not including the range boundaries: x>a AND x<b
Condition: Within a range, including the range boundaries: x≥a AND x≤b

Scope

By means of the scope, you define where the event is to be saved and for what it is to apply. See "*Scope* 1694" of events.

Notes and tips

Tip: Suppressing the timer

The timer is to be tripped only once a certain point in time is reached.

Solution: Use an AND-operator with a second condition (e.g. Virtual Bit=1).

Disadvantage: the timer continues to run in the background the whole time. The timer's actual start time can not be chosen.

Note: When the same event occurs too soon

The linked commands are all run in succession. Only once one has been concluded can the event be triggered again. If the event occurs again during processing, a warning appears in the logbook.

Note: Variable does not exist

Events whose variables either don't exist or don't exist yet are ignored. As soon as the variable does exist, it is monitored. No notes or warnings are posted to indicate that any error has occurred.

Reason: There are regular states in which this is normal; e.g. when a project-specific event is involved, but the variable only appears once a certain experiment has been loaded.

Note: Initialization / First value does not count as a change

There can only be a change to a trigger, variable or channel if a previous value had already been present. Thus for instance, if during the "*Prepare*" process for a measurement, a trigger's status changes from "*not defined*" to "*not released*", this is not a change. In consequence, no event will be triggered in this case.

This also applies if the event is defined as "*Channel < 5*". If the channel's value is < 5 already at the beginning of the measurement, the event is not triggered. And this also applies when measurement starts a second time and the channel's value was < 5 already at the end of the first measurement.

If you wish to apply the condition at the start of the measurement, then link the event to an appropriate trigger (AND conjunction), for example.

Changes to the variable are not recorded because it changes too fast

An event is to be triggered if, for instance, a DI-bit briefly transitions to "1". If the event is very short, the change is not even transferred to the PC.

Solution: You can prolong the event, for instance by using the imc Online FAMOS-function: "`Monoflops`". In this way, you create a virtual channel, which transmits the event to the PC for a sufficient amount of time to be monitored. Another possibility would be the trigger dwell time.

Notes: If the event (e.g. the impulse on a DI-bit) is shorter than the fastest sampling interval in the system, it may not be possible for the change to be reliably detected even in the device.

15.8.2 Order of events in response to actions

Device actions, such as "*Prepare*", prompt a variety of events to occur. The list below presents some device actions along with the respective events they prompt, in the order in which they occur.

Legend:

- ○ : For devices associated with the firmware imc DEVICEcore (ARGUS*fit*, EOS), the highlighted events are only initiated if there is something to be done , or if changes have been made to the device configuration.
  For devices associated with the firmware imc DEVICES, the events are initiated even if there is nothing to be done.
- ● : The highlighted events are only initiated if there is something to be done .

Actions and the events they prompt

	Action: Connect	Comment
○	Device_BeforeRequestConnect	
○	Devices_BeforeRequestConnect	
●	Device_Connected	only performed if no connection with the device was already present
●	Devices_Connected	only performed if no connection with the device was already present
○	Device_AfterRequestConnect	
○	Devices_AfterRequestConnect	

	Action: Process configuration	Comment
○	Device_BeforeCheckConfiguration	
○	Devices_BeforeCheckConfiguration	last opportunity at which the configuration can still be changed; also Automation
○	Device_PreCompile1	event used for internal purposes
○	Devices_PreCompile1	
○	Device_PreCompile1	event used for internal purposes
○	Devices_PreCompile2	
○	Device_Compile	event used for internal purposes
○	Devices_Compile	
○	Device_Link	event used for internal purposes
○	Devices_Link	changes to the device configuration made after this event are no longer applied in the device
○	Device_AfterCheckConfiguration	
○	Devices_AfterCheckConfiguration	

	Action: Prepare
(○)	*<as with "**Connect**">*
(○)	*<as with "**Process configuration**">*
○	Device_BeforeDownload
○	Devices_BeforeDownload
○	Device_Configured
○	Devices_Configured

	Action: Start
(○)	*<as with **Prepare**>*
	Device_BeforeRequestStart
	Devices_BeforeRequestStart
	Device_Started
	Devices_Started

	Action: Stop	Comment
	Device_BeforeRequestStop	Only when measurement has been stopped by a menu action. Not when, after a limited measurement duration, the device announces the conclusion of measurement.
	Devices_BeforeRequestStop	Only when measurement has been stopped by a menu action. Not when, after a limited measurement duration, the device announces the conclusion of measurement.
	Device_Stopped	
	Devices_Stopped	

	Action: Disconnect (both from a stopped and from a running device)
	Device_BeforeRequestDisconnect
	Devices_BeforeRequestDisconnect
	Device_Disconnected
	Devices_Disconnected
	Device_AfterRequestDisconnect
	Devices_AfterRequestDisconnect

	Action: Connect (measurement running - Re-connect)
	Device_BeforeReconnect
	Devices_BeforeReconnect
	Device_Connected
	Devices_Connected
	Device_Started
	Devices_Started
	Device_Reconnected
	Devices_Reconnected

	Action: Create (save) Diskstart
	*<as with "**Connect**">*
	*<as with "**Process Configuration**">*
	Device_BeforeCreateDiskStart
	Devices_BeforeCreateDiskStart

Notes

- The **Devices_x** (with "s") event is always initiated once the corresponding **Device_x** event has been initiated for each device in the measurement (this also includes DataProcessing tasks) (see example 1695).
- The ***_After*** events are initiated even when an error has occurred and the subsequent steps were canceled in consequence.
- The ***_Configured** events are initiated even when an error has occurred during "*Prepare*".

15.8.3 Description of the events

Event	Description
Device_AfterCheckConfiguration	After processing a device configuration
Device_AfterRequestConnect	After trying to connect to a device
Device_AfterRequestDisconnect	After trying to disconnect from a device
Device_BeforeCheckConfiguration	Before processing a device configuration
Device_BeforeCreateDiskStart	
Device_BeforeDownload	Before preparing a device
Device_BeforeReconnect	
Device_BeforeRequestConnect	Before trying to connect to a device
Device_BeforeRequestDisconnect	Before trying to disconnect from a device
Device_BeforeRequestStart	Before trying to start the measurement for a device
Device_BeforeRequestStop	Before trying to stop the measurement for a device
Device_Compile	Before processing a device configuration (step 3)
Device_Configured	After preparing a device
Device_Connected	When connected to a device
Device_Disconnected	When disconnected from a device
Device_Link	Before processing a device configuration (step 4)
Device_PreCompile1	Before processing a device configuration (step 1)
Device_PreCompile2	Before processing a device configuration (step 2)
Device_Reconnected	
Device_Started	Once the measurement has started for a device
Device_Stopped	Once the measurement has stopped for a device
Devices_AfterCheckConfiguration	After processing all device configurations
Devices_AfterRequestConnect	After attempting to connect to all devices
Devices_AfterRequestDisconnect	After attempting to disconnect from all devices
Devices_BeforeCheckConfiguration	Before processing all device configurations
Devices_BeforeCreateDiskStart	Once 'Device_BeforeCreateDiskStart' was the last event to have occurred for all devices
Devices_BeforeDownload	Before preparing all devices

Event	Description
Devices_BeforeReconnect	Once 'Device_BeforeReconnect' was the last event to have occurred for all devices
Devices_BeforeRequestConnect	Before attempting to connect to all devices
Devices_BeforeRequestDisconnect	Before attempting to disconnect from all devices
Devices_BeforeRequestStart	Before attempting to start the measurement for all devices
Devices_BeforeRequestStop	Before attempting to stop the measurement for all devices
Devices_Compile	Before processing all device configurations (pre-stage 3)
Devices_Configured	After preparing all devices
Devices_Connected	After connection with all devices
Devices_Disconnected	After disconnection from all devices
Devices_Link	Before processing all device configurations (pre-stage 4)
Devices_PreCompile1	Before processing all device configurations (pre-stage 1)
Devices_PreCompile2	Before processing all device configurations (pre-stage 2)
Devices_Reconnected	Once 'Device_Reconnected' was the last event to have occurred for all devices
Devices_Started	After the measurement has started for all devices
Devices_Stopped	After the measurement has stopped for all devices
Experiment_Loaded	After loading an experiment or "Save as"
Experiment_Saved	After saving an experiment
MeasurementFolder_Closed	Upon conclusion of a measurement (or of an interval), the event is triggered.
MeasurementFolder_Deleted	When a measurement is deleted, the event is triggered.
MeasurementFolder_New	When a new measurement appears in the Data Browser, the event is triggered.
MeasurementFolder_Updated	When the measurement is updated, the event is triggered.
OnAfterStartup	After starting imc STUDIO
OnBeforeQuit	Before attempting to exit imc STUDIO
OnError	After the occurrence of an error
Project_Loaded	After loading a project

15.9 Useful options

In the Options, you can activate additional useful functions: "*Sequencer*".

Reduce logbook entries

When you run the Sequencer, many commands generate entries for the logbook. By means of these entries you can analyze the signals' background situations. For instance, what commands started right before a problem, or in what loop the routine was currently located at the relevant point in time.

Option	Description
Reduce logbook entries 133	Reduces the number of logbook entries produced by the Sequencer. The following messages will be suppressed and not be reported to the logbook window or the logbook file: • Starting/Stopping command • Messages regarding loops (for/while) • Messages regarding if and switch

The advantage of this is that more important entries in the logbook may be noticed more quickly.

15.10 Information and tips

FAQ — The command-sequence is interrupted upon every error

The command-sequence is interrupted upon every error. What is the best way to prevent this?

Example: While the command: "*Wait*" is being executed, another action causes an error. In consequence, the sequence is cancelled.

Solution: For this command, deactivate the associated checkbox "*Stop on error* 1685" in the Sequences table. With this setting, any error messages posted at that position will be ignored during runtime. As a result, the subsequent command will be executed even if an error occurs. Please be aware that this means even the running command's errors will be ignored.

If it is possible, verify later that the action was performed correctly.

Here is an example involving the command "*Execute menu action*" > "*Start measurement*". The sequence is not intended to end even if the command fails. Using the IF-command, you can check the value of the variable "*Measurement status*" and thus verify whether the measurement is actually running. This gives you the ability to respond to the result in a targeted way and close the Sequencer properly, if appropriate.

15.11 Command reference

Commands allow the execution of targeted actions such as jumping to a Panel page, running a imc FAMOS sequence, or printing out a Panel page as a report.

Commands are available in various areas of imc STUDIO, imc WAVE and imc STUDIO Monitor:

- imc STUDIO Sequencer
- imc STUDIO Panel
- imc STUDIO Automation

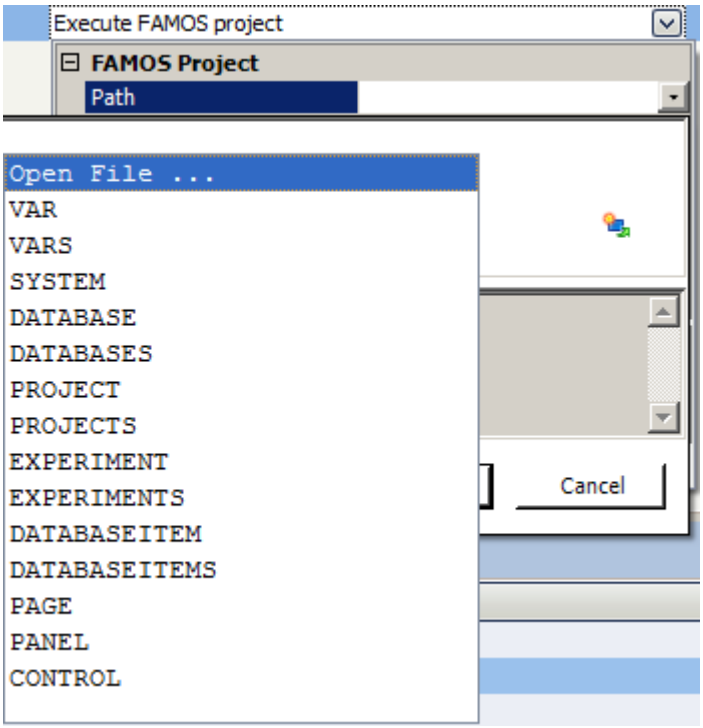
Note

Not all commands described here are available in all areas mentioned, since not all of them are relevant to every area. The commands respectively available are displayed in the associated command list.

15.11.1 Data analysis and Scripting

15.11.1.1 Execute imc FAMOS project

A imc FAMOS project is selected and started:



Command: Execute imc FAMOS project

Click on the symbol and select *Open File...*

In the project folder, select the file *Default.FamosPrj*.

15.11.1.2 Run imc FAMOS sequence

Status	Command
	Execute FAMOS sequence

From within imc STUDIO it is possible to transfer variable currently being measured to imc FAMOS and to process them with a imc FAMOS sequence there. The results of the imc FAMOS sequence can be transferred back to imc STUDIO and displayed in a Panel page, for instance.

Note — Prerequisite

The prerequisite is installation of imc FAMOS on the same (see "*Technical data sheet*" > "*Additional imc Software Products*").

Reference — Functions

Information on imc FAMOS' scope of functions is presented in the separate imc FAMOS manual

To **open the imc FAMOS Dialog**, click on the button (symbol with three dots) in the sequence table.

Reference — imc FAMOS dialog

More information on the imc FAMOS dialog is provided in the section:
"*Data analysis and signal processing*" > "*imc FAMOS dialog* 1088".

15.11.1.3 Run Python code file

The Python command supplies functions which provide a bridge to the Python programming language.
imc STUDIO generates an embedded instance of the Python runtime environment, which provides an interpreter for the Python programming language.

For the purpose of analyzing the measured data, a Python code file is selected. The variables to be analyzed with Python can be selected, and the results can be returned in variables.

System requirements

Support is provided exclusively for the Python reference implementation from the "*Python Software Foundation*" (CPython) in one of the versions listed below; these can be downloaded and installed from https://www.python.org/.

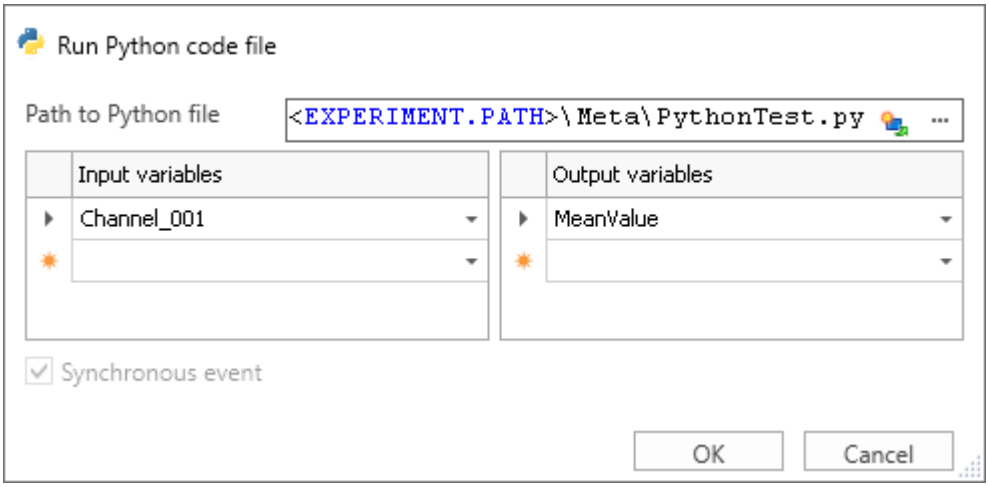
- Python 64 bit is installed
- Compatible Python versions: 3.11, 3.10, 3.9 and 3.8 (64 bit)

Note — Installation

- Python is not included on the installation medium.
- In order to facilitate subsequent installation of packets, it is recommended that you activated the option "*Add python.exe to PATH*".

Configuration

In the command configuration, the Python code file is selected and the transfer of variables to and from Python is configured.



Properties of the command "Run Python code file"

Parameter	Description
Path to Python file	Path to the executable Python code file (*.py). Ideally, the file is saved in the Meta-folder. The content of this folder is also included when performing Save As/Export. *Note*: For the file selection, the current experiment-path is replaced by the experiment-placeholder. In consequence, no modifications are necessary when the experiment is saved under a new name.
Input variables	The variables entered are passed to the Python-sequence. The variable names must match those in the Python-sequence. The spelling is case sensitive. Transfer and processing of the variables is performed on the basis of the "*Current Measurement*" and not on the basis of any saved (earlier) measurements.
Output variables	The variables calculated by Python are assigned to imc STUDIO via the table. The variable names must match those in the Python-sequence. The spelling is case sensitive. If the variable does not yet exist in imc STUDIO, it will be created. Note: A channel can not be transferred back to a "*User-defined Variable*" of the type "*Numeric*" (single value). For this purpose, please modify the type in Python if you wish to obtain a single value. If you require a channel, use as the target a variable created using Python. This will then always be assigned the appropriate data type.

Data type conversion to Python

The data type of the Python variables generated is automatically determined.

Here, the following rules apply:

imc STUDIO data type	Python data type
Single values	
8-Byte Integer (full-scale)	Integer ('int')
Miscellaneous numerical data formats	Real number, 8-Byte ('float')
Normal data sets	
8-Byte Real with length = 0 (constant 'EMPTY')	None ('NoneType')
8-Byte Integer (full-scale)	List ('list' [..'int'..])
1-Byte Integer unsigned (full-scale)	Byte array
Miscellaneous numerical data formats	Liste ('list' [..'float'..])
TimeStampASCII and other special formats	not supported
2-component data sets	
Complex, Cartesian, Length = 1	Complex number ('complex')
Complex, Cartesian, Length <> 1	List of complex numbers ('list' [..'complex'..])
Complex, Magnitude/Phase	not supported
XY	not supported
Structured data (Events/Segments)	
Segments ... Integer 1 Byte unsigned (full-scale)	List of byte arrays ('list' [..'bytearray'..]). Each byte array corresponds to one segment.
Segments ... miscellaneous formats	List of lists ('list' [..'list'..]). Each list element corresponds to one segment.
Events	not supported
Miscellaneous data types	
Text	not supported

Note

Channel properties

What can not be transferred are such channel properties as the sampling rate, time, unit, ...

Event-based data sets

It is not possible to apply calculations to event-based data sets. A channel automatically becomes an event-based data set when the parameter "*Available events*" under "*Data transfer*" is set to "*all*". Set this parameter to "*last*" in order to obtain a normal data set.

Special characters

The rules for special characters in Python-variable names must be observed. Some characters which are allowed for imc STUDIO-variables are not allowed. Such variables which have a period in their name, for instance, pv-variables, can not be transferred.

Data type conversion to imc STUDIO

The data type of the imc STUDIO-variable generated is determined automatically from the data type of the Python-variable.

Here, the following rules apply:

Python data type	imc STUDIO data type
Standard types	
'NoneType'	Data set of length 0; data format: 8-Byte Real (like constant 'EMPTY')
'int'	Single value; data format: 8-Byte signed Integer
'float'	Single value; data format: 8-Byte Real
'complex'	Complex data set (RI) of length 1; data format: 8-Byte Real
'bool'	Single value; data format: 8-Byte Real
'str'	not supported
Container types 'tuple', 'list', 'range', 'set', 'frozenset'	
with element number 0	Data set of length 0; data format: 8-Byte Real
all elements of type 'float'	Data set; data format: 8-Byte Real
all elements of type 'int'	Data set; data format: 8-Byte signed Integer
all elements of the type 'boole'	Data set; data format: 8-Byte Real
all elements of the type 'complex'	Complex data set (RI) of length 1; data format: 8-Byte Real
all elements of the type 'str'	not supported
all elements of a 1-dimensional container, where all containers have the same length N and all elements the same numerical type (float, int, complex)	Segmented data set; segment length: N
other element-types or various types	not supported
Miscellaneous container-types	
'dictionary'	not supported
'bytes', 'bytearray'	Data set; 1-Byte unsigned Integer

Notes User-defined variable as the target

These variables are automatically converted to an appropriate data set. If the target variable had previously been defined in imc STUDIO as a user-defined numerical variable, then a single variable can be assigned to this variable. The variable remains a single value.

Behavior

- If an input variable does not exist, an "empty channel" is transferred to Python. This will not appear in the Data Browser, but is only available in the Python-environment.
- When a variable which Python requires is not transferred, an error is thrown upon execution.
- If any output variable is defined which Python does not use, a warning appears in the logbook.

Additional notes

- The asynchronous mode is not yet available at this time, for which reason the option "*Synchronous event*" is disabled (appears in gray).

Examples

Calculating the sum of all samples

```
import math
pySummeSine = math.fsum(Sine)
```

15.11.1.4 Run script

Using the command **Run Script**, available scripts of type *Script* can be executed. When running the script, the script's method Run() is called. The programming language is C# (.NET 4.8).

Script selection

Parameter	Description
Name	Here, the script to be run is selected.
Add new Script	Using this button, a new script can be created. The Create Script dialog will subsequently open.

Reference Scripting

For further information see Scripting-manual:

- Create script 1812
- Run script 1816

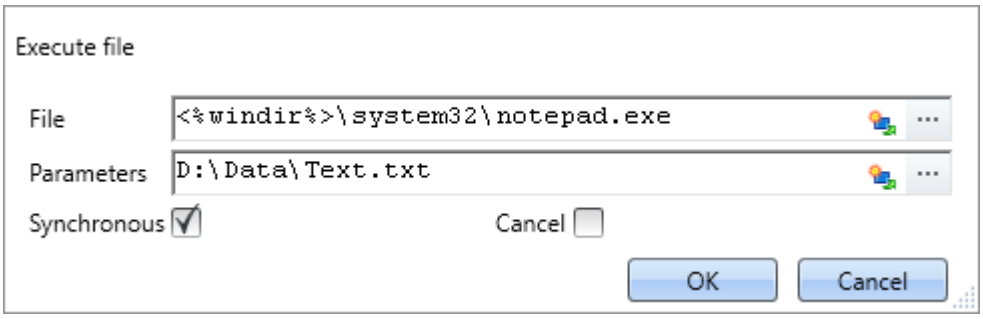
Note When is the command "Done"

The command is **Done** when the `Run()`-method has concluded.

15.11.2 External calls

15.11.2.1 Execute file

The command starts a Windows program.



Execute: In the example, Notepad is started

Parameter	Description
File	Display text in the sequence table (without effect on the dialog).
Parameters	The dialog's header.
Synchronous	When activated, the Sequencer remains on this command until either the program or the file is closed.
Cancel	Activates the Timeout function. At the latest, after a specified timeout time, the Sequencer runs the next line.
Timeout	Duration in seconds [s] until the timeout occurs.

15.11.2.2 Format converter

The imc Format Converter provides the ability to convert measured data to a different format, e.g. to EXCEL data format. For instance, the imc Format Converter can operate as a command performed automatically at the end of a measurement for the purpose of converting the data to the desired format.

Reference

You can also use the imc Format Converter as a **standalone** program, if imc STUDIO or imc FAMOS has been installed; see the chapter "*imc Format Converter* 1191"

Export formats

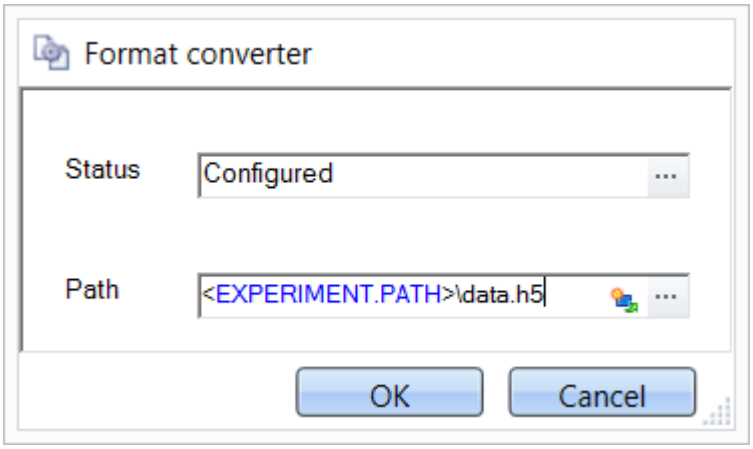
The same export formats are supported which are also offered by imc FAMOS. These include:

- user-defined ASCII export-formats
- user-defined EXCEL export-formats
- MDF2.0, MDF3.0, MDF4, ASAM ATFX, ASAM ATFX NVH, Catman 5.0
- Google Earth Export, HEAD acoustics (also 4.5 compatible)
- Matlab 4 and 5, nSoft-DAC, RPC-3, Somat SIF (nCode output)
- TEAC TAFFmat and TEDAM, DIAdem TDM and TDMS
- Binary Universal File Format
- Universal File Format (UFF)

15.11.2.2.1 Format converter as command

Calling

Via the Sequencer: "*External calls*" > "*Format converter*":



Configuration of the command: Format Converter

Status

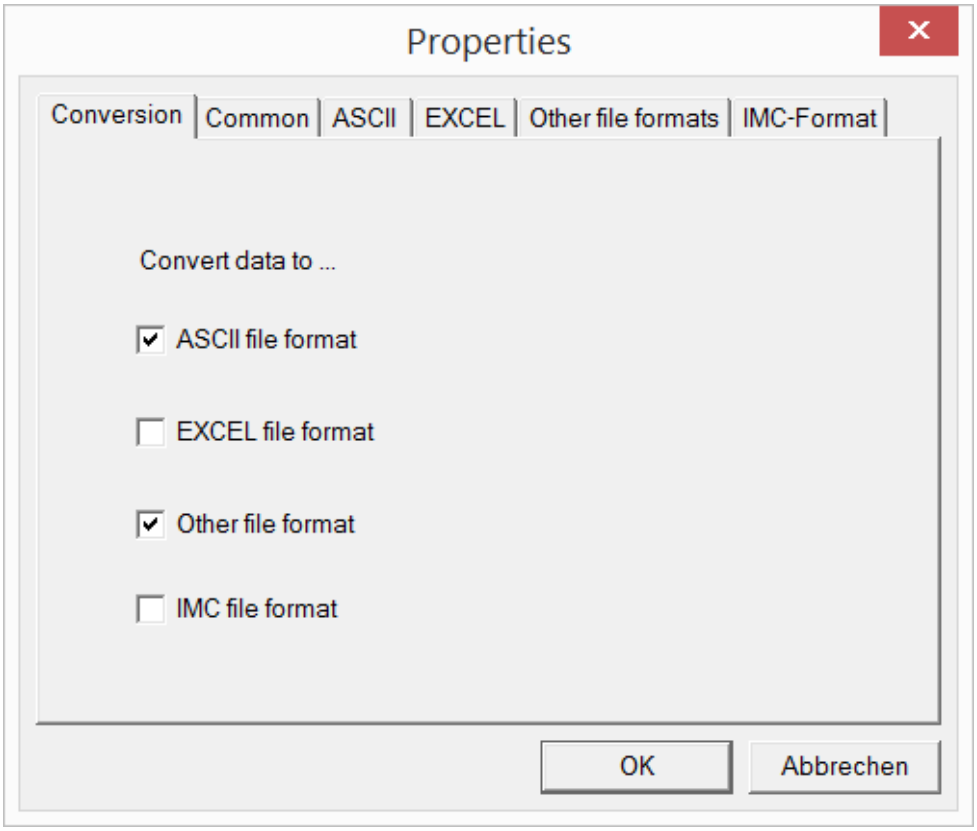
Immediately upon opening the command's configuration, the status is set to "*not configured*".

Path

- If just **one** file (*.dat, *.raw) of measured data is specified here, you can select whether to **convert either only the single file, or all files of measured data** which the **folder** contains.
- If a **folder** is specified here, then **all files of measured data in the folder** are converted:
  - If the option "*Save all files of a folder in one file*" among the settings on the tab page "*Common*" (see the segment on "*Status*") is selected, then **all files of measured data** present in the specified folder are converted to **a single target file**.
  - If the option is **not selected**, then each file of measured data which the folder contains, the data are converted to **a separate file**.

Using the button ··· in the box "*Status*" you can open a dialog via which you can make all the settings for the conversion:

- Conversion: Target format. It is also possible to select multiple formats.
- Common: Storage location and whether individual files or a multi-file is to be created.
- ASCII: Selection of the export template and specifying of the file extension.
- EXCEL: Selection of the export template and specifying of the file extension xls or xlsx.
- Other file format: Selection of the format and if necessary its formatting options.
- IMC-file format: Save the data to a file in the imc format, or as a copy to a different memory location.



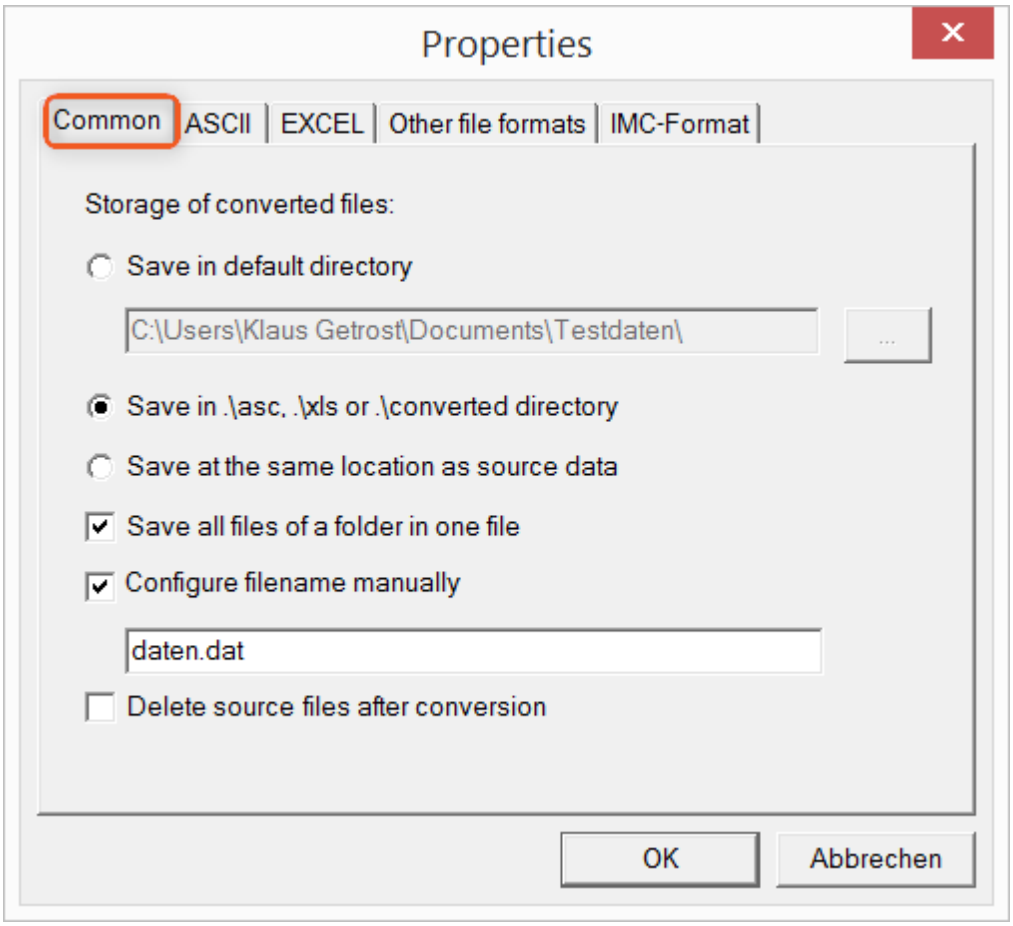
Setting the Format Converter

15.11.2.2.2 Settings

In the settings you specify whether the data are saved either individually or together, which templates are used for ASCII and EXCEL, or what other formats are used.

The settings remain intact, so that under normal circumstances you can immediately select data and convert them.

Data saving



Storage location for converted files:

Save in default directory: Free choice of a **target folder**. Even network drives are possible.

Save in .\asc, .\xls oder .\converted directory: The data are saved in their source folder but additionally in a **subfolder**. For ASCII in the *asc* folder, *xls* for EXCEL and *converted* for any other export formats.

Save at same location as source data: The data re saved on the same **level of the tree diagram as the source data, without subfolders**.

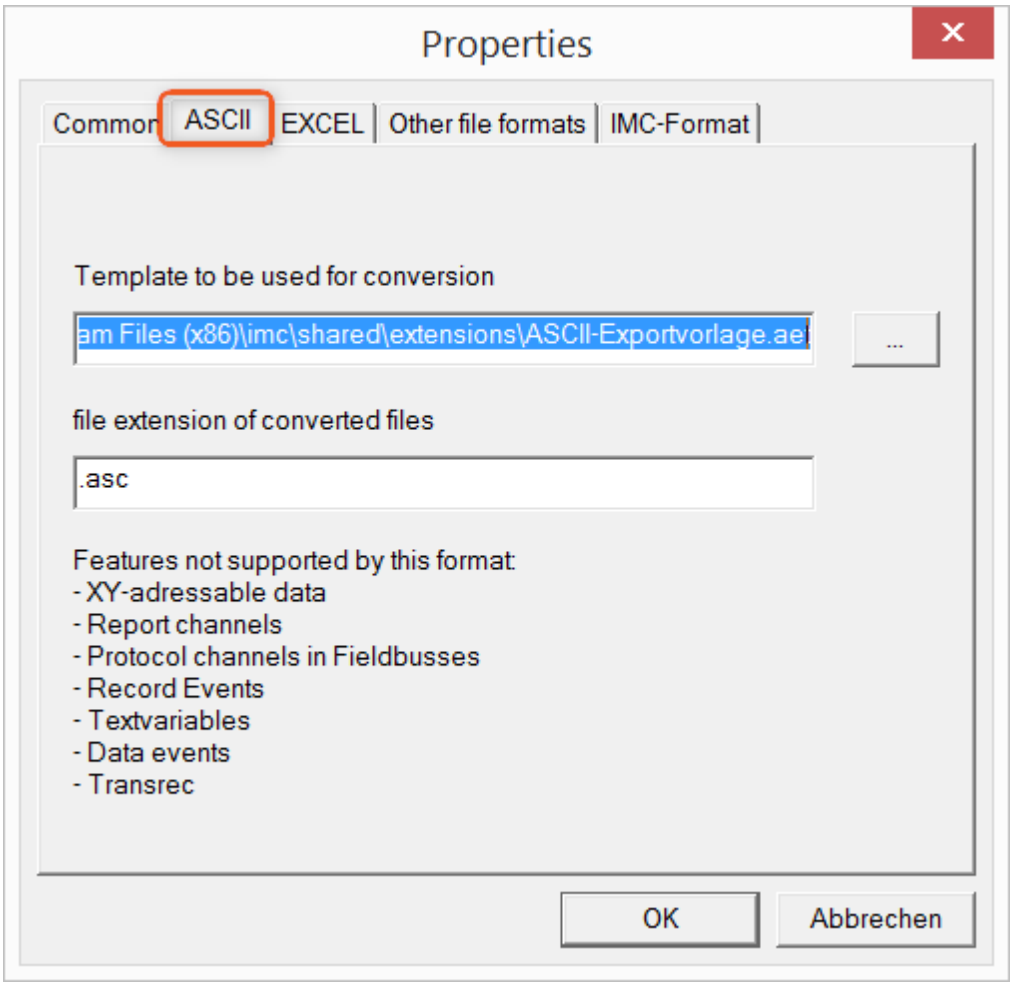
Manner of data saving:

Save all files of a folder in one file: All of a channel's files are saved **together in one file**.

Configure file name manually: **Name of the target file** in which all channel files are saved together.

Delete source files after conversion: Once the data have been converted, the **source files can be deleted automatically**, in order to make memory space free. This setting can make sense when all data are saved together in the imc format.

ASCII



Format template to be used for conversion:

By default, ASCII export templates are located in the folder *"C:\Program Files (x86)\imc\Shared\Extensions"* and use the file extension ***.aet***. Owners of imc FAMOS are also able to personally create or modify these templates. In this case, the aet-files are saved in the folder *"C:\ProgramData\imc\Common\Def"*.

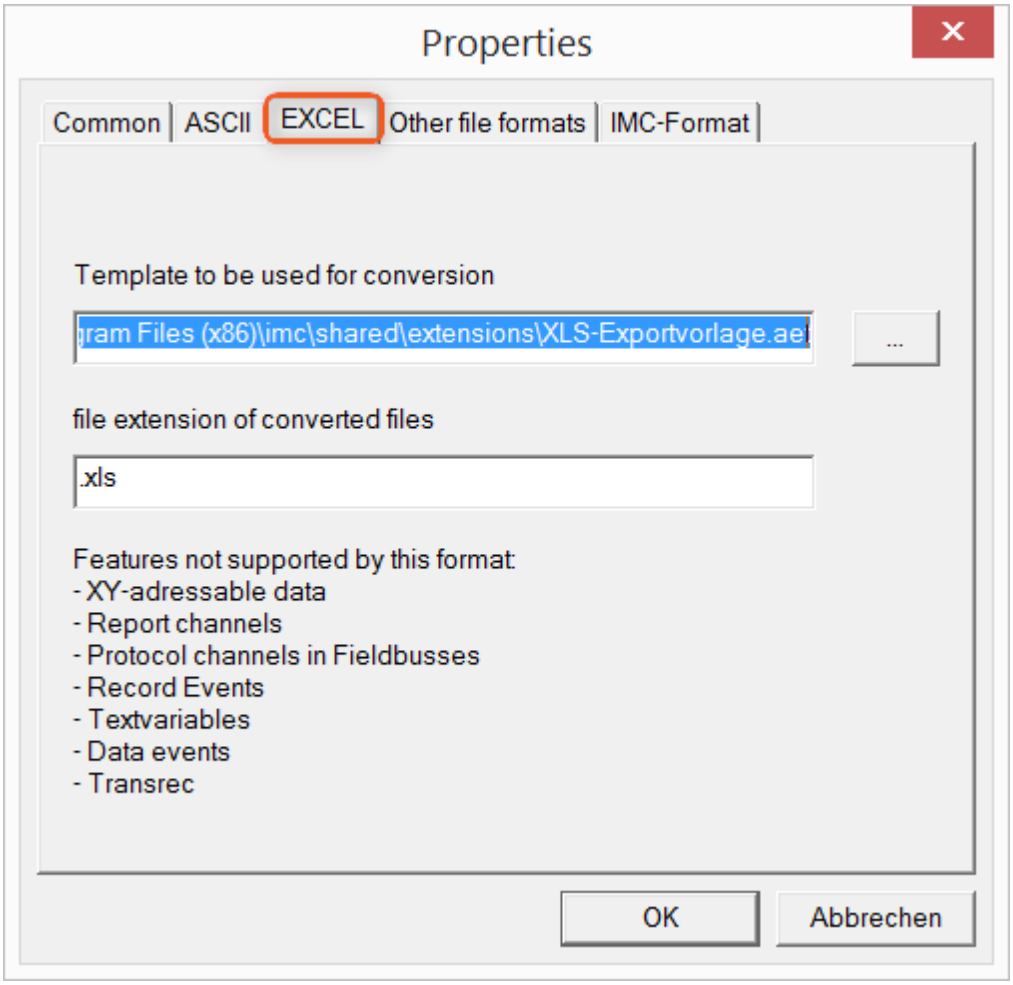
By default, the file extension is *"asc"*, but any arbitrary one can be specified here.

Special variable types which can **not** be exported as ASCII data are listed on this tab page.

Notes File extensions

In imc FAMOS the file extension is specified with the ASCII template. Within imc FAMOS this file extension is added to the ASCII file. With imc Format Converter the file extension specified here is used instead.

EXCEL



Format template to be used for conversion:

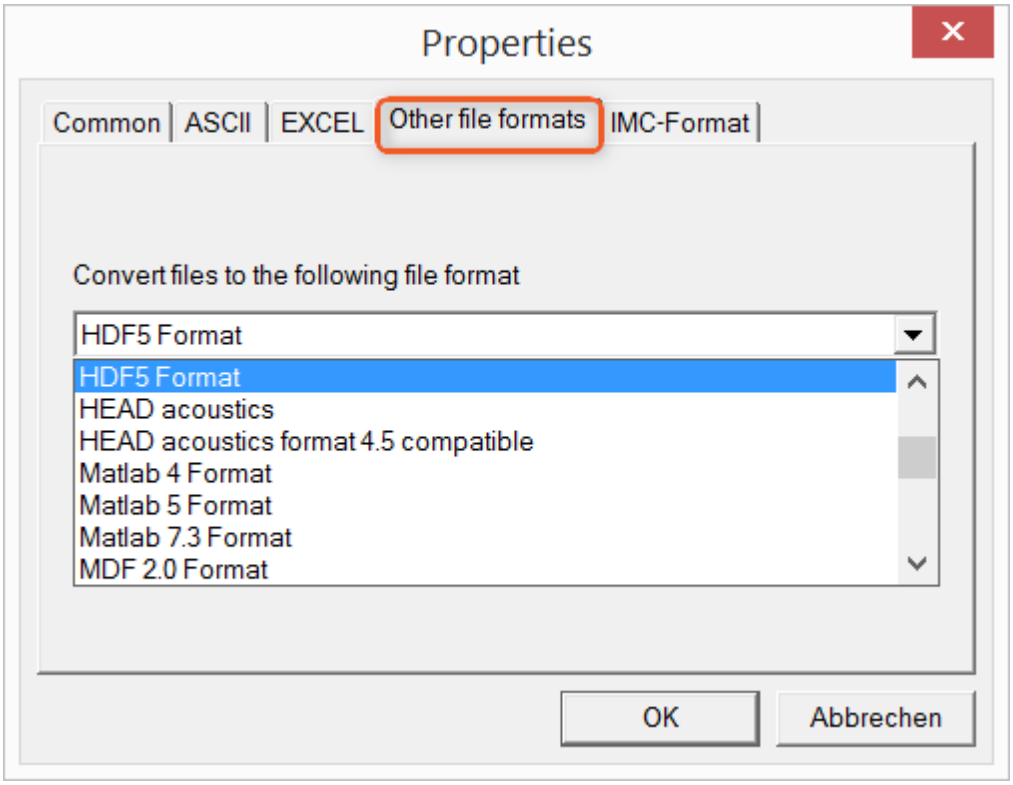
For EXCEL files, a similar technique is used as for ASCII 1717. By default, the file extension is *"XLS"*, but can be arbitrarily specified here.

Variable types which are **not** exported correspond to those for ASCII-export 1717.

Notes File extensions

If the XLS**x** format was selected as the export template, it must be here as the file extension. The imc Format Converter does not import the specified file extension from the AET file.

Other file formats



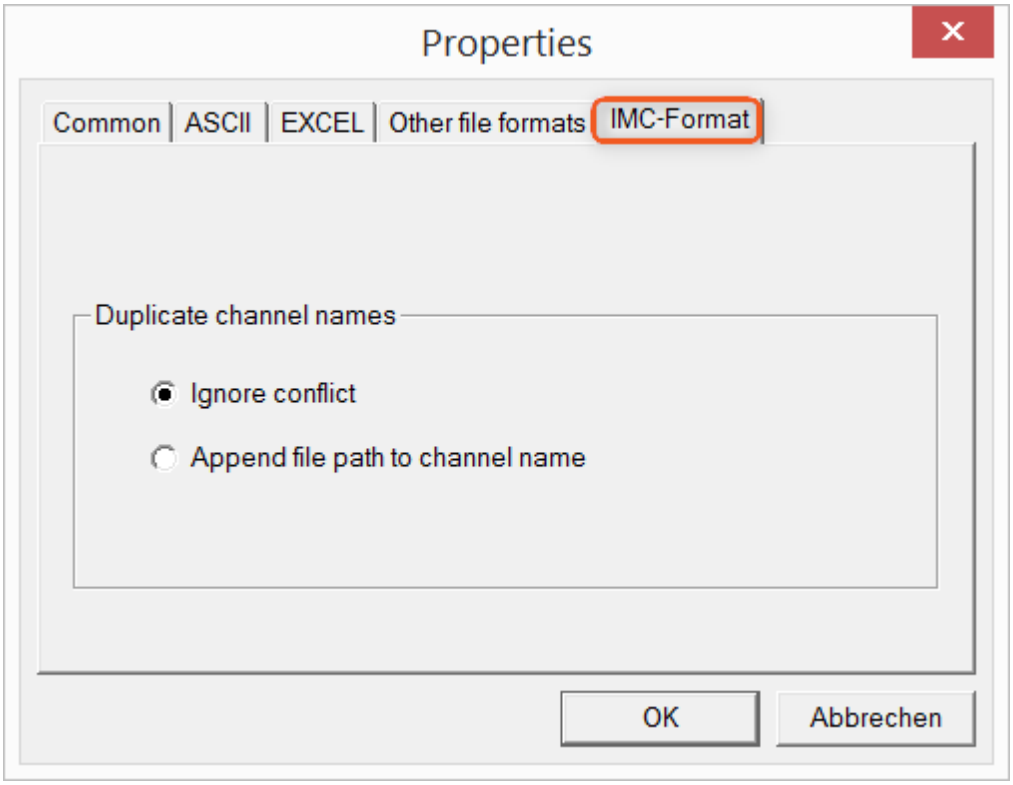
Convert files to the following file format:

Selection of the format in which the data are to be saved. Installation of the imc Format Converter makes certain export formats available.

Options

If the format selected comes with additional export options, these can be set by means of the dialog opened by clicking on the button "*Options*". Descriptions of the options appear in the document ***ImportExportFilter.pdf***, of which a copy is imported along with the imc FAMOS installation. Alternatively, the document is available from the imc homepage.

imc-Format



Handling duplicate channel names

Ignore conflict: If files are overwritten due to duplicate channel names, the conflict is not reported.

Append file path to channel name: A unique channel name is generated from the file path.

Notes Whay save in the imc-format?

The imc Format Converter is designed for imc data, so why is there an imc-Format option? imc devices allow complex data structures in which channels can be recorded at different moments in time (using triggers). For this reason, the channels are saved as individual files. Using the imc Format Converter, you are able to join the data sets accumulated into one single file after the measurement. One appropriate group of settings on the page "*Common* 1716" to accomplish this would be:

Save at same location as source data = active

Save all files of a folder in one file = active

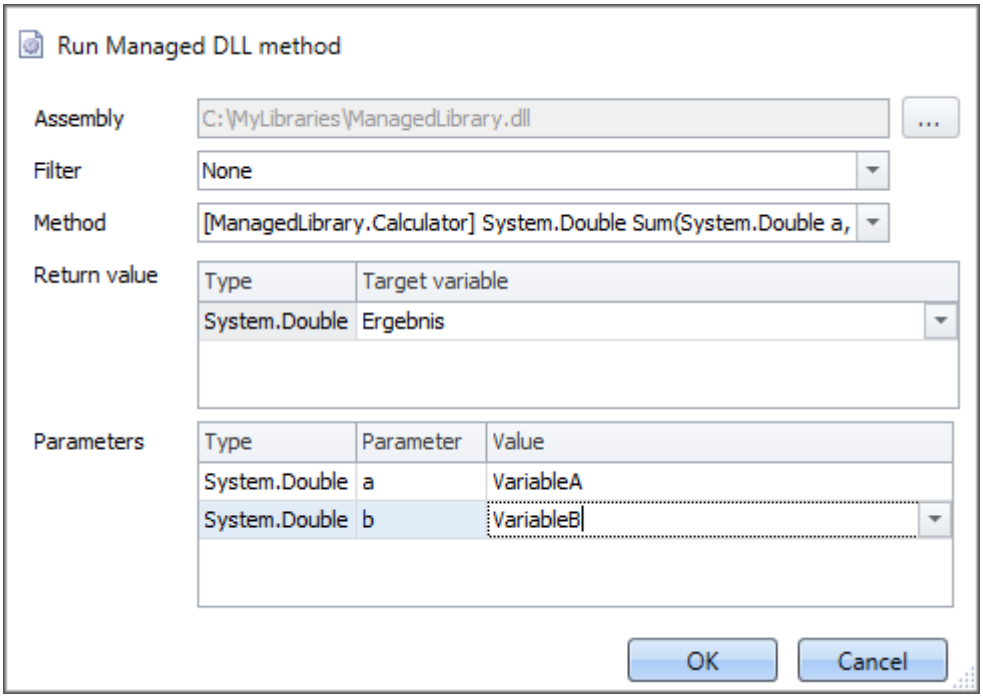
Delete source files after conversion = active

15.11.2.3 Run Managed DLL method

This command enables you to to use methods from external **managed** DLLs in imc STUDIO.

Note Notes on using this command

- Whether external DLLS and their methods **work correctly** or on the other hand even cause damage **can not be ascertained by** imc STUDIO, and thus also **not be prevented**!
- For the purpose of obtaining a good understanding of error messages and the necessary settings, skill in distinguishing between **unmanaged** and **.NET DLLs**, as well as basic programming experience, are **recommended**.
- Only primitive data types can be used.
- If you wish to use only one **WinAPI-DLL**, be **absolutely certain** that the **parameter signature** is correct. Otherwise, the imc STUDIO functionality could be affected.



Example of a configured command

Parameter	Description
Assembly	Clicking on the button ... causes a .NET Framework Assembly to be selected which usually is of the type .dll.
Filter	You can set up a filter which hides any unnecessary method names from a list of available ones. The filters listed here are permanently programmed and cannot be changed.
Method	Here, methods offered in the assembly are listed, naturally only the ones designated as public.

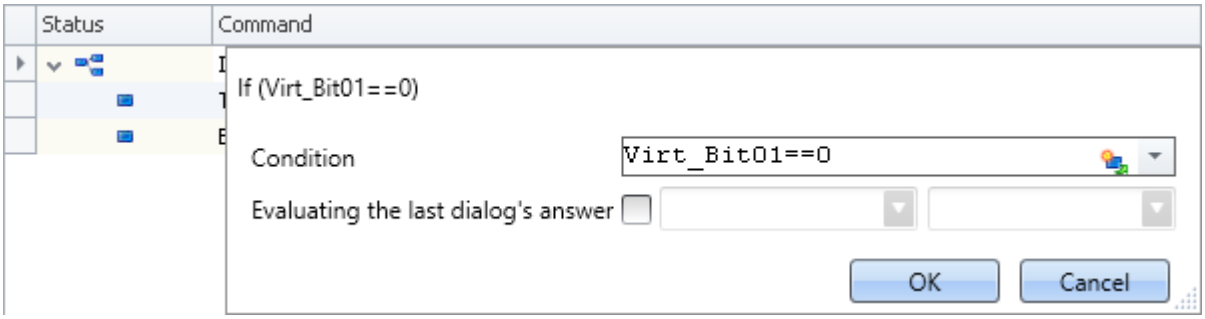
Parameter - Return value	Description
Type	The data type's **complete .NET name**, which is returned by the method. This name must be outputted in its entirety in order to distinguish between any multiple declarations in different locations (in the example shown: "System").
Return value	The name of the imc STUDIO variable in which the value is to be saved; this box can also remain empty, if the return value is not needed.

Parameter - Parameters	Description
Type	The data type's **complete .NET name**, which is expected by the method. This name must also be specified completely in order to distinguish between any multiple declarations.
Parameter	The parameter's name in the method declaration.
Value	The value of the parameter to be transferred; this box comes with the same functions as Set Variables. If a variable from imc STUDIO is used here, it must necessarily be **convertible** to the **target data types**. If this rule is violated, then an error message appears in the log book at the moment the command is run.

15.11.3 Flow control

15.11.3.1 If

"*If*" enables the routine to branch off in response to "*If*" and "*Else*" instructions.

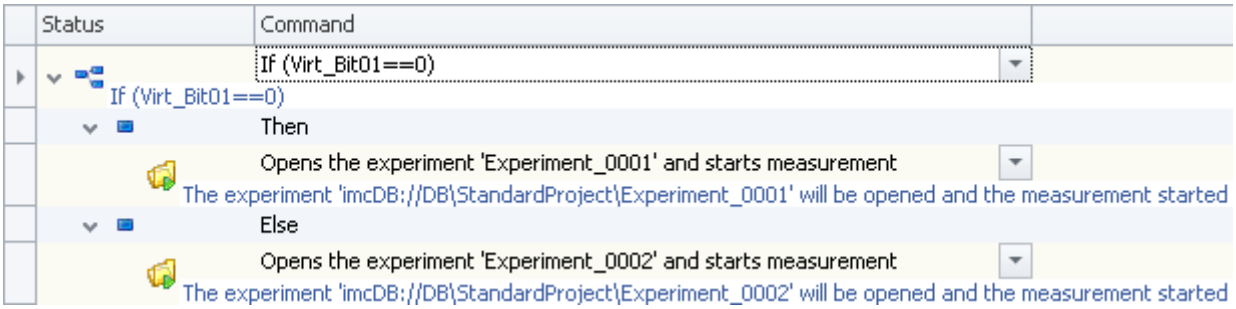


Note that a comparison is performed using the operator "==".

Note — Evaluating the last dialog's answer

Using this command it is possible to react to dialog responses. For more on this topic, see: "*Evaluating the last dialog's answer* 1727".

Example — Loading different experiments according to a variable's value



Example: Depending on the variable Virt_Bit01 an experiment is loaded and started.

In the example, an experiment is loaded and started, depending on the variable "*Virt_Bit01*". If the variable's value is "0", then "*Experiment_0001*" is loaded and started. For any other value, "*Experiment_0002*" is loaded and started.

Note — Note on loading an experiment from the Sequencer

Note that the command "*Open Experiment/Start measurement* 1731" does not overwrite the Sequencer. The Sequencer is ignored when the command "*Open Experiment/Start measurement* 1731" is selected.

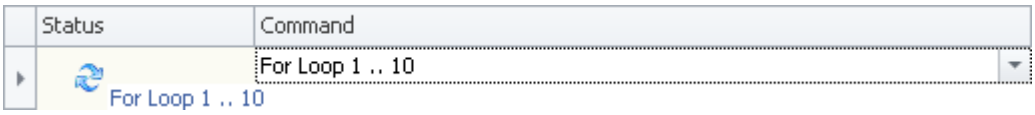
15.11.3.2 Loop

Repeatedly run sections of code are placed inside of loops. There is a distinction between the For Loop and the While Loop.

- A *For Loop* is repeated until a specific number of loops is reached.
- A *While Loop* is run until a specifically defined condition is met.

As the counter variable, a pv-variable or Display variable can be used, which can be displayed, for instance, in the Panel.

From the *commands* belonging to the group *Flow control*, drag the command *Loop* to the sequence table.

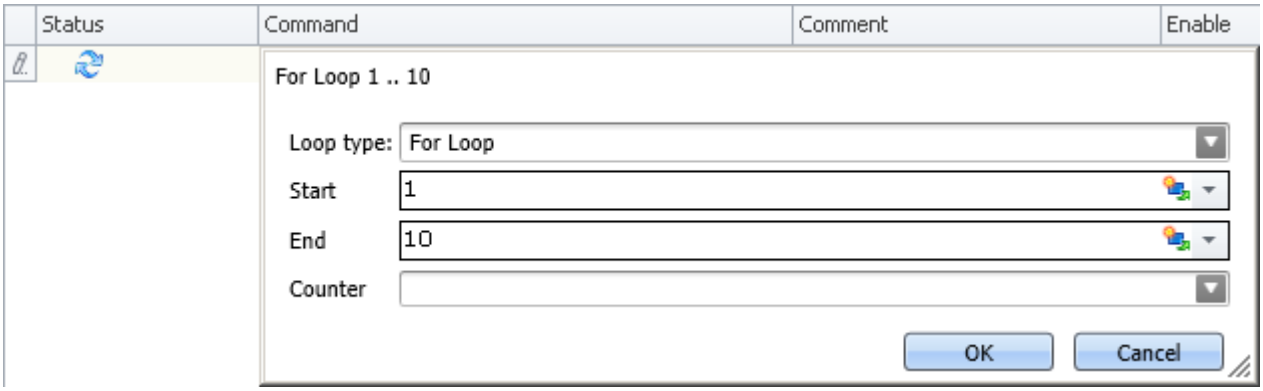


Flow control: Loop

Right-click next to the *For Loop* entry to open its Properties dialog:

- Select the *Loop type: For Loop* or *While Loop*.

For Loop

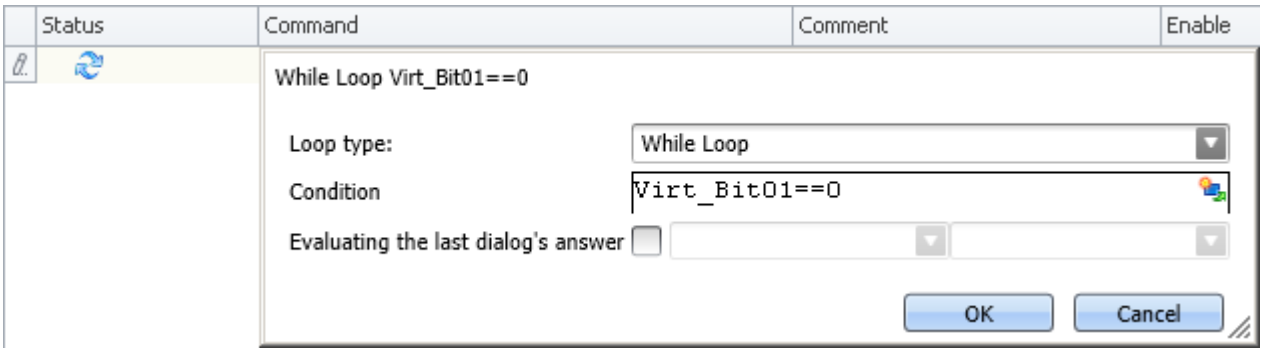


Flow control: For Loop

A *For Loop* has a *Start*- and an *End*- number. These can be constant numbers or variables (PVV or Display variables).

The current number of runs is specified under *Counter* as a Variable.

While Loop



Flow control: While Loop

A *While Loop* is repeated until the *Condition* is no longer fulfilled.

In the example, by means of the symbol under *Variable* the virtual Bit01 was used. As long as this is not set, the Sequencer stays in the loop.

Note that a comparison is performed using the operator "==".

Reference

An example of a For Loop 1783 and While Loop 1785 is found in the examples for the plug-in Sequencer.

Note Evaluating the last dialog's answer

Using this command it is possible to react to dialog responses. For more on this topic, see: "*Evaluating the last dialog's answer* 1727".

15.11.3.3 Stop Sequencer

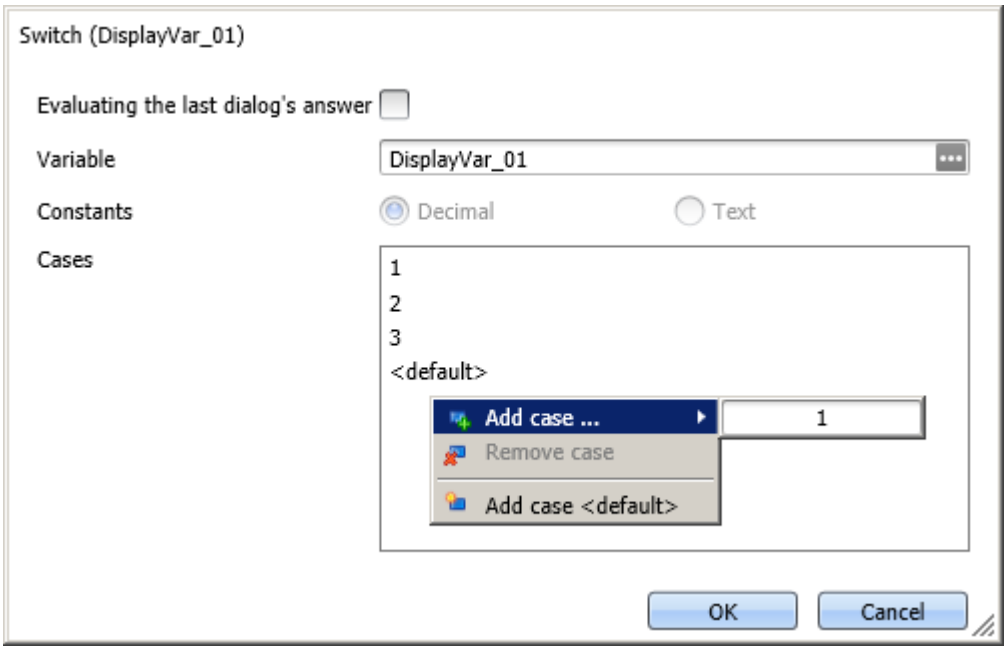
When this command is run, the Sequencer is stopped. The command only takes effect on the Sequencer running. Other command-lists, such as ones which are started by means of a Panel-button, are not stopped in that way.

15.11.3.4 Switch - Case

The command Switch enables multiple branching (Case) depending on a variable's value.

The *Variable* must be a single value (Display- or pv-variable) or a text variable.

The *Cases* are defined as concrete numbers (*decimal*) or texts (*Text*).



Flow control: Switch

Expressions such as "2<x<5" are not allowed here.

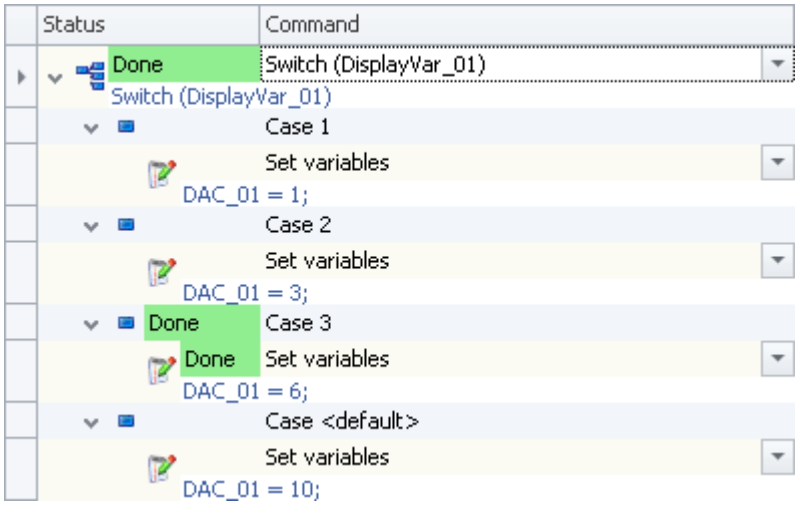
Additional *cases* are added using the right mouse button.

Note Evaluating the last dialog's answer

Using this command it is possible to react to dialog responses. For more on this topic, see: "*Evaluating the last dialog's answer* 1727".

Example

In the example, the Display variable DisplayVar_01 is linked to a pop-down list in the Panel.



Example of a Switch - Case structure

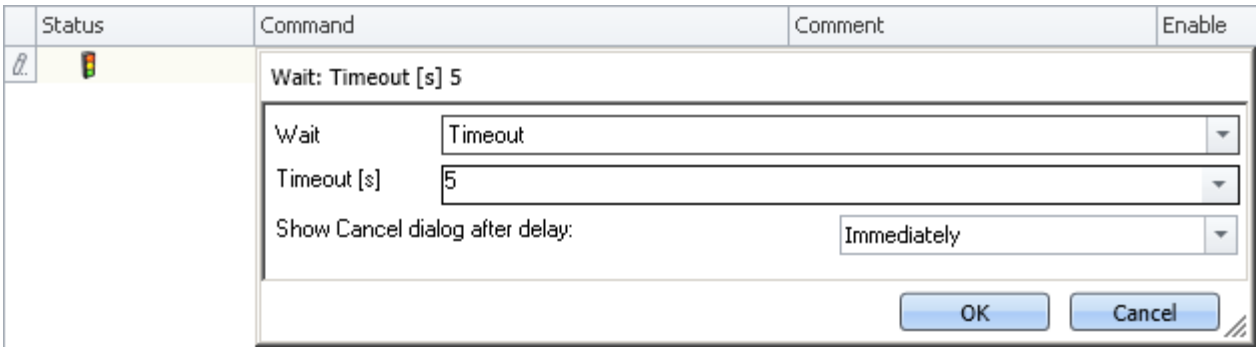
The selected line determines the value of the Display variable. The analog channel DAC_01 is set in response to the value.

Reference More than one command sequence

With the help of the switch command you can simulate several command sequences. See the example: "*More than one command sequence* 1780"

15.11.3.5 Wait

Using the command *Wait* you can halt the Sequencer. The waiting time can be set in the mode *Timeout* or be made to depend on a *Condition*.



Flow control: Wait

Parameter	Description
Wait	Sets the mode to either **Timeout** or **Condition**.
Timeout [s]	In the *Timeout* mode, enter the waiting time here, in seconds. This can be achieved by use of a variable, or it can be a fixed value.
Show Cancel dialog after delay	If you wish to cancel the timeout early, you can have a "Cancel dialog" after a specified time.
Condition	In the mode *Condition*, set the condition here, e.g. `Virt_Bit01==1`

Note that a comparison is performed using the operator "`==`".

15.11.3.6 Evaluating the last dialog's answer

By means of this property, some commands can react to dialog responses. This property is available for the following commands:

- If 1722
- Loop as While Loop 1723
- Switch 1725

For the evaluation, always the response to the last dialog is used. This includes *Show message box* 1754 and *Show Panel page as dialog* 1732.

The last dialog-response is reset when the Sequencer is re-started. In consequence, at every start of Sequencer, the same state is available: "No response provided yet".

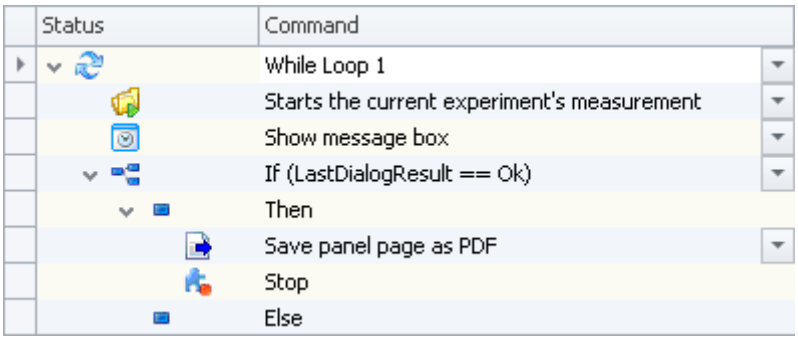
Is no response available yet?

If no dialog has received a response yet by the time a response must be evaluated, the response: "*OK*" is assumed. For example, a question is posed at the end of a While-loop, which decides whether the While-loop is repeated or concluded. In this case, no response is available upon entering the While-loop, so "*OK*" is assumed.

Example — Example for the If-command: Loading different experiments depending on variable's value

In a message box (command: *Show message box*), the user is asked whether the measurement was processed properly and the procedure can thus be concluded.

- For `Yes`, a report-page is to be exported and the Sequencer is stopped.
- For `No`, the measurement is to be repeated.



Example: Depending on a dialog result, the Sequencer is either stopped or run one more time

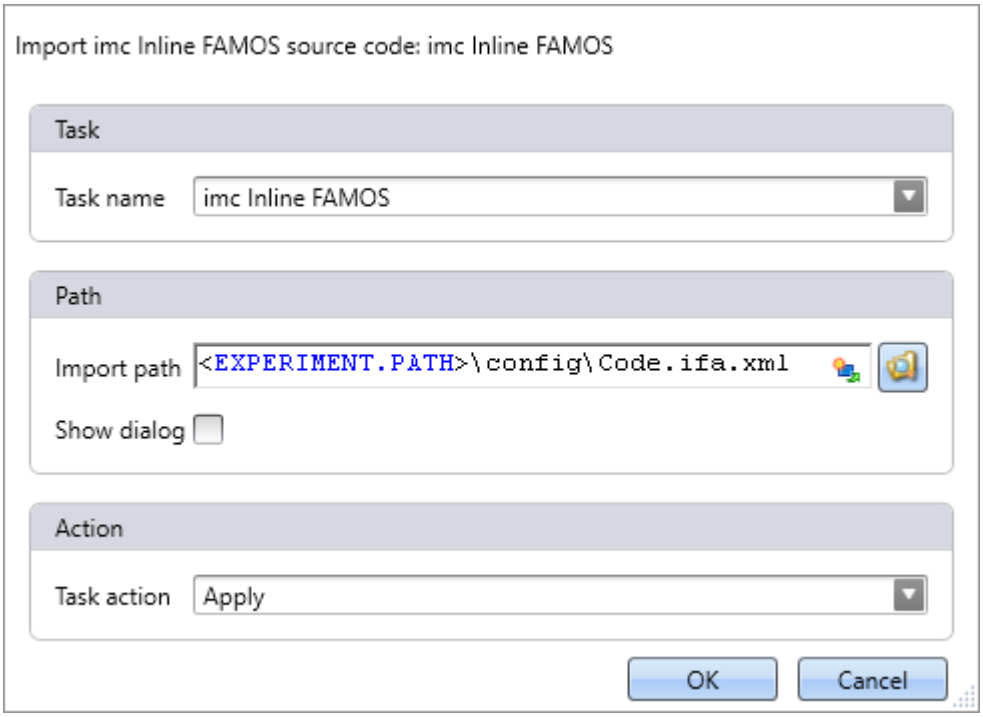
Note — Notes regarding the evaluation

- **Only such dialogs are evaluated which are called in the course of executing the Sequencer.** Any dialogs called by means of a Panel-button or by a another event are not evaluated.
- Please note that there is no assignment for what dialog to evaluate. Always the last dialog called by Sequencer is evaluated. For this reason, either evaluate dialogs immediately or ensure that between the dialog and its evaluation, no other dialog can be called by means of the Sequencer.

15.11.4 imc Inline FAMOS

15.11.4.1 Import imc Inline FAMOS source code

Using this command, you can import imc Inline FAMOS source texts. You can choose whether a text is to be immediately applied and run after the import, or initially only imported.



Configuration: Import imc Inline FAMOS source code

Parameter	Description
Task name	Target name of the Task. If a Task of that name exists already, it is overwritten without confirmation prompt. If no Task of that name is present, a new Task of that name is created.
Import path	Specification of the file to import, including the pathname.
Show dialog	If this option is activated, the file selection dialog appears when the command is executed. Select the file desired. If a path has been defined, the selection dialog starts there. The selection does not take effect on the configuration of the command in the command list, but only on the import currently performed.
Task action	The imported Task can either be applied and run immediately, or initially only imported. • **Apply**: The Task is applied and the calculation started immediately (see also the information on the "*Apply* 1100" Ribbon in the Data Processing) • **Edit**: The source text is imported, but the Task is not applied immediately. Thus, this does not cause the running calculation to be concluded. If the measurement is re-started, the Task is applied automatically.

15.11.5 Measurement

15.11.5.1 Set measurement number

The command *Set measurement number* assigns a measurement number to a measurement.

The available settings are:

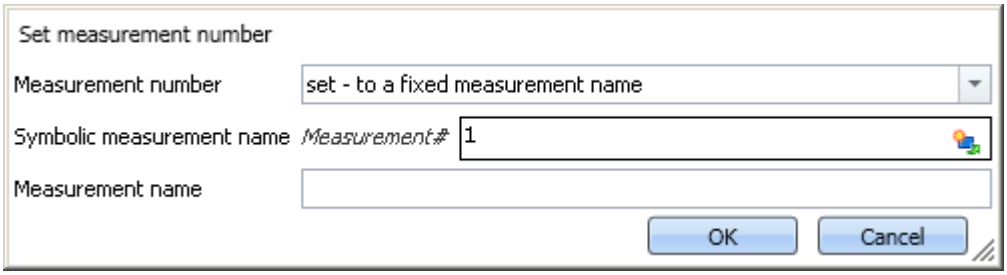
- Set measurement number **to a fixed measurement name**
- Set measurement number **to a fixed index**
- Set measurement number **to the last completed measurement**
- Set measurement number **to a fixed offset before the last completed measurement**
- Remove measurement number

The numbers **1 through 99** can be used for assigning measurement numbers.

Reference The last measurement

More detailed information on the "last measurement" is presented in the chapter:
"*Storage options and directory structure*" > "*The last measurement* 441"

set - to a fixed measurement name

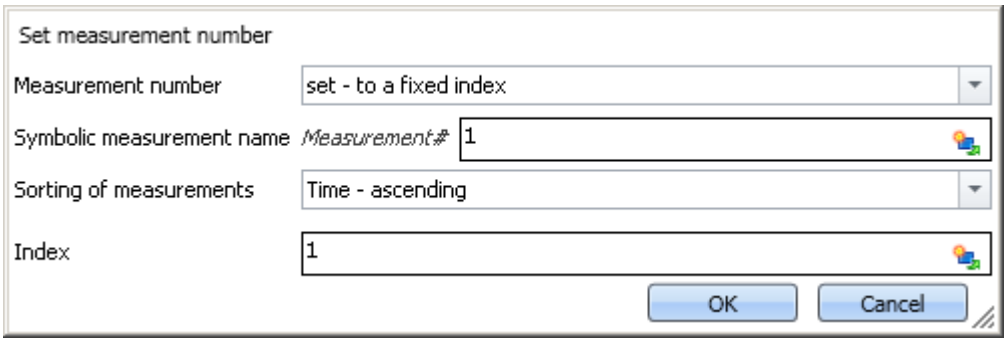


Command "Set measurement number to a fixed measurement name"

With this command, you can assign a number to a measurement which has a **particular measurement name** at the time the command is run.

In the box *Symbolic measurement name*, you specify the measurement number to be assigned. In the box *Measurement name* you can assign a name to the measurement receiving that number.

set - to a fixed index

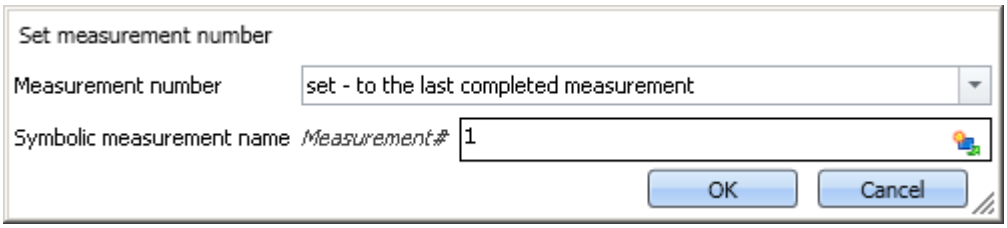


Command "Set measurement number to a fixed index"

With this command, you can assign a number to a measurement which has a **particular index** at the time the command is run.

In the box *Symbolic measurement name*, you specify the measurement number to be assigned. In the box *Sorting of measurements*, you can specify what kind of sorting to apply to the value entered in the box *Index*.

set - to the last completed measurement

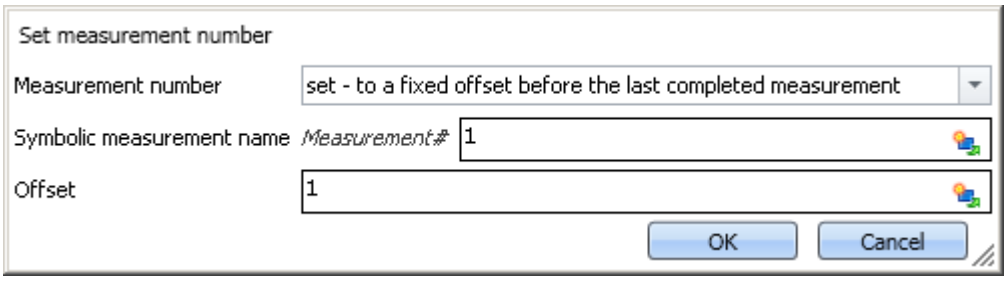


Command "Set measurement number to the last completed measurement"

With this command, you can assign a number to a measurement which is the **last completed measurement** at the time the command is run.

In the box *Symbolic measurement name*, you specify the measurement number to be assigned.

set - to fixed offset before the last completed measurement

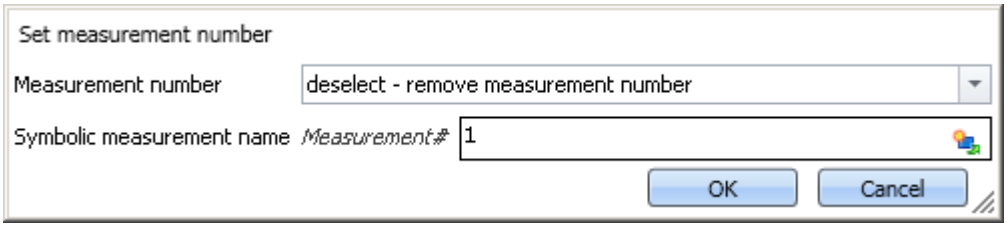


Command "Set measurement number to a fixed offset before the last measurement"

With this command, you can assign a number to a measurement which has a certain **offset to the last completed measurement** at the time the command is run.

In the box *Symbolic measurement name*, you specify the measurement number to be assigned. In the box *Offset*, you can specify how many completed measurements going back the measurement to be numbered is. For instance, an offset of 1 means that the next-to-last completed measurement is assigned the number.

deselect - remove measurement number



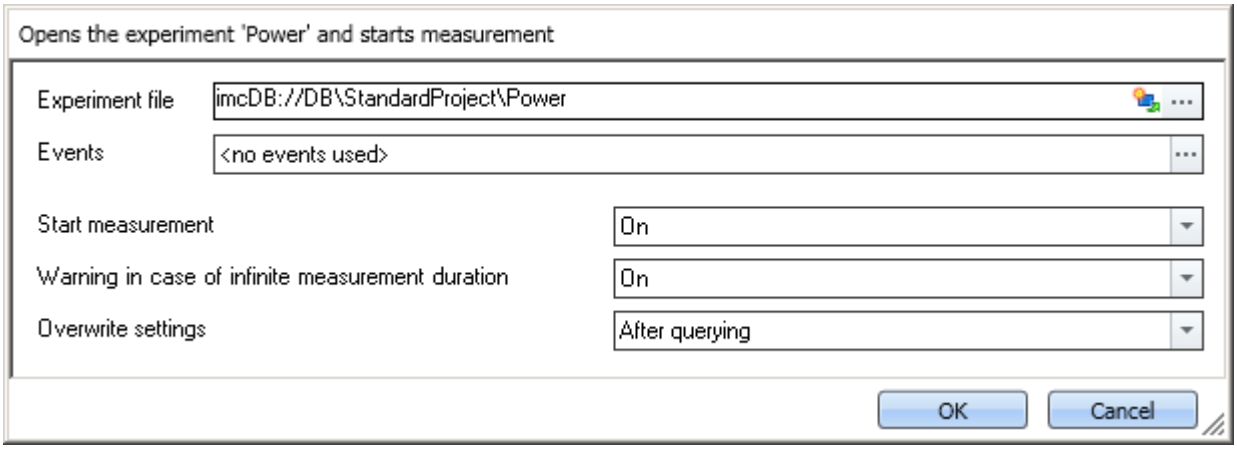
Command "Set measurement number - remove measurement number from a measurement"

With this command, you can delete an assigned measurement number from any measurement.

In the box *Symbolic measurement name*, you specify which number to delete. It does not matter to which measurement this number is assigned.

15.11.6 Open experiment / Start measurement

The command **Open Experiment/Start measurement** enables starting of the measurement from the Sequencer.



Properties of the command Open Experiment/Start measurement

Parameter	Description
Experiment file	Leave this entry empty if you wish to start the measurement with the current device settings. To load a different experiment's **experiment settings** to the **current experiment**, enter the experiment here. What is important to know is that the experiment is not loaded (like when using the ribbon *Home > Open*). Further, the current experiment remains loaded. All experiment settings are **replaced** with the settings of the experiment to be loaded. • **All current experiment settings are deleted**, such as the device configuration (Setup), Panel, user-defined variables with the validity range: Experiment, in accordance with saved scripts. • **All experiment settings are loaded from the selected experiment.** • All Sequencer-settings of the experiment to be loaded are ignored.
Events	Enables insertion of a command within the starting procedure.
Start measurement	For *On*, the measurement starts upon being opened. For Off, the experiment is only loaded. You can also enter a user-defined variable here.
Warning in case of infinite measurement duration	If a channel was configured so that its duration would be infinite, the Sequencer stays on this line until the measurement is stopped. With this option, you can notify the user that the measurement will have to be stopped manually.
Overwrite settings	Confirmation prompt for whether to overwrite settings. Alternatively, besides *After prompt* and *Without prompt*, you can use a *Variable*.

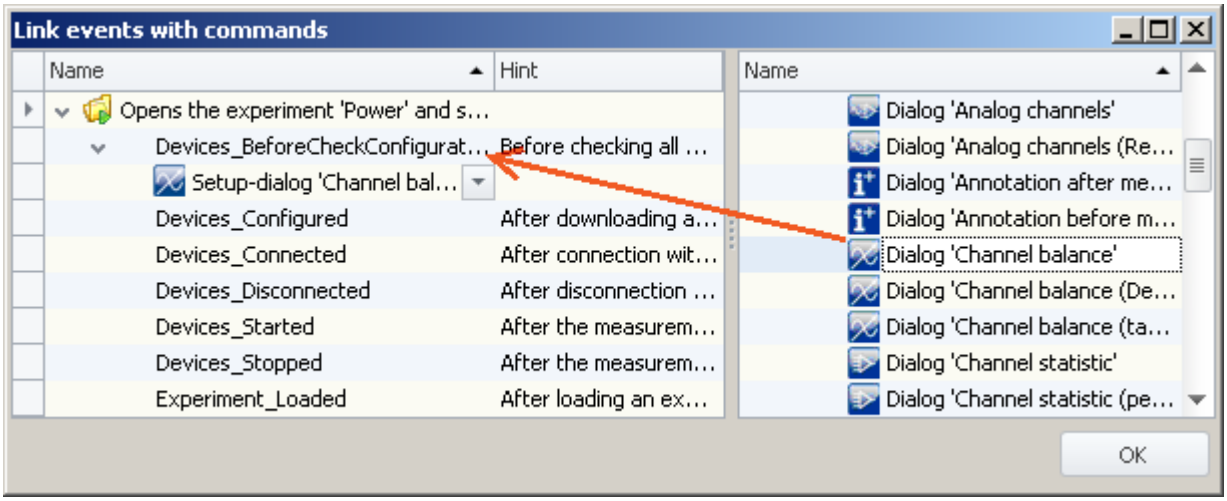
Note What happens if variables also exist in the experiment loaded?

Any "higher-level" variables are not overwritten if a variable of the same name but from a different validity range is loaded.
Example: Suppose a variable exists with the validity range "*Project*" or "*Sequencer*". An experiment is loaded by means of the command: "*Open experiment*". A variable having the same name exists in this experiment and has the validity range "*Experiment*". This variable is not loaded from the experiment when the command is run.

Events

Just as there **events** for each measurement, there are also events which can only be configured for this measurement.

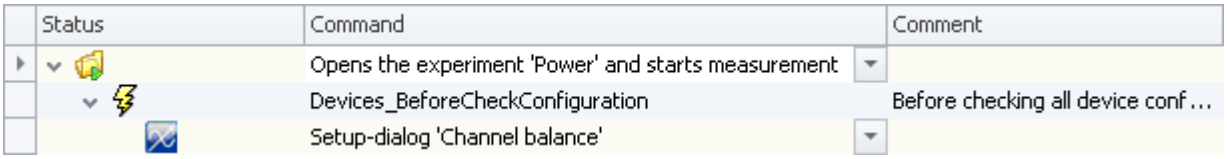
Using the parameter: *Events* opens an additional window:



In the example, the channel balancing dialog is called prior to processing the configuration

Here, you can attach commands to events of this measurement.

If events are used, they also appear in the Sequence table:



Example: Command associated with an event in the Sequence table

15.11.7 Panel

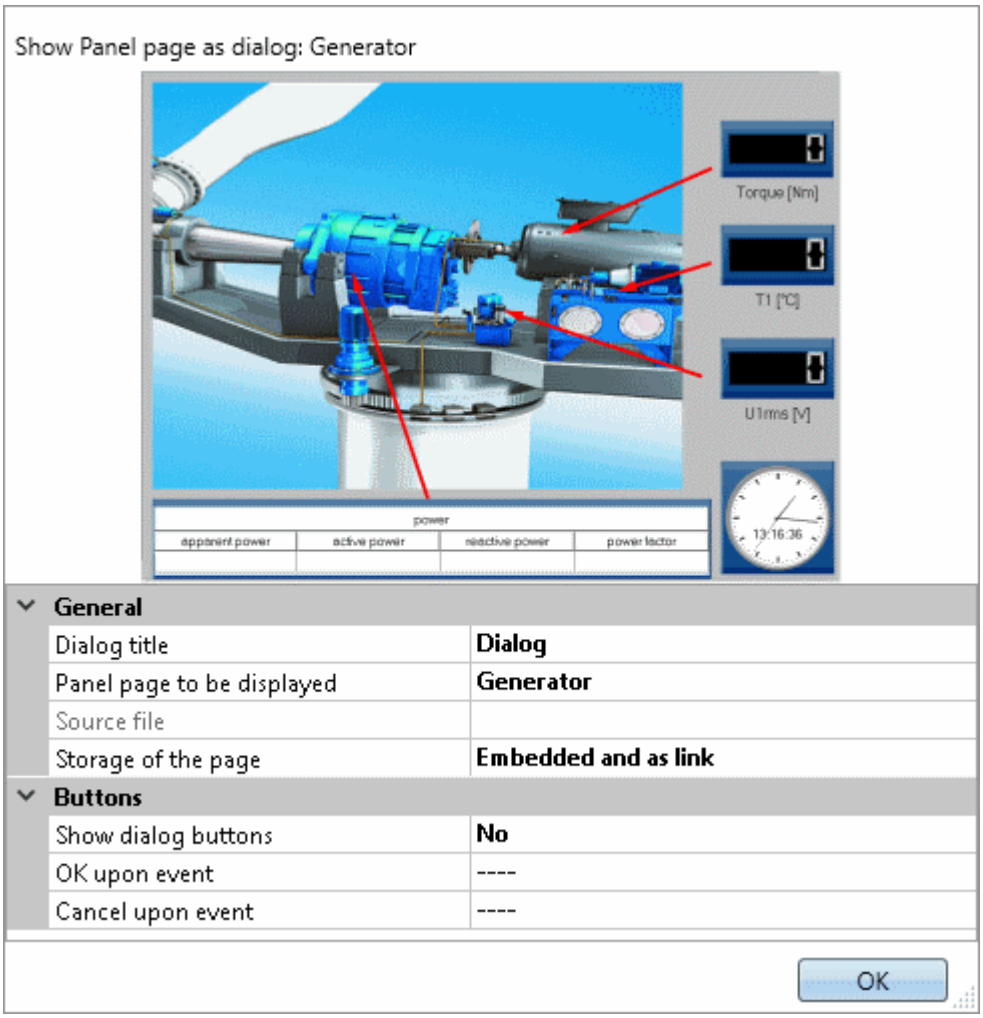
15.11.7.1 Show Panel page as a dialog

A Panel page can be displayed as a dialog.

The ways to close the dialog are:

- use existing buttons on the Panel page, or
- actuate the dialog's separate "OK" and "Cancel"-buttons.

If Setup pages are displayed on the Panel, the buttons can be hidden until all important boxes are filled.



Properties of the command "Panel-page as dialog"

Parameter: General	Description
Dialog title	Title of the dialog to be displayed
Panel page	Select the page to be displayed. The possible selections are existing pages and exported pages in the dbv format. Using "..." it is possible to select any arbitrary *.dbv file.
Source file	If this box is **empty**, the Panel pages of the current experiment are used. If you have selected a *.dbv-file in the box "*Panel page*", its name including its path is displayed here.

Parameter: General	Description
Storage of the page	Here you can define to which changes to the Panel pages the system responds. **Embedded:** This page is embedded into the command. There is no connection to the page in the Panel. • If this page is deleted, the command continues to function. • Changes to the page are not applied for the command. **As link:** Only a reference to the existing Panel page is saved. • If the page is deleted, the command no longer functions. If it is run, an empty page is displayed. • Changes to the page are applied for the command. **Embedded and as link:** The page is embedded into the command. Additionally, there is a linkage to the existing Panel page. As long as a page having this name is present, it is displayed. If the page is not present, the last state is displayed. • If the page is deleted, the command continues to function. • Changes to the page are applied for the command. • If a new page having the same name is created, then it is used.

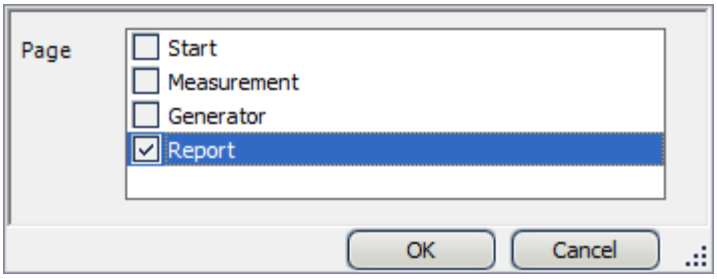
Parameter: Buttons	Description
Show dialog buttons	Separate "OK" and "Cancel"-buttons in the dialog can be shown or hidden. **Yes:** The buttons are displayed upon execution. The parameters for the OK- and Cancel-buttons are shown for the purpose of configuring the button. **No:** The buttons are not displayed. Instead, use certain buttons on the Panel page ("OK and Cancel upon event"). If neither of the two actions is defined upon event occurrence, the Close-button (X) is displayed. Then it is only possible to close the window using the "X". Closing this way is interpreted as "Cancel".
OK upon event	Select an event on the page, which closes the dialog with "OK". E.g. pushing on a switch. Some switches have two events, while others have only one. For this purpose, the switch must be present on the page.
Cancel upon event	Analogous to "OK upon Event". In this case, the dialog is closed with "Cancel".
OK button	This field is only enabled if "Show dialog buttons" is set to "Yes".
Visible	Shows/hides the button.
Active	The button can be deactivated in particular situations. **Always:** The button can be actuated at all times. **If all inputs are valid:** The button can only be actuated if all inputs are valid. For this purpose, embedded Setup pages are required. For details on this topic, see the chapter: "*Setup-Layout*" > "*Using Mandatory Boxes* 269".
Label	The button's caption.
Cancel button	This field is only available if "Show dialog buttons" is set to "Yes". For the Cancel-button, the same parameters are available as for the OK-button (Visible, Active, Label).

Note Evaluating the dialog's answer

Note that it is possible to have the dialog's response evaluated automatically.
See: Evaluating the last dialog's answer 1727.

15.11.7.2 Print Panel page

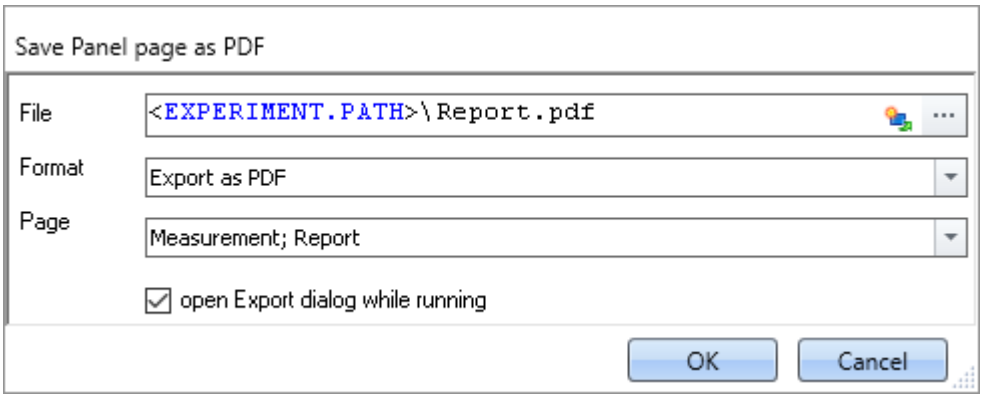
You can print out Panel pages. By this means, it is easy to compose reports. To do this, create Panel pages which are displayed as a report.



Example for printing out a Panel page

15.11.7.3 Export Panel page

Export Panel pages to a specified folder. As a graphic, PDF or in the form of Panel export.



Example of exporting Panel page

Parameter	Description
File	Specification of target file including the path.
Format	Target format for export. **Please always select the appropriate format!** Possible options:

Format	Description
Export Panel page	Exports the page in dbv-format. The file can be imported back into the Panel at another position as a page. For each page, a separate page is generated.
Export as PDF	Exports the pages in PDF. All pages are exported to one file.
Export as graphic	Exports the page as a graphics file in png-format. For each page, a separate page is generated.

Page	Selection of the pages to be exported. Depending on the parameter "File" and "Format", either one or more pages can be selected.

Parameter	Description
Open Export dialog while running	If this option is activated, then when the command is run the "Save as" dialog appears. The change has no effect on the configuration of the command in the commands list, but only affects the current export.

Note Name of the results file depending on the setting

For varying formats, there are different respective results unless there are clear specifications available for the parameter "File".

Panel page:

Parameter	Description
C:\tmp\Report.dbv	Generates the file: Report.dbv in the path C:\tmp Only possible when an individual page is selected.
C:\tmp\Report	Generates a separate dbv-file in the path C:\tmp\Report for each selected page
C:\tmp\Report\	The files are named for their respective page.

PDF:

Parameter	Description
C:\tmp\Report.pdf	Generates the file: Report.pdf in the path C:\tmp.
C:\tmp\Report	
C:\tmp\Report\	Generates the file .pdf in the path C:\tmp\Report (<-- *not recommended!*)

Graphic:

Parameter	Description
C:\tmp\Report.png	Generates the file Report.png in the path C:\tmp Only possible when an individual page is selected.
C:\tmp\Report	Generates a separate png-file for each selected page in the path C:\tmp\Report
C:\tmp\Report\	The files are named for their respective page.

15.11.7.4 Import Panel page

You can import a Panel page which was previously exported in DBV format.

15.11.7.5 Delete page

If the command is executed, a defined Panel page is deleted, without waiting for confirmation.

By means of the command's settings, you determine which Panel page is to be deleted. To do this, select an existing page, or enter the page name, if the page does not yet exist.

It is possible to use the list to select multiple pages. To select multiple pages by making an input, use the following syntax:

```
Site 1; Site 2; Site 3.
```

15.11.8 Parameter set

A parameter set is a group of parameters which can be used for various purposes. For instance, parameters may be **settings for the device and channel configuration**. They can also be **metadata** and **variables values**.

The command: "*Export parameters*" allows automated export of the parameters at desired times and according to previously adjusted settings.

The command: "*Import parameters*" allows automated import of the parameters at desired times and according to previously adjusted settings.

Reference

The description is provided in the section: "*Channel types, variable types, data types and data storage formats*" > "*Parameter set* 1159"

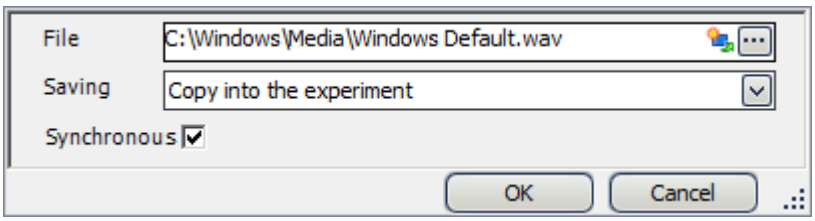
- "*Export parameters* 1167"
- "*Import parameters* 1172"

Note Import/Export of variables preferably via the variable commands

For **importing/exporting variables' values**, the command "***Load/Refill variable*** 1760" or "***Export variable*** 1758" should be used.These variable commands are specially designed for variable values.
Using the command: "*Import parameter*" can cause unintended alteration of the device control.
Certain variable types can only be exported by means of the command "*Export variables*"; this included text-variables.

15.11.9 Read/Write data

15.11.9.1 Play soundfile



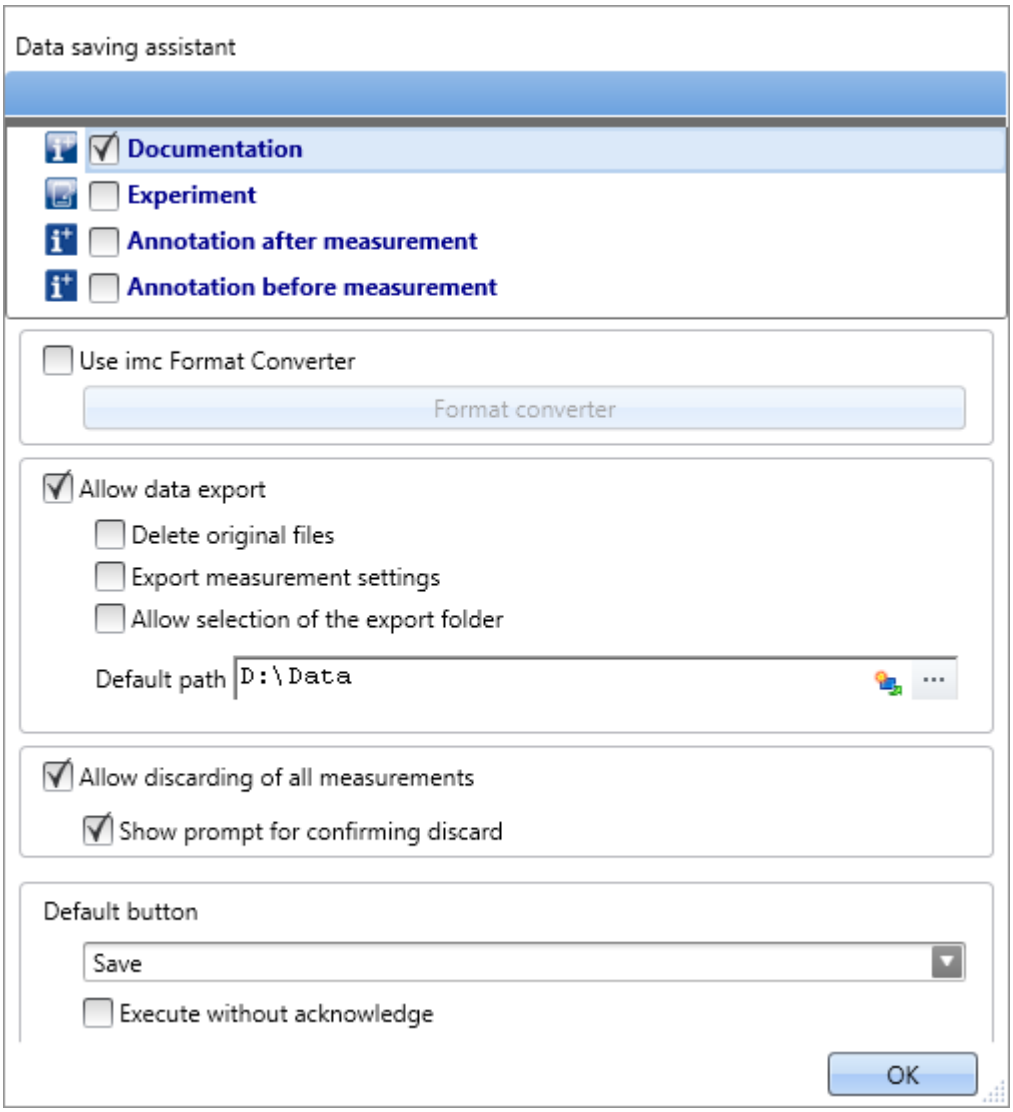
Properties of Play soundfile

Parameter	Description
File	Select an audio file in WAV format.
Saving	Sets whether the audio file is permanently saved in the experiment file *(As copy in the experiment)* or only the path to the file is saved *(Only as link).*
Synchronous	When activated, the Sequencer remains on this command until audio output was completed.

15.11.9.2 Data saving assistant

If data saving to the PC was selected for at least one active channel, then in the default case, the measurement results are saved in the database. With the help of the **Data Saving Assistant** command, one can continue to specifically process these measurements. This command must be started either through a Panel event (Widget) or by means of a sequencer event.

The Data Saving Assistant lists all new measurement folders which were created since the start of the last measurement. The number of folders depends on for instance the interval data saving or the interrupt/resume of data storage. Thus, these measurements, for example, can then be exported with or without deleting the original files.



Data Saving Assistant

Setup pages

It is possible to save **measurement comments** (metadata) which pertain to the measurement. In the list at the top, the Setup pages are listed. When you select one of the pages, this page's parameter set as saved with the measured data as a *.csv* file.

If the assistant is running, a new window appears. For each page selected, a tab is added at the top. There, you can fill the fields.

imc Format Converter

In addition to the saved or exported measurement results, you can transfer the results to other formats. In the options for the imc Format Converter, you can additionally specify the format, e.g. ASCII or EXCEL.

Data export

This settings allows you to export saved measurements.

Allow data export

When this option is active, the measurement results can be exported to a different location. When the command is executed, an additional button appears, by means of which the export is initiated. I.e. in that case the buttons *Export* and *Save* become available.

- If you click on *Save*, the measurement result only goes to the database.
- If you click on *Export*, the data are exported according to the settings for the other options.

Delete original files

The original files in the database can be deleted upon exporting. This means that the database only contains the explicitly saved measurements which were not exported. The exported measurements are only located in the export path.

Example: Supposing you don't need the measurement results to be in the database. If the measurements were performed correctly, the results are exported and the original files are deleted right away. In all other cases, you can delete the measurements right away by means of the button "*Discard measurement* 1739" (described further below).

Export measurement settings

For purposes of traceability, the measurement settings (in force for the current experiment) are saved along with the corresponding measurement results in the database (unless a different setting applies 132). If this option is active, the current experiment's settings are exported in parallel with the data export.

Allow selection of the export folder

- If a default path is specified, you can allow/forbid changes by the user.
- If this option is activated, a folder selection dialog always appears. If a default path is specified, it is the suggested target folder in the dialog.
- If this option is deactivated, export is always to the specified default path.

Default path

Here you set the target folder for export.

Discard measurement

You have the choice of discarding measurement results.

Allow discarding of all measurements

If this option is activated, it is possible to delete the measured results. When the command is executed, an additional button by means of which the results can be deleted appears. I.e., the buttons *Discard measurement* and *Save measurement* become available. If you click on *Save*, the measurement results remain in the database. If you click on *Discard measurement*, the measurement results are deleted.

Prompt to confirm discard

Here you set whether to post a confirmation prompt when *Discard measurement* is clicked.

Default button

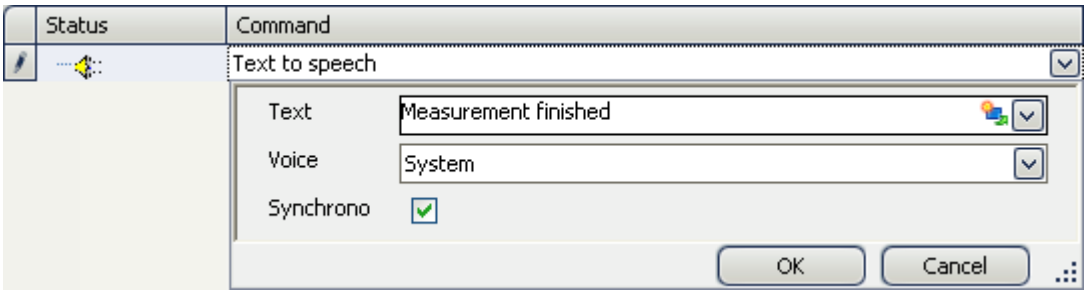
Here you set which button is "pre-selected". If the commando is executed, the selected action can be run by clicking on the Enter-key.

Execute without acknowledge

If the box "Execute without acknowledge" is checkmarked, the selected default action is executed automatically, without a delay for the purpose of confirmation. In this case, it is not possible to change the action if the command is executed.

You can only checkmark this box if the other settings do not require any acknowledgement. For instance, there would have to be a pre-set path for exporting, and no confirmation prompt would be required for deleting.

15.11.9.3 Text to speech



Properties of Text to speech

Parameter	Description
Text	Enter here the text to be spoken, e.g. "*Measurement finished*".
Saving	Character of the voice output. The available selection depends on the Windows system and usually only offers English voices.
Synchronous	When activated, the Sequencer remains on this command until audio output was completed.

15.11.10 Setup

15.11.10.1 Measure a user-defined characteristic curve

Characteristic curves of sensors provide the ability to convert measured electrical readings into physical values by referencing a plot of the relationship between those variables – the characteristic curve. Using this command, a characteristic curve is saved in a dat-file. This file can then later be imported as a supplemental file for imc Online FAMOS, for example.

Reference Sensors, Characteristic Curves and TEDS

A description of the sensors and how to generate their characteristic curves using the command is presented in the section:

"*Setup pages - Configuring device*" > .. > "*Sensors, Characteristic Curves and TEDS*" > "*Create a new sensor characteristic curve* 736"

15.11.10.2 Execute device action

By means of this command, you can perform a variety of device actions which are also available via the action buttons on the Setup pages. For instance, you can perform bridge balancing or a shunt calibration.

The list of available actions is unfiltered. For this reason, you also find actions here which have no effect unless you make additional settings on the Setup pages, or which have no effect as they are located in the list.

Parameter	Description
Enumerator Class	In an enumerator-class, the association with "object-groups" is defined (e.g. analog channels, Display variables, devices, ...). See also: "Enumeration Classes 246".
Action column	List of possible actions to be performed (corresponding to the buttons on the Setup pages)
Action	Some action columns contain various possible actions to be performed. For instance, with the action column: "Balance action", it is possible to perform a taring or bridge balancing.
Filter type	Here you define the target for which the action is to be performed. In many cases, multiple targets are available, e.g. all analog channels in the case of a balancing action. **All rows:** The action is performed for all targets possible. In some cases this can lead to error messages being posted if the particular action cannot be performed for certain targets. **Selection:** Here, select particular targets on which the action is to be performed. If any new targets are added, they are not already automatically selected. **Filter:** Create a filter according to whose criteria the targets are selected. Here are some examples:

Summary	Filter formulation	Interne designation
all active channels	[Channel status] Equals Active	`([eStatus] = 1)`
Only one balancing type	[Balance action] Equals Bridge	`([eBalanceAction] = 2012)`
Only one particular module type	[Module type] Equals Bridge, current and voltage amplifier and temperature measurement unit "UNI2-8"	`([eModuleType] = 47)`
Only channels belonging to a defined coupling	Coupling Is any of (Full bridge, Half bridge, Quarter bridge)	`[eCoupling] in (5,3,6))`

As an illustration, the following examples from practical applications

Example Balancing 1

Perform a balancing action: for all channels currently selected

Parameter	Setting
Enumerator Class	Channel
Action column	Balance action
Action	Currently selected
Filter type	All rows (here the use of a filter is recommended: see further below)

For all analog channels, whichever of the balancing types "Bridge", "Taring" or "Manufacturer's calibration" which is set on the Setup page: "Balancing" is performed.

Note: Since passive channels will report an error in consequence of this action, it is advisable to define a filter for all "Active" channels:

[Channel status]] Equals Active

`([eStatus] = 1)`

Example Balancing 2

Perform bridge balancing: for all active channels of the module type: "Bridge, current and voltage amplifier and temperature measurement unit "UNI2-8""

Parameter	Setting
Enumerator Class	Channel
Action column	Balance action
Action	Bridge
Filter type	Filter: Only active channels and only channels belonging to the amplifier: "UNI2-8" (([eModuleType] = 47) And ([eStatus] = 1)) (here, it is recommended to use an additional filter: Only channels for which a suitable coupling is set: see further below),

For all analog channels having the appropriate amplifier, a bridge balancing operation is performed.

Note: Since not all channel settings allow bridge balancing, it makes sense to modify the filter even more. Depending on the amplifier used, the coupling may need to be set to full bridge, half-, or quarter-bridge:

`(([eModuleType] = 47) And ([eStatus] = 1) And [eCoupling] in (5,3,6))`

Example Shunt calibration

Perform a shunt calibration: for all channels selected

Parameter	Setting
Enumerator Class	Channel
Action column	Shunt calibration
Action	--
Filter type	Selection

Example Starting measurement

Start a measurement: for all devices

Parameter	Setting
Enumerator Class	Device
Action column	Device control
Action	Start
Filter type	All rows (starting a specified device is also possible)

All devices are started.

Note: The difference from the menu action is that correct preparation (reconfiguration) is always performed previously.

Instead of starting measurement this way, please use the command: "*Execute menu action* 1756".

When the action "*Start*" is performed using the command "*Execute device action*", no calculations or analyses performed on the PC side, such as imc Inline FAMOS-calculations, are started. Only the devices are started.

15.11.10.3 Import MFB configuration

The command enables the **import** of Fieldbus-configurations. The following Fieldbusses are supported:

- **CAN:** *.cba; *.dbc
- **ARINC:** *.idb
- **ECAT-IF:** *.imcecatif.xml
- **AFDX:** *.xml

Reference Description of the configuration files

The files are generated using the Fieldbus-Assistant. For a detailed description, please refer to the associated chapters of the manual.

Parameter	Description
Import file	Specification of the file to import, including the pathname.

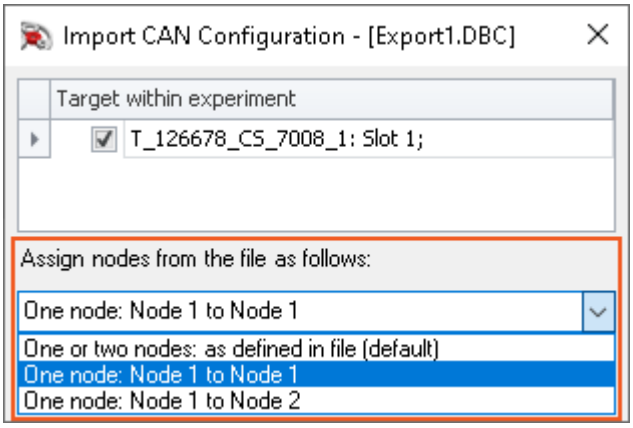
Parameter	Description
Mapping	Either specify the device so that no additional prompt will appear upon execution of the command, or wait to select the device until when you run the command. **Show mapping table:** Upon **execution**, a **prompt** appears which requests you to specify the device to use as the target. You can predefine the mapping table. **Use mapping table:** Specify the desired device by means of the additional parameter "*Mapping table*".
Mapping table	Here, select the desired device. The target is a device having an appropriate module.

Menu item: "Import/Export":

By means of this menu item, you can also import MFB-configurations. Here, the same options are available for your use.

CAN-import - Node selection

If the configuration only contains information on one node, you can select to which node to perform the import.



Selection	Description
One or two nodes: as defined in file	The file's content is processed sequentially. The first node (belonging to the file) is placed on the first node (in the Assistant). If there is any second node in the file (only cba), then this is set on the second. If there is no second node in the file, the configuration of the **second node** is **deleted** from the Assistant.
One node: Node 1 to Node 1	The file contains only one node - The content of the file is imported **to the first node** in the Assistant. The configuration of the **second node remains intact**.
One node: Node 1 to Node 2	The file contains only one node - The content of the file is imported **to the second node** in the Assistant. The configuration of the **first node remains intact**.

15.11.10.4 Reassignment of module positions

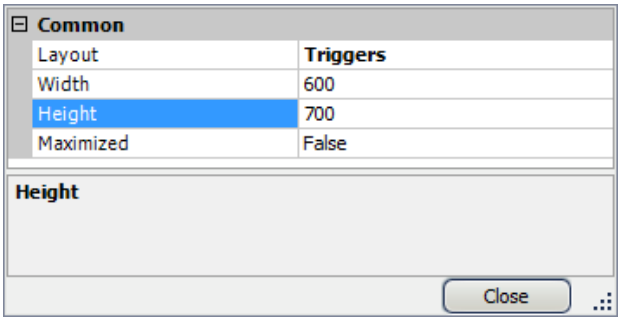
The module addresses are re-assigned. This matches the working of the function "*Reassign*" in the dialog "*Module Properties*". The existing module numbers are overwritten according to a gapless re-numbering process.

Reference

For more information on the module position and re-assignment, see the chapter: "*Module Properties* 226".

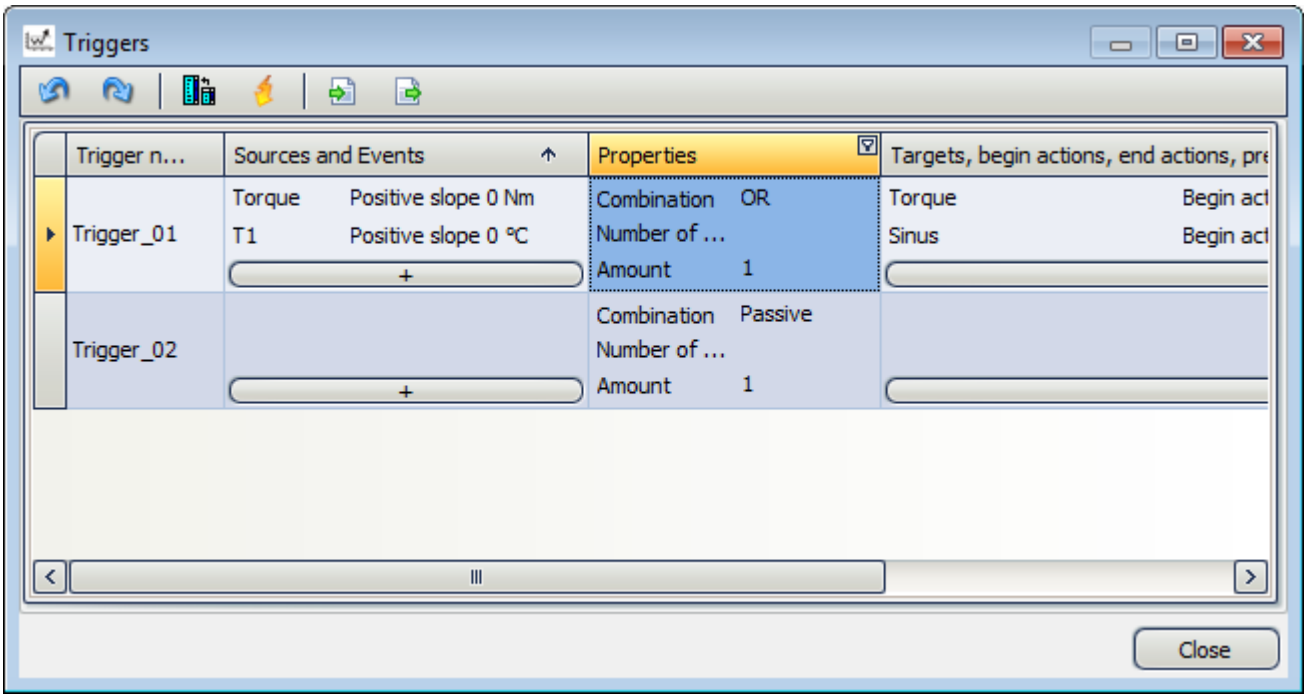
15.11.10.5 Display Setup dialog

The command group *Setup* offers all setup settings in dialog form.



Setup properties

Select *Layout* and set the *Width* and *Height*. Alternatively, the dialog's display can be *maximized*.



Example Trigger dialog, started by means of the command Setup

15.11.10.6 Import synthesizer-/controller configuaration

This command enables **import** of synthesizer and controller configurations for the **Synthesizer-module**.

Reference Description of the configuration files

The files are generated by means of the Synthesizer-Assistant. A detailed description is presented in the Synthesizer user's manual.

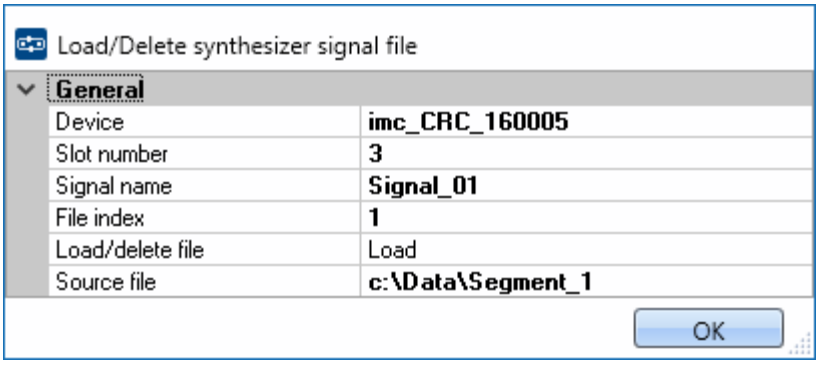
Parameter	Description
Import file	Specification of the file to import, including the pathname.
Mapping	Either specify the device so that no additional prompt will appear upon execution of the command, or wait to select the device until when you run the command. **Show mapping table:** Upon **execution**, a **prompt** appears which requests you to specify the device to use as the target. **Use mapping table:** Specify the desired device by means of the additional parameter "*Mapping table*". **auto:** The system searches for an appropriate target for the import. Only devices having a suitable module can be the target. • The import file contains a statement of the source device. If this device is selected, it is used. • If that device is not selected, the system uses a device among the ones which are selected which has the same number of slots. • If no such device is available either, the system selects the next best one. Assignment of the slot is performed in a similar way. If the same numbers are available, they are used. Otherwise the assignment procedure uses other numbers in numerical succession.
Mapping table	Here, select the desired device. The target is a device having an appropriate module.

15.11.10.7 Load/Delete synthesizer signal file

Using imc STUDIO you can load files for the Synthesizer module into the device, or delete them from the device. By means of an index you can import various segment-files. With an appropriate device, you can play back these segments in succession.

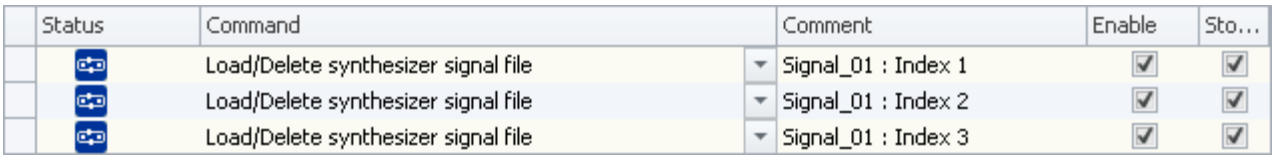
In the Properties you select the Synthesizer output. The outputs are distinguished by their respective slot numbers and signal names.

Parameter	Description
Device	Here, select the target device. The device requires a Synthesizer-module
Slot number	Synthesizer slot in the device
Signal name	See the Synthesizer-configuration for the signal name.
File index	A signal's segments can be changed during the measurement. By means of the parameter, the segment is assigned an index in reference to which it is accessed. The index indicates the number of the segment which is selected by means of the pv-variable <SignalName>_Slot<Number>_NextFile. For an detailed description, see the section "*Managing the segment files*" in the "*Synthesizer Manual*".
Load/delete file	• **Load:** Transfers the source file to the device • **Delete:** Deletes the file having the index specified from the device
Source file	Here, select the file you wish to import.



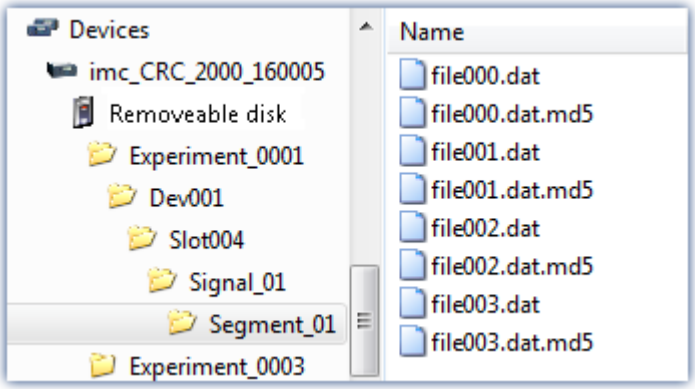
Assignment of the segment to the synthesizer output

It is thus possible to transfer multiple segments in succession:



Transfer of three segments

The segments are set up in the device's data carrier as follows:



File structure of segments on the device disk

Note

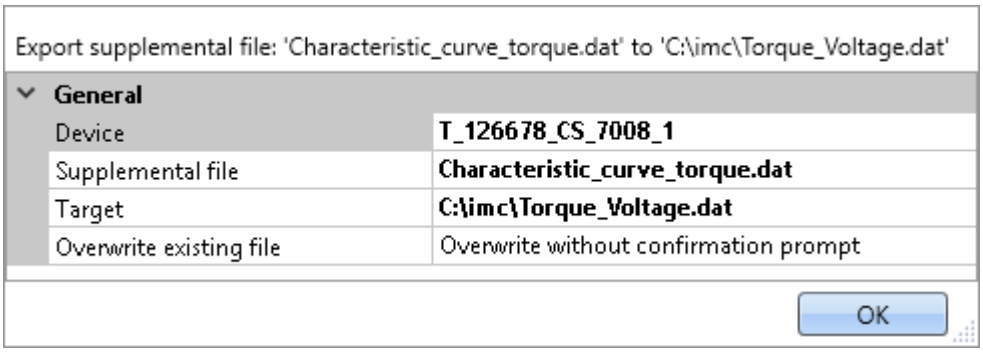
The data are saved on the device drive in a special format. It is **not** possible to copy the data manually!

15.11.10.8 Export supplemental file

This command enables export of a supplemental file. The files can, for example, be exported from the device.

Reference Description of the supplemental files

Supplemental files exist in various file types. For an exact description of the supplemental files, see the section: "*Setup - Device Configuration*" > *Ribbon:* "*Supplemental Files* 212".



Configuration: Export supplemental file

Parameter	Description
Device	Here, select the source where the file is located. The source may be a device. It is always only possible to make one selection.
Supplemental file	Specifies the file to be exported.
Target	Target file including the path.

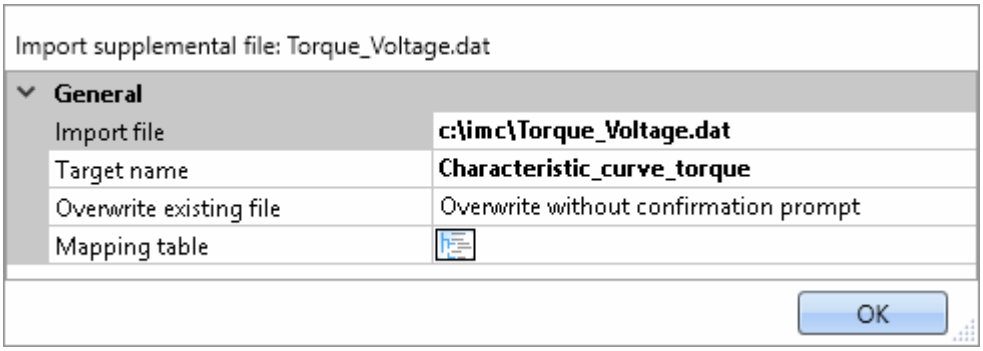
Parameter	Description
Overwrite existing file	Here, define the system's behavior for handling the files if the target file already exists. • **Overwrite without confirmation prompt:** Files in the target location which have the same name are overwritten without any confirmation prompt. • **Do not overwrite:** Files in the target location which have the same name are retained. The file is not exported and no notification is posted. • **Prompt for action:** A dialog concerning the subsequent proceedings is displayed. Available options are: "Overwrite", "Cancel". Each of these actions concludes the command with "Finished". This option is relevant for automated routines.

15.11.10.9 Import supplemental file

This command enables import of one supplemental file. The files can be used by imc Online FAMOS, for example.

Reference	Description of the supplemental files

Supplemental files exist in various file types. For an exact description of the supplemental files, see the section: "*Setup - Device Configuration*" > *Ribbon:* "*Supplemental Files* 212".



Example of a characteristic curve used in imc Online FAMOS with the Charact() function

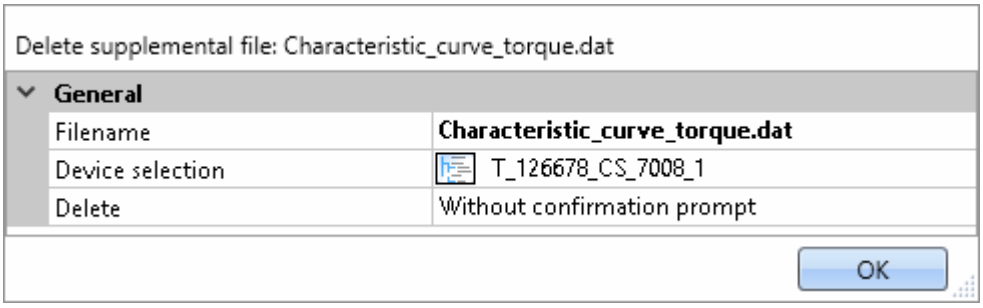
Parameter	Description
Import file	Specification of the file to import, including the pathname.
Target name	Optional entry. Here, specify a deviating name if the filename in the device is not supposed to match the original filename.
Overwrite existing file	Here, define the system's behavior for handling the files if the target file already exists. • **Overwrite without confirmation prompt:** Files in the target location which have the same name are overwritten without any confirmation prompt. • **Do not overwrite:** Files in the target location which have the same name are retained. The file is not imported and no notification is posted. • **Prompt for action:** A dialog concerning the subsequent proceedings is displayed. Available options are: "Overwrite", "Rename", "Cancel". Each of these actions concludes the command with "Finished". This option is relevant for automated routines.
Mapping table	Here, select the import target. The target may be a device . it is always only possible to make one selection.

15.11.10.10 Delete supplemental file

This command enables deletion of a supplemental file. The files can be deleted from the device, for example.

Reference Description of the supplemental files

Supplemental files exist in various file types. For an exact description of the supplemental files, see the section: "*Setup - Device Configuration*" > *Ribbon:* "*Supplemental Files* 212".



Configuration: Delete supplemental file

Parameter	Description
Filename	Specifies the file to be deleted.
Device selection	Here, select the source where the file is located. The source can be a device. It is always only possible to make one selection.
Delete	Define here whether a confirmation prompt is to appear. • **Without confirmation prompt:** The supplemental file is deleted without any confirmation. • **With confirmation prompt:** A dialog concerning the subsequent proceedings is displayed. Available options are: "Delete", "Don't delete". Each of these actions concludes the command with "Finished". This option is relevant for automated routines.

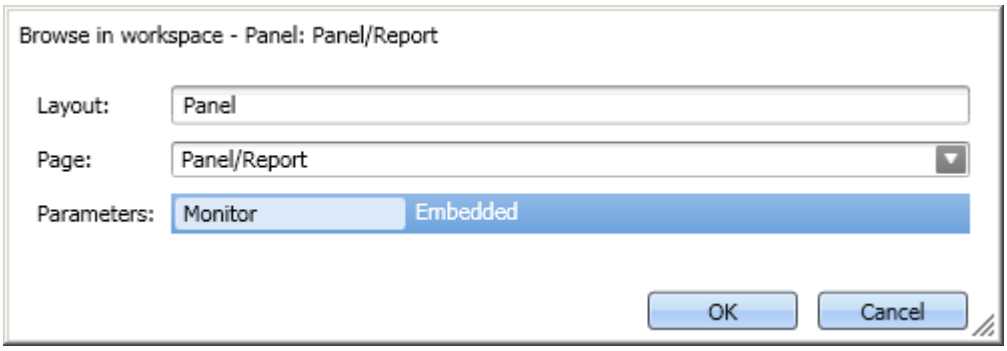
15.11.11 User interacion

15.11.11.1 Browse in workspace

If the command **Browse in workspace** is executed, imc STUDIO jumps to the selected page. Every main window can be selected as the target. With main windows which have their own pages, it is also possible to jump directly to a special page (e.g. Panel or Setup).

If a Panel page is defined as the target, the page can also be displayed on a monitor of the user's choice in fullscreen mode. In this case, imc STUDIO does not change the main window. Thus, for instance, the Sequencer can be viewed on the main monitor, while on the second monitor, the measured data can be viewed on a Panel page in fullscreen mode.

Using this command, it is always possible to open the appropriate window via the Sequencer, or to jump to, for example, the balancing dialog, by means of a button on a Panel page.



Selection of a specific page

Parameter	Description
Layout	Display of the main page's name. Adjusts itself according to the target page. Can not be edited.
Page	Here, you define the target page.
Parameters	If target page: Panel • **View: Panel Embedded Mode:** Jumps to the Panel. If necessary, exits fullscreen view. • **View: Panel Fullscreen Mode:** Jumps to the Panel in fullscreen mode (see Panel menu ribbon: *Control* > *Panel Fullscreen* 1206). If target page: Panel page • **Monitor: Embedded:** Jumps to the Panel and opens the selected page • **Monitor: <Monitor name>:** The page is displayed in fullscreen mode on the monitor selected. Independent of the opened main window (see Panel Context menu: The Panel page's tab: *Show Page on Monitor* 1231).

Reference

An example of the command: "*Browse in workspace*" is presented in the user's manual: Sequencer > Tutorial 1765.

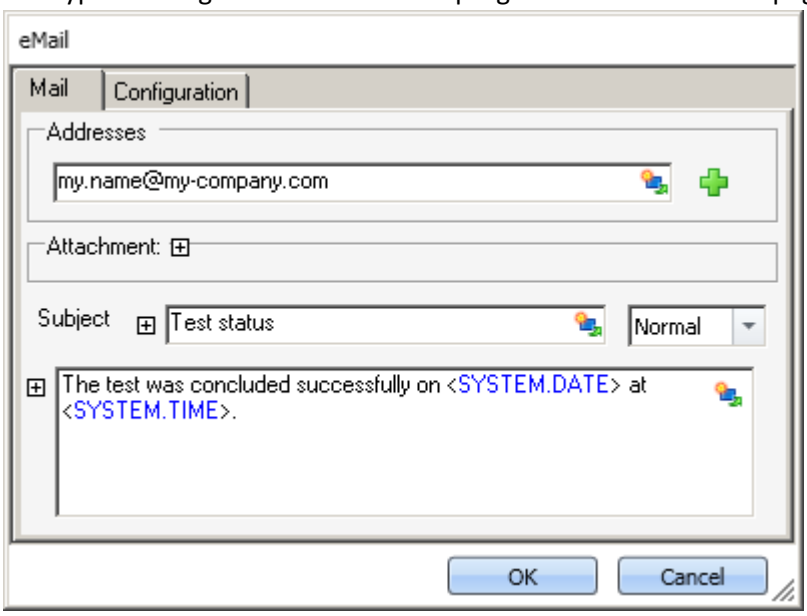
15.11.11.2 E-mail

The command **E-mail** enables automated sending of an e-mail in important situations. For this purpose, no external e-mail program is required.

The E-Mail-provider requires the login data of the sender's address.

Mail

The typical dialog boxes for an e-mail program are found on the page *Mail*.



Defining an E-Mail

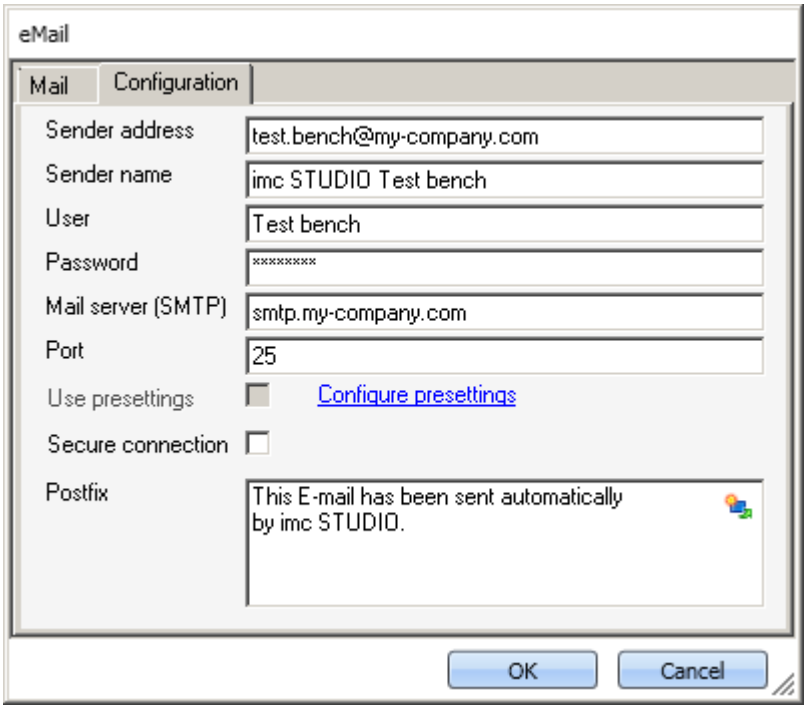
By right-clicking, or by clicking on the placeholder symbol ( ), you can apply a variety of placeholders.

Under *Addresses*, you can define one or more recipients. Additionally, an *Attachment* can be added and the *Priority* of your message specified.

The box *Subject* and the text to be sent can be defined in various languages.

Configuration

On the page *Configuration* you set the access data for the E-Mail provider.



Defining the E-Mail configuration

Parameter	Description
Sender address	Sender address (your e-mail address).
Sender name	The name of the sender of the e-mail (your name).
User	User name for logging in.
Password	Password for authentication.
Mail server (SMTP)	Outgoing mail server of the E-Mail provider used.
Port	SMTP server port used.
Use presettings	In the imc STUDIO option, it is possible to define a presetting for the E-Mail command (Ribbon *Extra > Options*). If "Use presettings" is activated, these settings are applied. This means that the setting can be defined a single time in the Options and does not need to be entered again for each E-Mail command.
Encoded connection	Encryption protocol for secure data transmission via Internet.
Postfix	This postfix will be appended to every e-mail.

FAQ

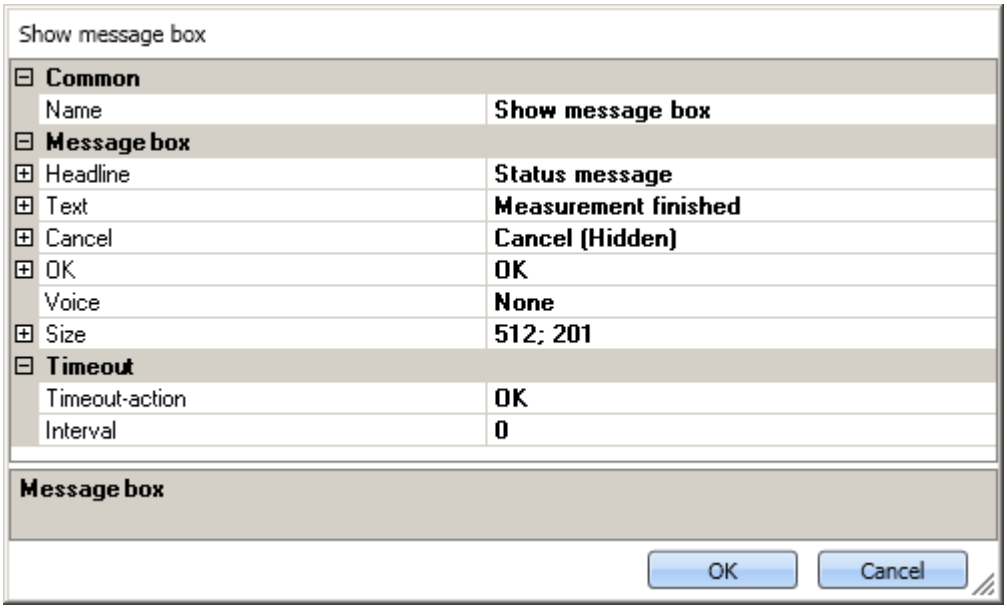
Why do e-mails not get sent, even though the settings are correct?

Answer: For reasons of security, some firewall programs prohibit the sending of e-mails when the firewall is not familiar with the program. Please check your firewall and configure it if necessary. **The process "imc.studio.exe" must be permitted to send e-mails if imc STUDIO is intended to send mails.**

15.11.11.3 Show message box

When the command *Show message box* was executed, a user-defined dialog was displayed. This can be used to display information, or to ask the user what to do next.

Along with a header and the information text, the dialog can also be displayed with a voice output.



Show message box

Parameter	Description
Name	Display text in the sequence table (without effect on the dialog).
Headline	The dialog's header.
Text	Text displayed in the dialog. With "\r\n", it is possible to force a line break.
Cancel OK	Settings for the buttons: "OK" and "Cancel" **Text:** Button caption **Visible:** The button can be either shown or hidden. • **Show:** The button is shown. In this way, the dialog can be closed with the corresponding dialog response. • **Hide:** The button is hidden In this way, the dialog can not be closed with the corresponding dialog response.
Voice	Character of the voice output. The available selection depends on the Windows system and usually only offers English voices.
Size	Size of the dialog.
Timeout-action	The selected action is performed upon elapse of the timeout time. The dialog is closed with "OK"/"Cancel".
Interval	Duration in seconds [s] until the timeout occurs. "0" switches the timeout off.

Note — Evaluating the dialog's answer

Note that it is possible to have the dialog's response evaluated automatically.
See: Evaluating the last dialog's answer 1727.

Reference Tutorial

An example of the command: "*Show message box*" is presented in the user's manual:
Sequencer > Tutorial 1765.

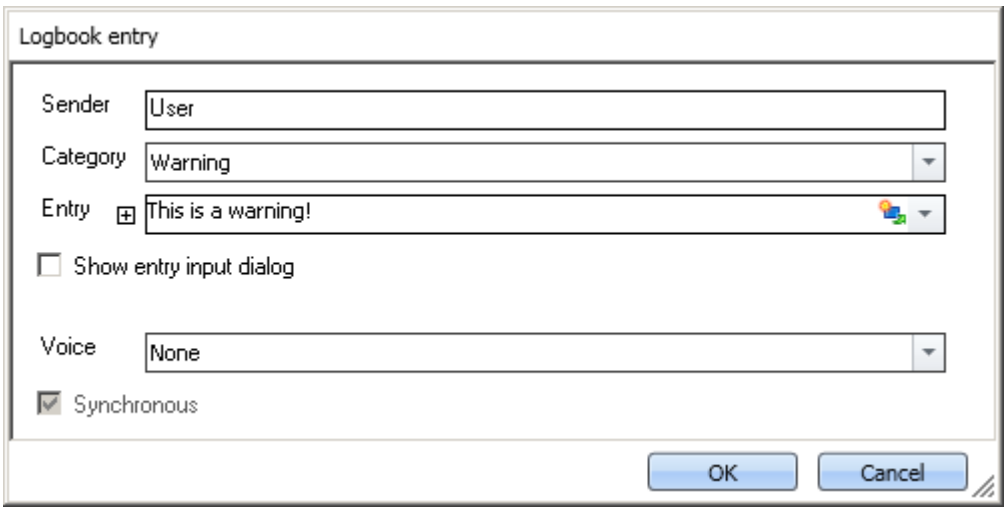
15.11.11.4 Open logbook-viewer

If the command is executed, the Logbook-Viewer is opened. How to operate the Logbook-Viewer is described in context of the tool window: Logbook 142.

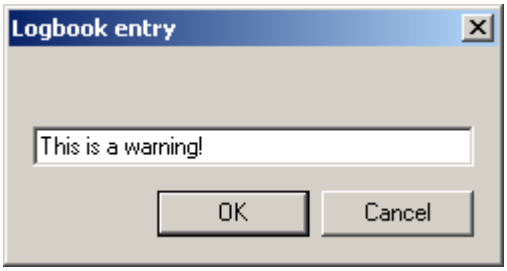
The Logbook-Viewer can be started with defined filter settings. For example, messages of the category: *Information* 143 and all Duplicates 144 can be hidden.

15.11.11.5 Logbook entry

Using this command, you can create an entry in the logbook and stop the Sequencer if appropriate.



Properties of the log book entry

Parameter	Description
Sender	Entry of information on the message's origin.
Category	There are four different categories: • **Information:** Informative messages • **Warning:** Warning text entry in the log book, otherwise no other effects. • **Error:** Error message, which interrupts the running Sequencer if appropriate. • **Fatal:** Fatal error message, which interrupts the running Sequencer if appropriate.
Entry	Text entered as message in the log book.
Shows entry input dialog	Enables a text input during runtime.  *With the option: "Show entry input dialog"*

Parameter	Description
Voice	The message text can be spoken by the computer voice.
Synchronous	When activated, the Sequencer remains on this command until audio output (Voice) was completed.

Output in the Logbook:

The output is logged to the logbook.

	Time	Code	Message	Sender
	13.02.2015 16:42:48	0	Sequencer ...done	imc STUDIO
	13.02.2015 16:42:48	0	This is a fatal error!	User
	13.02.2015 16:42:48	0	This is an error!	User
	13.02.2015 16:42:47	0	This is a warning!	User
	13.02.2015 16:42:47	0	This is some information	User
	13.02.2015 16:42:47	0	Sequencer ...starting	imc STUDIO

Logbook with generated entries

15.11.11.6 Execute menu action

This command lets you execute a menu action. A list contains the actions available in the menu ribbon. When this command is run, it replicates clicking the mouse over the corresponding button.

This command is often linked with switches on the Panel. In particular when the Panel is run in fullscreen. The menu ribbon is not present then.

Reference Widget: "Execute menu action"

The action can be directly linked with the synonymous Widget: "*Execute menu action* 1526". This Widget adapts itself to the status of the menu action. E.g. when the action is currently not present, the button can not be clicked.

Note When is the command "Done"

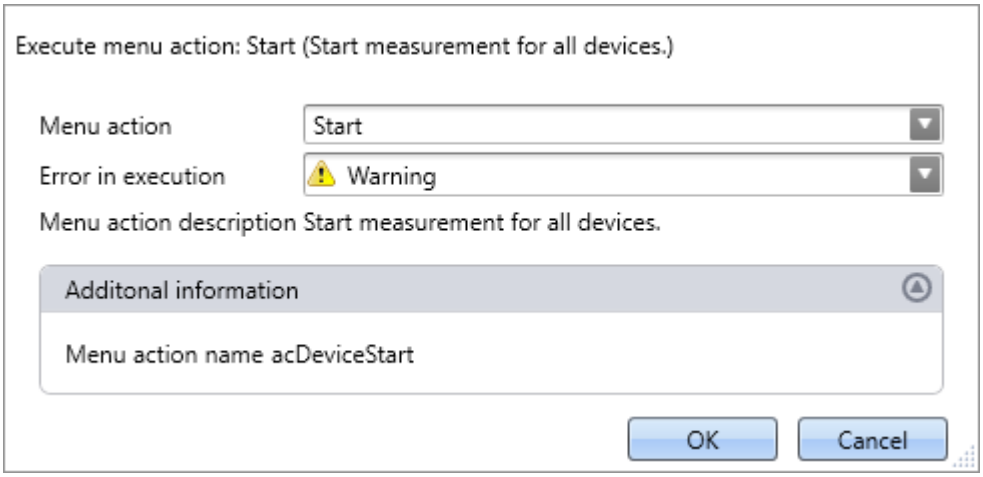
The sequence does not wait for the function behind the action to be concluded. For the sequence, the action is "done" when the action is acknowledged.

Examples:

- Menu action: Connect
  The command does not wait until the connection can be established.
  The command does not care whether the connection even can be established.
- Menu action: Start Measurement
  The command does not wait until the measurement has been started.
  The command does not care whether the measurement even can be started.

Once the action has been initiated, the command is "done". Even if error messages are posted afterward.

The menu action can not be executed if it is not available or not enabled. Some menu actions are attached to specific user roles, device components or loaded imc STUDIO-components. When such actions are initiated, a corresponding error message is posted.



Configuration: Execute menu action

Parameter	Description
Menu action	Here, select the desired menu action. Open the drop-down list. The menu actions appear in groups. Scroll to the appropriate group and select the desired action. Filtering: You can enter the first letter of the action's name in the input box. Then the list will be restricted to actions starting with that letter.
Error in execution	If the action is not enabled or not available, the command may respond in various ways. • Error: The command returns an error message in the logbook. If "Stop on error" is activated, the sequence is stopped. • Warning: The command returns a warning message in the logbook. The sequence continues to run. • Ignore: The command is concluded with "Done". No info appears in the logbook.
Menu action description	The action's description text is displayed.
Additional information	The menu action's internal designator is displayed.

15.11.12 Variables

15.11.12.1 Export variable

Description of the command "*Export variables*" and of the corresponding function accessed via the Data Browser 1222.

The function **enables variables** and their respective values **to be saved** at any desired location. It is possible to select one or more variables to be exported.

The following applies to the command: All options can be either specified, or selected manually while the Sequencer is running.

Parameter - File options	Description
Folder	Specifies the target folder. If no target folder is specified and the option: "*Show Dialog*" (only applicable to command) is not activated, a folder selection dialog appears when the function is executed.
Save to one file per file format (Save all in a single file)	If this option is activated, all variables are saved to a single (multi-channel) file. The filename can be specified. The file format for the file does not need to be specified. However, if no filename is specified, all variables are automatically saved to the file *data.dat*. Furthermore, for each file format a separate file is created. If the option is not activated, each variable is saved in a separate file. In that case, the filename can be specified separately for each variable.

Parameter - Mapping instruction	Description
Mapping instruction	Here you can select pre-defined export settings. • **No selection:** The user-defined settings are used (e.g. selection of variables) • **Default Export Mapping:** It is not possible to select variables. Instead, all variables (but not single-value variables) are exported. Thus, the same functionality is implement as for the menu action: "**Save current Data**". Additionally, it is possible to select a measurement whose variables are to be exported.
Measurement	Here you specify which measurement's variables to export. The option can only be used in conjunction with a mapping instruction. • **No selection:** Variables from the current measurement are exported. • **Selection or entry possibilities:** Measurement name or Symbolic measurement name (with measurement number)

Parameter - Variables	Description
Add variables	Here, the variables to be exported are selected and entered in the Variables list. • If the variables exist in the Data Browser, they can be selected by means of the button ⋯. • If a variable does not yet exist, it is still possible to add it. Enter its future name and confirm by clicking on Enter. Selected and entered variables appear in the Variables List.

Parameter - Variables list	Description
By means of the checkbox, it is possible to de-select for export a variable already entered. This means that it still exists in the list, but is not exported. It can later be activated for export again at any time.	
Variable name	Name of the variable as it is shown in the Data Browser
Measurement name	Measurement from which the variable is to be exported • **No selection:** The variable is selected from the current measurement • **Selection or entry possibilities:** Measurement name or Symbolic measurement name (with measurement number)
File format	Target format of the file to export. The system selects the file format automatically on the basis of the variable type. Alternatively, you can specify the file format in the settings under Options 137.
Filename	(not for: "*Save all in a single file*") By default, the file name is formed from the variable's name. You can change this here.

Parameter - Other options	Description
Treat error as warning	If the function is executed even though not all selected variables exist, error messages are posted. In normal cases, the function and the Sequencer is exited when an error message appears. This also causes not all variables to be exported. The variables in the list are exported in succession and all variables below the non-existent variable are thus not exported. In order that always exports all existing variables, it is possible to activate this option. In this case, no error messages appear in the logbook, but warnings instead. Warnings do not cause the function or the Sequencer to close.
Show file options and Show variables options (Only applicable to command)	If one of the two options is activated, then when the command is executed, the settings dialog opens for the respective scope: • **Show file options:** File options • **Show variable options:** Mapping instruction, Variables and Variables list. All settings can be checked and edited. Any changes do not take effect on the configuration of the command in the command list, but only on the export currently performed.
Always overwrite existing files	If this option is activated, files having the same name in the destination folder will be overwritten without a confirmation prompt. This option is useful for automated routines.

File comment	A file comment can be added.
	In imc FAMOS, the file comment is called as shown below:
	`path = FileName?(Channel_001)`
	`id = FileOpenDSF(path, 0)`
	`comment = FileComm?(id)`
	`FileClose(id)`
	Supported:
	• Export in RAW/DAT-format, as |NO Key
	• Export to *.aet files for the placeholder %FILECOMMENT%

15.11.12.2 Load/Refill variable

Load Variable: The command enables **creation or replace** of user-defined **variables with pre-defined values** from saved files. The user-defined variable inherits all of the properties of the variable from the saved file. (Possible targets: User-defined variables)

- If the variable doesn't exist yet, it is created.
- If the variable exists already, the existing variable is replaced with the new one.

Refill variable: The command enables **setting the values of an existing variable** from saved files. The variable must already exist before the command is run. Only the values are adopted. (Possible targets: among others device or user defined variables)

The values must be saved in a suitable file, for example one which was created using the commando: "*Export Variable* 1758". It is possible to select one or more files. If you wish to use multiple files, they must all be located in one folder.

All options can be either specified, or selected manually while the Sequencer is running.

Note Background information

A function "***Load variable***" is now implemented, which resembles "Load Data" in imc FAMOS. Overwriting existing variables is allowed. A confirmation prompt appears for each conflict of variables. If any of the confirmation prompts is canceled, the entire import automatically cancels. In this way, the system prevents the loading of an incomplete set of controller parameters, for example.

When an existing variable is overwritten in the process of loading, then it is effectively completely replaced, including all of its properties. In consequence, the old variable is neither deleted nor is any new one created, so that no "events" are triggered by the deletion or creation. This is important for scripts, curve windows, etc.

In the process, the user-defined variables' scope is also overwritten. But when a variable for which no scope is defined is imported, the variable becomes "*temporary*".

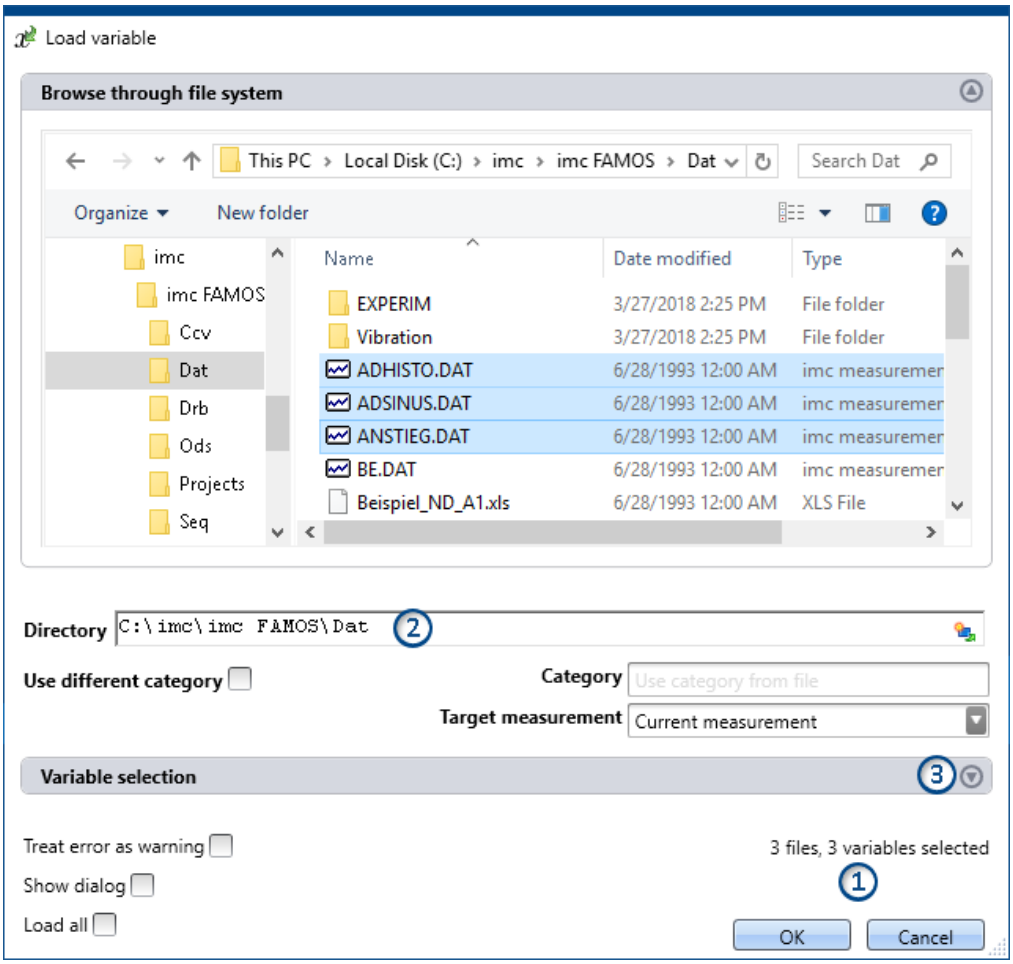
In contrast to imc FAMOS, there are variables which cannot be overwritten, e.g. channels or pv-variables. The reason is that, for instance, these variables belong to the specific device, or possess specific data formats and other properties which must be retained. If the user attempts to overwrite them, an error message is posted indicating that they can't be overwritten.

With the "*Load variable*" command, there is an option for determining whether existing variables can be overwritten with or without a confirmation prompt. By default, this option is activated, meaning that there is no confirmation prompt for overwriting.

By using "***Refill variable***", it is possible to change the content of existing variables. In this way it is possible, for instance, to change the content of pv-variables or user-defined variables. Since this only changes the variables' content and not their type, the target variable and the variable to be loaded must have the same properties, including, for example, the unit. Thus, only values and the sample count may be different. Therefore, please always use "*Load variable*" in cases where changes are possible.

Brief description

In the upper region, there is an Explorer for selecting files:



Here you can use the "Explorer"-functions to **navigate to the files and select them**. After making a selection, you can close the dialog by clicking "OK". **All data selected are loaded** (or, when using the command, selected for loading).

Exception: If you don't select any file in the Explorer, all importable files belonging to the current folder are automatically selected.

Note — Adding variables subsequently

Please be aware that the **variables are no longer in the selected state** in the Explorer once the command has been concluded by clicking "*OK*". If you open the configuration again and **select an additional file**, then **only this file is selected**; all others are de-selected.

Add more files by means of either the <CTRL>- or <SHIFT>-key. In this case, the files are added to the list.

Remove the variables via the "Variable selection" area.

File count (1): At the bottom right there is a readout of how many files are selected and how many variables belonging to these files are selected.

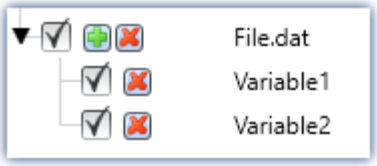
Directory (2): Below the file selection window, there is an input box for the folder path, which can be used as an alternative for specifying the file(s) desired. Here, you can enter placeholders ( ) or paths which only exist once the command is run.

Note Utilizing a placeholder

The placeholder can be used at all positions (start, middle, end). E.g. "`<EXPERIMENT.PATH>\Data\`"
When possible, the path is resolved in the "Explorer". There, you can also select files.
However, if you open a subfolder or a different folder in the Explorer, the field "Directory" is filled with the current value. The placeholder is no longer present.

Variables-box (Variable selection) (3): Using the buttons ( ) near the right edge, you can, for example, expand the Variables area. You can also make additional settings such as

- **de-/selecting** specific files or **variables** using the checkbox (☐) , for cases where not all variables belonging to a file are to be loaded. This means that the are present in the list but will not be loaded. They can be activated again at any time.
- **adding more elements**, for the case where the variable only exists once at the command's runtime. Use the "+"-symbol ( ) to add files and variables. Be aware that a file must always contain at least one variable.



Example: a file having two variables

Variables list	Description
Variable name	Name of the file/variable as found in the file.
Target variable name	Name of the variable as it is to be found in the Data Browser following load.

The element-specific configuration is discarded once the file/variable is deleted from the list.

Other options

Category	Description
(Only for the command: Load Variable)	
Use different category	Put a checkmark in the box if you wish to create the variables in a deviating category. An input box for the new category appears. This category is applicable for all variables loaded. • **deactivated:** The variable is displayed **in the variable's category, or without any category** under "no category". • **activated:** The variable is displayed **in the specified category or without any specification under "User-defined variable"**. If you wish to create a category within the category "*User-defined variables*", do the following: write a "\" before the category name. Category: "MeasurementPoint_1". A category called "MeasurementPoint_1" is created in **parallel with** the other categories, such as "*Analog Inputs*". It contains the variable. Category "\MeasurementPoint_1". This creates a category "MeasurementPoint_1" **within the category** "*User-defined variables*". It contains the variable. This applies, for example, to variables created by means of the command: "*Load variable*" or to generating a user-defined variable 1149 by means of the Data Browser.

Target measurement	Description
Current measurement	**Load:** The variable appears in the Data Browser under "*Current measurement*". **Refill:** The variable overwrites the value of any variable existing under "*Current measurement*".
Last completed measurement	**Load:** The variable appears in the measurement which is in the Data Browser and is saved in that measurement's folder. The variable remains always available once the measurement has been loaded.
Measurement number Measurement#<No>	
Fixed measurement name	

Other options	Description
Treat error as warning	If the command is executed even though not all selected variables exist, error messages are posted. In normal cases, the Sequencer is exited when an error message appears. This also causes no variable to be imported. In order that the command always imports all variables, it is possible to activate this option. In this case, no error messages appear in the logbook, but warnings instead. Warnings do not cause the Sequencer to close.
Show dialog	If this option is activated, the settings dialog appears when the command is executed. All settings can be checked and edited. Any changes do not take effect on the configuration of the command in the command list, but only on the import currently performed.
Always overwrite existing variables	**Load:** If this option is activated, variables (and files, if applicable) having the same name in the destination folder will be overwritten without a confirmation prompt. This option is useful for automated routines.
Load/Import all	When this option is activated, then when the command is run, all variables are imported in accordance with the file selection: • **if files are selected**, then all variables belonging to those files are imported at runtime. • **if no file is selected**, then all files and their variables are imported at runtime.

Note Changes to files are not applied

If configuration of the command is completed, the Variable list's configuration is permanent.
This means that if the file is subsequently changed, the changes will be ignored.

Example: A file contains two variables: Var_1 and Var_2.

The command is configured in such a way that the file is selected. When the command is run, both variables are imported.

Later the file is replaced. The two variables still exist in it, but now with an additional variable: Var_3. If the command is executed, only Var_1 and Var_2 are imported. The new variable is not imported.

If you wish to make changes to the files actually applicable, please activate the setting: "*Import/Load all*".

15.11.12.3 Delete variable

The command ***Delete Variable*** enables existing variables to be deleted. When the command is executed, all selected variables are deleted. It is possible to select one or more variables to be deleted. All options can be either specified, or selected manually while the Sequencer is running.

Variables list

To add a variable to the list, first click in the empty box and then

- enter a variable's name or
- click on the button ••• and select an already existing variable from the Data Browser.

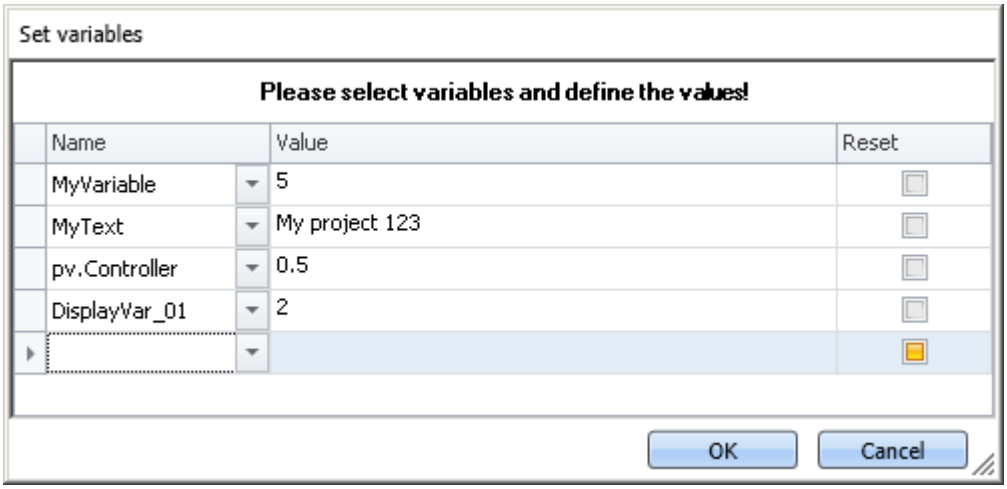
Note Fault condition - Unable to delete variable

If a variable cannot be deleted during running of the command, as is the case with device-variables and system-variables, a pertinent warning is posted. The same applies when a certain variable does not exist. However, all variables which can be deleted are deleted.

Parameter - Other options	Description
Show dialog	If this option is activated, the settings dialog appears when the command is executed. All settings can be checked and edited. Any changes do not take effect on the configuration of the command in the command list, but only on the deletion currently performed.

15.11.12.4 Set variables

The command **Set variables** assigns values to variables.



Command: Set variables

It is possible to set pv-variables, Display variables, virtual bits, or user-defined variables. You can also set triggers to 0 or 1, as well as assign data to whole channels.

To select an **existing** variable, click on ⌄, the Data Browser opens. All variables used in this command must be present at the time the command is run.

In the column "*Value*", you can use placeholders; see the chapter for more Placeholders 166 information.

Note Automatic resolution

As described below, this command automatically resolves a number of expressions. To prevent this automatic resolution, place the assigned value in quotations marks.

It is recommended to write texts in the box *Value* in quotation marks, as a rule, in order to prevent unintended resolution.

The quotation marks are automatically deleted before the assignment, so that they do not appear during subsequent processing and display of the variables.

If you wish to include quotation marks in a variable, use \".

Resolving existing variables

If the name of an existing variable is used in the "*Value*" box, then at runtime it is replaced by its current value. In order to prevent a new name being replaced by a value, place the text in quotation marks. If you wish to resolve a targeted variable within a text which is in quotation marks, please use the following placeholder: <VARS["My_Variable"].VALUE>.

Mathematical operators

In the "*Value*" box, it is possible to use mathematical operators such as +, -, / and *. Thus, you could set the variable `a`, for instance, to the value `a+1`.

Logical operators

In the "*Value*" box, it is possible to use logical operators such as && and ||.

Comparisons

It is possible to perform comparisons in the "*Value*" box such as <, >, <=, >=, ==, !=. In order for the comparison operators to be recognized as comparison operators, there must be numerical values (either a number or a resolved variable) to their right and left.

Example: In order to toggle a variable between the values 1 and 0, you can use either of the two following methods:

```
MyVar == 0
```

or

```
MyVar != 1
```

Reset

If the checkmark: "*Reset*" is set, the variable is set to the initialization value. The value entered in the box "*Value*" is ignored.

User-defined variables have an initialization value which is set when the variable is created.

Device variables have no initialization value. Here, the value is set to "0".

15.12 Tutorial

The following are examples involving the plug-in **Sequencer**. The examples use **Widgets** 1235 from the plug-in **Panel** 1205, which may not come with your installation (depending on the product configuration).

- Starting the Sequencer 1766
- Output box 1769
- Scrolling through the Panel pages 1770
- Dialog 1772
- Use of imc FAMOS 1777
- More than one command sequence 1780
- Conditional branches 1783

- Example: Wind power 1786

15.12.1 Starting the Sequencer

Assignment:

Start the Sequencer via a button on a Panel page. By means of the Sequencer, a 10-second measurement is to start. The measurement is to be displayed in a curve window and the computer announces the measurement's conclusion via the sound card.

Procedure:

To begin, the measurement device is selected and configured.

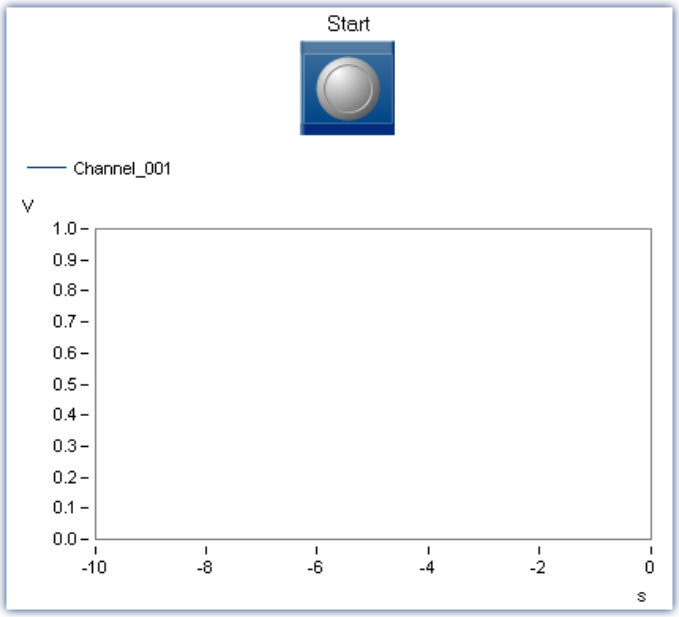
- Go to the **tab**: "*Devices*" .
- If the device is not yet in the list, run a device search.
- Select the measurement device desired and set one channel to active with a sampling duration of 10 s.
- To **apply the changes**, click on the button "*Process configuration* 241" ( ).

Ribbon	View
Home > Process ( )	all
Setup-Control > Process ( )	Complete

By this means, the device settings are available to other components.

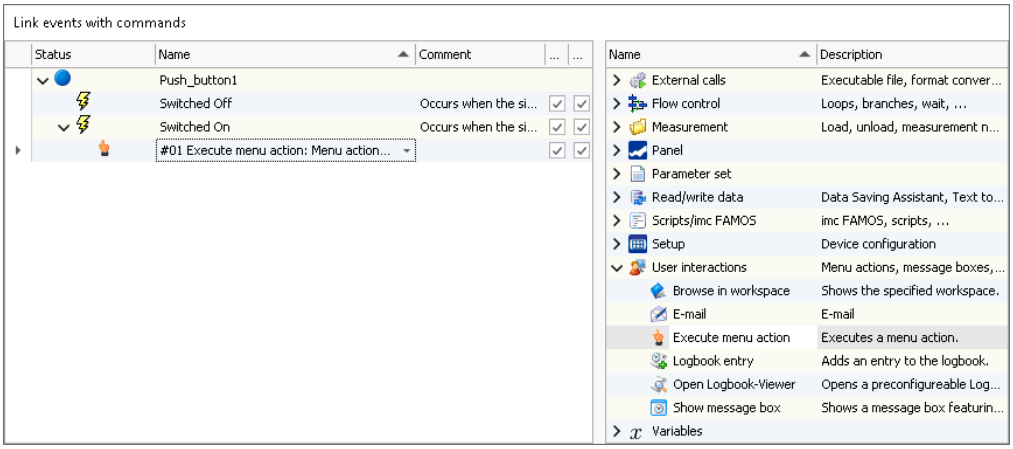
Next, the Panel page is filled.

- Open the page: "*Panel*".
- Drag the activated channel from the Data Browser to the Panel page and have it displayed as a curve window.
- Create a push-button on the page; e.g. "*Switches/buttons*" > *Push button*". Ensure that the "*Switching behavior*" is set to "*Push-button*".
- In the Properties, give the push-button the label "*Start*".



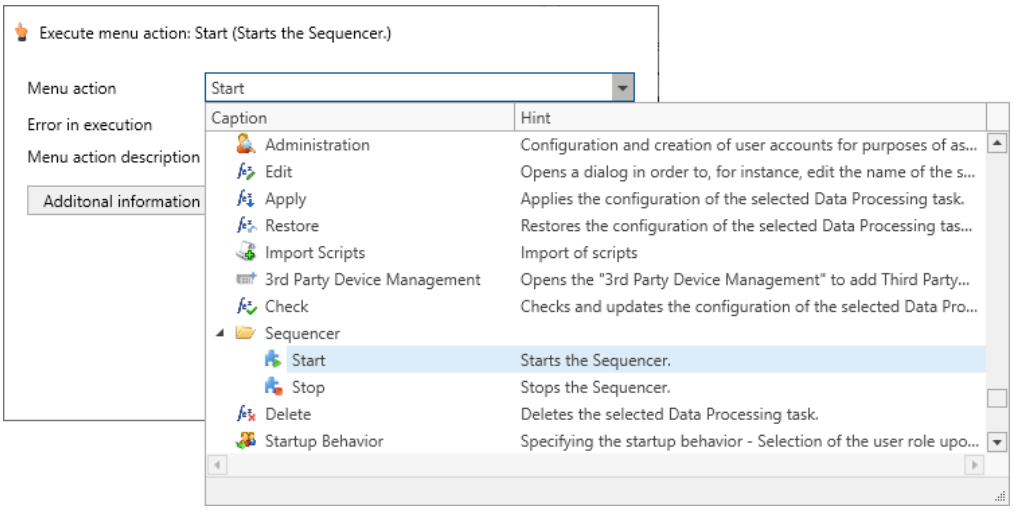
Curve window with push-button

- Open the push-button's property "*Events*" and use drag&drop to move the command "*Execute menu action*" from the group "*User interactions*" into the Sequences table below the property "*Switched On*".



The push-button's events

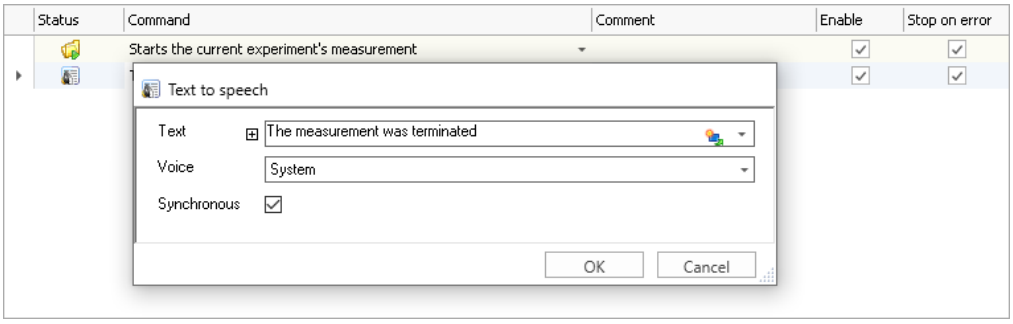
- Click on the dropdown arrow at the right of the command inserted.
- Type in the entry "*Start*" in the "*Menu action*" combobox and select the suggested command "*Start - Starts the Sequencer*". Click on "*OK*" and close the events window and properties window.



Execute menu action - Starts the Sequencer

Sequencer-procedure

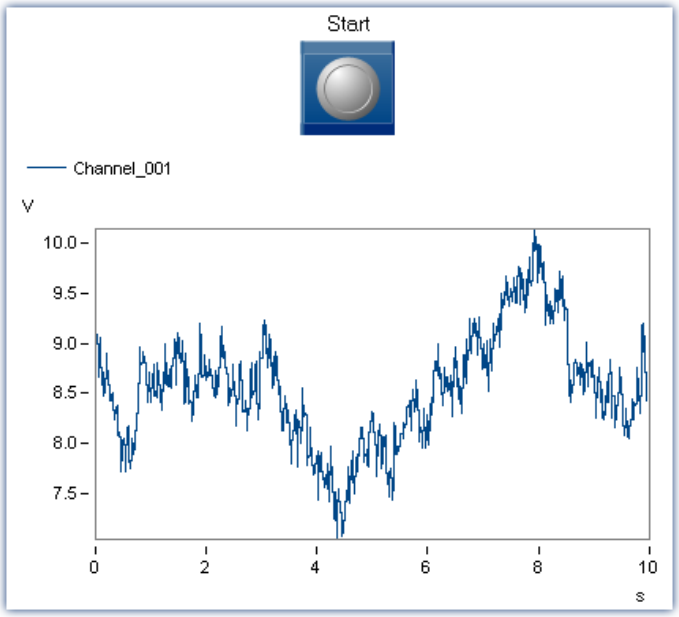
- Go to the Sequencer
- Use Drag&Drop to insert the command "*Open Experiment/Start Measurement*" into the sequence table. The default setting for the command starts the currently open experiment.
- Drag the command Text to Speech from the group Read/write data to the sequence table as well, placing it below the existing command.
- Configure the text-to-speech command as follows:
  - Text: "The measurement was terminated".
  - Voice: "System"



The computer announces the measurement's conclusion via the sound card

Starting via the button

- Check the volume level in Windows.
- Go back to the Panel page and switch "*Design Mode*" off, if necessary.
- Click on the "*Start*"-button on the Panel page:
  - The measurement starts.
  - After 10 seconds, the voice output begins.



The measurement is concluded

Optional: Instead of voice output, it is possible to play a sound by using the command "*Play soundfile*" belonging to the group Read/write data. With this configuration, a sound file can be selected from the folder "*c:\Windows\Media*".

Result:

The measurement can now be started by means of the push-button on the Panel page. By means of the Sequencer and the voice output command, you can notify the user when the measurement ends.

Remark:

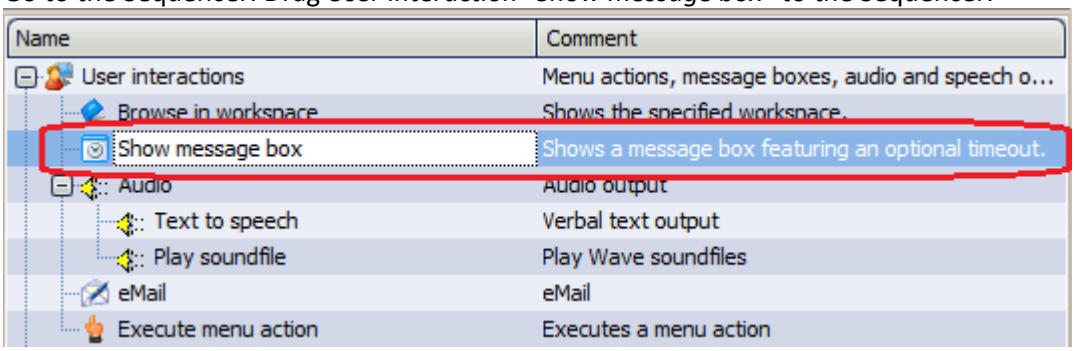
It is additionally possible to start the measurement independently of the Sequencer, as before. In this case, the Sequencer does not start along with it.

15.12.2 Output box

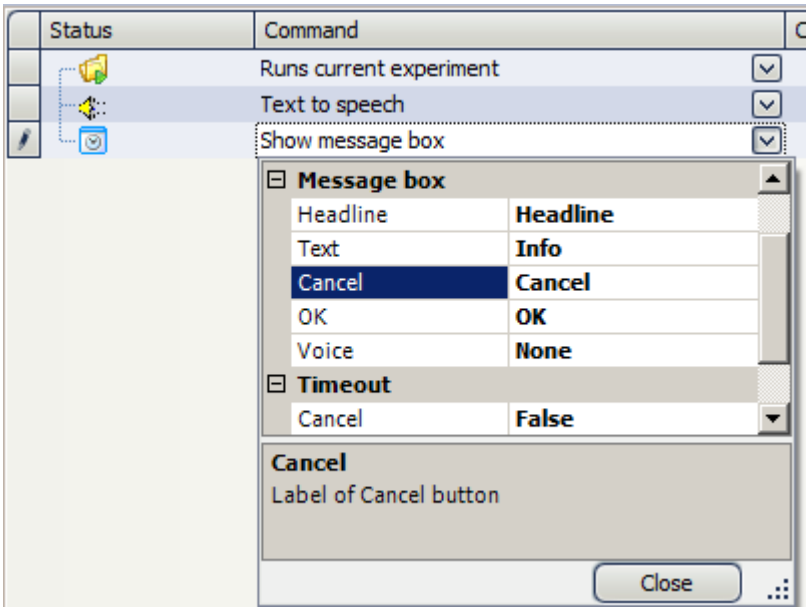
The preceding exercise is expanded to include a message box at the end of the measurement. Subsequently, an additional message box with sound output is displayed.

Assignment: Create a message to be displayed at the end of the measurement. Also, have the text spoken by voice output.

1. Load the experiment "Start Sequencer" if it is not open yet and **save** the experiment under the name "**Output box**".
2. Go to the Sequencer. Drag *User interaction "Show message box"* to the Sequencer.



3. Enter in the properties:



4. **Save** the changes made under the experiment name "**Output box**"
5. Go back to the Panel and switch off the Design mode. Make sure that the sound volume is set to high. Click on the Start button you previously made.

Result: At the end of the measurement, an output box appears, whose text is additionally outputted acoustically.

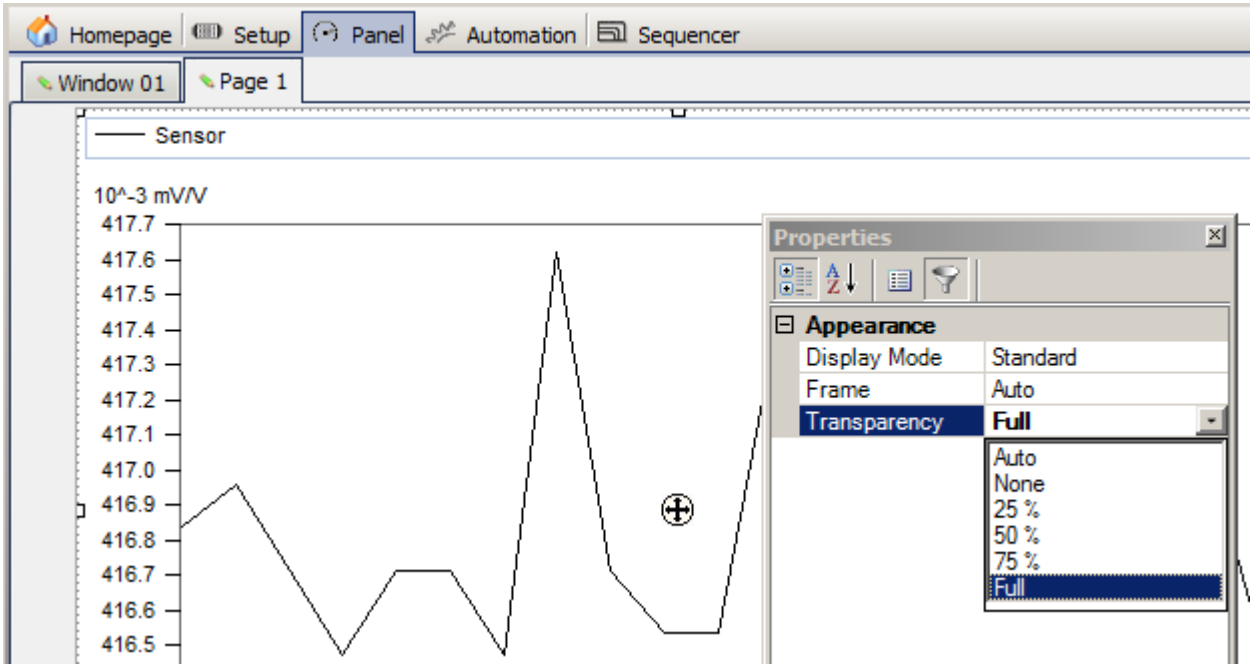
Remark: The voice output can be switched off by the entry Voice: none.

15.12.3 Scrolling through the Panel pages

A new Panel page is create. At the end of the measurement, the system automatically goes to this Panel page.

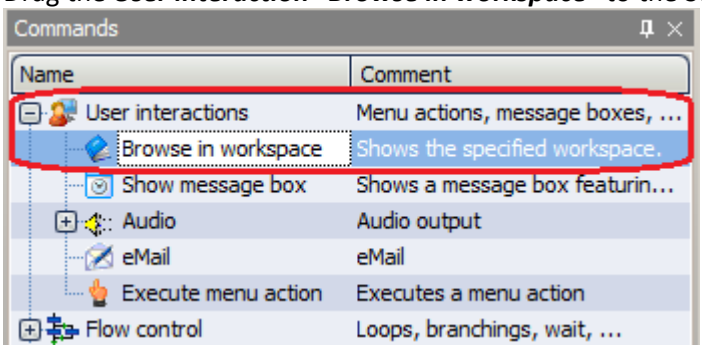
Assignment: Change the example "Start Sequencer" so that at the end of the measurement, a second Panel page is displayed instead of the sound output.

1. Load the experiment "**Start Sequencer**" and **save** the experiment under the name "Scroll".
2. Go to the Panel. Create a **new Page** as a ***Report (Portrait)***. To do this, right-click the mouse over the Panel and select "*New Page*". Name it "*Report".*
3. Create a **Standard curve window** and set the property *Transparency* to *Full*.

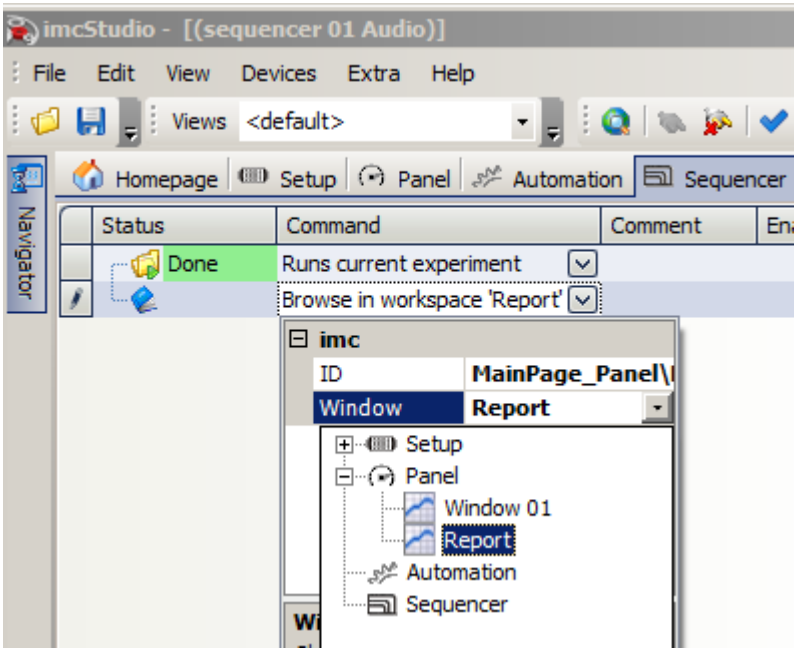


4. Go to the Sequencer. **Delete all Sequencer steps** except the first step, which starts the measurements. To do this, highlight the steps and right-click the mouse to open the context menu. Select the context menu item *X Delete*.

5. Drag the ***User interaction*** "***Browse in workspace***" to the Sequencer steps.



6. As the **Target**, select the new Panel page **"Report"**.



7. **Save** the changes made under the experiment name "**Scroll**".
8. Go to the Panel and switch off the Design mode. Click on the Start button you previously created.

Result: At the end of the measurement, imc Studio automatically pages through to the Report page.

Remark: Using the command *Browse in workspace*, you can scroll to any pages which can also be reached manually, e.g. even the analog channel settings on the Setup page.

15.12.4 Dialog

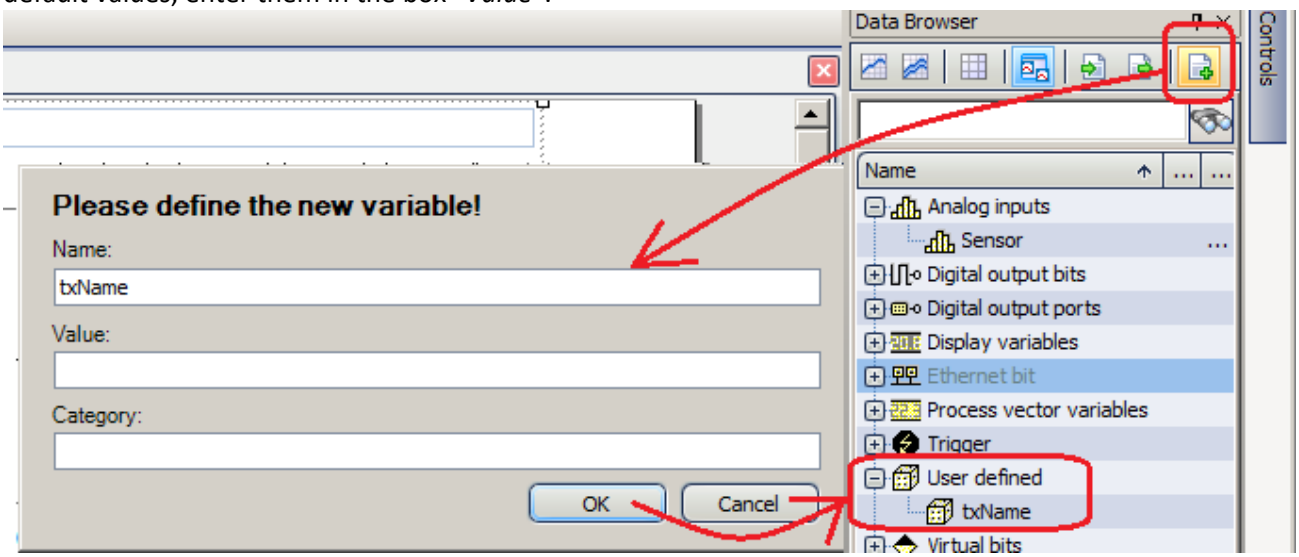
An additional Panel page is displayed as a dialog.

Assignment: Expand the experiment "Scroll". Create a dialog by entering your name and test object number before the measurement.

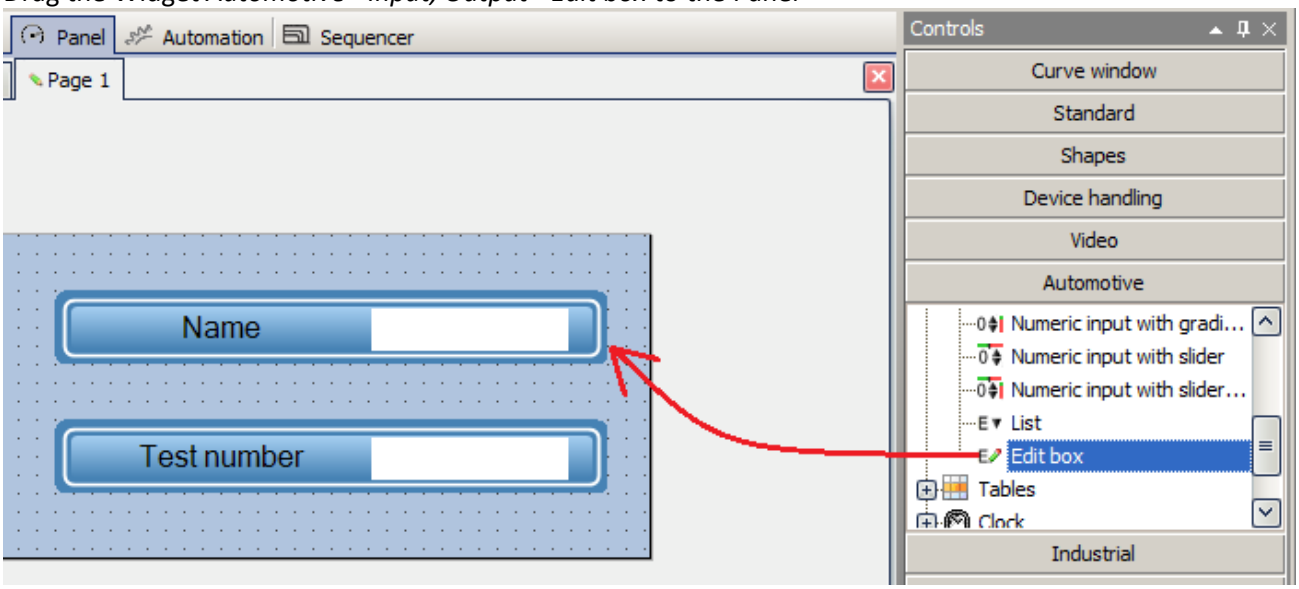
1. Load the experiment "**Scroll**" if it isn't open yet and **save** the experiment under the name "Dialog"

2. Go to the Panel and create a **new page as *Standard Dialog***. Name the dialog *"Input"*

3. In the Data Browser, add the *user-defined* variables TxName and TxTestObject. If you wish to provide default values, enter them in the box "*Value*".



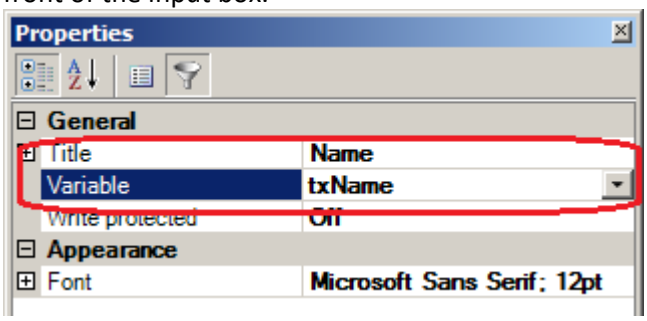
4. Drag the Widget *Automotive - Input, Output - Edit box* to the Panel



Input dialog with two text boxes

5. Set the Widget's properties to :
   Title= Name
   Variable= txName (this is under the 3rd variable created)

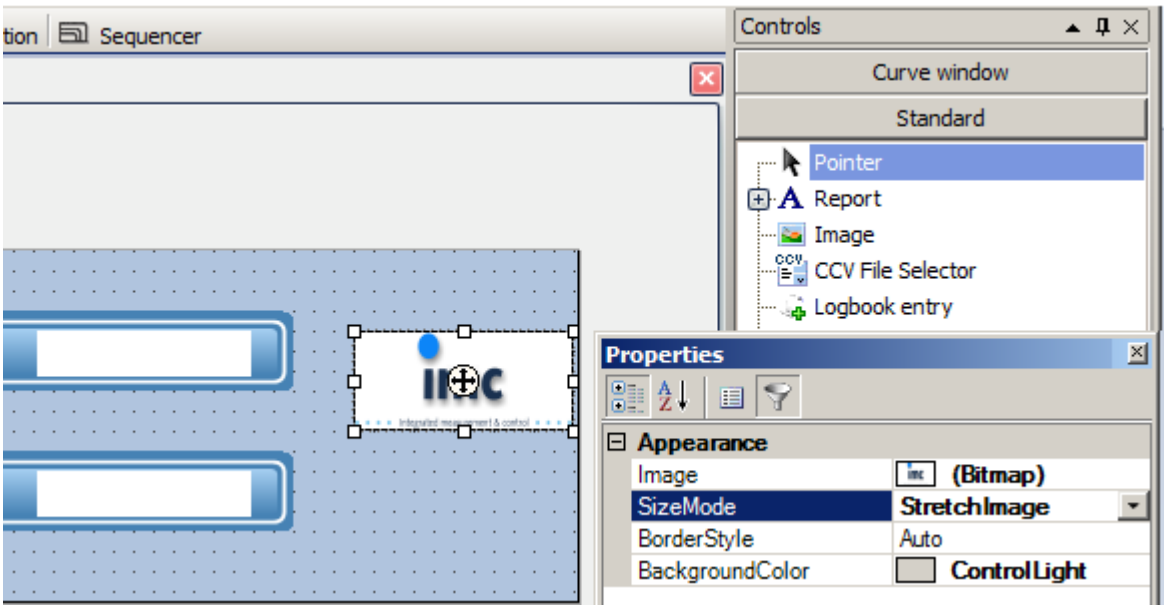
   Click on the small triangle next to the title entry and select the left arrow. Then the name will appear in front of the input box.



6. Repeat the last step with
   Title= Test Object
   Variable= txTestObject

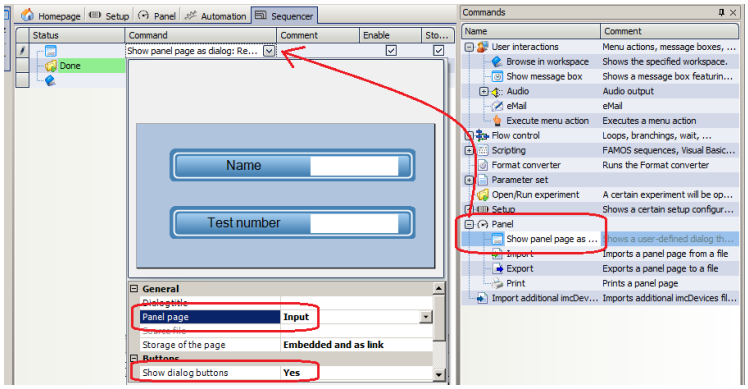
7. Make the Panel page smaller using the sizing handle at the lower right corner.

8. Drag the Widget *Image* from the group *Standard* to the dialog. Select the imc logo in the property *Image* and set the property *SizeMode* to *Stretch Image*.

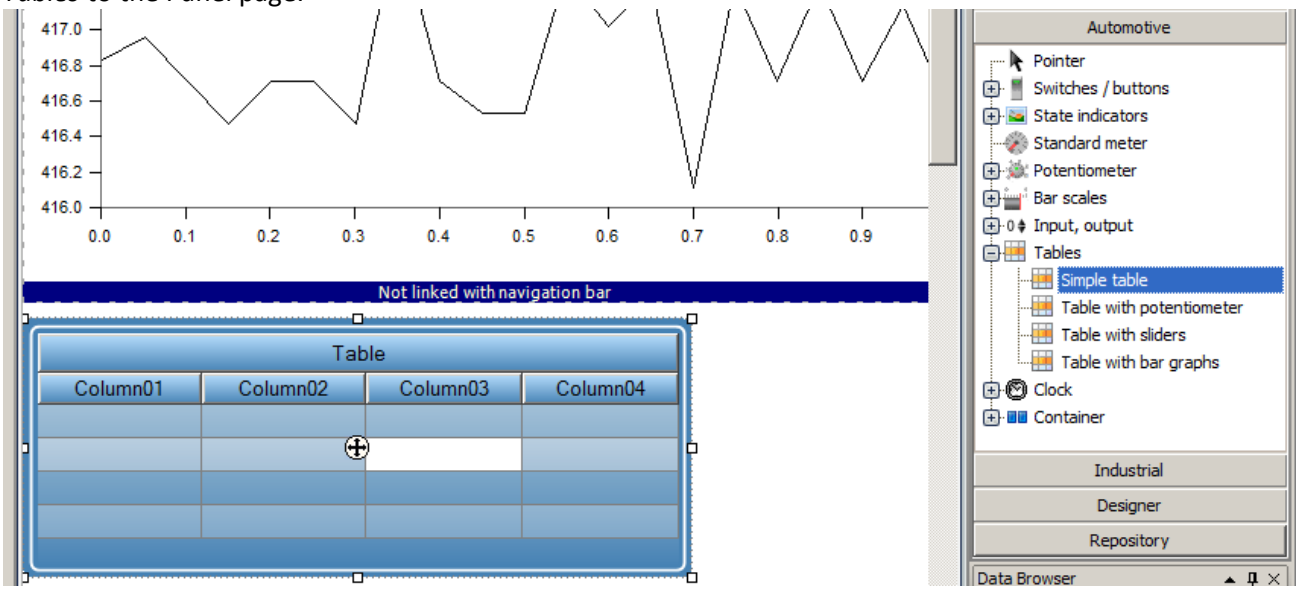


9. Go to the Sequencer

10. In the next step, the Panel page "Input" is entered into *Show panel page as dialog*. Select the Panel page "Input" as the *Panel page*. Activate the setting *Show dialog buttons* by entering *Yes*. Then in the branch *OK-button* activate the property V*isible* by entering *Yes*.

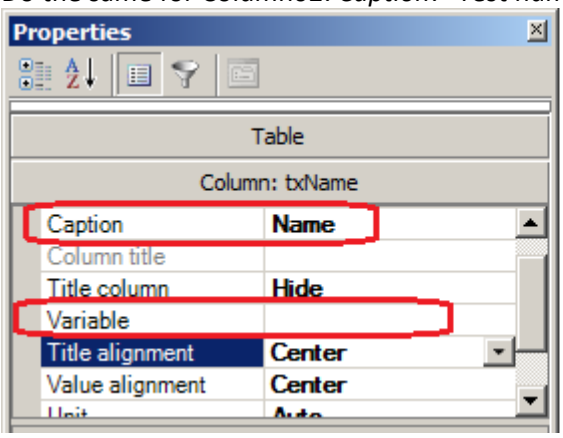


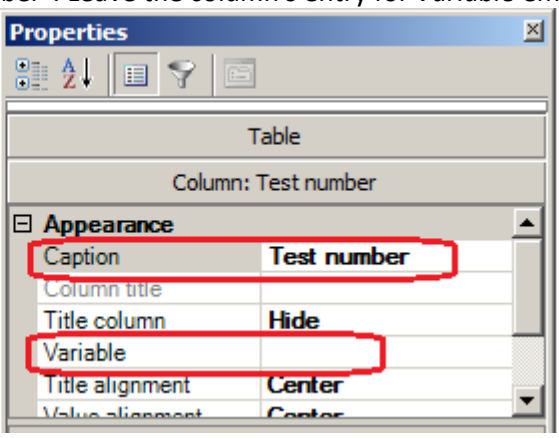
11. Go to the Panel page "Report". From the group *Automotive*, drag the table object *Table* from the branch *Tables* to the Panel page.

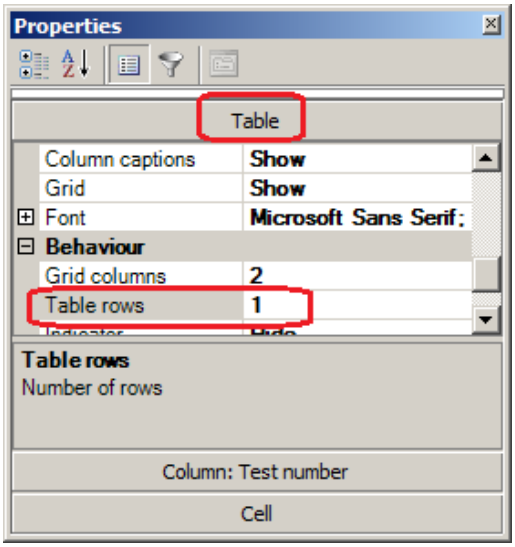


12. Remove *Column03* and *Column04* by pulling them down.
    From the Data-Browser, drag the user-defined variable txName to the first cell in the column (not the column header). Drag the variable txTestObject to the cell in the second column.

    Open the *properties* and click on the title cell Column01. Select as *Caption* 'Name'.
    Do the same for Column02: *Caption*: "Test number". Leave the column's entry for *Variable* empty.







Set the number of *Table rows* to 1. This entry is visible when you select *Table* in the Properties.

13. Remove the table's title cell by clicking on the cell *Table*. In the Properties, expand the branch *Title* ad select for *Position* "-".
    Make the table's size small enough that only the column header and the input box are visible.
    Adapt the table's colors to your wishes. The best way to get white is by selecting *Background color: Transparent*, visible only, if the properties of *Table* has been selected.

Name	Test number
	123456

14. **Save** the changes made under the experiment name "**Dialog**".
15. Go back to the Panel and turn off the Design mode. Click on the Start button which you had made.

Result: Before the beginning of the measurement, an input template appears for extra information displayed at the end of the Report page.

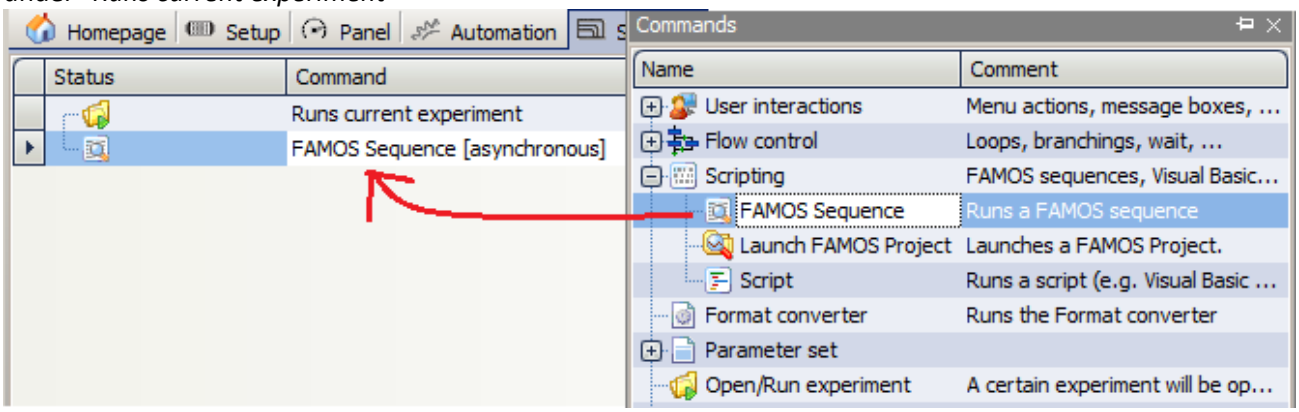
15.12.5 Use of imc FAMOS

Following the measurement, the data are automatically transferred to imc FAMOS for statistical analysis. The results are displayed on a Panel page and saved together with the measured data.

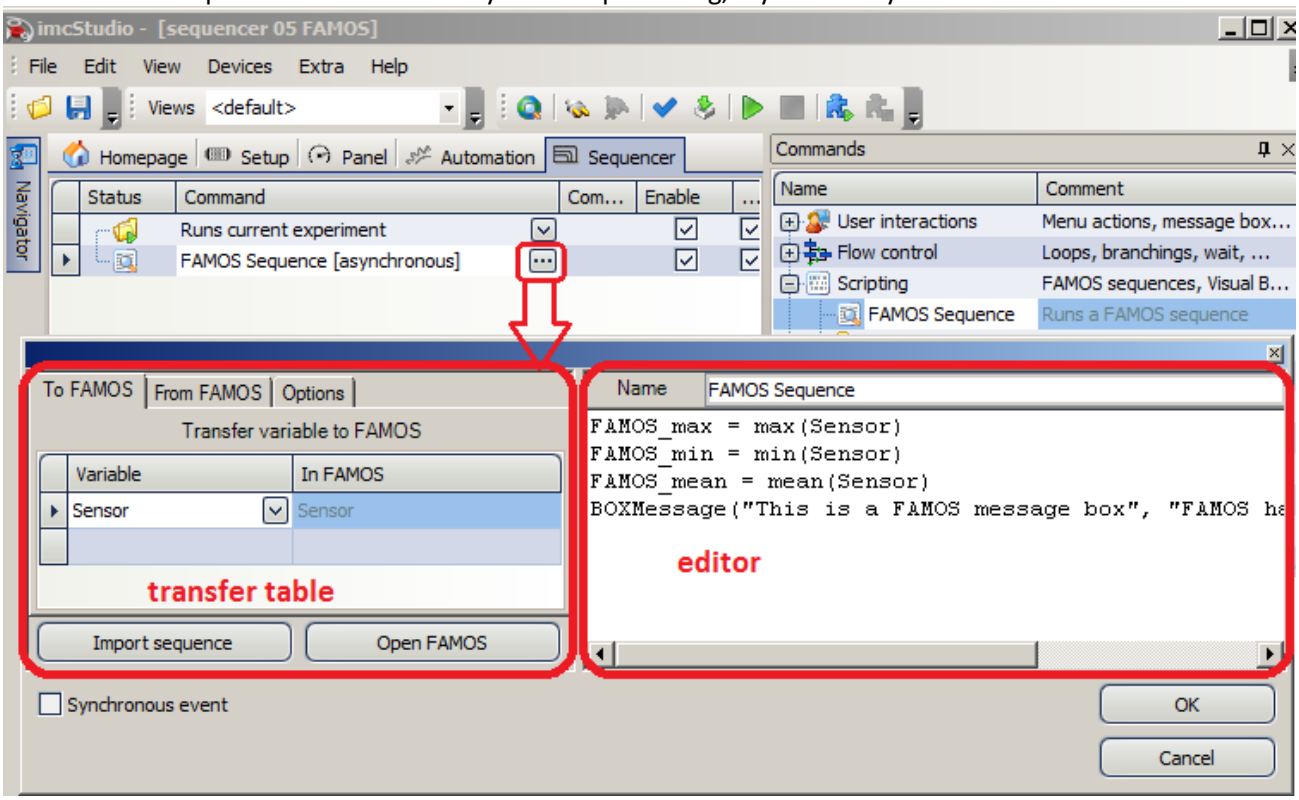
ATTENTION: For this exercise, an installed version of imc FAMOS is required.

Assignment: The previously created experiment "Start Sequencer" is to be enhanced with an automated analysis by imc FAMOS. imc FAMOS determines the maximum, minimum and mean values. The results are to be displayed in the report. The report is to be printed out at the push of one button.

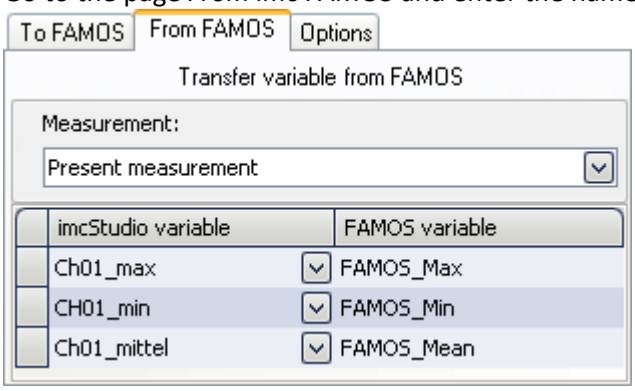
1. Load the experiment "**Start Sequencer**" if it isn't open already and save it under the name "**FAMOS_Analysis**".
2. Go to the Sequencer page and select from the commands *Scripting - imc FAMOS Sequence*, and drag it to under "*Runs current experiment*"



3. Click on the symbol with the three periods "...". An input dialog for the imc FAMOS sequence appears. You can enter the sequence to be run directly in the input dialog, if you already know the functions.

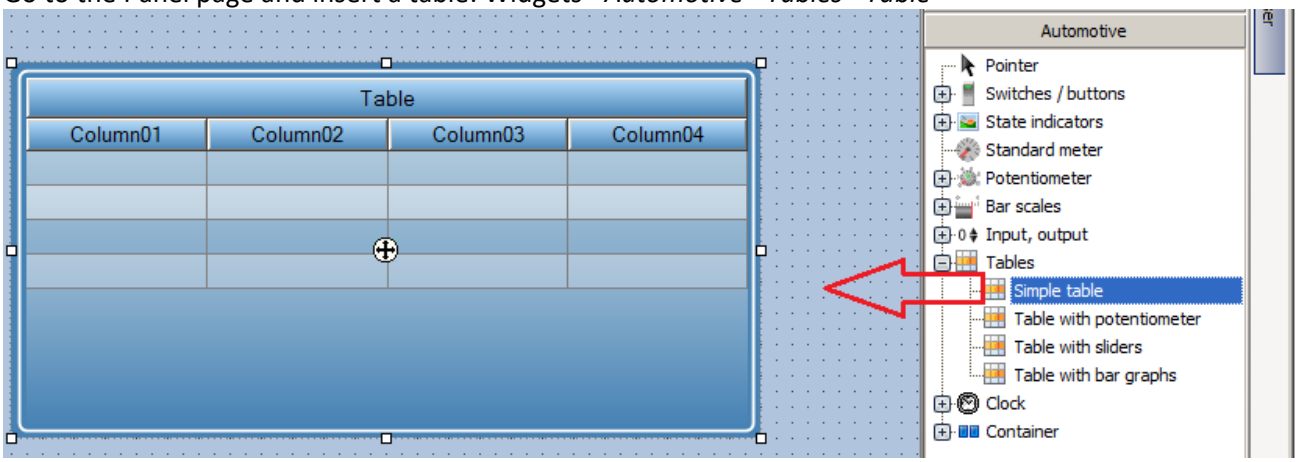


4. Enter the command lines as shown in the picture. As of imc FAMOS 6.1, the language in which the function names are expressed no longer matters.
```
FAMOSMax = Max(Sensor)
FAMOSMin = Min(Sensor)
FAMOSMean = Mean(Sensor)
BOXMessage("This is a imc FAMOS Box","imc FAMOS has done calculation.","!1")
```
5. Transfer to and from imc FAMOS is accomplished using the tables at left. On the page *To imc FAMOS* , enter the channel *Sensor* for Variable and give it the same name under imc FAMOS.
Go to the page *From imc FAMOS* and enter the names of the variables:

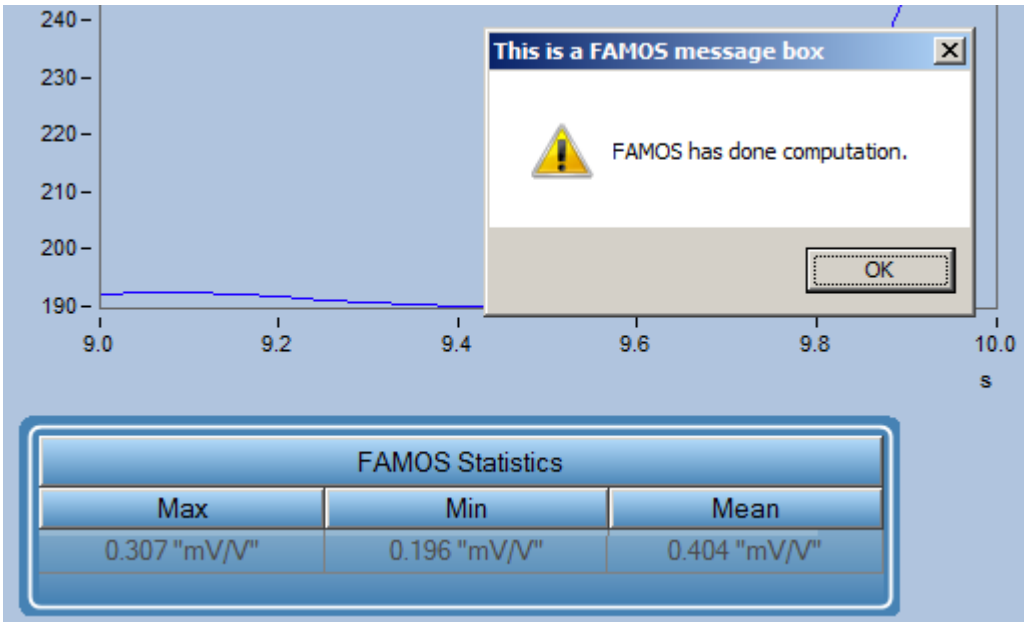


These variable name were already used in the Results table. After the calculations by imc FAMOS, they are automatically applied.

6. Go to the Panel page and insert a table: Widgets - *Automotive - Tables - Table*



7. Delete Column04 by dragging it downward. Set the property *Table rows* to 1.
8. Click in the gray property Table and change the title column "*Table*" to "*Results*".
9. Drag&drop the results from the Data browser\ branch imc FAMOS into the table.
10. **Save** the changes made under the experiment name "**FAMOS_Evaluation**".
11. Go back to the Panel and deactivate the Design mode. Click on the Start button you earlier created.



Result: At the end of the measurement, an output box appears which was created using imc FAMOS. The results are subsequently displayed in the table.

Remark: Instead of simple using the Sequence Editor (see point 8), you can start imc FAMOS by means of the button "Open imc FAMOS" and there create the sequence using the imc FAMOStools. When you close it, the sequence is adopted.

15.12.6 More than one command sequence

Learning goal:

Depending on the respective current status of the overall system, different command sequences are to be performed in the Sequencer. Since there is only one routine, it is subdivided into individual subroutines by means of the "*Switch*"-command.

Example from practice:

By means of imc Online FAMOS or of the "*Events*", the state of a motor control system and the gear currently engaged are to be detected. A certain variable reflects the system state and the Sequencer is used to start the appropriate command sequence in response to that variable's value.

For instance, one could use imc Online FAMOS to monitor the RPMs. If this value remains stably within some range for a few seconds, then a particular command sequence can be started by means of the Sequencer, depending on the respective RPM value.

Assignment:

There is a variable "*Status*" which we will use to simulate the example described above. This variable has four states: "*Initialization*", "*Conclusion*", "*Rotation stable at 1000 rpm*" and "*Rotation stable at 2000 rpm*".

When one of the states is entered, the associated command sequence is started in the Sequencer.

Implementation

Create a user-defined variable with the name "*Status*"

Preparing Sequencer

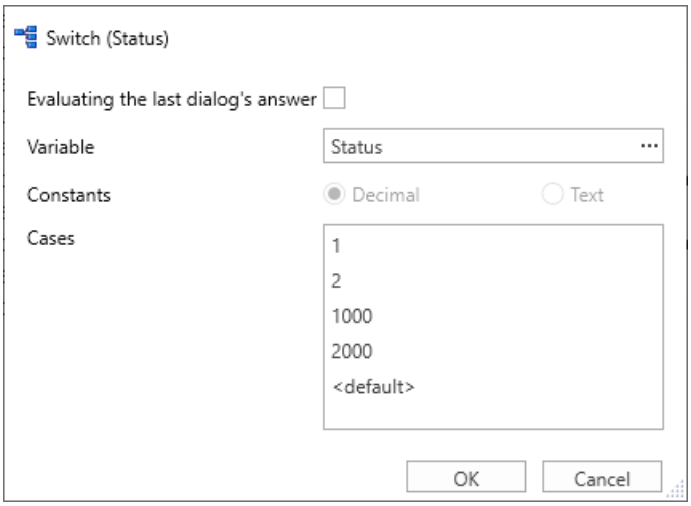
Open the Sequencer

Insert the command: "*Switch*"

Select the variable "*Status*". The individual cases are selected on the basis of this variable's value.

In the box "*Cases*", open the context menu and create for more cases:

- 1: click Enter
- 2: click Enter
- 1000: click Enter
- 2000: click Enter

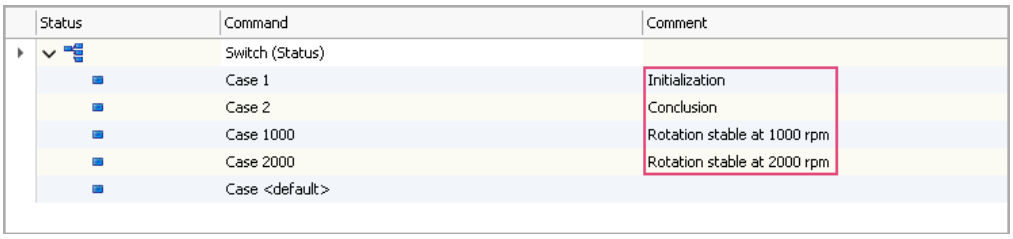


Four created cases + the default case

Close the configuration.

Use the column "*Comment*" to enter unique descriptions for the various cases in this application:

- 1: Initialization
- 2: Conclusion
- 1000: Rotation stable at 1000 rpm
- 2000: Rotation stable at 2000 rpm



The cases in the Sequence-table with unique names in the Comments column

Preparing the events

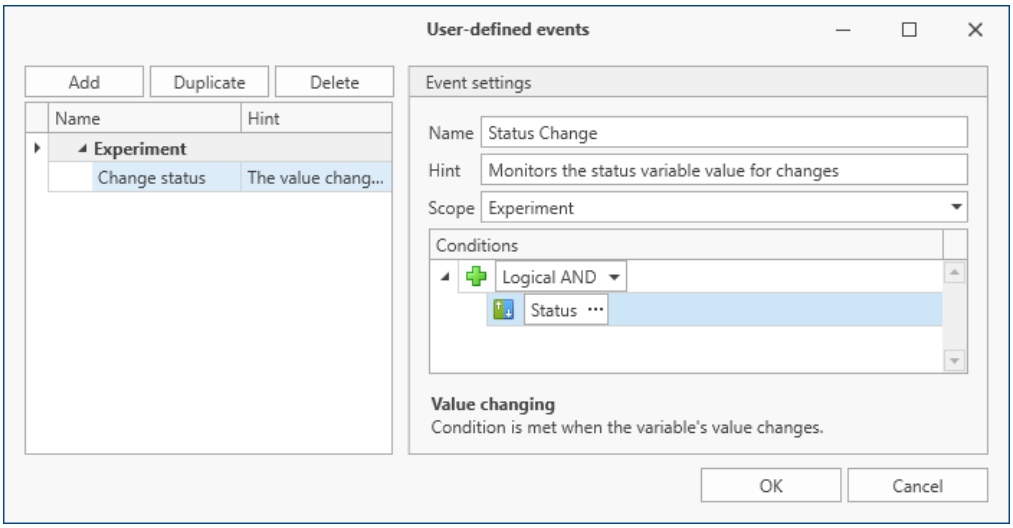
Each time when the value of the variable "*Status*" changes, the Sequencer is to start once and run the associated Case-sequence.

For this purpose, define a "*User-defined event*" by means of the button of that name at the bottom with the "*Events*".

In the dialog, click on "*Add*".

- Enter a name for the new event: "*Status Change*"
- Hint: "*Monitors the status variable value for changes*"
- Scope: "*Experiment*"

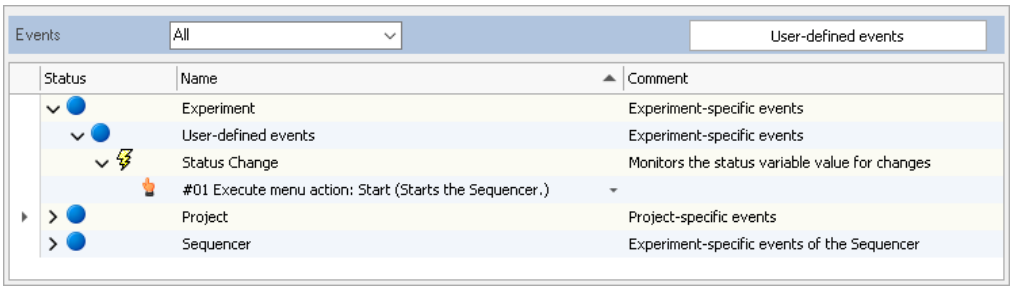
Click on the + -symbol and select the condition"*Value changing*". Select the variable to monitor: "*Status*".



Configuration of the variable monitoring

Next, close the dialog.

In the Events-list, a new entry appears under "*Experiment*". Add the command "*Execute menu action*" to the new event and select the action "*Start (Starts the Sequencer)*".



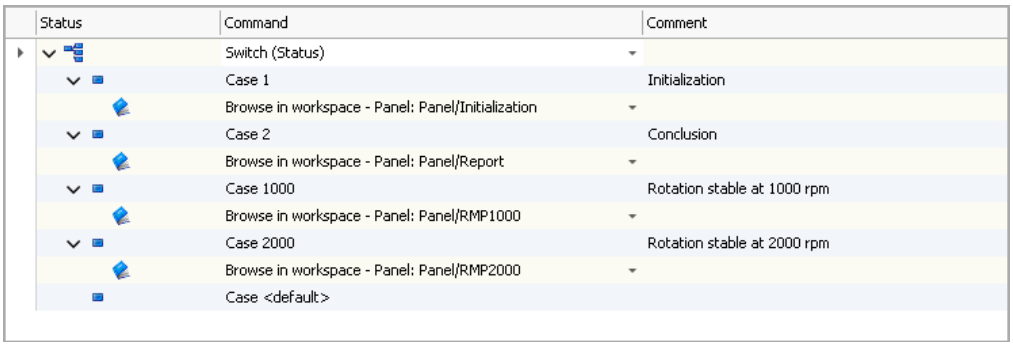
Preparing Panel

In the very beginning, an appropriate Panel page is to open for the values 1, 2, 1000 and 2000.

To do this, create three dialog pages and one report page. Give these the names : "*Initialization*", "*RPM1000*", "*RMP2000*" and "*Report*" for the report page.

Conclusion of the Sequencer

In the Sequencer, place the command "*Browse in workspace*" in each of the four "*Case*"-segments and in each one select the matching Panel page.



Opens the Panel page as soon as a change to Status is made

Testing the routine

Next, perform a test run:

Double-click in the Data Browser (Panel) on the variable "*Status*". The "*Current value*"-window opens.

- Change the value to "2" → The Panel jumps to the "*Report*"-page
- Change the value to "1000" → The Panel jumps to the "*RPM1000*"-page
- Change the value to "2000" → The Panel jumps to the "*RPM2000*"-page
- Change the value to "1" → The Panel jumps to the "*Initialization*"-page
- Change the value to "100" → Nothing happens

Conclusion

You have now constructed a little example in which the Sequencer was restructured into multiple small sequences.

One variable determines which sequence is to start and the event-monitoring starts the Sequencer upon any changes to the value. The Switch-command starts the associated sequence.

You are now able to enhance the individual sequences as desired and modify them for your own purposes.

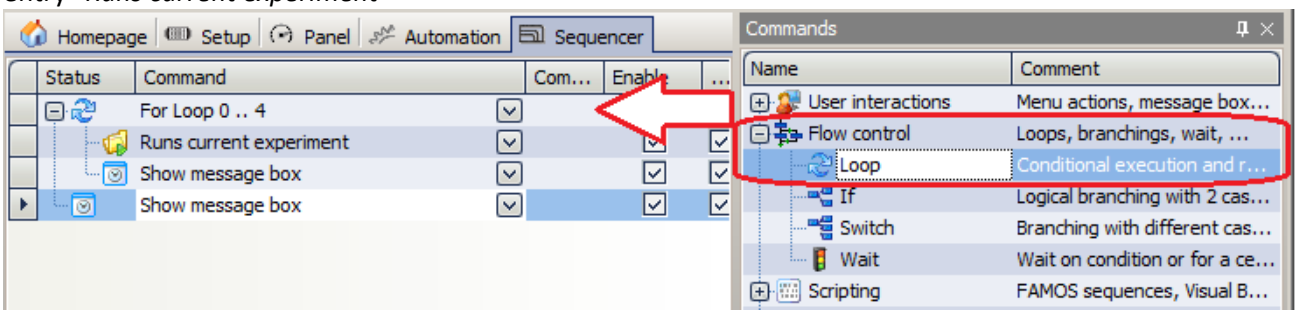
15.12.7 Conditional branches

This exercise shows how the conditional branches (For, While, Switch, If, Else etc.) are used in the Sequencer.

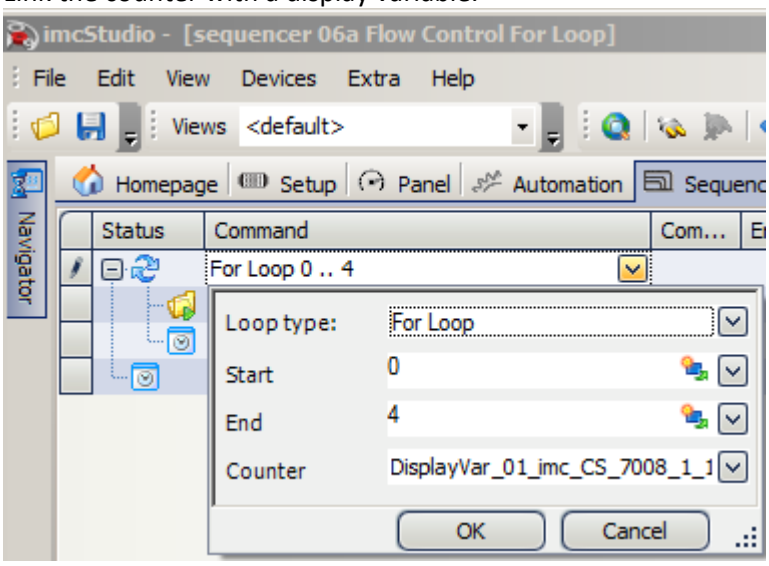
15.12.7.1 For loop

Assignment: Run the current experiment 4 times in succession. A Seven-segment-displaydisplays the number of runs on a Panel page.

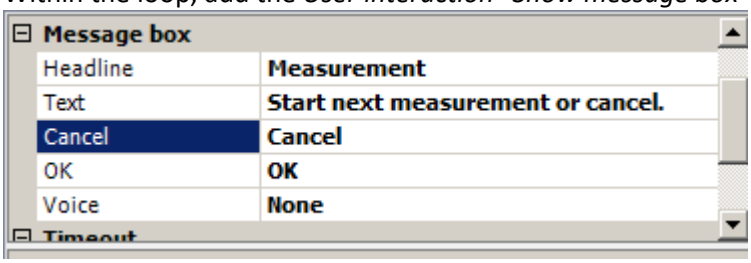
1. Load the experiment "**Start Sequencer**" and save it under the name "**Sequencer_For**".
2. Reduce the channel's measurement duration to 3s.
3. Go to the Sequencer and drag from the command group *Flow control* the command *Loop* to before the entry "*Runs current experiment*"



4. Right-click the mouse next to the *For Loop* entry and make specifications for the loop stages 0 through 4. Link the counter with a display variable.



5. Within the loop, add the *User interaction* "*Show message box*". Make the following entries:



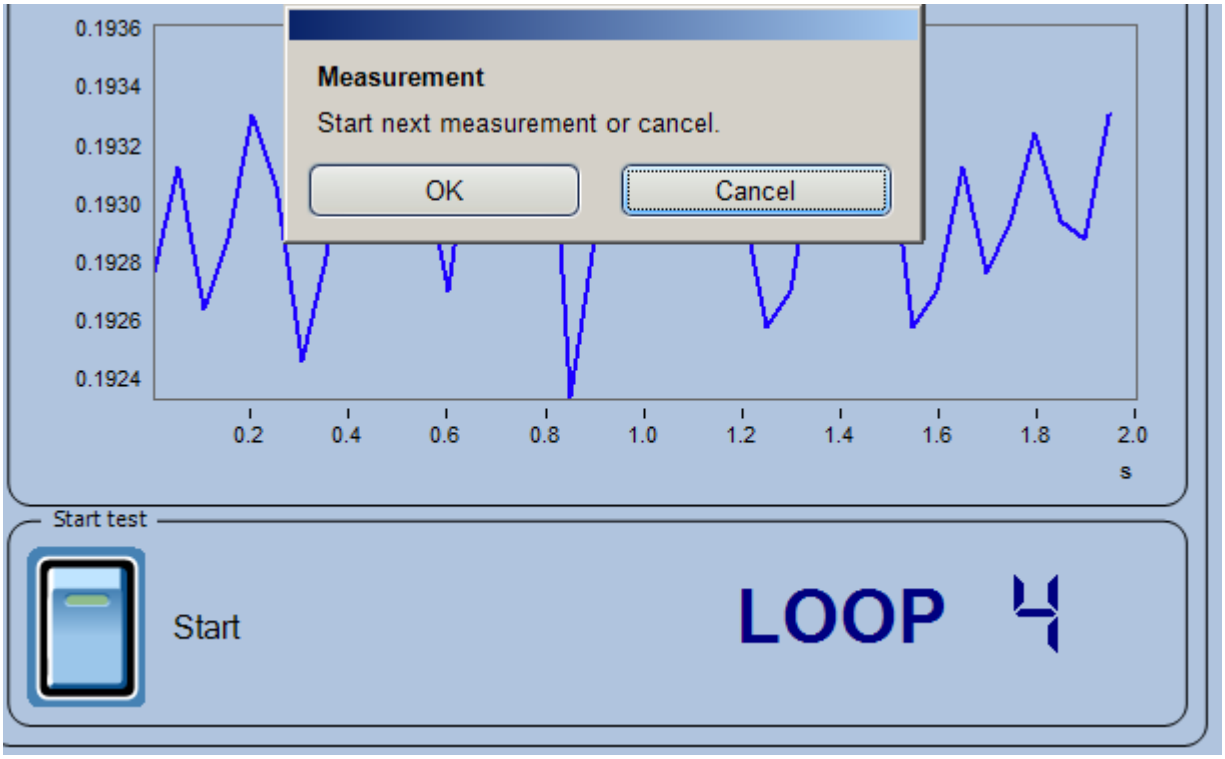
6. Add another *User interaction* "*Show message box*", but after the loop

Status	Command
	For Loop 0 .. 4
	Runs current experiment
	Show message box
	Show message box

with the following contents:

Common	
Name	**Show message box**
Message box	
Headline	**End**
Text	**Sequencer ends here**
Cancel	**Cancel**
OK	**OK**
Voice	**None**
Timeout	
Cancel	**False**
Interval	**0**

7. Go to the Panel and place a *Seven-segment-display* there from Widgets - *Automotive* - *Status display*. Link it to the Display variable previously linked in the *For Loop*.
8. **Save** the changes you made under the experiment name "**Sequencer_For"**.
9. Go back to the Panel and switch off the Design mode. Click on the Start button you made.



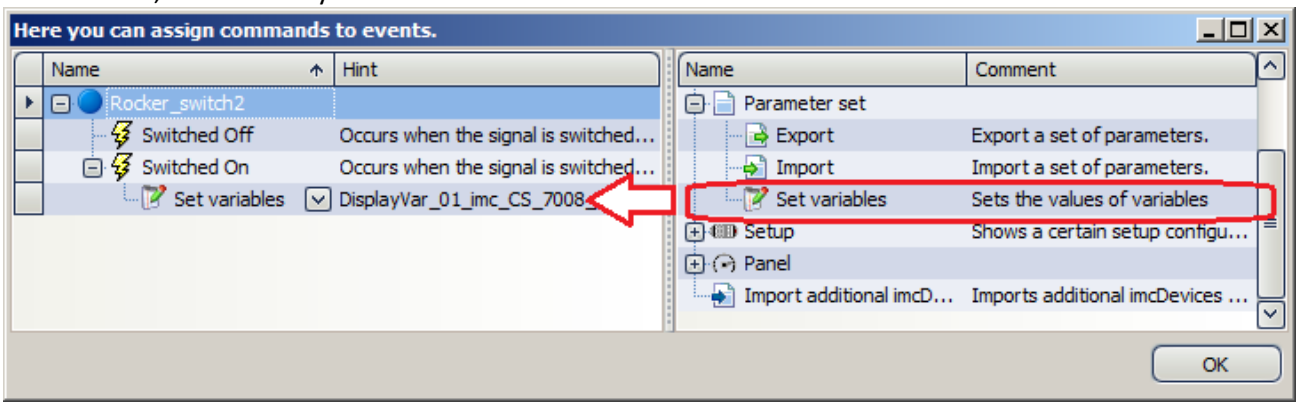
Result: A message appears after each run. The seven-segment-display indicates how many runs have been performed.

15.12.7.2 While loop

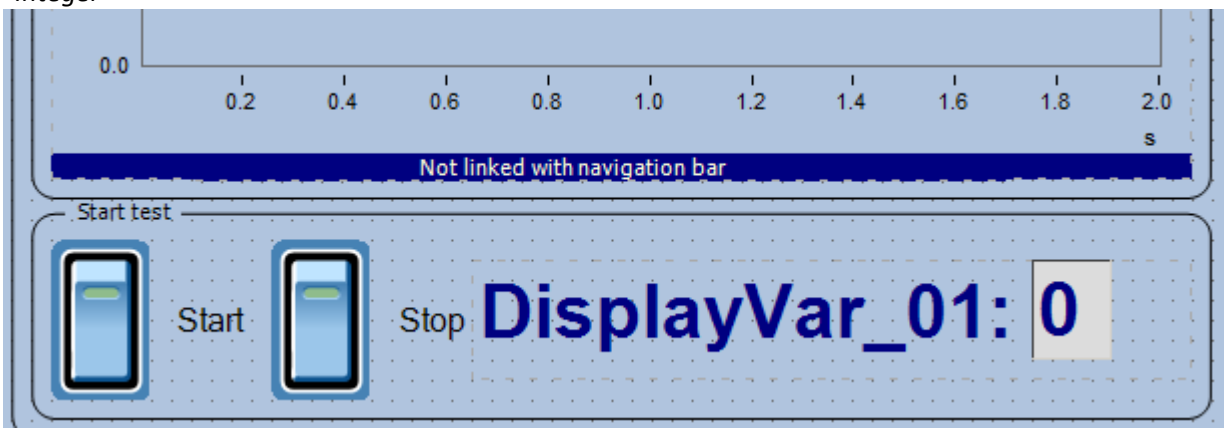
Assignment: Run the current experiment repeatedly until a Display variable contains a nonzero value. Use a pushbutton which assigns a value of 1 to the Display variable when pressed. The Display variable value is displayed on a Panel page.

Based on the experiment "Sequencer_For", an experiment "Sequencer_While" is created.

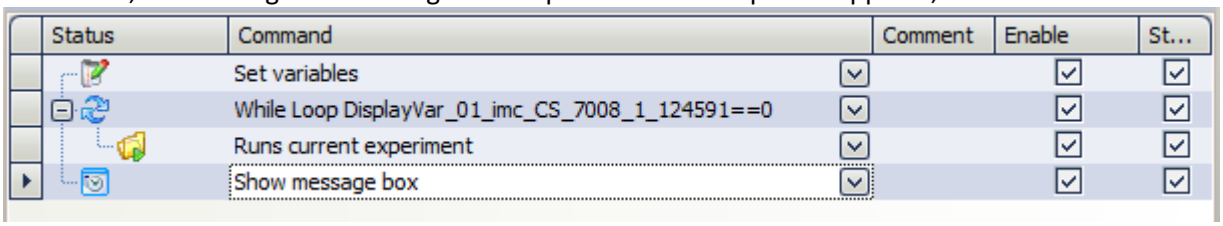
1. Load the experiment "**Sequencer_For**" and save it under the name "**Sequencer_While**".
2. Go to the Panel and create a pushbutton. Give it the caption "Stop". Select "Events" and in the branch Switched On, add the entry Set Variables from the branch Parameter set.



3. Create an input box (*Standard - Report - Editbox*) and link it to a Display variable. Select as the format "*Integer*"



4. Go to the Sequencer and drag from the command group *Flow control* the command *Loop* to in front of the entry "*Runs current experiment*". In the *Properties*, switch the *Loop type* to *While Loop*. For the condition, enter "DisplayVar_01==0". The Display variable name may be supplemented with the device name.
5. So that the loop isn't interrupted at the beginning, you must ensure that the Display variables are set to zero.
   This is accomplished using the entry *Set variables* from the branch *Parameter set.* Insert this and there, set the Display variable = 0
6. The experiment is called within the loop.
7. At the end, the message announcing the completion of the sequence appears, as before.



8. **Save** the changes made under the experiment name "**Sequencer_While**".
9. Go back to the Panel and switch off the Design mode. Click on the Start button you created earlier. Check whether the measurement starts multiple times and the click on the Stop button.

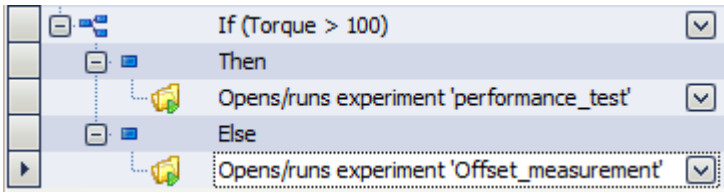
Result: When you click on the Stop button, the Display variable is set to 1 and the measurement does not start again.

15.12.7.3 If - Else

Assignment: The exercise Sequencer_While 1785 is expanded. At the end of every measurement, the system checks whether the **mean value is positive or negative**. The result is indicated in a text message.

You can enhance the sequence yourself. To do this, you need the knowledge presented in the exercise Use of imc FAMOS 1777.

One solution could appear as shown below:



15.12.7.4 Switch-Case

Assignment: The exercise Sequencer_While 1785 is enhanced. At the end of each measurement, the mean value is calculated. Depending on this statistic, one of four different messages is to be displayed.

To do this, use the Switch Case operator. For the lowest value, the message "All OK" appears. For the next higher value, the message "Pressure slightly elevated", then "Pressure too high" and finally "Danger: Pressure Out Of Control"

You can enhance the sequence yourself. To do this, you need the knowledge presented in the exercise Use of imc FAMOS 1777.

Reference More Tutorial

See "*More than one sequence* 1780"

15.12.8 Example: Wind power

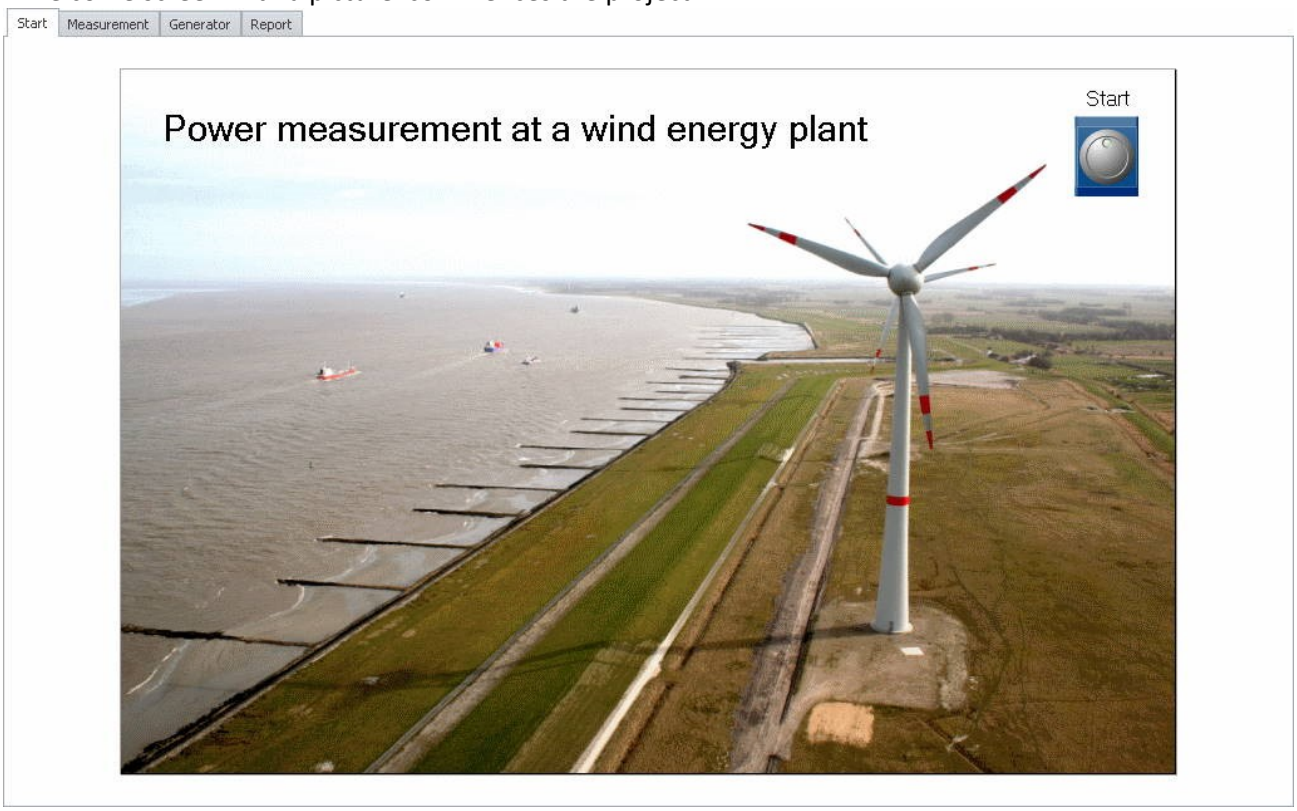
Assignment: Based on the project WEP, an example of skills presented in all the previous exercises working in concert is shown.

The measurement variables for 3x U and 3x I are simulated using imc Online FAMOS. All variables relevant to the power are computed with the function Power3.

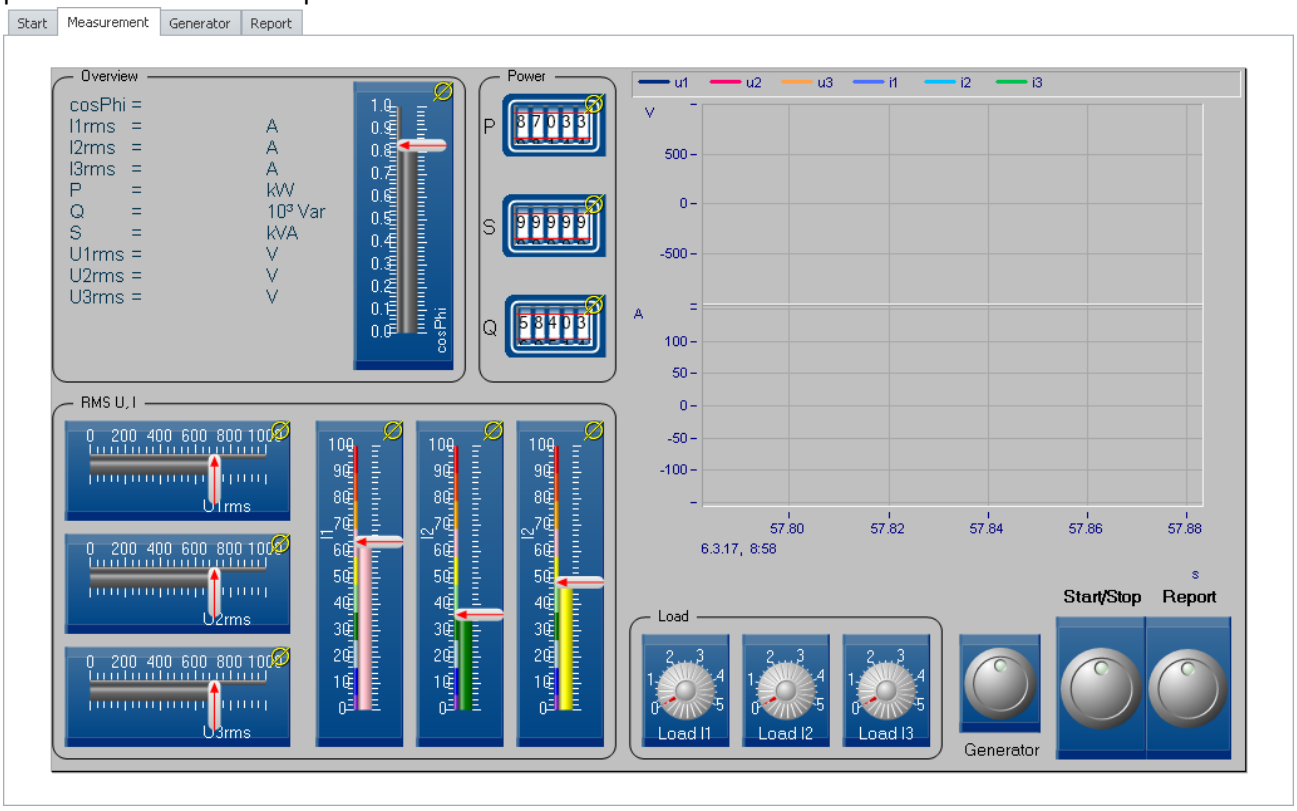
Additionally, a torque (simulated by a tuning fork) is measured, and a temperature.

The following steps are performed:

1. A welcome screen with a picture commences the project.

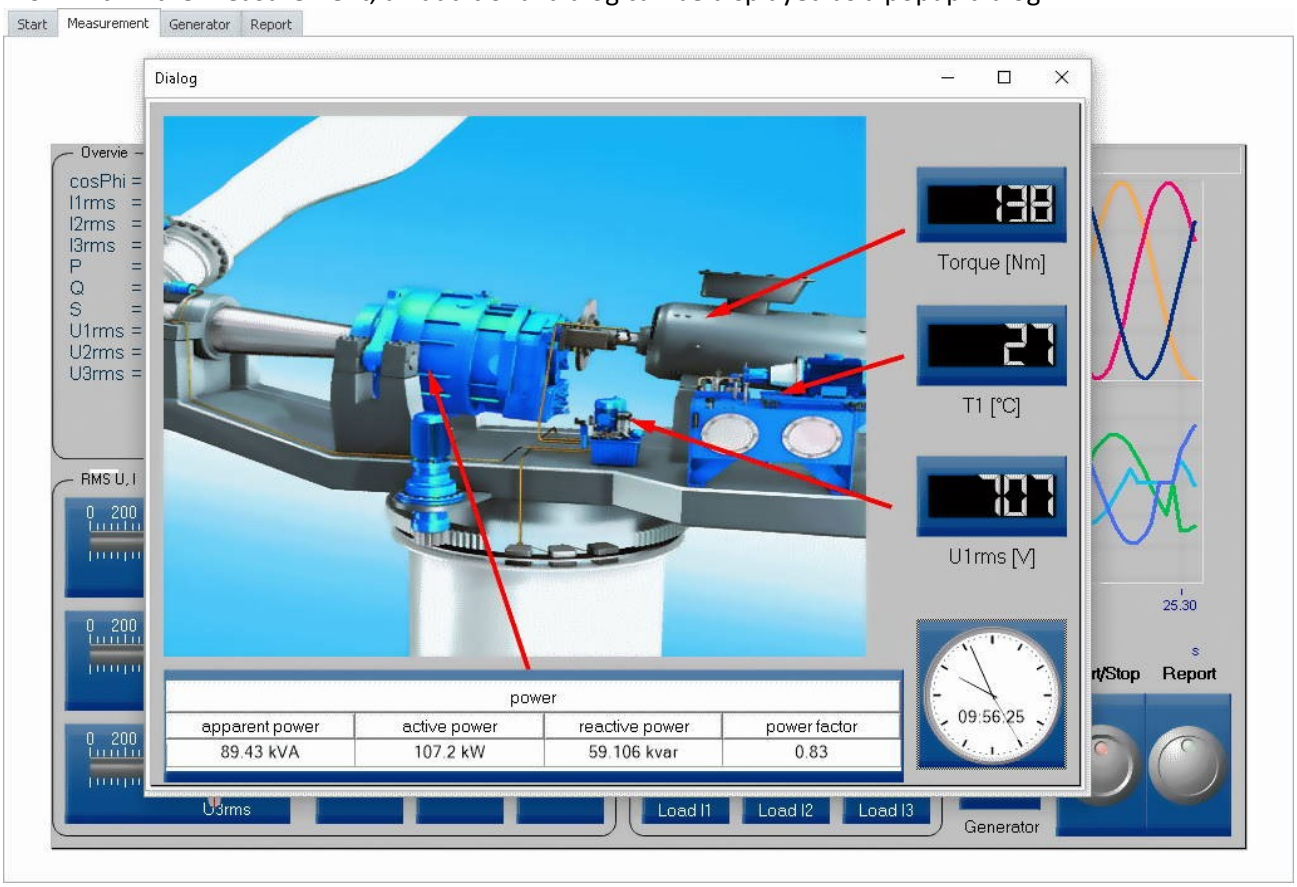


2. On a measurement page, both time and power data are displayed. The load can be set using potentiometers for each phase conductor.

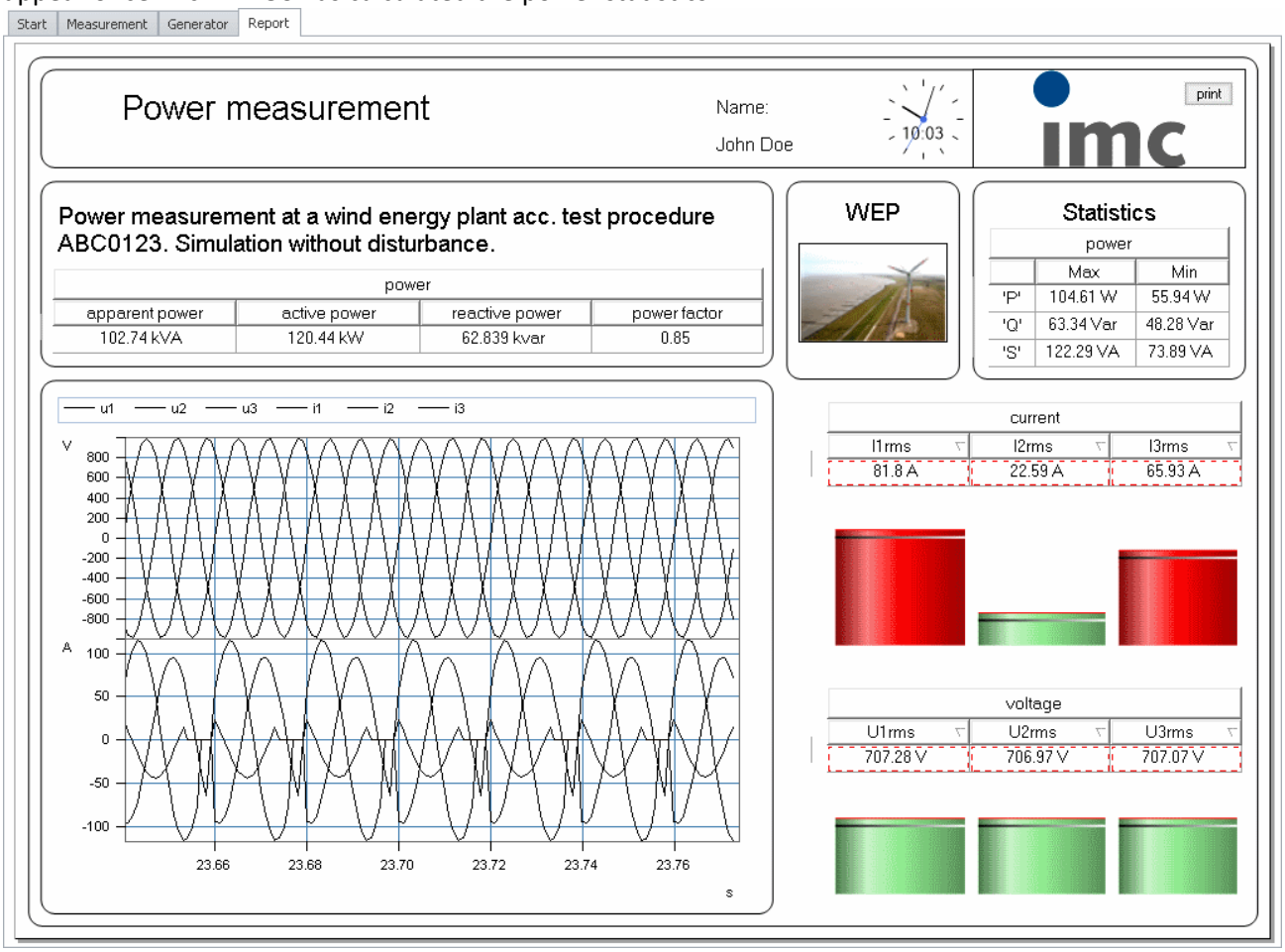


Measurement is started and stopped by a switch. At the end of the measurement, additional results are calculated by imc FAMOS: P max/min, Q max/min and S max/min.

3. From within the measurement, an additional dialog can be displayed as a popup dialog:



4. The results are displayed with the associated data in a Report page and as a PDF, which automatically appear once imc FAMOS has calculated the power statistics.



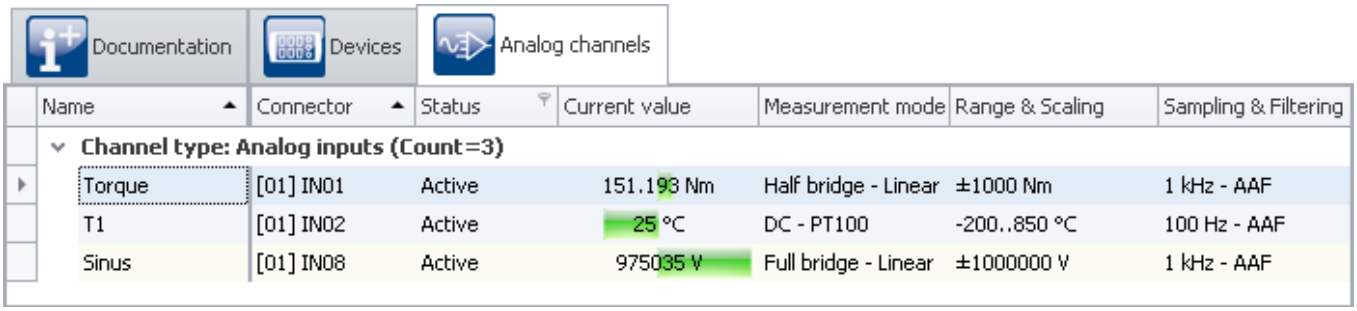
imc Online FAMOS Code: This code is based on the existence of a channel "Sine" which is recorded at a rate of 1kHz.

```
; Three phase simulator
_r =  6.28*SawTooth(Sinus, 0, 0.05, 20) ; Ramp
_r2=  6.28*SawTooth(Sinus, 0, 0.0005, 2000) ; Ramp
u1= 1000 * Sin(_r) ; three phases with 120° offset
u2= 1000 * Sin(_r+ 2.0941)
u3= 1000 * Sin(_r+ 4.18)

; the currents with different phases
; the unit A must be set by means of the Properties (toolbar "i").
i1= DspV_i1 * 20 * Sin( less(_r, 5.3) * (_r+ 0.53))      *  (2  + Sin(_r2))
i2= DspV_i2 * 20 * Sin( less(_r, 4.2) * (_r+ 0.47+2.09))*  (2  -  Sin(_r2))
i3= DspV_i3 * 20 * Sin( less(_r, 5.7) * (_r+ 0.32+4.18))*  (1.5+Sin(_r2))

Power3(p_t, P, S, Q, cosPhi, U1eff, I1eff, U2eff, I2eff, U3eff, I3eff, 0.02, u1, i1, u2, i2,
u3, i3)
properties
yUnit( i1, "A" )
yUnit( i2, "A" )
yUnit( i3, "A" )
yUnit( p_t, "W" )
yUnit( P, "W" )
yUnit( S, "VA" )
yUnit( Q, "Var" )
yUnit( cosPhi, "" )
yUnit( I1eff, "A" )
yUnit( I2eff, "A" )
yUnit( I3eff, "A" )
```

One possible solution is illustrated on the following pages.

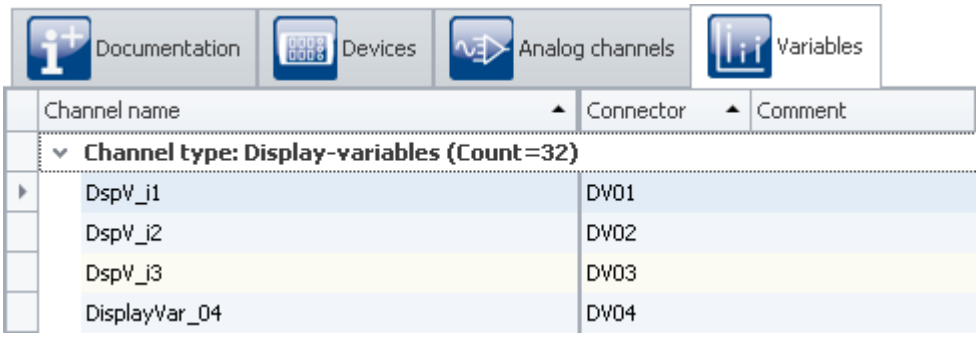
15.12.8.1 Setup

1. Go to *Setup \ Analog channels*
2. First activate three channels. Name one of the channels, which does not need to have an applied signal, **"Sine"**, with a sampling rate of 1kHz.
3. Connect the tuning fork to Channels 1 and 2. Name Channel 1 "Torque" and Channel 2 "T1".
   Settings:
   Torque: Half-bridge, scaling: 1 "Nm"/"mV/V", range: 1000 Nm, sampling rate: 1kHz
   T1: coupling: DC, correction: Pt100, sampling rate: 100Hz

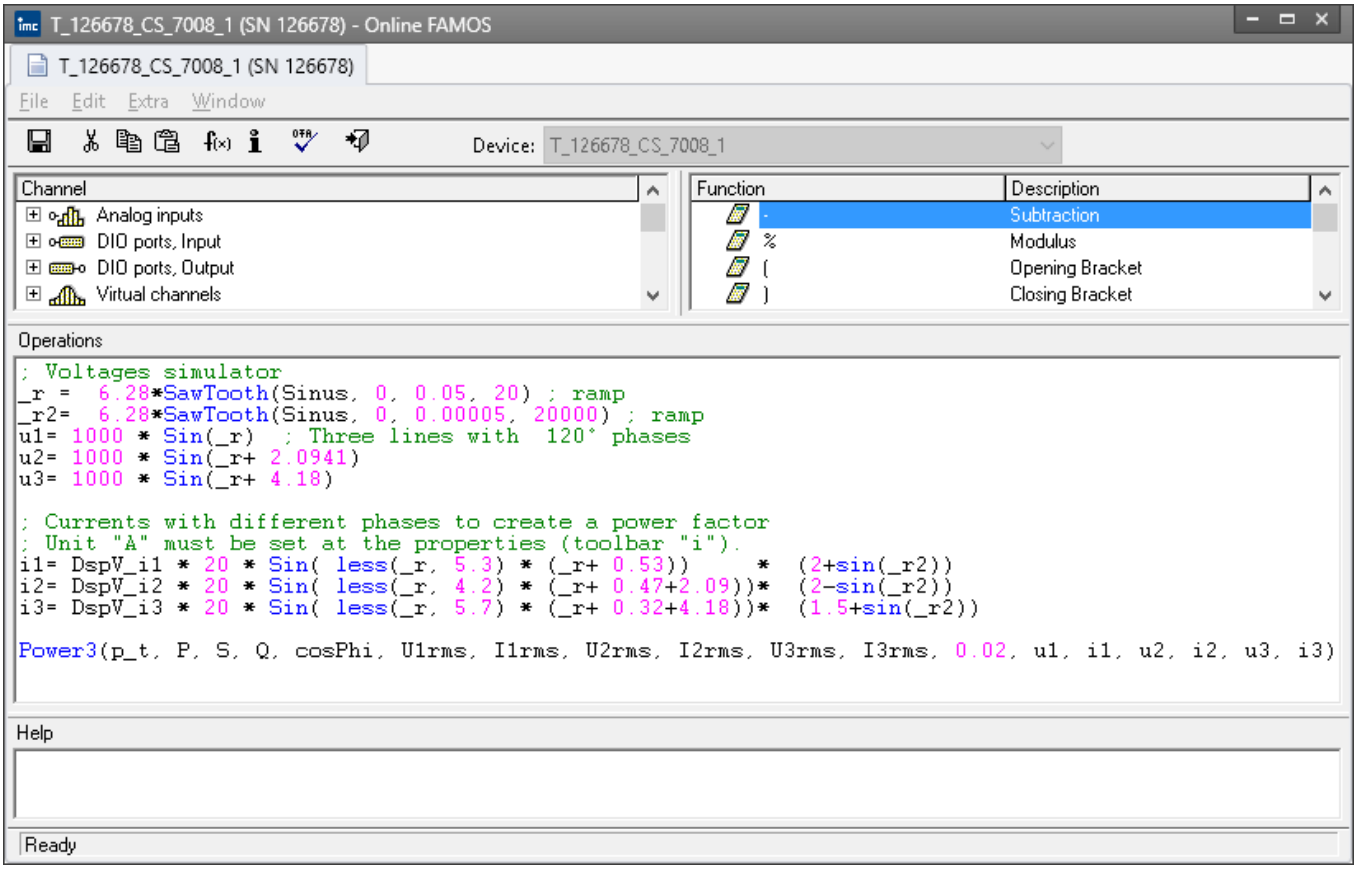


4. Go to the Setup / Variabes

5. Name the three Display variables: DspV_i1, DspV_i2 and DspV_i3



6. From the Device menu, open imc Online FAMOS and enter the imc Online FAMOS source text. From this text it is possible to copy script and paste it into imc Online FAMOS. If the imc Online FAMOS source text is available as a file, you can import it directly to imc Online FAMOS.



7. Click on the blue checkmark button to prepare the configuration, which sets up the channels in the Data Browser.

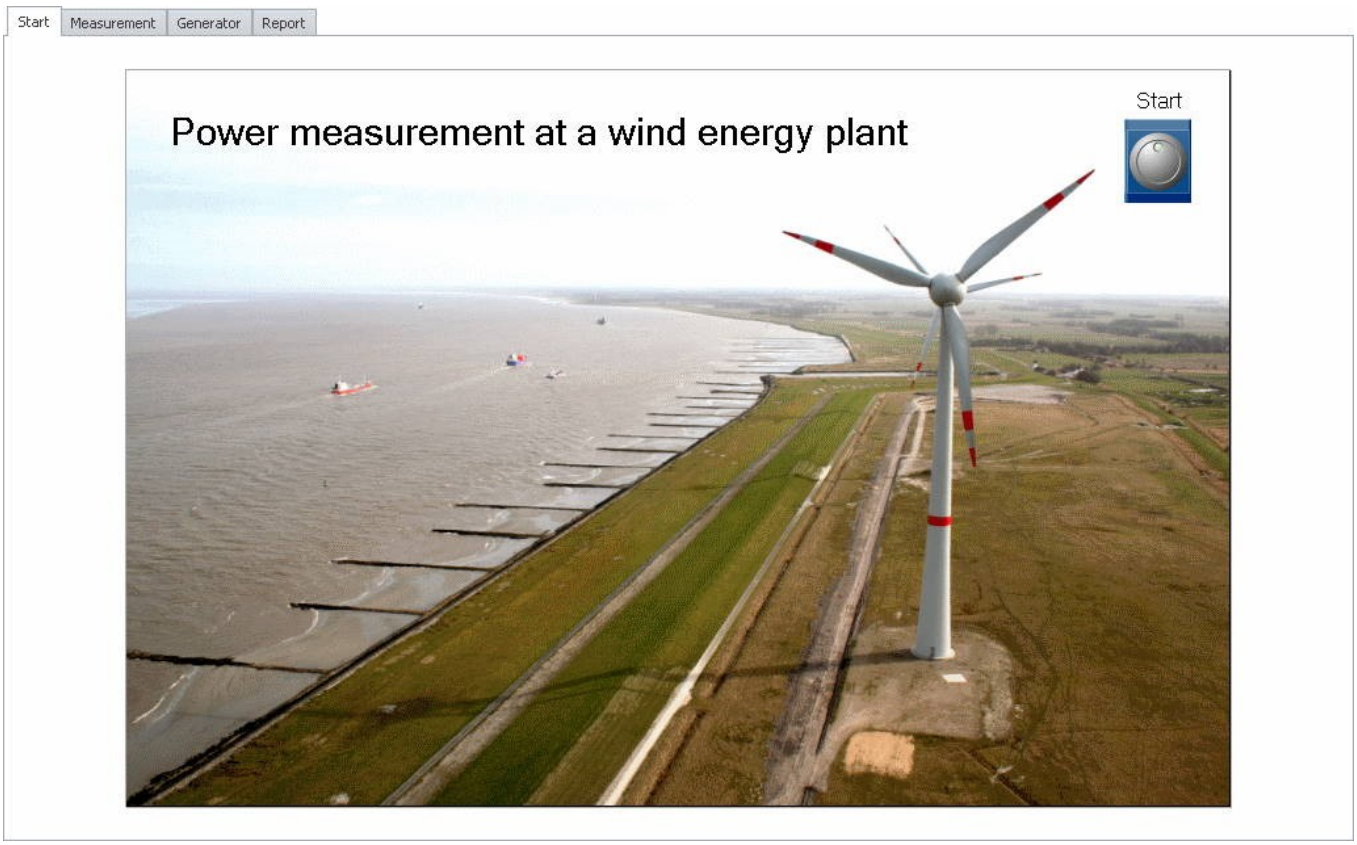
15.12.8.2 Panel pages

All Panel pages in the example were created with the **Skin** "*Gray skin*". To do this, right-click the mouse over the page and use the appropriate context menu items.

First create **four Panel pages**: "*Start*", "*Measurement*", "*Generator*" as standard dialogs and "*Report*" as a report in landscape format. These will be configured in the next steps.

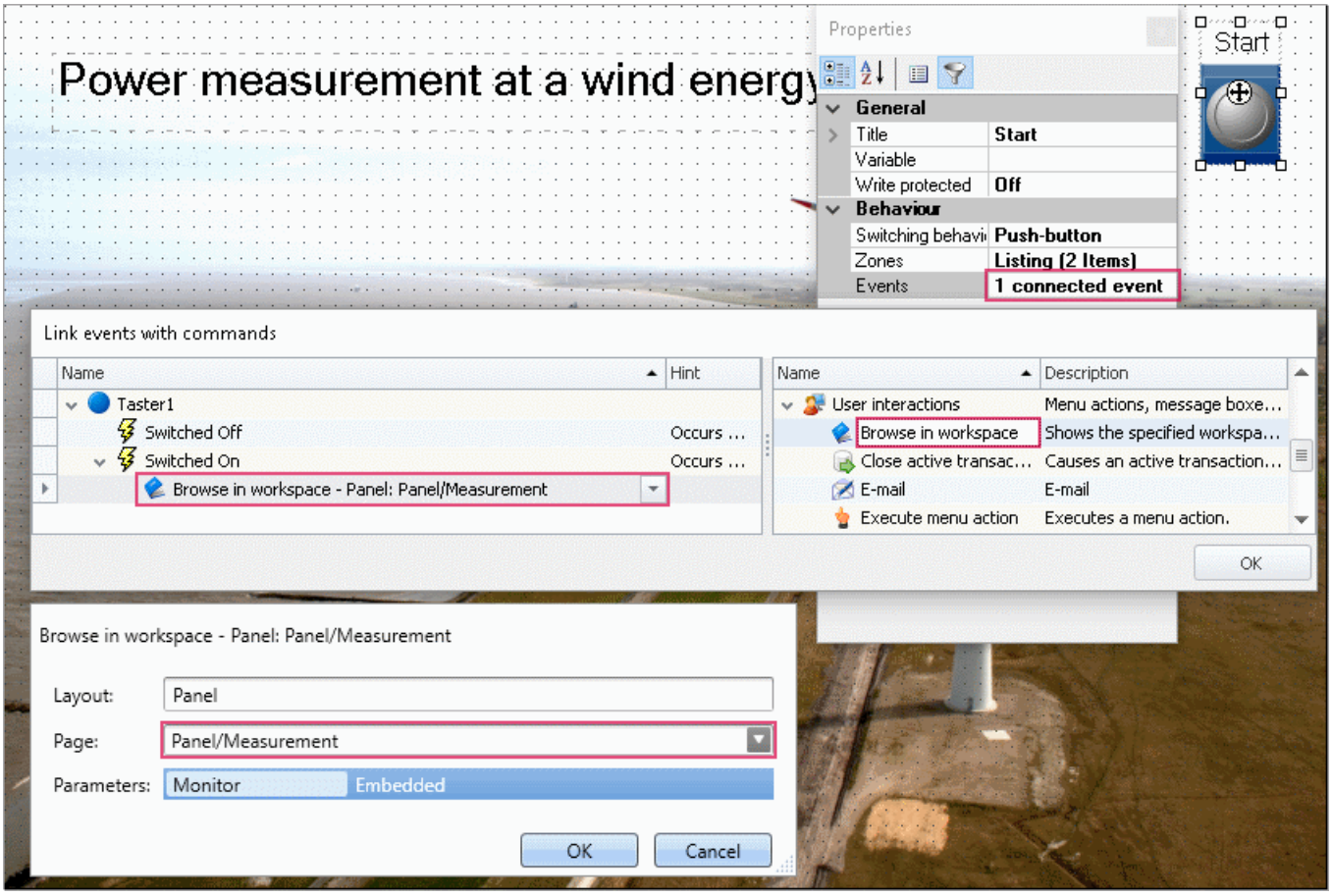
15.12.8.2.1 Panel - Start

The Panel page initially contains only a start button and the picture "Windanlage.jpg".

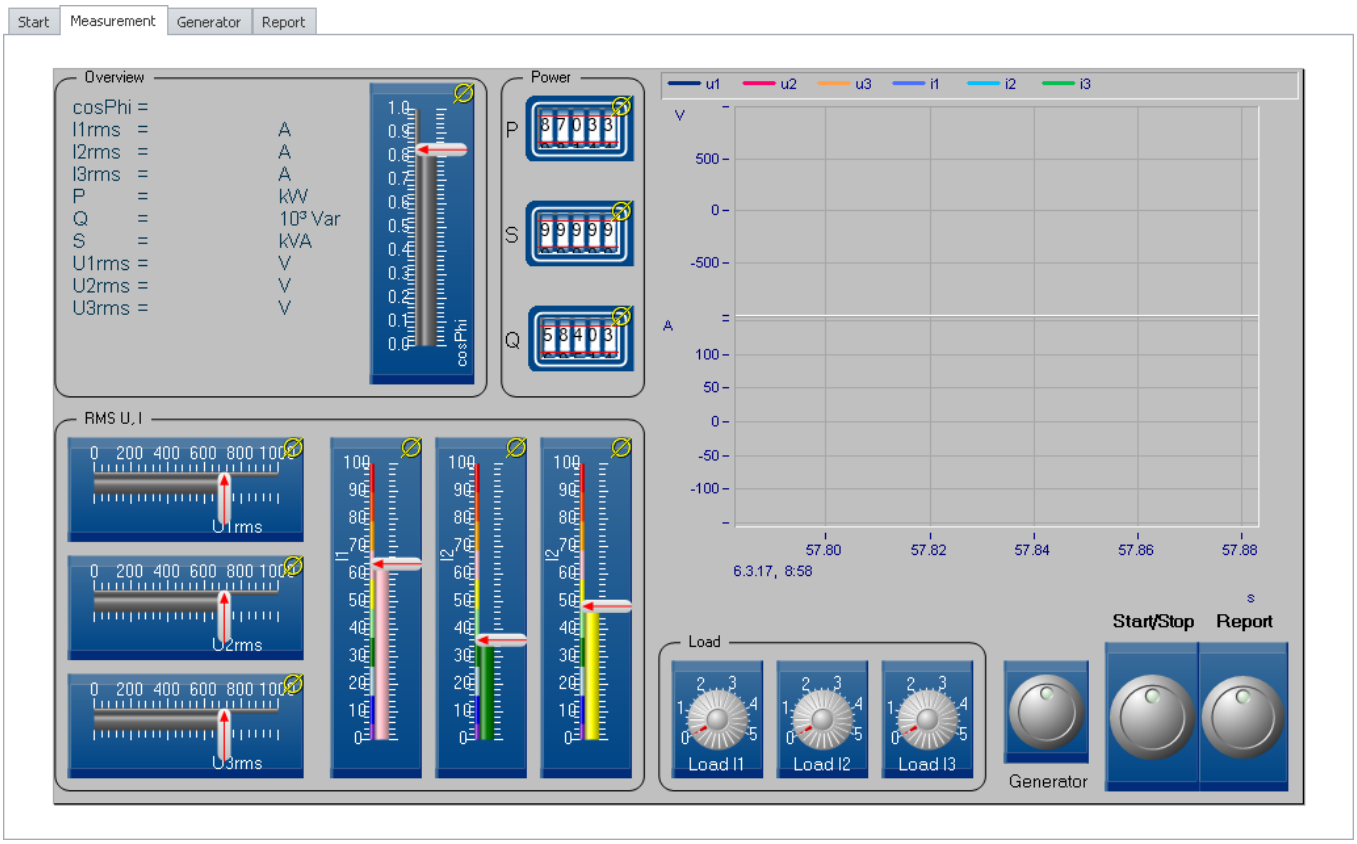


In the example, the Widget "*Industrial - Switch/buttons- Push-button*" is used. Give the *push-button* the caption "Start".

The *push-button* when pressed is to browse to the Panel page *Measurement*. This is arranged by setting the *Events* in the button's Properties:



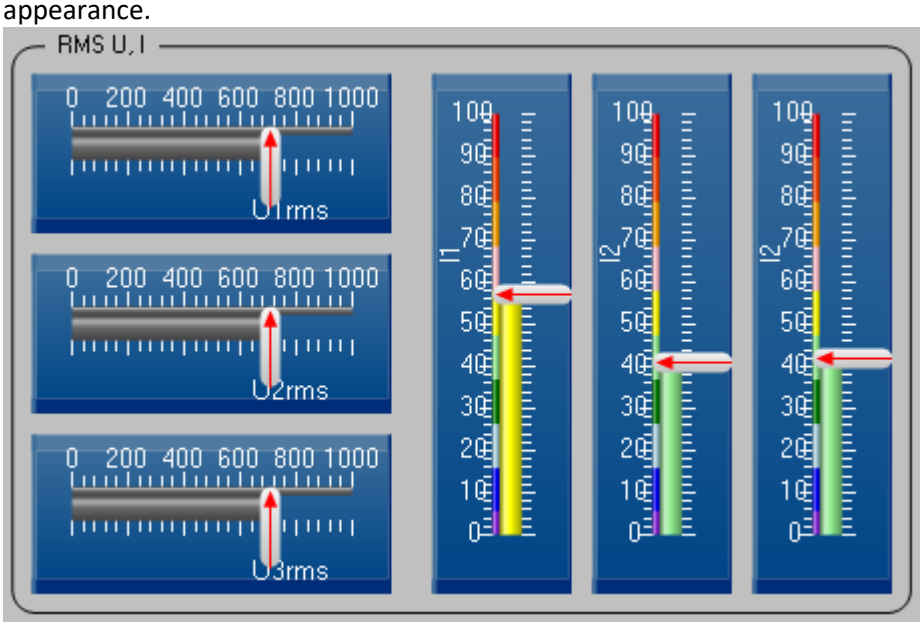
15.12.8.2.2 Panel - Measurement



This page has multiple bar meters for U and I as well as cosPhi, gauges for the power readings, potentiometers for setting the load, pushbuttons and two curve windows.

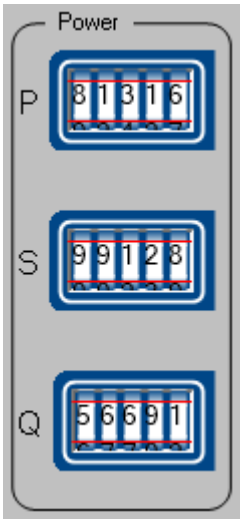
1. Begin with the **curve windows**. The window at right shows the instantaneous U and I values. Determine the variable. Create two Y-axes by first dragging thre voltages to the Y-axis. Then drag the three currents to under the X-axis. Add a grid.
2. Set up the left curve window. Drag the power signals, cosPhi and RMS-values from the Data Browser to the curve window, as illustrated in the picture. Set the curve window to "*Last value as number*" (right-click to open context menu: *Configuration\Display*). Format the numbers as desired: *Double-click\Numerical format*).

3. Next create the bar meters. In the example, the Widgets *Bar meters* from *Automotive - Bar scales* have been used. Also, their design was switched to *Industrial*. Set the input range as shown in the picture. The currents were supplemented with zones. Simply by adding the zones, you should obtain the following appearance.



Tip: First create a bar meter and make all desired settings. Next, make two copies of the Widget. From the Data Browser, drag the other channels to the copies.

4. The power signals are represented by the Widget *Odometer* from the group *Industrial - Potentiometer*.



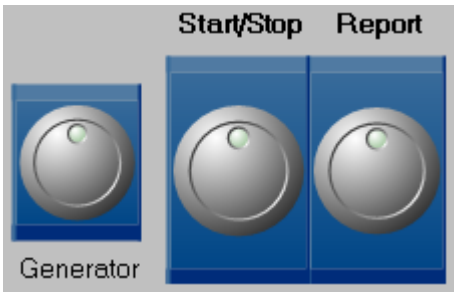
A *GroupBox* is dragged from the group *Shapes* to a position around the power signals.

Ensure that the variable name is visible to the left of the readout.

5. The loads are simulated by having one Display variable in the imc Online FAMOS programs govern each current amplitude. *(DspV_i1* through *DspV_i3*).
In the Panel, the Display variables are linked with potentiometers. For this example, the potentiometers from the *Industrial* group are used.

The range extends from 0 to 5. Name the potentiometers *Load I1* through *Last I3*

6. Finally, create the *push-buttons*; in this example, also using the Widgets from the *Industrial* group.



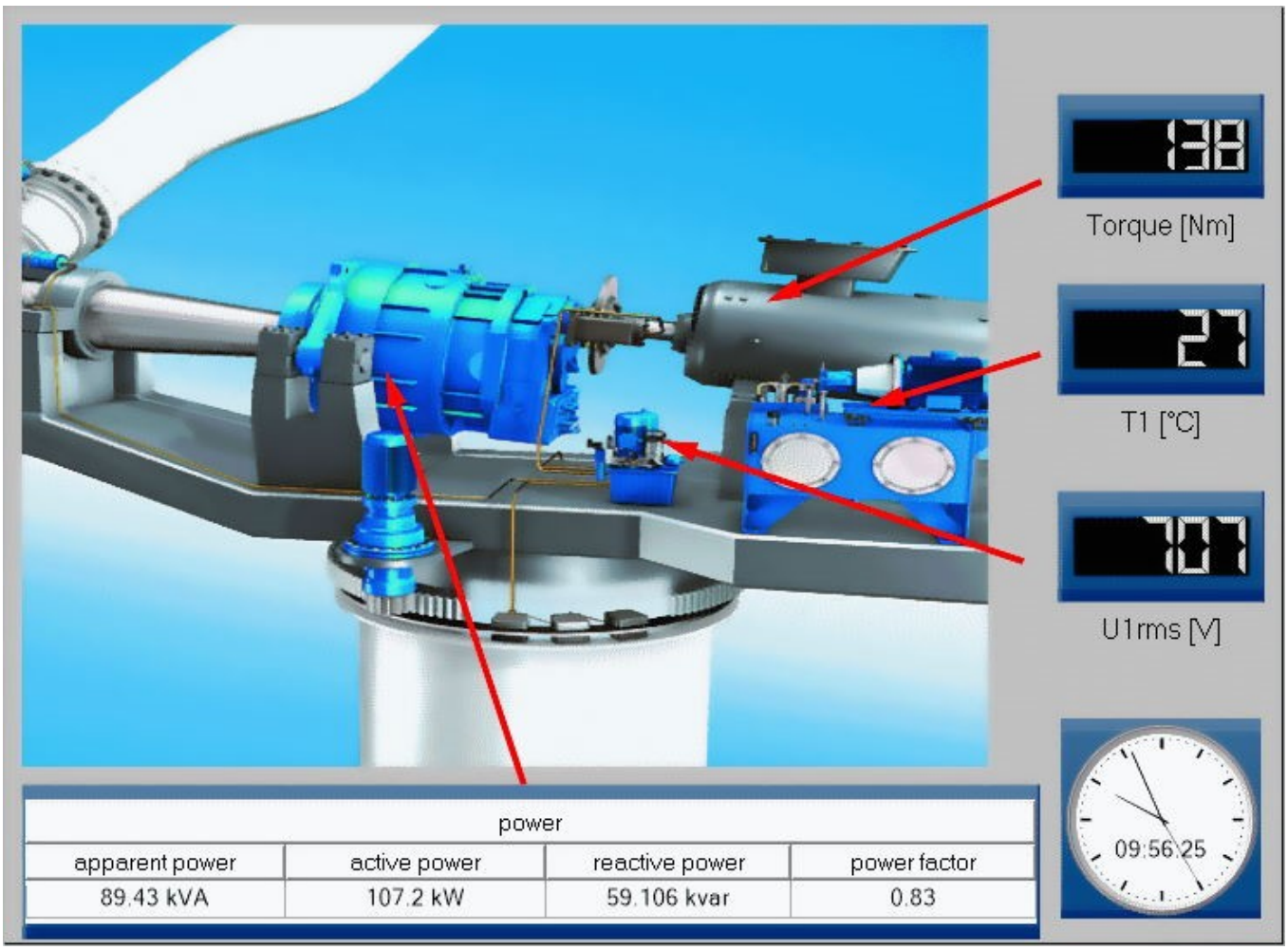
The **Generator-push-button** displays the dialog "*Generator*" as a popup dialog. If you have not yet created one, do so now. Ensure that the title "Generator" is visible. The push-button's events for the branch "*Switched On*" are set as "Panel" - "Panel *page as dialog*". Click on the arrow next to "Panel *page as dialog*" and select the Panel "Generator" as the Panel page to be displayed.

The **Report-push-button** takes you to the Panel page *Report*. To do this, the *User interaction "Browse in workspace*" under Events is needed.

The **Start/Stop** switch Widgets the Sequencer. To do this, the property Switching behavior must be set to *Switch*. What is essential is that the Sequencer is now governed by the switch. This is accomplished by means of the *Events*:
Switched On: *Execute menu action: Sequencer - Start Sequencer;*
Switched Off: *Execute menu action:* ***Device*** *- Stop.*
The Sequencer is now started. The measurement is concluded for the device. In the Sequencer, we will later wait for the end of the measurement and then run the imc FAMOS evaluation.

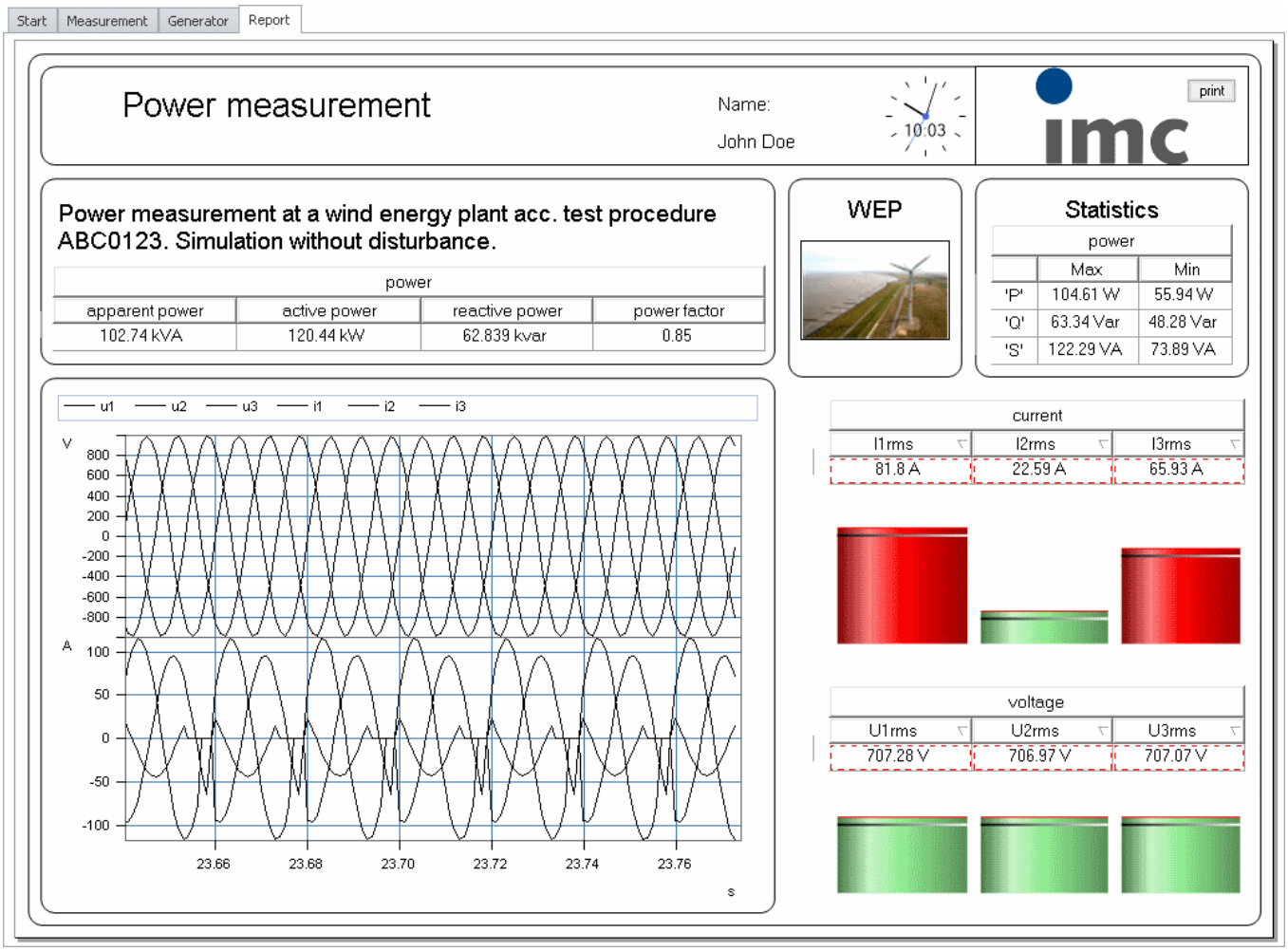
15.12.8.2.3 Panel - Generator

The Generator page is displayed as a popup dialog from the Panel page Measurement.



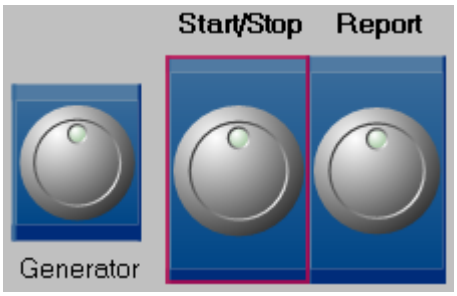
1. This page shows an additional picture "Generator.jpg".
2. From the group Industrial, **table** for the power signals is inserted. The row count is reduced to 1. By means of Drag&Drop, the cells are filled from the Data Browser with power values. Also, enter headers for the column and table.
3. Add three "*Seven-segment-displays*" from the group *Industrial\Status indicators*. Change the property *Background* to *Industrial*.
   Link the *Torque* and the temperature *T1*, as well as an RMS voltage value, with the displays, and display their titles as well.
4. Drag the clock Widget to the Panel page.

15.12.8.2.4 Panel - Report



1. Copy the **curve window** from the Panel page Measurement to the report. Set the curve window's Transparency to *Full*.
2. Copy the **power table** from the Panel page Generator to the report. There, too, switch the Transparency to *Full*.
3. For the "Statistics", create an additional table with 3 rows and 3 columns. Give a name to the table caption and column headers as shown in the picture. The cells are linked with variables which have not yet been created. Nevertheless, enter for the property *Variable* of the number entries "P_max", "P_min", "Q_max", "Q_min" and "S_max", "S_min".
4. For the Properties under *Cell*, select the appropriate *Unit* (W, Var, VA) and the *Factor Kilo*.
5. Create additional *Tables with Bar meter* for the RMS-values of the current and voltage.
6. As finishing touches for the report, add pictures and the remaining texts.

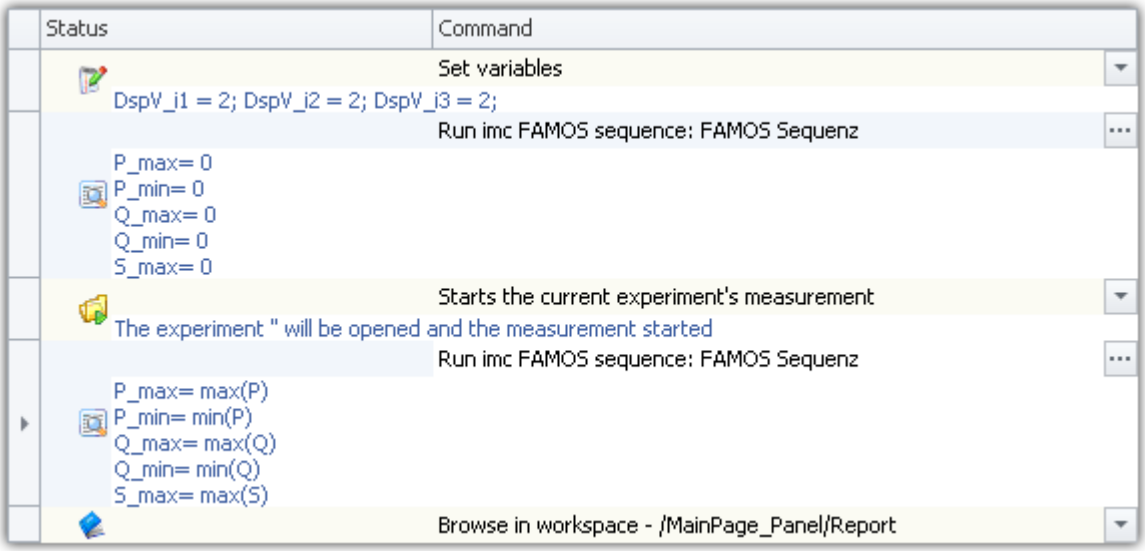
15.12.8.3 Sequencer



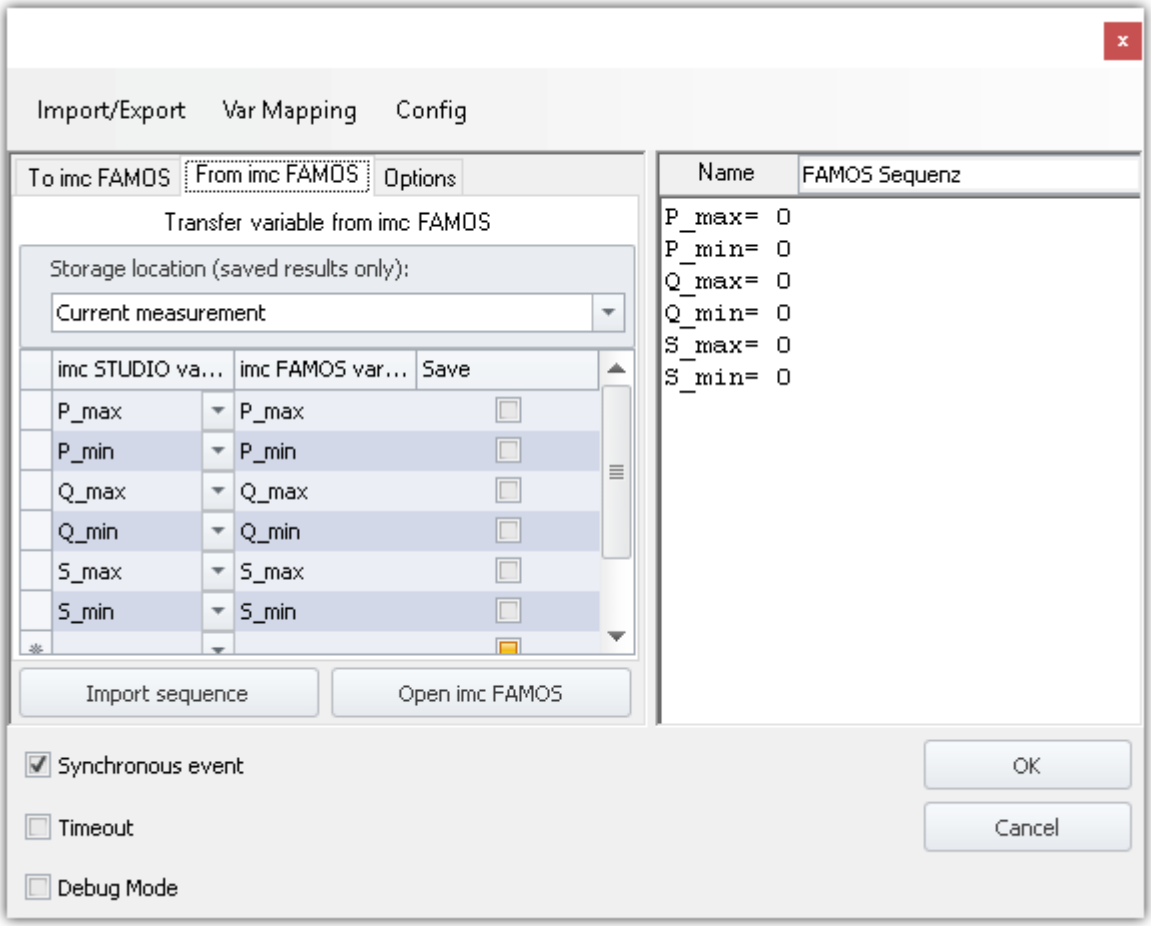
To start the Sequencer push the *Start/Stop* button in the Measurement Panel (see in fig.).

The following steps are now carried out automatically:

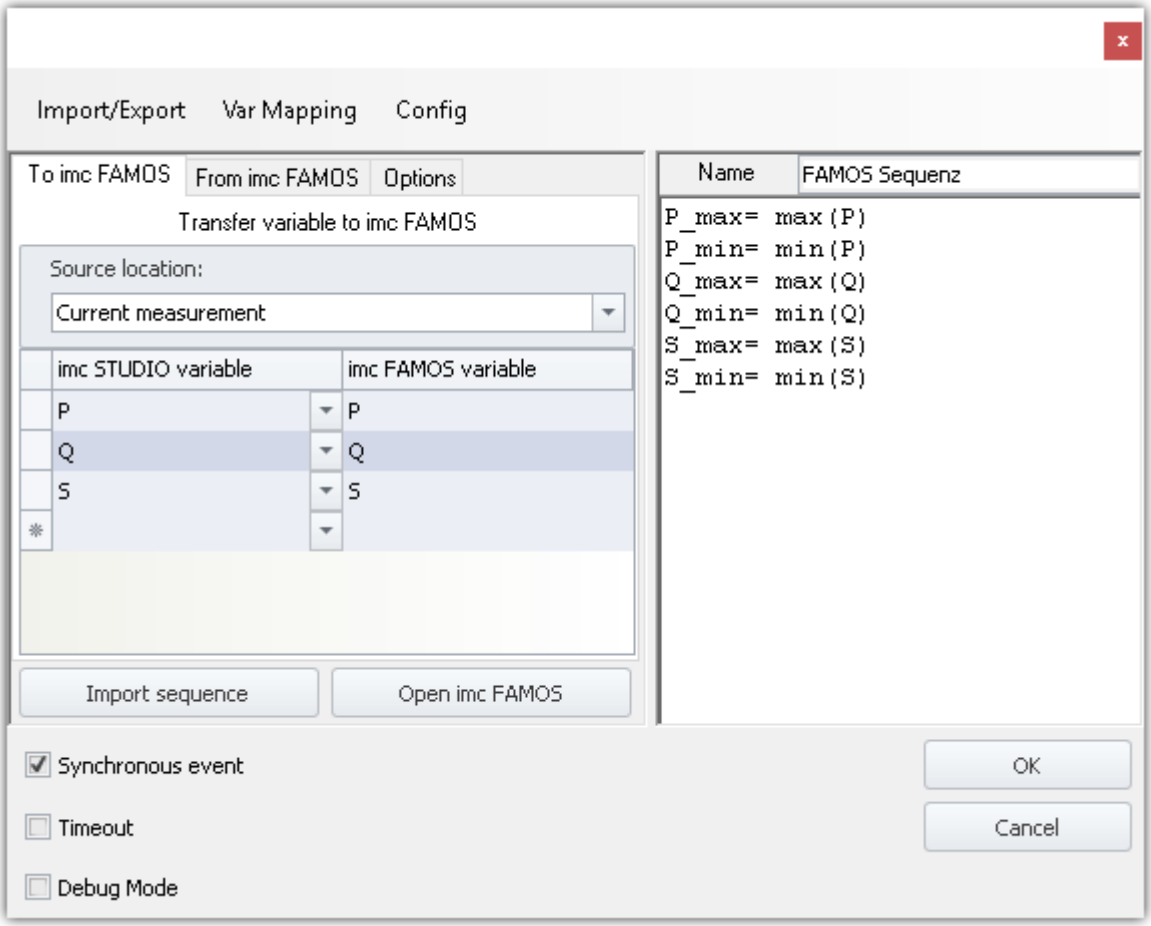
- initializing the variables to be transferred to Famos directly after the measurement
- measurement starts
- measurement stops (*Start/Stop* switch in panel)
- Famos sequence runs. This computes P_max, P_min, Q_max, Q_min ..... etc.
- report page pops-up, showing the calculated results.



First sequence, for initialization:



Second sequence, for calculating the power values.



Remark: The measurement can also be started using the normal Start button (green triangle), however, in that case the Sequencer steps will not be carried out.

16 Video interface

Section	Description
imc STUDIO GoPro 1802	imc STUDIO GoPro enables the acquisition of video data with a GoPro video camera.

16.1 imc STUDIO GoPro

imc STUDIO GoPro enables the capture of video data using the GoPro video camera.

The camera is connected via Bluetooth and WLAN with the controlling PC on which imc STUDIO is running.

The GoPro-plugin includes a remote control mechanism for the camera. Video data from a GoPro and measurement data from imc measurement devices can be captured and saved simultaneously.

The video data are captured by the camera and saved in the camera. After conclusion of the measurement, the video file is downloaded automatically. This process often takes as much time as the capture itself.

During its capture, there is no live transmission of the video image. Only after the measurement is it possible to analyze the video (e.g. using imc FAMOS) in parallel with the measured data.

Since GoPro works independently, direct synchronization of the video data with the imc measurement device is not possible. There is always a small offset which varies with the particular measurement and camera type and which can not be determined automatically. For purposes of synchronizing the video data with the measured data, it is possible to use the imc FAMOS Video-Kit, for example.

Note — Recommended: Use of the codec "LAV Filters"

For the playback of MPEG 4 video formats, the use of "LAV-Filters" is recommended. This package includes all necessary codecs.
You can obtain the software from the Internet or from our tech support 8.

imc STUDIO Product configuration

Activate the option "*imc STUDIO GoPro*" in the product configuration 29.

16.1.1 Startup operations

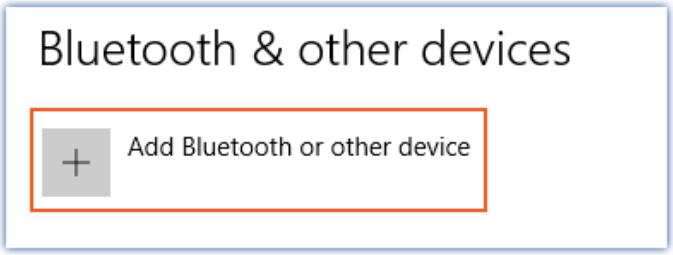
Prior to the first connection, the following steps normally must be performed:

- Prepare the camera in Windows - but do not connect with it.
- Find the camera in imc STUDIO and connect with it.
- Set up the connection with the camera via WLAN.

Prerequisites
The GoPro-plugIn must be activated in the Product Configurator.
Video data belonging to a measurement can only be recorded once either an imc measurement device, or a 3rd-party device, is participating in the measurement.
The GoPro contains a memory card with sufficient write-speed: SD-card with V30 or higher.
The PC is able to establish a connection both via Bluetooth and via WLAN.

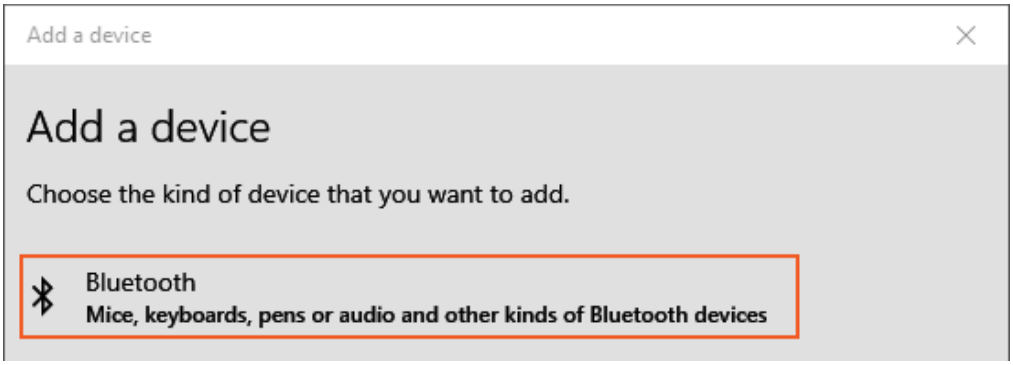
Setting up the camera for connection in Windows

- Open Windows "*Settings*" (`<Win> + i`)
- "*Devices*" > "*Bluetooth & other devices*"
- Click on "*Add Bluetooth or other device*"



An additional dialog appears.

- Click on "*Bluetooth*" to start a search in Windows.



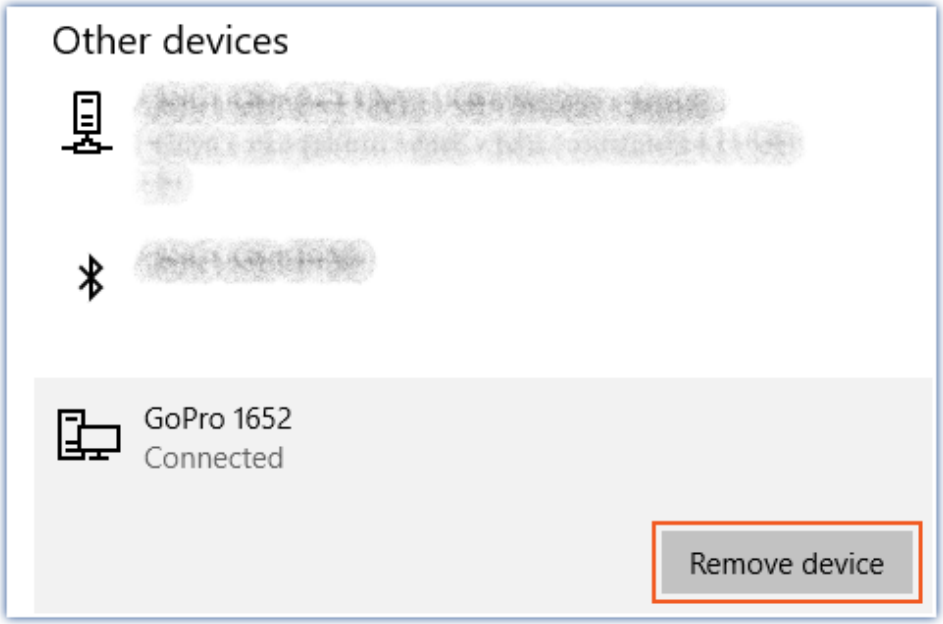
- If the camera was found, **click on "*Cancel*"**. Do NOT connect with the camera.

Upon the next device search in imc STUDIO, the camera should be found.

Notes Connect GoPro via imc STUDIO, and not via Windows

In order to be able to connect imc STUDIO with the GoPro, the camera may not already be paired or connected with a device via Bluetooth.

Therefore, before connecting imc STUDIO with the GoPro, remove any GoPro present from Windows under: "*Settings*" > "*Devices*" > "*Bluetooth & other devices*" > "*Other devices*" > "*Remove device*". Otherwise, connection will not be possible.



Finding and connecting the camera in imc STUDIO

As soon as GoPro is activated, it can be discovered via Bluetooth.

For purposes of the first connection, the camera must be in **pairing mode**. Depending on the model, the corresponding menu item in the camera can have various names:

- HERO8: "*Preferences*" > "*Connections*" > "*Connect Device*" > "*GoPro App*"
- HERO9: "*Connections*" > "*Connect Device*" > "*Quik App*"
- HERO10: "*Preferences*" > "*Wireless Connections*" > "*Connect Device*" > "*GoPro Quik App*"
- HERO11: "*Preferences*" > "*Wireless Connections*" > "*Connect Device*" > "*GoPro Quik App*"

If necessary, consult the user's manual for the camera.

Under imc STUDIO, run the "*Device Search*". If the GoPro is found, it appears in the Devices list. If no other imc device was found, a message will appear which will state that no device has been found, even though the GoPro actually was found.

Select the camera and click on "*Connect*".

Only one GoPro can ever be selected at the same time.

Notes If the GoPro doesn't appear in the list

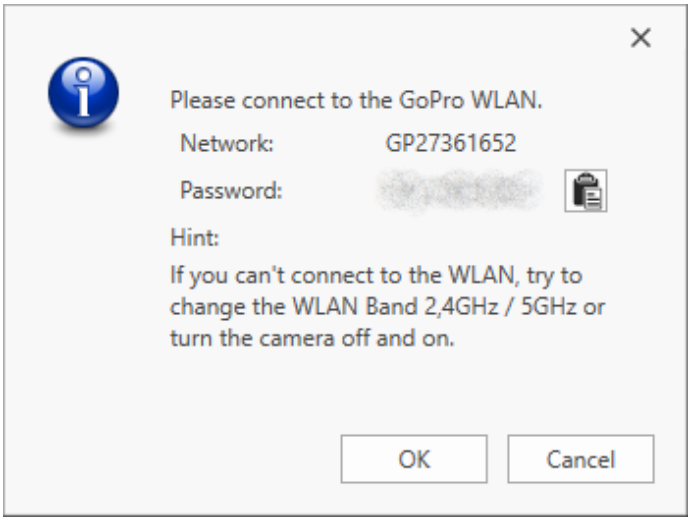
- Repeat the device search. Occasionally it is necessary to run the device search multiple times until the GoPro has been found.
- If appropriate, observe the note above: "*Connect GoPro via imc STUDIO, and not via Windows*"
- It is also possible that the GoPro will appear only in the Devices list only after a delay.

Connection with the camera via WLAN

In order for connection with the GoPro to succeed, there may not be any other WLAN-connection.

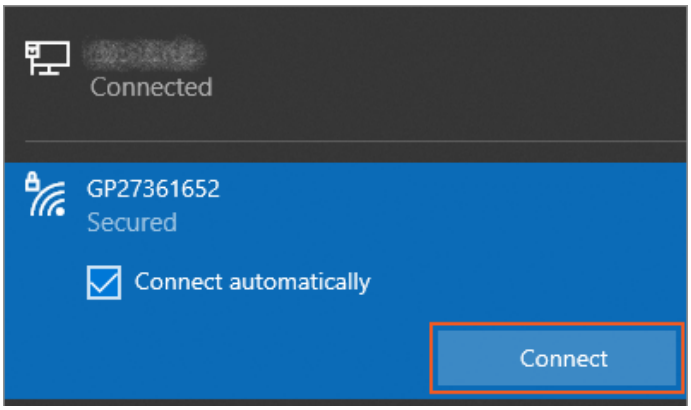
Once connection via Bluetooth has been established, the camera automatically activates the GoPro's WLAN.

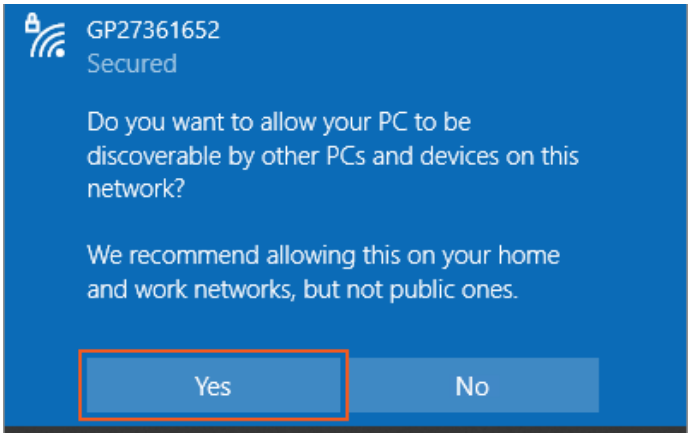
As long as the WLAN connection with the camera has not yet been established, the following dialog appears:



Here you see the network name and the password for the camera's WLAN. Leave this dialog open for now.

The GoPro's WLAN eventually also appears in the WLAN list in Windows. There, select connection with the camera's WLAN.





Respond to the following prompt with "*Yes*":
"*Do you want to allow your PC to be discoverable by other PCs and devices on this network?*"

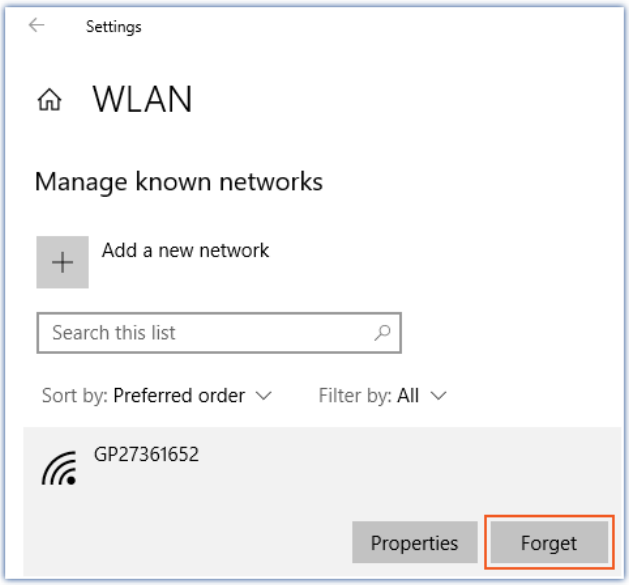
Once the WLAN connection has been established, click in the imc STUDIO's dialog on the button "*OK*". Then the imc STUDIO will attempt for 10 seconds to connect with the GoPro via WLAN. If connection with the WLAN is not possible, the dialog shown above will appear again. In that case, repeat this procedure. Under some circumstances, this process may take some time.

Notes — If connection via WLAN not possible

If connection via WLAN can't be established, it may help to remove the WLAN as a known network.

- Open Windows "*Settings*" (`<Win> + i`)
- "*Network & Internet*" > "*WLAN*" > "*Manage known networks*"

Select the network and click on "*Forget*" in order to delete it:



More solutions to problems

Problem: When connecting via Bluetooth, the error 20000332 would repeatedly appear.

Solution: It can help to perform the device search again. The GoPro is then not added again, but instead is updated internally.

16.1.2 Running the measurement

In the GoPro, **Video-mode must be activated**. Recording of video data in the GoPro begins upon starting the measurement.

At least one regular measurement channel (imc measurement device, AudioDevice or ChannelLoader) **must be active** in order for the camera to record.

- The camera **can not be trigger-driven**.
- For this reason, at least one active channel must be set with an "immediate trigger" (e.g. the "*BaseTrigger*"). "Immediate triggers" are triggers which fire upon starting the measurement, and have no other associated sources.

During the measurement, a red "*Record-Button*" on the display indicates that the GoPro is currently recording.

Downloading the video

Once the measurement has stopped, download of the video to the measurement folder begins after a delay of some seconds.

Additionally, a URL-file (e.g. "*GoPro 1619.URL*") is saved to the measurement folder. If the download fails for some reason, it is possible to identify the video file on the memory card by its respective URL-file. Following successful download of the video, the URL-file in the measurement folder as well as the video-file on the camera are deleted.

Downloaded videos can be displayed in imc FAMOS, for example. Prerequisite: Installation of an LAV-filter for MP4-files.

Notes

File size

Maximum size of video file: 4 GB. When the file size is reached, an additional video file is created in the device; e.g. "*GoPro 1619_002.MP4*", "*GoPro 1619_003.MP4*".

Video download

- Downloading the video takes about as much time as the video itself. As soon as the download is concluded, a message appears in the logbook.
- Simultaneous downloading and recording are not possible.
- When imc STUDIO is closed, the download is canceled.

16.1.3 The Livestream-Widget

The Livestream-widget is located in the Panel under the grouping "*Video*". A Livestream-Widget can be positioned (not one on each page).

Before and after the measurement, the Widget returns the current video transmission from the camera. During the measurement, no live streaming is possible (exception: GoPro HERO8).

The live stream is also visible during the download.

17 Programming interface

Section	Description
Scripting 1808	Interface for using Scripts in imc STUDIO.
Third Party Device Interface 1917	Interface for using third party devices in imc STUDIO.
API 1946	Interface for developing custom application with access to imc STUDIO functions.

17.1 Scripting

The **imc STUDIO Scripting** is an **imc STUDIO** plug-in providing a programming interface (C#, .NET) to imc STUDIO. The development environment *SharpDevelop* (Editor) is included.

Besides user-defined programming, **fundamental functions** of imc STUDIO are available, e.g.

- read and write device and channel settings/parameters (Setup),
- access to the Panel and the Widgets,
- access to the Data Browser: create, read and write variables,
- application of imc FAMOS-functions to channel data,
- execution of menu actions (ribbon),
- execution of Sequencer commands,
- respond to events.

The following **script types** are available:

- Script
- Panel Script
- Context script
- Type library script
- Event script
- Third Party Device script.

It is possible to export and import the scripts, also as DLL.

To **run a script**, the following mechanism are available depending on the script type:

- as command
  - in the Sequencer
  - at Widget
  - on events
- in the background
  - linked to a Panel page
  - linked to the experiment or project

This manual illustrates entry points into **Scripting**.

Note

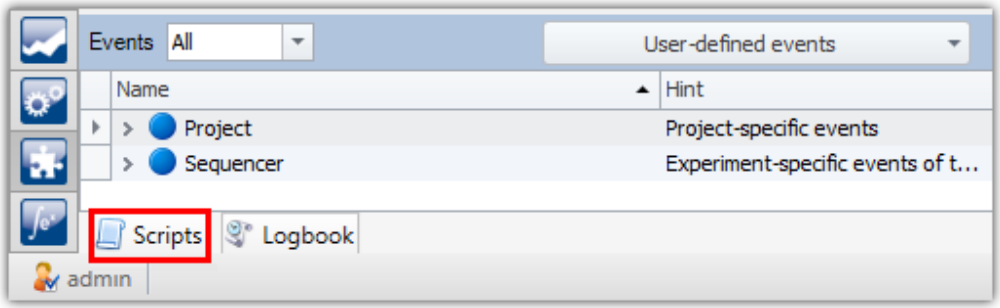
- For **creating** and **editing** a script an imc STUDIO *Developer* license is needed.
- Scripts[1] can be **run** in all editions without a special license.
- The programming language is C#.

[1] with the exception of Third-Party-Device scripts

17.1.1 Menu bar and context menu

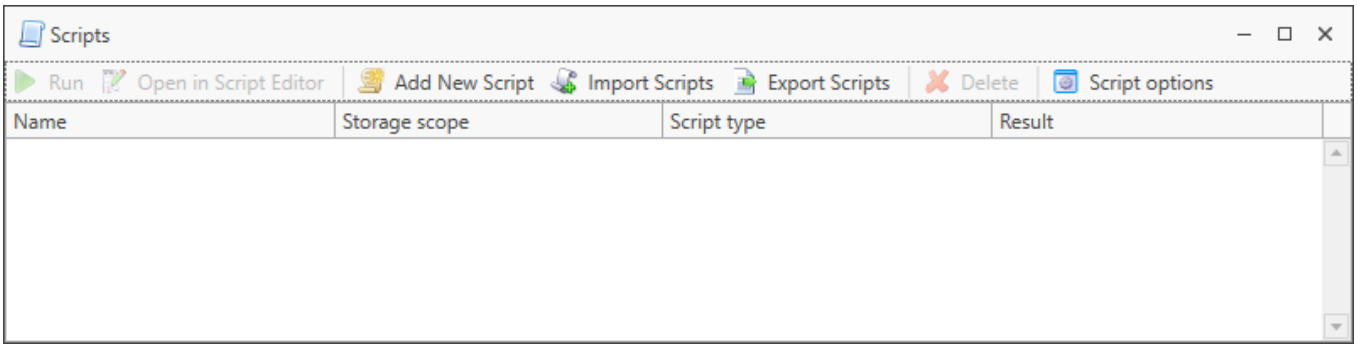
Tool window

To use **Scripting**, go to the Sequencer page. There is a tab on the bottom called **Scripts**.



Sequencer: Tool window "Scripts"

In the tool window the scripts can be sorted by clicking on the column names.



Tool window "Scripts"

Note

If the *imc STUDIO Scripting Editor* (Developer-Edition) is activated and the tool window *Scripts* is not available, please check in the ribbon *View > Choose Tool Windows* if the tool window *Scripts* is activated.

Menu

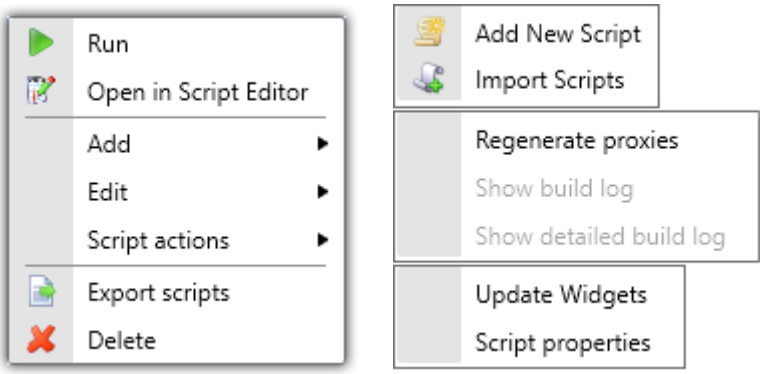
Run | Open in Script Editor | Add New Script | Import Scripts | Export Scripts | Delete | Script options

Menu action	Description
Run	Runs 1816 the selected script.
Open in Script Editor	Opens 1815 the selected script in the editor.
Add New Script	Opens the dialog for creating 1812 a script.
Import Scripts	Opens the Import-Dialog 1818.
Export Scripts	Opens the Export-Dialog 1817.
Delete	Deletes the selected script.
Script options	Opens 1906 the script options in the imc STUDIO options dialog.

To edit a script, select the menu item *Open in Script Editor* Open in Script Editor.

Only one script can be edited at a time. For this reason, the script editor must be closed before editing another script.

Context menu

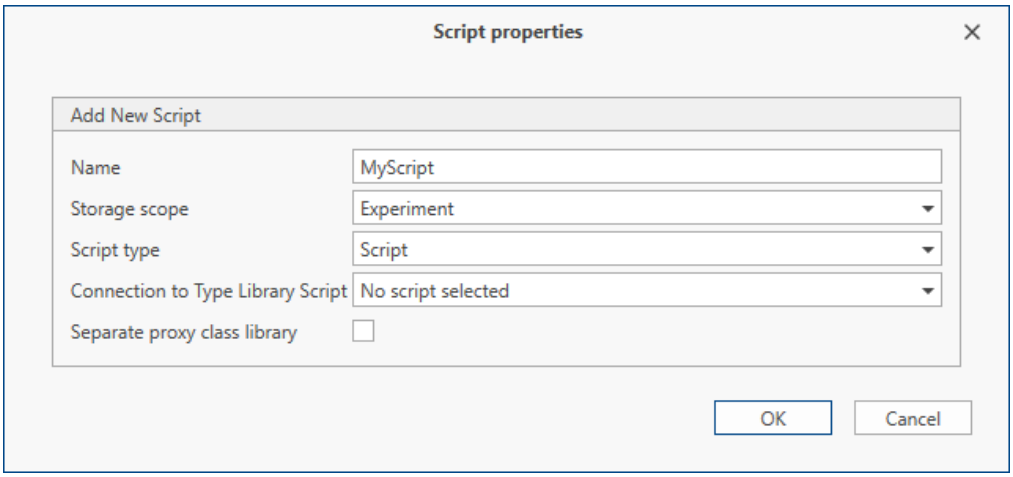


Menu actions	Action/Sub-menu
Run	Runs 1816 the selected script. (Script types: Script, Context script, Event script)
Open in Script Editor	Opens 1815 the selected script in the editor.
Add	• Add new Script 1812 • Import Scripts 1818
Edit	• Regenerate proxies 1819 • Show build log • Show detailed build log
Script actions	• Script Properties: New Storage scope 1814 • Regenerate Script Widgets 1879 (Panel script)
Export scripts	Opens the Export-Dialog 1817.
Delete	Deletes the selected script.

17.1.1.1 Create scripts

To create a new script (C#), switch to the Sequencer and open the tool window **Scripts.**

Click on *Add new script* Add New Script . The **Script Editor** has to be closed before adding a new script.

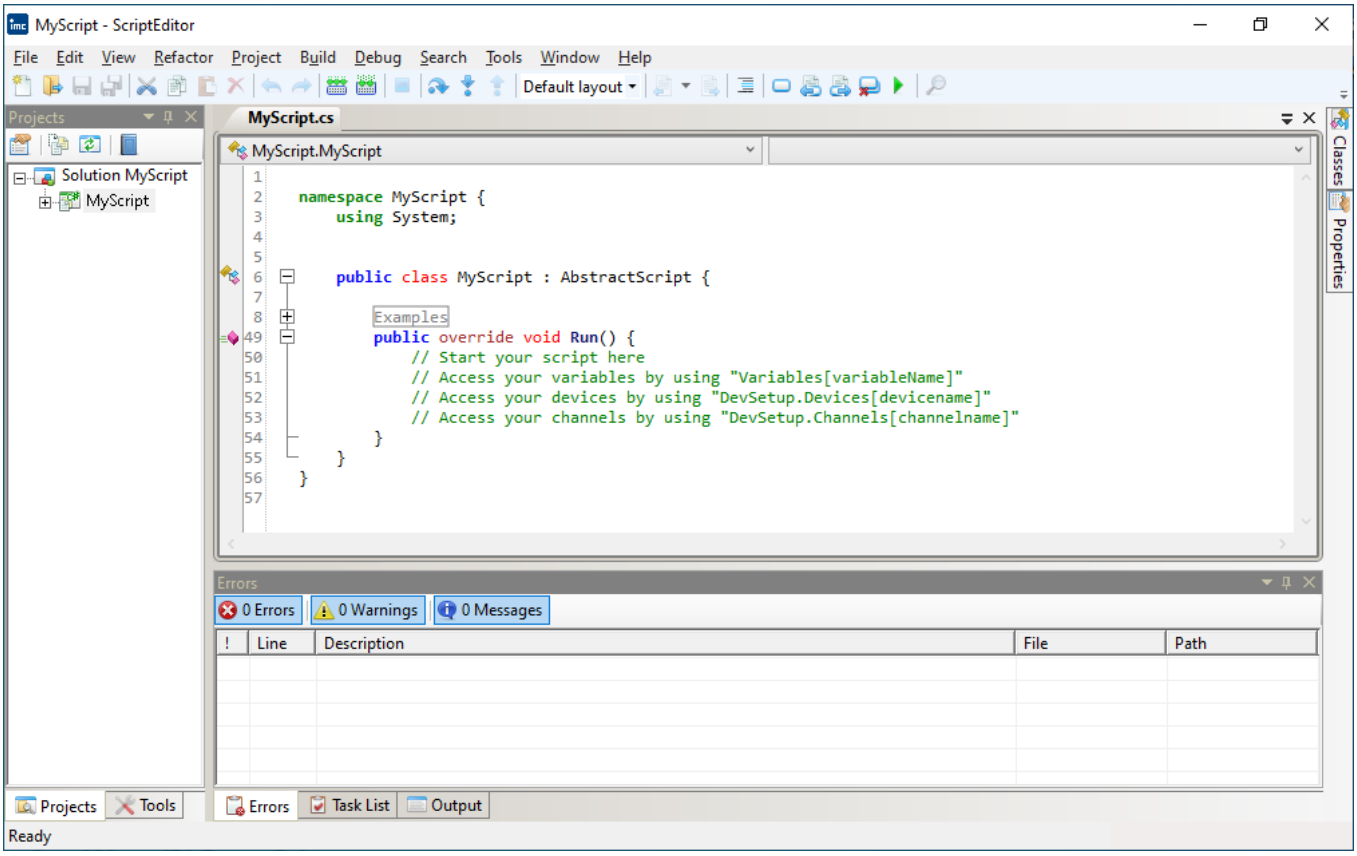


Dialog: Script properties

For the other script types 1813, this dialog and the editor template looks different. See also the corresponding sections: Panel-script 1879, Context-script 1883, Type-Library-script 1887, Event-script 1889.

If the dialog is exited with *OK*, the **Script Editor** opens. The programming language is C#.

Helpful examples are provided in the *Examples* portion of the source text . Before a script can be used, it must be compiled (*Build > Build Solution*).



ScriptEditor

Note Changing the language of the editor

The **language** of the editor can be changed via *Tools > Options > General > UI Language.*

17.1.1.1.1 Script type

The following **script types** are available. The **script type** should be selected which accords with the intended application.

Script Type	Description
Script 1878	• A **Script** is called and run by means of a *command*.
Panel Script 1879	• When creating a **Panel Script** an according *Panel page* has to be selected. • A **Panel Script** is linked to the corresponding Panel page. • The `Run()` method in the script is called when **browsing to** the corresponding page or when the page is already opened and the *Design mode* is deactivated. This is also the case when loading an experiment. • The `Stop()`-method is called when **browsing to other** pages or when the experiment is closed, or the *Design mode* is activated.
Context Script 1883	• A **Context Script** is linked to a *scope* (Experiment, Project, ...), • The `Run()`-method is called when **opening** an experiment (Scope *Experiment*). • The `Stop()`-method is called when **exiting** the experiments or when imc STUDIO is closed or another experiment is opened. (Scope *Experiment*). • Here, some of the possible actions are to register ("*hook*") actions with *events* and to de-register ("*unhook*") them from the events.
Type Library Script 1887	• The **Type Library Script** is a way to provide one's *own classes* and *methods* for the other script types as a *DLL*.
Event Script 1889	• The **Event Script** is appended to the associated events as a *command*. • In the script, there is the possibility to access *arguments* of the events triggered, e.g. the measurement's name or its path for interval storage.
Third Party Device Script 1918	• With the help of the **Third Party Device Script**, it is possible to integrate and use Third Party Devices in imc STUDIO.

When registering actions to, or de-registering them from events, it is preferable to use a **Context Script** (or for Widget-actions, a **Panel-script**) instead of a **normal script**.

For further information, see the corresponding sections: Panel-script 1879, Context-script 1883, Type-Library-script 1887, Event-script 1889 or rather Third Party Device Script 1918

17.1.1.1.2 Script properties

Script Properties	Description
Name	The script's name must not start with a *number*. No *spaces* or *special characters* are allowed, except for the underline. It also may not be the designation of any C# command (for, foreach, if, else, class, private, public, int, double, string ...).
Storage Scope	The script's *availability*, e.g. Experiment, Project, Application or Sequencer. To change this setting, click on *Script actions > Script properties > New Storage Scope* 1814.
Type	The selected *script type* 1813.
Connected Type-Library-Script	Selection from a list of created *Type-Library-Scripts* 1887. The Type-Library-Script is available within the script as a library.
Separate proxy class library	The Proxy class is added as separate reference (DLL) and can be also used in other scripts, e.g. for global use of objects from the script's namespace, using the Script clipboard 1875.
Activity Scope	*for Context scripts only:* Scope (Experiment, Project, Application), in which the script is active.

Note — Name assignment

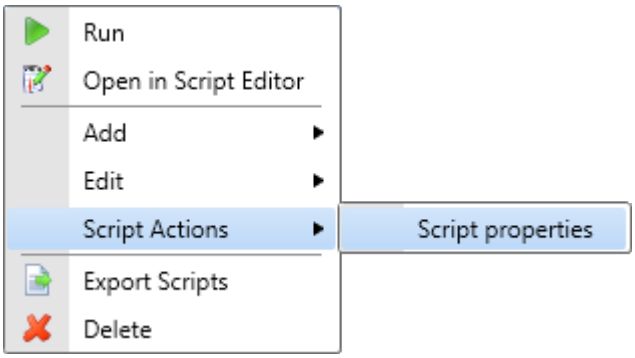
- Note that the same script **name** may not be used multiple times, even when using different **Script Storage Scopes.**
- **Example:**
  - In the **Experiment** *Exp_001*, a **script** "*Test*" is created with the **Storage Scope** *Experiment*.
  - In another **experiment**, *Exp_002*, a **script** "*Test*" is created with the **Storage Scope** *Project.*
  - The **experiment** *Exp_001* is loaded, and a warning appears in the logbook, indicating that the **script** "*Test*" cannot be loaded since it already exists (in the project).

17.1.1.1.3 Storage scope

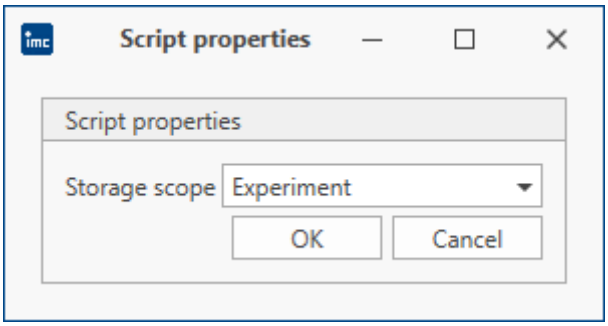
The option **Storage Scope** (*Save With / Script storage)* specifies the script's location in memory.
In order to use scripts in other experiments belonging to the same project, the script is created with the setting *Save with > Project.*

Storage Scope	Description
Experiment	The script is saved in the *experiment folder* and is also only available within the experiment.
Project	The script is saved to the *project folder* and is listed and available to all experiments belonging to the project. If you wish to transfer the experiment to another system, the project-script must also be exported and transferred along with it.
Application	The script is saved to the a*pplications-folder* (e.g. C:\ProgramData\imc\imc STUDIO\Applications_1\Scripts) and is available for all projects and all experiments. Here, too, for the purpose of transferring the experiment to a different system, the script must be exported and transferred along with it.
Sequencer	If a (different) experiment is loaded by means of a command, then this script remains available.

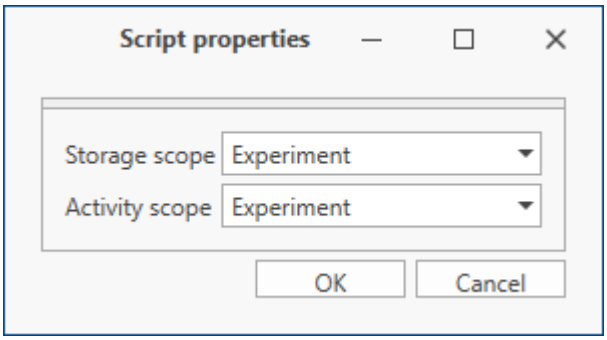
This option can be changed in the **context menu** *Script Actions > Script properties*.



Script Actions: Script properties



Dialog: Script properties



Dialog: Script properties (for Context-Scripts)

Warning Save Project and Experiment

Please **save** your **project** and **experiment** after **changing** the S*torage Scope*, otherwise changes might be lost.

17.1.1.2 Edit scripts

To edit a script, please open the tool window **Scripts** and click on *Open in Script Editor* Open in Script Editor. You can also double-click on a script to edit the script. Before editing another script, the Script Editor must be closed.

Note

The scripts are deposited beyond imc STUDIO.
The save state of the scripts is only transferred, if the corresponding *storage scope* (Experiment, Project) is **saved**.

17.1.1.3 Run scripts

To run a script, the following mechanism are available depending on the script type:

Script type	Mechanism

Script 1878	• Command in theSequencer
	• Command on an event.
	• Tool window scripts
	  ○ Menu
	  ○ Context menu

Panel script 1879	• Browsing to a page (`Run` method) • Leaving the page (`Stop` method)
Context script 1883	• Loading/opening the corresponding context (`Run` method) • Leaving/closing the context (`Dispose` method)
Type library script 1887	• Cannot be executed. • Is used in other scripts.
Event script 1889	• Command on an event.
Third Party Device Script 1918	• Cannot be executed • Refresh of the Third Party Device after closing the editor.

After running the script using **Run**, you can display the Build-Log by selecting *Edit > Show Build log* in the context menu.

If the script runs successfully, it will be indicated by a green dot symbol in the tool window.

Otherwise, if there is any error, a red dot symbol will be displayed.

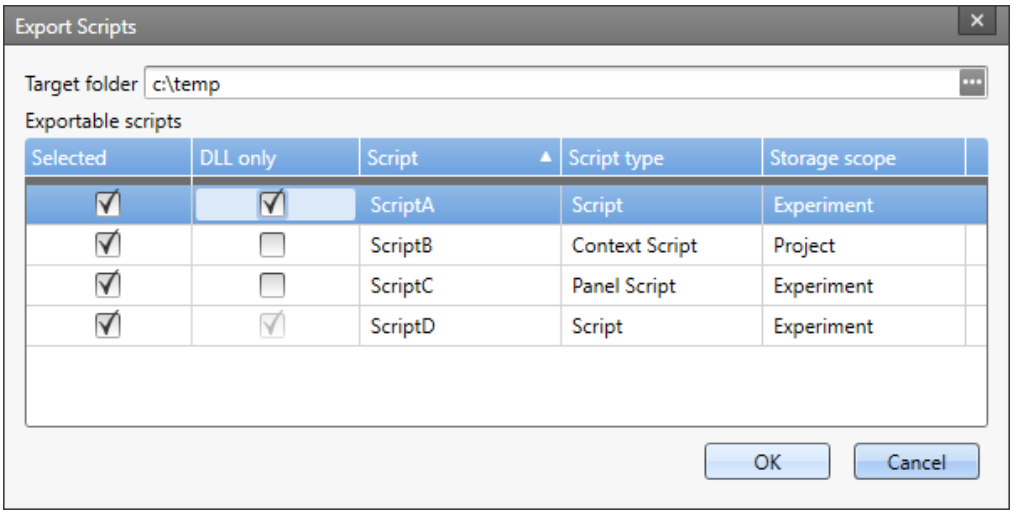
17.1.1.4 Export and import

Note

If **Resources** are used in the *C#-Project*, please get sure, that they the property "Build action" is set to "*Resource*" or "*Embedded Resource*" and/or "Copy to output directory" is set to "*true*".
With these settings the Resources folder is considered when using **DLL-Expor**t.

Export

To export scripts, click in the **Scripts** tool window on *Export scripts* Export Scripts . Subsequently, a dialog showing all scripts opens.



Dialog: Export scripts

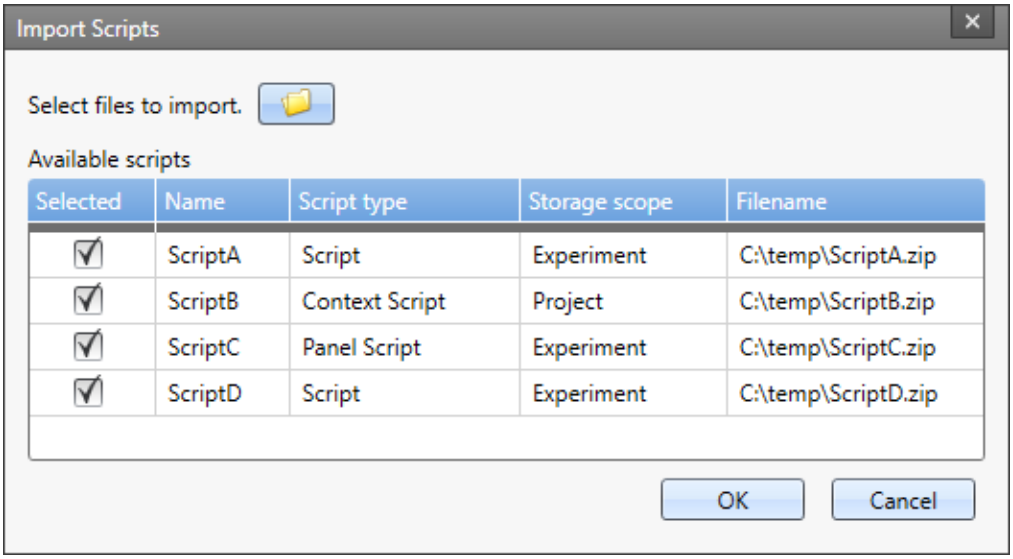
Here, select the scripts to export and enter a target path.
if only the library (DLL) of a script is to be exported, then make the corresponding selection in the column "only DLL".
The scripts are exported as an archive (*.zip). Along with the script-files, some information is also recorded on the type and storage type of the ZIP-file. This also applies to the DLL-exports.

Warning

If a script is exported as DLL, there might be a dependency to assemblies of the current version.
If an experiment with DLL-scripts is used in a newer version, there might be an FileNotFoundException error.
The developer of the script must recompile it in the newer version. See also, Important Information 1916.

Import

Click in the tool window **Scripts** on *Import Scripts* Import Scripts . The following dialog will appear:



Dialog: Add existing scripts

Choose the file at *Name.*

Depending on the script type, the dialog will change and further settings can be changed, e.g. Name and Storage Scope. Click on *OK* to add the script to the list. When scripts are imported which contain only a library (DLL), then of course these can not be edited but only executed.

Also, single script files (*.cs) can be added in the **ScriptEditor** to the C# project via *Add > Existing item.* Former script files with the extension *.csscript has to be renamed with the extension *.cs.

Warning

If you import a Panel script, you have to link the script to the corresponding Panel page in the page properties.

Note Add "Import Scripts" to the ribbon

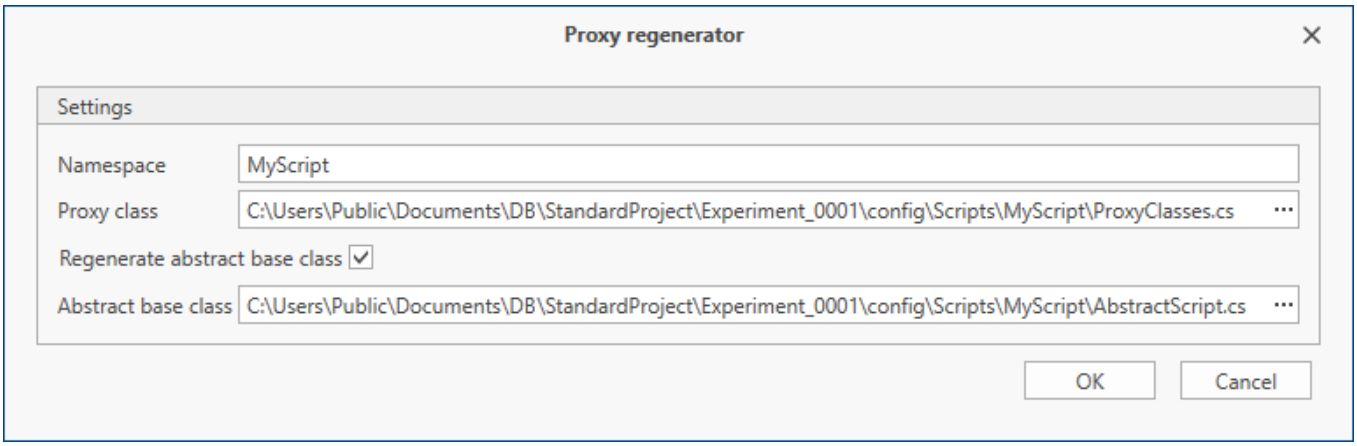
You can customize the ribbon and add the menu action "Import Scripts".
After adding the menu action the view needs to be saved.

17.1.1.5 Regenerate proxies

If a script of an old imc STUDIO version is imported and new functions are needed in this script, the proxy classes have to be regenerated.

Select the script in the tool window **Scripts** and click in the **context menu** on *Edit > Regenerate proxies.*

The following dialog opens.



Dialog: Proxy regenerator

Warning

Due to new functions, internal structures may have changed. Therefore after regenerating proxies, you have to check and change the source code to fix the compiler errors.
You will find hints in the chapter Important Information 1916.

17.1.1.6 Debugging

To **debug** a script, the following steps must be performed after compiling.

Step 1 : Add breakpoint

Press **<F7>** or use the menu item *Debug > Toggle Breakpoint* to add a breakpoint. The program counter will stop in this line, when reaching this statement.

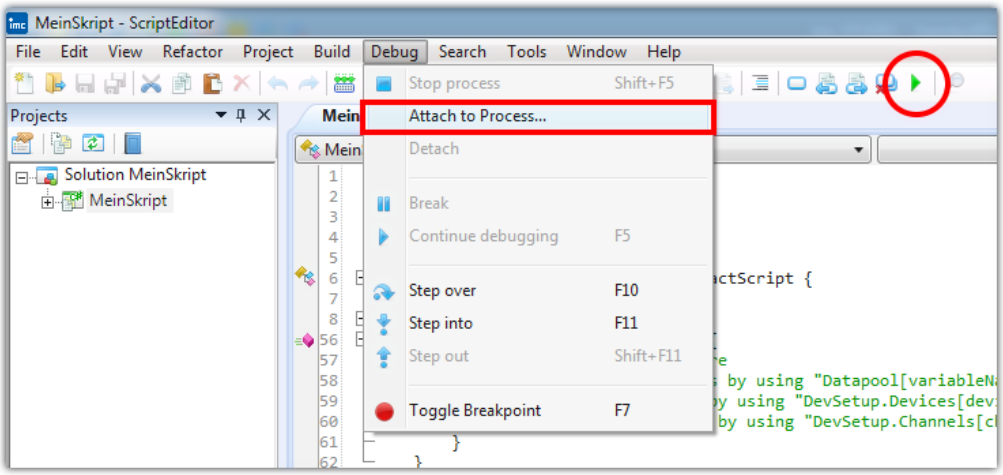
Afterward there is a red point left to the line number.

Editor: set Breakpoint

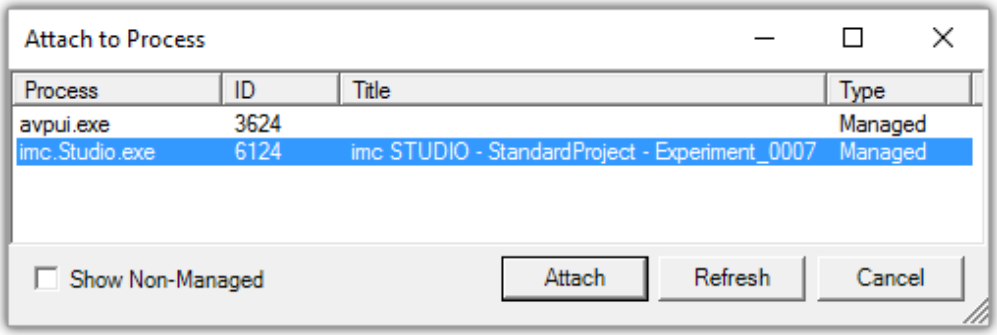
Step 2 : Attach to imc STUDIO process

Attach the Debugger to the running imc STUDIO process by using the menu item *Debug > Attach to Process* In the dialog, select *imc.Studio.exe* (if multiple processes are listed, select the task whose name matches the experiment's name).

The Play button (green triangle) in the menu bar can be used instead to attach the debugger directly to *imc.STUDIO.exe*.
Subsequently, the breakpoint's color is yellow .



Editor: attach to process



Process selection

Editor: Debugging is attached to process

Step 3 : Run the script

Run the script in imc STUDIO according to the script type.
Perform the actions needed to reach the breakpoint.

Step 4 : Program counter

The *Program Counter* (yellow arrow ➡) stops at the breakpoint, if this line is reached.

Editor: Program Counter

Step 5 : Navigate

Navigate through the script by using the keys **<F10>** and **<F11>**, or **<SHIFT>+<F11>**, continue debugging by using **<F5>**.

Key		Description (englisch)
	F5	Continue debugging
	Shift+ F5	Stop process
	F7	Toggle breakpoint
	F10	Step over
	F11	Step into
	Shift + F11	Step out

Warning Always use 'Continue Debugging' before detaching

Do not *detach* from the process while debugging. Depending on the user's programming, imc STUDIO might be in an undefined or non-operable state. Do always use *Continue Debugging* (blue "Play" symbol) to end debugging before detaching.

17.1.1.7 Classes

Class name	Description
Actions	Access to the menu actions: Commands 1870
Commands	Access to the Commands 1870
Core	Access to classes (needed in other script types) *also see table on the below*
DataProcessing	* Access to Data Processing
DevSetup	Device and channel settings (parameters): Device functions 1824, Channel functions 1832
Documents	Informationen about the project and the experiment.
Environment	Informationen about the software environment: Product Information 1873
Events	Access to events.
InlineFamos	Access to InlineFamos: Inline FAMOS functions 1868
Logbook	Access to the logbook: Logbook 1864
Notifications	Access to the notification system.
Panel	Access to the Panel and the Widgets: Panel and Widget functions 1840
ScriptClipboard	Script Clipboard 1875
Scripts	Access to the scripts
SharedComponents	*
StorageManager	Used in an Event Script 1889
UserRequest	Default dialog response 1874
VariableViewer	* Current values window
Variables	Access to variables: Variables functions 1848
Windows	Access to tool windows. Show and hide.

* Access not possible or restricted.

Core

Some of the classes above can also be accessed via the Core class. In Addition there are the following classes.

Class name	Description
Accessibility	* Access to the user administration
Options	*
Services	*

* Access not possible or restricted.

17.1.2 Access to functions

17.1.2.1 Device functions

Note

- For setting the parameters, class *EClassID* is used. Setting values of the parameter are available in the *ParameterValues*-Class. See also Device parameters 1891.
- The method `ExecuteActions()` is only needed for device and channel actions. For parameters the method `SetValue()` is used.

To use needed constants/classes the following **libraries** have to be added:

```
using imc.Studio.Tools.DevSetupGraphGenerator; // EClassID
using imc.Studio.Interfaces.DevSetup; // EDeviceCoreType_V2
```

The following line is needed for the following examples. The device *imcDev__17123456* is used. The device name is case sensitive.

```
var device = DevSetup.Devices["imcDev__17123456"];
```

The parameters are used as follows:

```
// EClassID_Parameter as string or from EClassID-Class
//var parameter = channel.Parameters["eMyParameter"];
var parameter = device.Parameters[EClassID.eDeviceName]; // Example
```

```
// set parameter value
parameter.SetValue("SetValue");
// get parameter value
string parameterValueAsText = parameter.GetValue().ToString();
// get printout of the parameter value
string parameterText = parameter.Text;
// select list of parameter values, if available
string [] paraTextArray = parameter.TextList;
```

For further information, please read the following chapters:

- Devices control and parameters 1825
- Storage 1827
- Diskstart 1828
- Misc 1830

17.1.2.1.1 Devices control and parameters

Search for devices

The device search is performed by the following command:

```
DevSetup.SearchDevices();
```

Add Device by DNS-Name or IP

Add a device by DNS-Name or IP by the following command:

```
using imc.Studio.Interfaces.DevSetup; // add Library
```

```
// IP
DevSetup.SearchDevice("192.168.0.2", DeviceAdapters.imcDevcies2x);
```

```
 // DNS
DevSetup.SearchDevice("imcDev__17123456", DeviceAdapters.imcDevcies2x);
```

```
// The return value of type "EnumItem" is the according device.
EnumItem device = DevSetup.SearchDevice("0.0.0.0", DeviceAdapters.imcDevcies2x);
```

Select a device

If the device name is unknown, the name can be determined by using the device's serial number.

```
// getting the device object via serial number
string serialNumber = "123456";
EnumItem device = DevSetup.Devices.GetItemBySerialNumber(serialNumber);
```

```
// getting the device identifier
string deviceName = device.Parameters[EClassID.eDeviceName].GetValue().ToString();
```

```
// getting the device name
string deviceNickName = device.Parameters[EClassID.eDeviceNickname].GetValue().ToString();
```

Actual selection of the device is based on the device name (*EClassID.eDeviceNickname*) or the device identifier (*EClassID.eDeviceName*).

```
using imc.Studio.Interfaces.DevSetup;
```

```
// selecting device via nickname
DevSetup.SelectDevice(deviceNickName, true, EDeviceCoreType_V2.Measurement);
```

```
// selecting device via name
DevSetup.SelectDevice(deviceName, false, EDeviceCoreType_V2.Measurement);
```

The device name can also be obtained by using the device list.

New parameters for the device are available, therefore the device object has to be assigned afterwards.

```
// via device name
EnumItem device = DevSetup.Devices[deviceName];
```

```
// via serial number
EnumItem device = DevSetup.Devices.GetItemBySerialNumber(serialNumber);
```

Deselect a device

```
DevSetup.DeselectDevice("DeviceNickName", true, EDeviceCoreType_V2.Measurement);
```

```
DevSetup.DeselectDevice("imcDev__17123456", false, EDeviceCoreType_V2.Measurement);
```

Remove device from List

```
DevSetup.RemoveDevice("DeviceNickName", true);
```

```
DevSetup.RemoveDevice("imcDev__17123456", false);
```

Device control actions

To connect to the device or start and stop the device, the following functions have to be executed.

```
var status = device.Parameters[EClassID.eDeviceControlAction];
// for action values see ParameterValues.eDeviceControlAction
```

```
// connect to device
status.SetValue(ParameterValues.eDeviceControlAction.Connect);
status.ExecuteAction();
```

```
// start measurement
status.SetValue(ParameterValues.eDeviceControlAction.Start);
status.ExecuteAction();
```

```
// stop measurement
status.SetValue(ParameterValues.eDeviceControlAction.Stop);
status.ExecuteAction();
```

```
// disconnect from device
status.SetValue(ParameterValues.eDeviceControlAction.Disconnect);
status.ExecuteAction();
```

For starting/stopping all selected devices:

```
DevSetup.ExecuteAction(ParameterValues.eDeviceControlAction.Start);
```

```
DevSetup.ExecuteAction(ParameterValues.eDeviceControlAction.Stop);
```

or

```
Actions.FireAction(this, "acDeviceConnect"); // connecting
```

```
Actions.FireAction(this, "acDeviceStart"); // starting
```

```
Actions.FireAction(this, "acDeviceStop"); // stopping
```

```
Actions.FireAction(this, "acDeviceDisconnect"); // disconnect
```

Query, if device is selected

```
bool isSelected = device.Parameters[EClassID.eConnectionStatus] != null;
```

List of selected devices (serial number)

```
using System.Collection.Generic;
using System.Linq; // may needs to be added to the references
using imc.Studio.Tools.DevSetupGraphGenerator;
```

```
List<string> serialNumbers = DevSetup.Devices
   .Where(device => device.Parameters[EClassID.eConnectionStatus] != null &&
device.Parameters[EClassID.eDeviceSN] != null)
   .Select(device => device.Parameters[EClassID.eDeviceSN].Text)
   .ToList();
```

Measurement state

```
int mState = (int)device.Parameters[EClassID.eMeasurementStatus].GetValue();
```

```
string stateText = device.Parameters[EClassID.eMeasurementStatus].Text;
```

Manufacturer

```
string manufacturer = device.Parameters[EClassID.eManufacturer].Text;
```

Warning

If *ExecuteAction* is used the according parameter/device should be read/assigned again due to internal changing instances. The consequence is a *NullReference* Exception. This could happen when selecting a device or reading TEDs, for instance.

Note

If several actions should be executed, e.g. multi-selection of devices, the methods `DevSetup.BeginTransaction()` and `DevSetup.EndTransaction()` can be used.

Example

Use the first device in the list

```
var deviceList = DevSetup.Devices.GetEnumerator();
deviceList.Reset();
deviceList.MoveNext(); // get next element in the list
EnumItem device = deviceList.Current;
if (device != null){
   string deviceName = device.Parameters[EClassID.eDeviceName].GetValue().ToString();
}else{
   // do something, e.g.
    deviceName = String.Empty;
}

// Alternative
using System.Linq;
// ...
EnumItem device = DevSetup.Devices.First(); // first known device in the list
```

17.1.2.1.2 Storage

Device specific options - Device

```
// Storage interval ; 0 = End of measurement
device.Parameters[EClassID.eUMIntervalTime].SetValue(60d); // 60 seconds
// Storage interval count
device.Parameters[EClassID.eUMIntervalCount].SetValue(128); // all = 0
// Storage locaction: 0: Removable medium, 1: Hard disk (HD); 2: Network drive
device.Parameters[EClassID.eUMDrive].SetValue(0);
```

```
// Save trigger events in individual files (valid for ALL devices!)
device.Parameters[EClassID.eExtraTriggerDirUMHD].SetValue(1);
```

Delete files on selected disk

All data on the selected storage devices are **deleted**.

Please assure, that the device storage location is set correctly, e.g. to the *removable medium*.

```
device.Parameters[EClassID.eUMDrive].SetValue(0); // Removable medium
device.Parameters[EClassID.eDeleteFilesOnSelectedDisk].ExecuteAction();
```

Experiment options (applies to all devices) - PC

These options can be set by setting the parameter of a device, but it will be applied for **all** devices in the experiment.

```
// Storage interval ; 0 = End of measurement
device.Parameters[EClassID.ePCIntervalTime].SetValue(60d); // 60 seconds
// Storage interval count
device.Parameters[EClassID.ePCIntervalCount].SetValue(128); // all = 0
// Save trigger events in individual files
device.Parameters[EClassID.eExtraTriggerDirPCHD].SetValue(true);
device.Parameters[EClassID.eExtraTriggerDirUMHD].SetValue(true);
```

17.1.2.1.3 Diskstart

Write Diskstart to destination

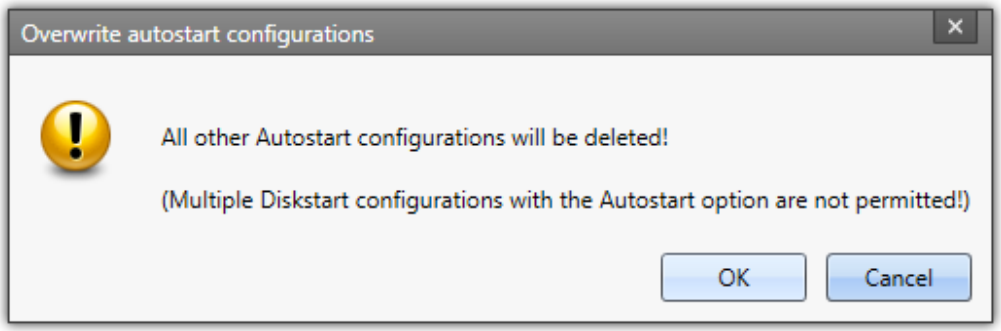
```
device.Parameters[EClassID.eDiskStartSelfStartOption].SetValue(true); // Self-start
```

```
var status = device.Parameters[EClassID.eDiskStartAction];
// See ParameterValues.eDiskStartAction for setting values, e.g.
// 2503: Device HD; 2505: Device intern;
```

```
// saving to removable device
status.SetValue(ParameterValues.eDiskStartAction.Device_PCMCIA);
status.ExecuteAction();
```

```
// saving to PC-HD
status.SetValue(ParameterValues.eDiskStartAction.Device_PC_HD);
status.ExecuteAction();
```

When writing an self-start experiment to the device, a dialog which has to be confirmed **manually** appears.



Dialog: Autostart

If you want to suppress this dialog, open the *Options-Dialog* and go to *General Options > Default dialog response > Overwrite autostart configurations* and choose **OK** from the selection.

This can also be done with Scripting (see chapter Default dialog response 1874):

```
using imc.Studio.Interfaces.Gui;
```

```
UserRequest.SetDefaultResultValue("Dev2x_OverwriteSelfStart", EMessageBoxResult.OK);
```

Deletes Diskstart in every selected device

```
foreach(var config in DevSetup.Items[EClassID.eConfig])
{
   var deleteAction = config.Parameters[EClassID.eDiskStarDeleteConfig];
   if((deleteAction != null) && (deleteAction.Base != null))
   {
       // deleteAction.SetValue(ParameterValues.eDiskStarDeleteConfig.Delete);
       deleteAction.ExecuteAction();
    }
}
```

Get Diskstart information

```
// Experiment-Name
var names = DevSetup.Items[EClassID.eConfig]
   .ToList()
   .Where(config => config.Parameters[EClassID.eDiskStartConfigName] != null)
   .Select(config => config.Parameters[EClassID.eDiskStartConfigName].Text);
```

```
// Storage location
var location = DevSetup.Items[EClassID.eConfig]
   .ToList()
   .Where(config => config.Parameters[EClassID.eDiskStartLocation] != null)
   .Select(config => config.Parameters[EClassID.eDiskStartLocation].Text);
```

```
// Show information
if (names.Any() && location.Any())
{
   MessageBox.Show(names.First() + "\n" + location.First());
}
```

17.1.2.1.4 Supplemental Files

Access to **supplemental files management** is available through `DevSetup.SupplementalFiles`.

If an errors occurs, an exception is thrown and must be handled accordingly.

Hardware ID

Via the parameter of the device name the hardware ID of the device can be queried:

```
string hardwareID = device.Parameters[EClassID.eDeviceName].HWID;
```

File names

To get a list of file names currently present in the device, one of the following functions is used.

```
string[] fileNames = DevSetup.SupplementalFiles.GetFileNamesByDeviceSerialNumber(string
deviceSerialNumber);
string[] fileNames = DevSetup.SupplementalFiles.GetFileNamesByDeviceHWID(string deviceHWID);
```

Import

The import of additional files (e.g. *.ofa) is done via one of the following functions.

```
DevSetup.SupplementalFiles.ImportByDeviceSerialNumber(string deviceSerialNumber, string
configurationFileName);
DevSetup.SupplementalFiles.ImportByDeviceHWID(string deviceHWID, string configurationFileName);
```

Export

The export of supplemental files (e.g. *.ofa) is done using one of the following functions. It is recommended to check the correct name of the file beforehand using `GetSupplementalFileNames`.

```
DevSetup.SupplementalFiles.ExportByDeviceSerialNumber(string deviceSerialNumber, string
supplementalFileName, string targetFileName);
DevSetup.SupplementalFiles.ExportByDeviceHWID(string deviceHWID, string supplementalFileName, string
targetFileName);
```

Delete

The deletion of supplemental files is done using one of the following functions. It is recommended to check the correct name of the file beforehand using `GetSupplementalFileNames`.

No Online FAMOS files should be deleted, otherwise an error message will occur when starting the device.

To "deactivate" the OFA code, the additional file should be replaced by an **empty** OFA code file.

```
DevSetup.SupplementalFiles.DeleteByDeviceSerialNumber(string deviceSerialNumber, string fileName);
DevSetup.SupplementalFiles.DeleteByDeviceHWID(string deviceHWID, string fileName);
```

17.1.2.1.5 Misc

Clock type and clock state

```
// Clock type
var clockType = device.Parameters[EClassID.eDeviceClockType].Text;
// Clock state
var clockState = device.Parameters[EClassID.eDeviceClockState].Text;
```

Synchronization

Parameter	ID
Signal input	eSynchSignalInput
Signal output	eSynchSignalOutput
Synchronization master (device)	eSynchMaster
Clock type	eDeviceClockType
Clock state	eDeviceClockState
External synchronization sources	eExternalSynchSignal

You will find the setting values for signal input and output, clock type and state in the ParameterValues class.

Timed start

Parameter	ID
Start option	eStartOption
Start date	eStartDate
Start time	eStartTime
Time offset	eStartTimeOffset
Synchronous start	eSynchStartFlag

You will find the setting values for the start option in the ParameterValues class.

NTP parametes

Parameter	ID	Type
Synchronization NTP Server (1)	eSyncNTPServer1	string
Synchronization NTP Server (2)	eSyncNTPServer2	string
Synchronization waiting time on self start max. [s]	eSyncMaxWaitingTimeOnSelfStart	int
Synchronization NTP deviation time max. [ms]	eSyncNTPMaxDeviation	int
NTP synchronization interval [s] (min;max)	eSyncNTPSyncInterval	string

Example Code:

```
device.Parameters[EClassID.eSyncNTPServer1].SetValue("99.99.99.99");
device.Parameters[EClassID.eSyncNTPServer2].SetValue("11.11.11.111");
device.Parameters[EClassID.eSyncMaxWaitingTimeOnSelfstart].SetValue(1); // int
device.Parameters[EClassID.eSyncNTPMaxDeviation].SetValue(1); // int
device.Parameters[EClassID.eSyncNTPSyncInterval].SetValue("128;1024"); // string
```

Get the fieldbus slot number

Alternative 1 (LINQ):

```
using System;
using System.Linq;
using System.Windows.Forms;
using imc.Studio.Tools.DevSetupGraphGenerator;
```

```
var serialNumber = "123456";
var device = DevSetup.Devices.GetItemBySerialNumber(serialNumber);
var moduleType = ParameterValues.eModuleType.MT_MFB_CANBUS;

var slotID = device.ChildItems
   .Where(child => child.Parameters[EClassID.eModuleType] != null &&
      (int)child.Parameters[EClassID.eModuleType].GetValue() == moduleType)
   .Select(child => child.Parameters[EClassID.eMFBSlotId].Text);

// Gets the first fieldbus slot number
if (slotID != null && slotID.Any()){
   MessageBox.Show("LINQ: " + slotID.First());
}
```

Alternative 2 (foreach):

```
using System;
using System.Linq;
using System.Windows.Forms;
using imc.Studio.Tools.DevSetupGraphGenerator;
```

```
var serialNumber = "123456";
var module Type = ParameterValues.eModuleType.MT_MFB_CANBUS;
var slot= GetSlotID(moduleType, serialNumber );
System.Windows.Forms.MessageBox.Show("(): " + slot);
```

```
private string GetSlotID(int eClassID, string serialNumber){
  string slot ="unkown Slot ID";
  var device = DevSetup.Devices.GetItemBySerialNumber(serialNumber);

  foreach (var para in device.ChildItems ) {

      var paraModuleType = para.Parameters[EClassID.eModuleType];

      if (paraModuleType != null){
         int moduleType = (int)paraModuleType.GetValue();
         if ( eClassID == moduleType) {
             slot = para.Parameters[EClassID.eMFBSlotId].Text;
             return slot;
         }

      } // ChannelTypeVal
  } // end of foreach
  return slot;
}
```

17.1.2.2 Channel functions

Note

- For addressing the parameters, class *EClassID* is used. Possible values of the parameter are available in the *ParameterValues*-Class. See also Channel parameters 1893.
- The method `ExecuteActions()` is only needed for device and channel actions. For parameters the method `SetValue()` is used.
- The `Channels` class can only be used to access the settings of the channels. The data is available via the VariableManager in the `Variables` class, see Variables functions 1848.

The following line is needed for the following examples. The channel *Channel_001* is used. The channel name is case sensitive.

```
var channel = DevSetup.Channels["Channel_001"];
```

Setting parameters of bits (e.g. comment):

```
var port = DevSetup.Channels["DIN001"];
var bit = port.ChildItems["DIN001_Bit01"];
bit.Parameters[EClassID.eChannelComment].SetValue("MyComment");
```

To use the class *EClassID* the following **library** has to be added:

```
using imc.Studio.Tools.DevSetupGraphGenerator;
```

The parameters are used as follows

```
// EClassID_Parameter as string or from EClassID-Class
//var parameter = channel.Parameters["eMyParameter"];
var parameter = channel.Parameters[EClassID.eSampleTime]; // Example
```

Set parameter value

```
parameter.SetValue("SetValue");
```

Get parameter value

```
string parameterValue = parameter.GetValue().ToString();
```

Get printout of the parameter value

```
string parameterText = parameter.Text;
```

Select list of parameter values, if available

```
string [] paraTextArray = parameter.TextList;
```

Metadata columns can be edited via the identifier (here: *MyIdentifier*).

```
var parameter = channel.Parameters["MyIdentifier"];
```

```
// get or set a parameter value
parameter.SetValue("MySettingValue");
string parameterText = parameter.GetValue().ToString();
```

For further information, please see the following chapters:

- Parameter 1833
- Data transfer 1838
- Curve window 1839

17.1.2.2.1 Parameter

Channel name

```
var deviceParameter = DevSetup.Devices["DeviceName"].Parameters;
var channelParameter = DevSetup.Channels["ChannelName"].Parameters;
```

```
// Change Channel name
channelParameter[EClassID.eChannelName].SetValue("Channel_001_new");
```

Balance actions

```
var paraBalanceAction = channel.Parameters[EClassID.eBalanceAction];
```

```
// set offset
paraBalanceAction.SetValue(ParameterValues.eBalanceAction.Offset);
paraBalanceAction.ExecuteAction();
```

```
// set bridge
paraBalanceAction.SetValue(ParameterValues.eBalanceAction.Bridge);
paraBalanceAction.ExecuteAction();
```

```
// set tare
paraBalanceAction.SetValue(ParameterValues.eBalanceAction.Tare);
paraBalanceAction.ExecuteAction();
```

```
// set manufacture's calibration
paraBalanceAction.SetValue(ParameterValues.eBalanceAction.Reset);
paraBalanceAction.ExecuteAction();
```

Balance state

```
string balanceState = channel.Parameters[EClassID.eBalanceStatus].Text;
```

Cable compensation

```
var paraCalibAction = channel.Parameters[EClassID.eCalibrationAction];
```

```
paraCalibAction.SetValue(ParameterValues.eCalibrationAction.CableCompensation); // Cable
channel.Parameters[EClassID.eCalibrationAction].ExecuteAction();
```

```
double cableResistance = (double)channel.Parameters[EClassID.eCableResistor].GetValue();
```

Channel comment

```
channel.Parameters[EClassID.eChannelComment].SetValue("My Channel");
```

Channel name via connector

The channel name of connector "[01] IN01" is queried here. The channel object with its parameters is In the variable *query*.

```
using System.Linq;
```

```
string plugInName = "[01] IN01";
var channels = DevSetup.Channels
   .Where(channel => channel.Parameters[EClassID.ePlugInName] != null &&
channel.Parameters[EClassID.ePlugInName].Text == pluginName);
```

```
// read channel name
string name = string.Empty;
if (channels != null){
   name = channels.First().Parameters[EClassID.eChannelName].Text;
}
```

```
// write channel name
string name = "MyChannel";
if ( channels != null ){
   channels.First().Parameters[EClassID.eChannelName].SetValue(name);
}
```

If there is a problem with the *where* query, please use *foreach* instead.

Channel status

- The channel name must exist and the value to be set must be valid for this type of channel.
- For each active channel, a variable with the same name is created and made available in the `Variables` class after the Process action.

```
// setting status of channel "Channel_1" to "active"
channel.Parameters[EClassID.eStatus].SetValue(ParameterValues.eStatus.Active);
```

```
// setting status to "passive"
channel.Parameters[EClassID.eStatus].SetValue(ParameterValues.eStatus.Passive);
```

```
// setting status to "write"
channel.Parameters[EClassID.eStatus].SetValue(ParameterValues.eStatus.Write);
// setting status to "read/write"
channel.Parameters[EClassID.eStatus].SetValue(ParameterValues.eStatus.Read_Write);
```

Connector / Enumerated channel number

```
// Connector name
string connectorName = channel.Parameters[EClassID.ePlugInName].Text;
```

```
// Enumerated channel number
string channelNumber = channel.Parameters[EClassID.eEnumeratedChannelNumber].Text;
```

Coupling

```
channel.Parameters[EClassID.eCoupling].SetValue(ParameterValues.eCoupling.AC);
channel.Parameters[EClassID.eCoupling].SetValue(ParameterValues.eCoupling.DC);
// See ParameterValue.eCoupling for more setting values
```

Duration

Duration of measurement can be set by the following code whereas the argument is time in seconds.

```
double duration = 60d;
channel.Parameters[EClassID.eDuration].SetValue(0); // undefined
channel.Parameters[EClassID.eDuration].SetValue(duration); // 60 seconds
```

Filtering

The filtering is set to a Butterworth-Bandpass with cutoff-frequencies 10 Hz and 50 Hz.

```
var paraFilterChar = channel.Parameters[EClassID.eFilterCharacteristic];
paraFilterChar.SetValue(ParameterValues.eFilterCharacteristic.Butterworth);

var paraFilterType = channel.Parameters[EClassID.eFilterType];
paraFilterType.SetValue(ParameterValues.eFilterType.Bandpass);

channel.Parameters[EClassID.eFiltCutoff1].SetValue(10); // 10 Hz
channel.Parameters[EClassID.eFiltCutoff2].SetValue(50); // 50 Hz
```

Input range

Input range is set to ±2.5V.

```
string str = "\u00b1" + "2.5 V"; // \u00b1 = PlusMinus-Symbol
channel.Parameters[EClassID.eRange].SetValue(str);
```

Input range of channel *Channel_001* is set to a text variable.

```
using imc.Studio.PlugIns.DevSetup.Common;
using imc.Studio.Tools.DevSetupGraphGenerator;
```

```
var paraRange = DevSetup.Channels["Channel_001"].Parameters[EClassID.eRange];
string sRange = paraRange.Text; // Display value of range, e.g. "±5 V"
```

Remarks**:** The only values which can be set are available in the selection list on the Setup page.

A list of possible ranges can be obtained by the following code:

```
var pRange = DevSetup.Channels["Channel_001"].Parameters[EClassID.eRange];
string [] sRange = pRange.TextList;
```

Module type: UNI2-8
Input range
±50 V
±25 V
±10 V
±5 V
±2.5 V
±1 V
±0.5 V
±0.25 V
±0.1 V
±0.05 V
±0.025 V
±0.01 V
±0.005 V

Values for other module types are listed in the corresponding data sheets.

Module type

```
// Return value is language dependent
string sModuleType = channel.Parameters[EClassID.eModuleType].Text;
// -> Bridge, current and voltage amplifier and temperature measurement unit "UNI2-8"
```

Preprocessing

In the following code an arithmetic mean value over 5 sample points is set as preprocessing.

```
var paraProcess = channel.Parameters[EClassID.eProcessing];
paraProcess.SetValue(ParameterValues.eProcessing.ArithMean);
channel.Parameters[EClassID.eProcessingPoints].SetValue(5);
```

Sample time / sample rate

Sample Time is set to 1 ms:

```
// setting Sample Time to 0.001s = 1ms = 1000 Hz
channel.Parameters[EClassID.eSampleTime].SetValue(0.001);
```

Remarks: The only values which can be set are available in the selection list on the Setup page.

A **module-specific** list of possible sample times can be obtained by the following code:

```
var pSampleTime = DevSetup.Channels["Channel_001"].Parameters[EClassID.eSampleTime];
string [] sSampleTimeList = pSampleTime.TextList;
```

Module type: UNI2-8	
Sample rate	**Sample time**
100 kHz	10 µs
50 kHz	20 µs
10 kHz	100 µs
5 kHz	200 µs
1 kHz	1 ms
500 Hz	2 ms
200 Hz	5 ms
100 Hz	10 ms
50 Hz	20 ms
20 Hz	50 ms
10 Hz	100 ms
5 Hz	200 ms
2 Hz	500 ms
1 Hz	1 s
0.5 Hz	2 s
0.2 Hz	5 s
0.1 Hz	10 s
0.05 Hz	20 s
0.0333333 Hz	30 s
0.0166666 Hz	60 s

Values for other module types are listed in the corresponding data sheets.

Sensitivity/Scale and Offset

The scaling and offset of an channel are set in the following code.

```
channel.Parameters[EClassID.eUserUnit].SetValue("Pa"); // Pascal
channel.Parameters[EClassID.eUserOffset].SetValue(1); // 1 Pa
channel.Parameters[EClassID.eUserScalingFactor].SetValue(5); // = 1/Sensitivity
```

Strain gauge

```
Coupling
channel.Parameters[EClassID.eCoupling].SetValue(ParameterValues.Coupling.HalfBridge;
```

```
Bridge resistor
channel.Parameters[EClassID.eBridgeResistor].SetValue(120);
```

```
Bridge mode
channel.Parameters[EClassID.eBridgeBridgeMode].SetValue(1);
// 1=>strain; 2=>stress
```

```
Bridge factor N
channel.Parameters[EClassID.eBridgeN].SetValue(1);
// 1=>1; 2=>2; 4=>3; (1-v)=>4; (1+v)=>5; 2*(1-v)=>6; 2*(1+v)=>7
```

```
Gauge factor k
channel.Parameters[EClassID.eBridgeFactor].SetValue(1);
```

```
Bridge unit
channel.Parameters[EClassID.eBridgeUnit].SetValue(1);
```

```
Transverse strain coefficient v
channel.Parameters[EClassID.eBridgeEps].SetValue(0.3);
```

```
Modolus of elasicity
channel.Parameters[EClassID.eBridgeEModule].SetValue(1);
```

TEDs: Read/Write sensor information

Warning

If *ExecuteAction* is used the according parameter/device should be read/assigned again due to internal changing instances. The consequence is a *NullReference* Exception. This could happen when selecting a device or reading TEDs, for instance.

```
// read
var paraReadSensor = channel.Parameters[EClassID.eSensorAction];
paraReadSensor.SetValue(ParameterValues.eSensorAction.ReadSensorFlash);
paraReadSensor.ExecuteAction();
```

```
// write
var paraReadSensor = channel.Parameters[EClassID.eSensorAction];
paraReadSensor.SetValue(ParameterValues.eSensorAction.WriteSensorFlash);
paraReadSensor.ExecuteAction();
```

```
// delete
var paraReadSensor = channel.Parameters[EClassID.eSensorAction];
paraReadSensor.SetValue(ParameterValues.eSensorAction.ResetChanWithSensorh);
paraReadSensor.ExecuteAction();
```

Two-Point scaling

Set the two-point values as `string`. If they are set as `double`, the scaling factor and offset is considered.

```
channel.Parameters[EClassID.ePoint1SetValue].SetValue("20");
channel.Parameters[EClassID.ePoint1BalanceAction].ExecuteAction();
```

```
channel.Parameters[EClassID.ePoint2SetValue].SetValue("50");
channel.Parameters[EClassID.ePoint2BalanceAction].ExecuteAction();
```

```
var paraCalibAct = channel.Parameters[EClassID.eCalibrationAction];
paraCalibAct.SetValue(ParameterValues.eCalibrationAction.TwoPoint); // Two Point
channel.Parameters[EClassID.eCalibrationAction].ExecuteAction();
```

Unit

To use the types CValueArray the following reference must be added:
imc.Studio.PlugIns.DevSetup.Common

Also add the following using-Directive to the script file: `using imc.Studio.PlugIns.DevSetup.Common`

Escaping of quotation marks (") by using backslash (\).

```
// setting User Unit to N/m
channel.Parameters[EClassID.eUserUnit].SetValue("\"N/m\"");
```

```
// reading unit
var paraUnit = DevSetup.Channels["Channel_001"].Parameters[EClassID.eUserUnit];
object oUnit = paraUnit.GetValue();
CValueArray cUnit = oUnit as CValueArray;
string sUnit = cUnit[0].ToString(); // unit is the element with index 0
```

Balance at Device start

Set the following parameter to perform a balance action at the device start (Diskstart/Self-start):

```
channel.Parameters[EClassID.eBalanceAtDeviceStart].SetValue(true);
```

17.1.2.2.2 Data transfer

Save data (Device)

```
channel.Parameters[EClassID.eUMTransferToHD].SetValue(true);
```

Buffer time (RAM)

```
channel.Parameters[EClassID.eBufferTime].SetValue(60d); // 60 seconds; auto = 0
```

Transfer to PC

```
// activate
channel.Parameters[EClassID.eTransferToPC].SetValue(true);
// deactivate
channel.Parameters[EClassID.eTransferToPC].SetValue(false);
```

Save data (PC)

```
// activate saving to Hard Disk
channel.Parameters[EClassID.ePCTransferToHD].SetValue(true);
// deactivate
channel.Parameters[EClassID.ePCTransferToHD].SetValue(false);
```

Display, calculations of measured data

```
// Available Events (Display) ; 0 = all; 1 = last
channel.Parameters[EClassID.eDisplayAvailableEvents].SetValue(0);
```

```
// Circular buffer time (Display)
channel.Parameters[EClassID.eDisplayRingTime].SetValue(60d);
```

17.1.2.2.3 Curve window

Display Time / Circular Buffer Time (Curve Window)

```
channel.Parameters[EClassID.eDisplayRingTime].SetValue(60d);
```

Graph Color

Does not change during measurement.

```
// preset Color
int color = Color.Tomato.ToArgb();
// via RGB
int color = Color.FromArgb(0,255,255).ToArgb();
// set
channel.Parameters[EClassID.eCurveColor].SetValue(color);
```

y-Axis (Scale)

```
channel.Parameters[EClassID.eCurveYAxisOption].SetValue(2);
// Input-Range (1), user-defined (2), auto (3)
channel.Parameters[EClassID.eCurveYAxisMax].SetValue(20d);
channel.Parameters[EClassID.eCurveYAxisMin].SetValue(-20d);
```

17.1.2.2.4 Examples

Saving all active analog channels to hard disk drive

```
using System.Linq;
using imc.Studio.Tools.DevSetupGraphGenerator;

public override void Run()
{
   DevSetup.Channels
      .Where(channel => IsAnalogChannel(channel) && IsActive(channel))
      .ToList()
      .ForEach(channel => SaveToHardDiskDrive(channel));
}

private bool IsAnalogChannel(EnumItem channel)
{
   bool isAnalog = false;
   var channelTypeParameter = channel.Parameters[EClassID.eChannelType];
   if (channelTypeParameter != null)
   {
      var channelTypeValue = channelTypeParameter.GetValue();
      isAnalog = (int)channelTypeValue == ParameterValues.eChannelType.AnalogInput;
   }

   return isAnalog;
}

private bool IsActive(EnumItem channel)
{
   bool isActive = false;
   var statusParameter = channel.Parameters[EClassID.eStatus];
   if (statusParameter != null)
   {
      var statusValue = statusParameter.GetValue();
      isActive = (int)statusValue == ParameterValues.eStatus.Active;
   }

   return isActive;
}

private void SaveToHardDiskDrive(EnumItem channel)
{
   var saveParameter = channel.Parameters[EClassID.ePCTransferToHD];
   if (saveParameter != null)
   {
      saveParameter.SetValue(true);
   }
}
```

17.1.2.3 Panel and Widget functions

Here you find the function to access the Panel and the Widget:

- Panel 1841
- Widget 1843
- Tables 1845
- Curve window 1847

17.1.2.3.1 Panel

Get the active Panel page

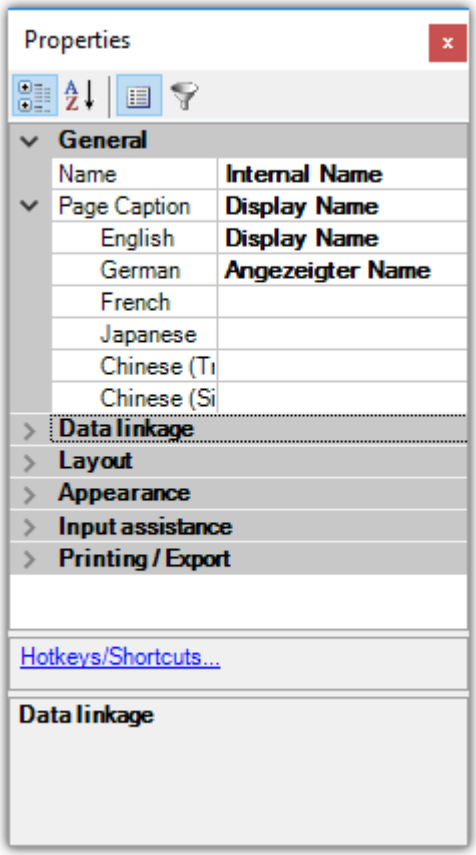
```
PanelPage page = Panel.GetActivePage();
```

Set Panel page with name "Page 1" as active

```
string pageName = "Internal Name";
Panel.SetActivePage(Panel.GetIndexOf(pageName)); // index is needed
```

Note

- For the page name `pageName` the internal name (*Name*) is used and not the displayed name (*Page Caption*) is not used.
- When renaming the *Page Caption* of a side, the *Name* property should also be adapted.



Page properties

Export Panel page

```
using imc.Common.Interfaces.Panel.Scripting;
```

```
Panel["Page 1"].Export(eFileFormat.PDF, @"c:\temp\myPage.pdf");
Panel["Page 1"].Export(eFileFormat.BMP, @"c:\temp\myPage.png");
Panel["Page 1"].Export(eFileFormat.DBV, @"c:\temp\myPage.dbv");
```

Import Panel page

```
string filename = @"c:\temp\myPage.dbv";
int pageIndex = 0; // page index, after the page has to be imported
Panel.InsertPageAfter(pageIndex, filename, "Page 1", "Page 1");
```

```
string filename = @"c:\temp\myPage.dbv";
int pageIndex = 0; // page index, before the page has to be imported
Panel.InsertPageBefore(pageIndex, filename, "Page 1", "Page 1");
```

Delete Panel page

```
Panel["Page 1"].DeletePage();
```

Rename Panel page

```
string error = string.Empty;
Panel["Page 1"].SetPageName("Page renamed", out error);
```

Create a Windows.Forms on the Panel page

It is recommended to create this form in a Panel script in the Initialize-method.

```
using System.Windows.Forms;
```

```
Control MyForm = new Control();
// ... configure form
Panel["Page 1"].Controls.Add(MyForm); // add form
```

The Windows.Forms are deleted when entering the design-mode. It is recommended to use a **Panel Script**.

Events for "Page added" and "Page deleted"

The Events "PageCreated" and "PageDeleted" can be used to react on adding or deleting a Panel page

These can be used in a **Context script**.

```
// Run
Panel.PageCreated += new Action<object, PanelPageEventArgs>(OnPageCreated);
Panel.PageDeleted += new Action<object, PanelPageEventArgs>(OnPageDeleted);
```

```
void OnPageDeleted(object arg1, PanelPageEventArgs arg2)
{
   string text = "New page: " + arg2.PageCaption + " (" + arg2.PageName + ") was delete from position "
+ arg2.Position.ToString() + ".";
   Core.Logbook.LogEntry("CheckPanel", text, 0 , ELogbookEntryCategory.Information);
}

void OnPageCreated(object arg1, PanelPageEventArgs arg2)
{
   string text = "New page: " + arg2.PageCaption + " (" + arg2.PageName + ") was added at position " +
arg2.Position.ToString() + ".";
   Core.Logbook.LogEntry("CheckPanel", text, 0 , ELogbookEntryCategory.Information);
}
```

```
// Dispose
protected override void CheckPanelDispose() {
   // Release all your resources and event hooks here.
   Panel.PageCreated -= OnPageCreated;
   Panel.PageDeleted -= OnPageDeleted
}
```

Example — Adding a button to a Panel page

A button with the text "MyText" is added to a page at a specific position with a given size. An action can be added, when clicking on the button.

```
using System.Windows.Forms;
```

```
Button button = new Button();
button.Text = "MyText";
button.Bounds = new Rectangle(100, 50, 100, 50);
// button.Click += new EventHandler(eventFunction);
Panel["Page 1"].Controls.Add(button as Control);
```

Subsequently, the Windows-control can be copied to the script clipboard.

17.1.2.3.2 Widget

The properties of a Widget can be changed by using `Panel["Page name"]["Widget name"]`. The page name and Widget are case sensitive.

Be sure, that the property is available for the Widget.

Position and size

```
int x = 10; // x-position
int y = 50; // y-position
int h = 100; // height
int w = 200; // width
Panel["Page 1"]["Standard_meter1"].Bounds = new Rectangle(x,y,h,w);
```

Several properties

```
Widget w = Panel["Page 1"]["Standard_meter1"];
//Panel["PageName"]["WidgetName"].PROPERTY = VALUE;
```

```
// Range
w.Max = 30;
w.Min = 10;
w.Interval = 10;
```

```
// Active
Panel["Page 1"]["Rocker_switch1"].Enabled = true;
// Visibility
w.Visible = true
```

```
// Texte setzen
w.Text.SetText("en", "MyText");
```

```
// Connect variable to widget
Panel["Page 1"]["Standard_meter1"].Variable.Name = "MyVariable";
```

The first parameter of `SetText()` is a country code for setting texts in different languages.

Standard Widget

To set the text of a Standard Widget the property `StandardText` has to be set. `SetText()` is not available for Standard Widget.

```
Panel["Page 1"]["CheckBox1"].StandardText = "MyText";
Panel["Page 1"]["CheckBox1"].StandardVariable = "MyVariable";
Panel["Page 1"]["Label"].StandardText = "Hello World!";
```

Create and edit Zones

```
using System.Drawing;
using System.Linq;
```

```
var widget = Panel["Page 1"]["List1"];
widget.ZonesController.ZonesClear();
widget.ZonesController.ZoneAdd();
var zone = widget.Zones.Last(); // Alternative: widget.ZoneController.ZoneGet(0);
zone.Text.SetText("en", "Entry_1");
zone.SetValue = 42;
zone.UpperLevel = 100;
zone.LowerLevel = 0;
zone.Color = Color.Blue;
```

Curve window: add Signal

```
// add signal to a new y-axis
Panel["Page 1"]["Curve1"].AddSignal("Channel_001",eAppendConstants.AppendNewAxis);
// add Signal to a new coordinate system
Panel["Page 1"]["Curve1"].AddSignal("Channel_001",eAppendConstants.AppendNewCosy);
// add signal to existing y-axis
Panel["Page 1"]["Curve1"].AddSignal("Channel_001",eAppendConstants.AppendNewLine);
```

Properties of signals can be set via `Panel["Page 1"]["Curve1"].Signals`.

Zones changed (e.g. list)

The event handling methods are run **asynchronously**.

```
public class MyPanel : AbstractMyPanel {

   /// List1 : GaugeWidget
   public override void Run() {
         List1.ZoneChanged += new Action<object, EventArgs>(List1_ZoneChanged);
   } // run

   void List1_ZoneChanged(object arg1, EventArgs arg2) {
         MessageBox.Show("Hello World!");
   } // List1_ZoneChanged

   public override void Stop() {
         List1.ZoneChanged -= List1_ZoneChanged;
   } // stop
} // class
```

Events

If errors (exceptions) occur in the registered functions, they are catched during processing and not thrown further.

More information about this can be found in the *ProxyClasses.cs* file.

17.1.2.3.3 Tables

Link variables to a table cell

The following code links the 1st cell in the 1st column to the variable *Var_01.*

```
using System.Linq;
```

```
var table = Panel["Page 1"]["Table1"];
var cell = table.Columns.ElementAt(0).Cells.ElementAt(0); // 1st col, 1st element
cell.Variable.Name = "Var_01";
```

Access to table cells

The folling example sets the text of the first column in the first row to "Hello World".

```
using System.Linq;
```

```
var cell = Panel["Page 1"]["Table1"].Columns.ElementAt(0).Cells.ElementAt(0)
cell.Text.SetText("en", "Hello World");
```

Table events

Syntax	Description
OnCellClick	Invoked, when a cell is clicked.
OnCellDisplayValueChangedByUser	Invoked, when the user changes the value of a cell.
OnCellDoubleClick	Invoked, when a cell is double-clicked.
OnColumnHeaderClick	Invoked, when the head of a column is clicked.
OnColoumnHeaderDoubleClick	Invoked, when the head of a column is double-clicked.
OnRowIndicatorClick	Invoked, when the indicator is clicked.
OnRowIndicatorDoubleClick	Invoked, when the indicator is double clicked.
OnSelectionChanged	Invoked, when the selection of the cells has changed.

Table functions

Syntax	Description
Region: Tabelle	
CellClick()	Performs a mouse click to the given cell.
CellDisplayValue()	Gets the display value of the cell.
CellDoubleClick()	Performs a double-click to the given cell.
GetCell()	Gets the cell as an CellWidget object.
GetColumn()	Gets the column as an ColumnWidget object.
SelectionClear()	Clears the current selection.

Table properties

Syntax	Description
Region: Table (TableWidget)	
SelectedCells	Gets the selected cells.
RowCount	Gets the amount of rows.
ColumnCount	Gets the amound of columns.
Region: Columns (ColumnWidget)	
Selected	Returns true, if the column is selected.
Region: Cells (CellWidget)	
Selected	Returns true, if the cell is selected.

17.1.2.3.4 Curve window

The following line is needed for the following examples.

```
Widget w = Panel["Page 1"]["Curve1"];
```

The following **libraries** are needed for constants and colors:

```
using imc.Common.Interfaces.Panel.Scripting;
using System.Drawing;
```

Add signals

In the constant *eAppendConstants*, the options for adding as a new y-axis or new coordinate system are included.

```
w.AddSignal("Channel_001",eAppendConstants.AppendNewAxis);
w.AddSignal("Channel_002", eAppendConstants.AppendNewLine);
w.AddSignal("Channel_003", eAppendConstants.AppendNewCosy);
```

Display mode

```
w.DisplayMode = eDisplayMode.Stacked;
```

```
// Display mode
public enum eDisplayMode {
   Default = 1,Stacked, Waterfall, ColorMap, LastValue, Barmeter, Table, ThreeD
}
```

Loading and saving curve window configurations

```
w.Save(@"c:\Test\MyCurve"); // ccv is appended automatically
```

```
w.Load(@"c:\Test\MyCurve"); // ccv is appended automatically
```

Axis settings

```
CurveAxis xAxis = w.xAxis;
CurveAxis yAxis = w.yAxis;
```

```
// Scale
xAxis.ScaleType = eAxisScaleConstants.ccwAbsTime; // ccwUnit
// Display range
xAxis.MinMaxType = eAxisMinMaxConstants.ccwFix;
// Minimum / Maximum
xAxis.SetMinMax(-10;10);
double max = xAxis.CurrentMax;
double min = xAxis.CurrentMin;
```

Signal settings

```
CurveSignalLine line = w.Signals[0];
```

```
// Color
line.Color = Color.Blue;
// Line shape
line.Shape = eLineShape.Line; // Bar, Bar3D, Dots, LineCubic, None, Steps, VertLine
//
line.ShowInLegend = eAutoBool.Auto; // Yes , No
//
string name = line.SignalName;
//
line.width = 3;
// Delete a signal from the curve window
line.Delete();
```

Delete all signals

```
foreach (var signal in w.Signals)
{
   signal.Delete();
}
```

17.1.2.4 Variables functions

Changes

Warning — Fundamental change in access to variables

The access to variables and measurements has been changed **significantly**.
Manual adjustments to *Scripts* from version 4.x and 5.x are therefore necessary.
The *Datapool* class no longer exists and is replaced by *Variables*.

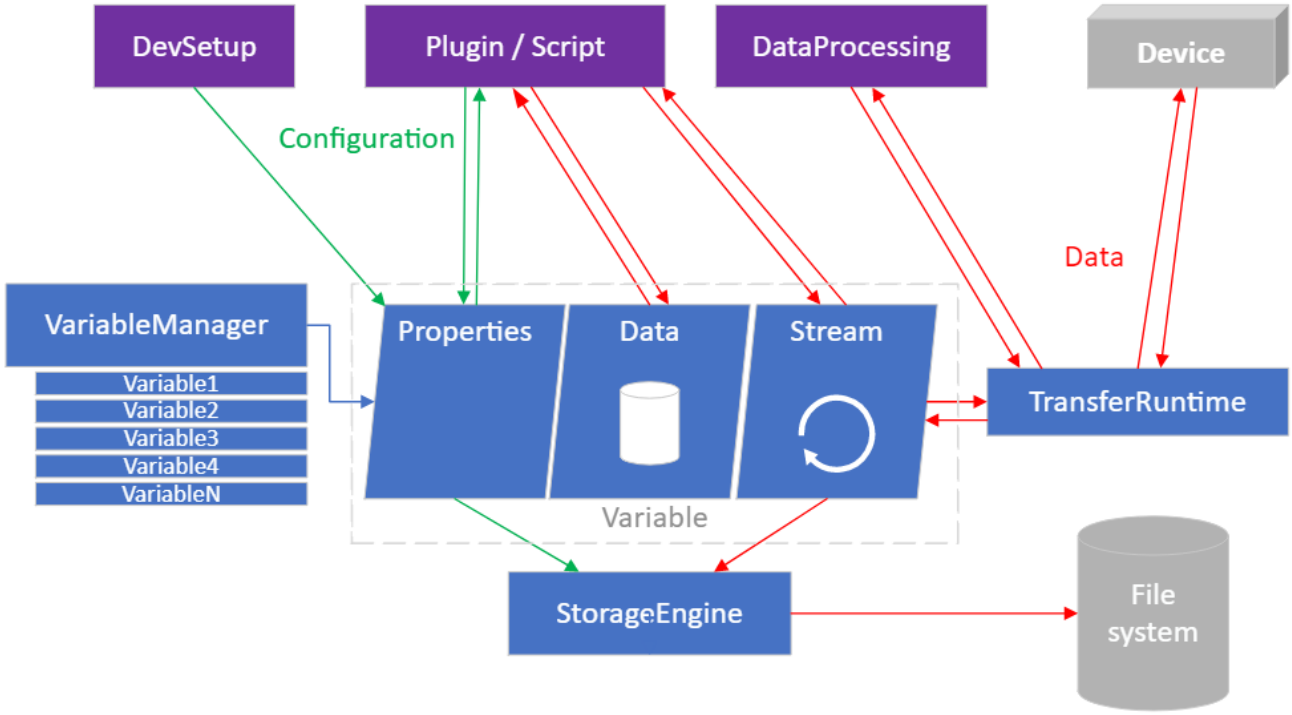
Note — Measurements

The *measurements* in the **Data Browser** are treated as *variables*. The *channels* belonging to a *measurement* are the corresponding **children** of this variable.

Note — Variables and Context scripts

When using **Context scripts**, note that not all variables are available when the `Run()` method is called.

If variables are to be used immediately, the *VariableCreated* event of the variable should be used in the **Context script**, see Events 1862.



Overview to variables access

Overview

The *DevSetup* creates the device *variables* including their properties in the *VariableManager*.

These *variables*, their properties and the *data* can be accessed via the *Scripting* interface (or a plug-in). Likewise *streaming data* can be processed.

The data communication with the *device* itself is done by a *TransferRuntime* component. Data can also be exchanged via *DataProcessing* (including Inline FAMOS Task).

FAMOS Task) kann ebenfalls ein Datenaustausch erfolgen.

The *StorageEngine* saves the variables and their *streaming data* on the *file system*.

Concept

Exceptions must be expected when accessing variables. It is therefore recommended to use the `Try` functions. These return their own `Result` type, which contains the property "IsSuccessful". In case of success the property Value contains the requested value. In case of an error the property Exception contains the corresponding exception.

```
var displayName = new DisplayName("NotExistendVariableName");
var variable = Variables[displayName]; // Exception is thrown
```

```
var result = Variables.TryGet(new DisplayName("ExistendVariableName"));
if (result.IsSuccessful)
{
   var variable = result.Value;
   // ...
}
else
{
   // result.ThrowExceptionIfAny();
   Logbook.LogEntry("Script", result.Error);
   // ...
}
```

The different variable types can be found in the corresponding chapter:

- Variables 1850
- Measurements 1854
- Channels 1854
- Streams 1857
- Complex Variables 1860
- Events 1862

17.1.2.4.1 Variables

Access to variables

```
var variableName = "MeineVariable";
IVariable variable;
```

```
variable = Variables[new DisplayName(variableName)];
variable = Variables[variableName];
```

```
variable = Variables[new DisplayNamePath(new[] {"Trigger_01", "State"})];
variable = Variables["Trigger_1\\State"];
```

```
variable = Variables["meineVariable@Measurement#1"];
variable = Variables["meineVariable@MeineMessung"];
```

```
Result<IVariable> result = Variables.TryGet(variableName);
if (result.IsSuccessful) {
   variable = result.Value;
} else {
   // Log(result.Error);
}

result.ThrowExceptionIfAny();
```

Create variables

```
var variable = Variables.Create(new DisplayName("VariableName"));
```

```
// Text
var variable = Variables.Create(new DisplayName("TextVariable"));
variable.Properties.Set(EClassID.eDataType, ParameterValues.eDataType.ASCII);
variable.Properties.Set(EClassID.eXFormat, ParameterValues.eXFormat.SingleValue);
variable.Properties.Set(EClassID.eObjectType, typeof(System.String));
```

```
// Numeric
var variable = Variables.Create(new DisplayName("NumericVariable"));
variable.Properties.Set(EClassID.eDataType, ParameterValues.eDataType.IEEEDOUBLE);
variable.Properties.Set(EClassID.eXFormat, ParameterValues.eXFormat.SingleValue);
variable.Properties.Set(EClassID.eObjectType, typeof(System.Double));
```

For creating several variables (or properties, see Variable properties 1853) it is recommended to use a **constructor task**.

```
var constructorTask = Variables.Factory.NewConstructorTask();
var variableWriterOne = constructorTask.AddNewVariableConstructor(new
DisplayName("NumericVariableOne"));
var variableWriterTwo = constructorTask.AddNewVariableConstructor(new
DisplayName("NumericVariableTwo"));
var variableWriterThree = constructorTask.AddNewVariableConstructor(new
DisplayName("NumericVariableThree"));
// ... do create properties
constructorTask.Process().ThrowExceptionIfAny();
```

Create variables in variables in variables ...

```
Variables.Create(new DisplayName("A")).Children.Create(new DisplayName("B")).Children.Create(new
DisplayName("C"));
var name = Variables[new DisplayNamePath(new string[]{"A", "B", "C"})].Name
```

Create variables with scope

- By default, the variables have the scope "volatile" (temporary).
- It is advisable to take care of creating and deleting variables yourself. Only then is it ensured that the appropriate properties are available.
- The following option should therefore be used with caution.

```
using imc.Data.Core.Contract.Identifiers;
using imc.Data.Core.Contract.Tasks;
using imc.Studio.PlugIns.CoreExtensions.UserVariables.Model;
using imc.Studio.Tools.DevSetupGraphGenerator;
```

```
var variableName = "TestVariable";
var accessGroup = "UserVariable";

var constructorTask = Variables.Factory.NewConstructorTask();
var variableWriter = constructorTask.AddNewVariableConstructor(new DisplayName(variableName));

variableWriter.Access.SetAccessGroup(accessGroup);

variableWriter.Properties.Set(EClassID.eXFormat, DevSetupCommonValues.eXFormat.SingleValue);
variableWriter.Properties.Set(EClassID.eDataType, DevSetupCommonValues.eDataType.IEEEDOUBLE);

variableWriter.Properties.Set("VariableScope", VariableScope.Experiment.ToString())
    .Attributes.Set(EClassID.eIsPersistent.ToString(), true);

variableWriter.Properties.Set("InitValue", 42);
variableWriter.Properties.Set(EClassIDExtention.eDefaultValue.ToString(), 42);

constructorTask.Process().ThrowExceptionIfAny();
```

VariableScope	
Selection	**Description**
Volatile	Variable exists only for the running instance of imc STUDIO.
Experiment	Variable exists only in Experiment context.
Sequencer	Variable exists only in Sequencer context.
Project	Variable exists only in Project context.
Presistent	Variable exists only in Application context.

Set the category

```
variable.Properties.Set("eCategory", "CustomCategory");
variable.Properties.Set("eCategory", ""); // User-defined variables
```

Delete variables

```
Variables.Delete(new DisplayName(variableName));
```

Set the value of a variable

```
variable.Value.Set(23d); // numeric
```

The variable must have the properties *eXFormat* and *eDataType.*

Get the variable's content

```
var content = variable.Value.Get<double>(); // whereas double is the desired type
```

```
var result = variable.Value.TryGet<double>();
if (result.IsSuccessful)
{
   var content = result.Value;
}
```

If you want to process many variables, use a ReaderTask:

```
List<DisplayName> displayNames = new() {
      new DisplayName("Variable_1"),
      new DisplayName("Variable_2"),
      new DisplayName("Variable_3")
};

List<IDataTaskResult<int>> valueResults = new();

IVariableManagerReaderTask readerTask = variableManager.Factory.NewReaderTask();

foreach (DisplayName displayName in displayNames) {
   IVariableReader reader = readerTask.AddNewVariableReader(displayName);
   valueResults.Add(reader.Data.Get<int>());
}

ProcessValues(readerTask, valueResults);
```

```
private void ProcessValues<T>(IVariableManagerReaderTask readerTask, List<IDataTaskResult<T>>
valueResults) {
   var result = readerTask.Process(); // imc.Data.Contract.Result
   if (!result.IsSuccessful) {
         // Error message
         return;
   }

   foreach (IDataTaskResult<T> valueResult in valueResults) {
         if (!valueResult.IsProcessed || !valueResult.IsSuccessful) {
               // Error message
               continue;
         }

         var variableValue = valueResult.Get();
         // do something
   }
}
```

Variable properties

When creating variable properties it is recommended to use a constructor task.

The following (user-defined) properties exist:

```
var constructorTask = Variables.Factory.NewConstructorTask();
var variableWriter = constructorTask.AddNewVariableConstructor(new DisplayName("NumericVariable"));

variableWriter.Properties.Set(EClassID.eCategory, category);
variableWriter.Properties.Set(EClassID.eChannelComment, "");
variableWriter.Properties.Set(EClassID.eDataType, ParameterValues.eDataType.IEEEDOUBLE);
variableWriter.Properties.Set(EClassID.eXFormat, ParameterValues.eXFormat.SingleValue);
variableWriter.Properties.Set(EClassID.Scope, VariableScope.Experiment);
variableWriter.Properties.Set(EClassID.eObjectType, typeof(System.Double));

// ... further
variableWriter.Properties.Set(EClassID.eCategory, "");
variableWriter.Properties.Set(EClassID.eChannelComment, "");
variableWriter.Properties.Set(EClassID.UserDefinedDataType, UserDefinedDataType.Numeric);
variableWriter.Properties.Set(EClassID.eUserUnit, "");
variableWriter.Properties.Set(EClassID.InitValue, "");
variableWriter.Properties.Set(EClassIDExtention.eDefaultValue, "");

variableWriter.Properties.Set(EClassID.FifoActive, false);
variableWriter.Properties.Set(EClassID.FifoEvents, 0);
variableWriter.Properties.Set(EClassID.FifoRingBufferTime, 0);
variableWriter.Properties.Set(EClassID.StoreActive, false);
variableWriter.Properties.Set(EClassID.StoreEvents, 0);
variableWriter.Properties.Set(EClassID.StoreRingBufferTime, 0);
variableWriter.Properties.Set(EClassID.StoreTriggerName, "");

constructorTask.Process().ThrowExceptionIfAny()
```

EVariableProperty (Extract)	
Property	**Description**
eCategory	Variable's category
eDisplayColor	Display color in the curve window
eYUnit	y-Unit
eYOffset	y-Offset
eXUnit	x-Unit
eXOffset	x-Offset
eSampleTime	SampleTime
eComment	Comment

Save user-defined properties

If user-defined properties are to be stored on variables or channels, the "IsPersistent" attribute must be set.

```
var constructorTask = Variables.Factory.NewConstructorTask();
var variableWriter = constructorTask.AddNewVariableConstructor(new DisplayName("NumericVariable"));
variableWriter.Properties.Set(EClassID.eXFormat, DevSetupCommonValues.eXFormat.Normal);
variableWriter.Properties.Set(EClassID.eDataType, DevSetupCommonValues.eDataType.IEEEDOUBLE);

var property = variableWriter.Properties.Set("UserProp_Persistent", "A property which is stored");
property.Attributes.Set(EClassID.eIsPersistent.ToString(), true);

variableWriter.Properties.Set("UserProp_Temporary", "A property which exists only at runtime");

var taskResult = constructorTask.Process();
```

17.1.2.4.2 Measurements

MeasurementManagerService

```
var measurementManagerService = Variables.Services.Get<IMeasurementManagerService>();
```

Last measurement's name

```
var result = measurementManagerService.FindLastMeasurement();

if (result.IsSuccessful)
{
   var measurementName = result.Value.DisplayName.ToString();
   // ...
}
```

Load measurement

```
Result<IMeasurementService> measurementResult = measurementManagerService.TryGet("FirstMeasurement");
if (measurementResult.IsSuccessful)
{
   var measurement = measurementResult.Value;
   measurement.Load().ThrowExceptionIfAny();
}
```

Unload measurement

```
Result<IMeasurementService> measurementResult = measurementManagerService.TryGet("FirstMeasurement");
if (measurementResult.IsSuccessful)
{
   var measurement = measurementResult.Value;
   measurement.Unload().ThrowExceptionIfAny();
}
```

Delete measurement

```
measurementManagerService.Delete(new DisplayName("SecondMeasurement")).ThrowExceptionIfAny()
```

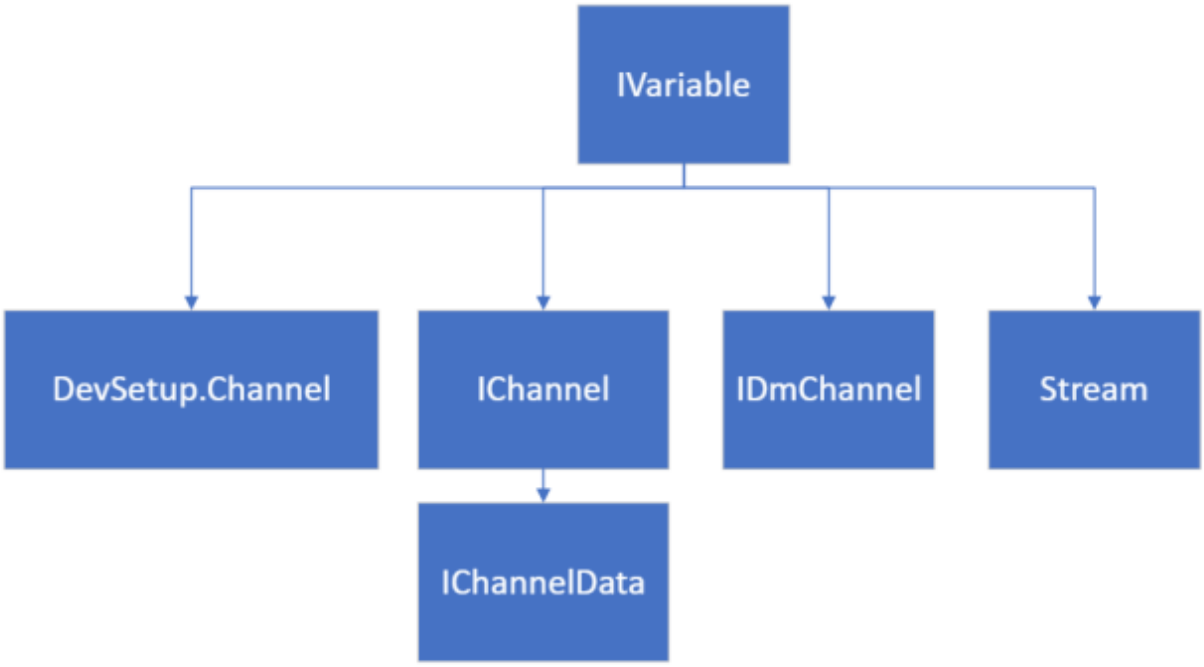
Rename measurement

```
Result<IMeasurementService> measurementResult = measurementManagerService.TryGet("FirstMeasurement");
if (measurementResult.IsSuccessful)
{
   var measurement = measurementResult.Value;
   measurement.Rename(new DisplayName("RenamedFirstMeasurement")).ThrowExceptionIfAny();
}
```

17.1.2.4.3 Channels

For each channel in DevSetup.Channels (see: Channel functions) 1832 a variable with the same name is automatically created upon processing the configuration.

```
var channelVariable = Variables["Channel_001"]; // or child from measurement
```



Channel overview

IVariable:	Variables and properties
DevSetup.Channel:	Device-Channel configuration
IChannel:	(saved) channel
IChannelData:	Data of a channel
IDmChannel:	Channel and data
Stream:	Streaming data of a channel

Access to the data of a channel

The measurements must be loaded in the data browser. See Measurements 1854

Access via IChannel:

```
using imc.Data.Core.Contract.IChannel
```

```
var result = variable.QueryInterface<IChannel>();
if (result.IsSuccessful)
{
   var channel = result.Value;
   long n = channel.Data.TotalLength;
   var offset = TimeSpanCore.FromTicks(0);
   var data = channel.Data.Read<double>(offset, n);
   // ..
}
else
{
   Logbook.LogEntry("Script", resultChannel.Error);
}
```

Access via IDmChannel:

```
using imc.Common.Components.DataManager.Interfaces;
```

```
public double[] GetData(IVariable variable)
{
    var result = variable.QueryInterface<IDmChannel>();

    if (result.IsSuccessful)
    {
        var channel = result.Value;

        return channel.Data;
    }
    else
    {
        // Error handling
        Logbook.LogEntry("Script", result.Error);
    }
}
```

Usage of IDmChannel/IChannel

The IDmChannel interface can be used arbitrarily for read operations. For write operations it is to be avoided, since it bypasses the management in the VariableManager and therefore inconsistencies can arise.

```
using imc.Common.Components.DataManager.Interfaces;
```

```
var result = Variables[variableName].QueryInterface<IDmChannel>();
```

The interface imc.Data.Storage.HDD.Contract.**IDmDataObjectVariableService** is provided as support for exchanging data and properties between **DmChannel** and **IVariable**.

```
var dataObjectService = variable.Services.Get<IDmDataObjectVariableService>();
```

If all data and properties are to be taken from a channel "newDataDmChannel", the following call helps:

```
var configResult = dataObjectService.Configure(newDataDmChannel, DataInclusion.WithAllData);
```

Set data (all existing in the variable will be deleted)

```
dataObjectService.SetData(newDataDmChannel)
```

Data is attached via the IChannelData interface:

```
var channelInterface = variable.QueryInterface<IChannel>().Value;
IChannelData data = channelInterface.Data.QueryInterface<IChannelData>().Value;
data.Append(newDataDmChannel.Data);
```

Usage of IDmGroup/DmGroup

All **DmChannels** passed into a **DmGroup** will be destroyed by it at the end. Therefore the following is forbidden:

```
var dmChannelofVariable = Variables[variableName].QueryInterface<IDmChannel>();
DmGroup.AddObject( dmChannelofVariable )
// ..
DmGroup.Dispose();
```

The only possible current way is to create a clone of the channel beforehand and pass it:

```
group.AddObject(DmChannel.CloneOf(dmChannelofVariable));
```

DmChannel objects, which are the result of Famos operations or were created by yourself, must only be cloned, if you still want to access them after the action with DmGroup.

Set a channel

Create variable

```
using imc.Data.Storage.HDD.Contract;
```

```
var service = Variables.Services.Get<IDmDataObjectVariableManagerService>();
var resultVariable = service.Create(resultChannel); // resultChannel of type DmChannel
```

Set variable

The variable must have suitable properties (eDataType, eXFormat, eSampleTime, eObjectTime, ...) before setting.

```
using imc.Data.Storage.HDD.Contract;
```

```
var resultVariable = Variables[new Displayname(name)];
var variableService = resultVariable.Services.Get<IDmDataObjectVariableService>();
variableService.SetData(resultChannel);
```

Create and set variable

```
var service = Variables.Services.Get<IDmDataObjectVariableService>();
var resultVariable = service.Configure(resultChannel, DataInclusion.WithAllData);
```

17.1.2.4.4 Streams

The following basics have to be considered:

There are *cycles, events* and the actual *data packages*. A cycle usually corresponds to a measurement, which consists of events, which in turn consist of data packages.

When reading data, it is recommended to use a state machine.

A `StreamReader` is registered to the variable. The `StreamReader` provides a `StreamDataEventReader` via the current event `CurrentEvent`, with which the actual data is read. First it must be checked whether new data are available for reading. If new data is available for **reading** (State 2), it can be read. Otherwise you are in a **wait position** (State 1).

Basic procedure:

1. Attach StreamReader via `AttachNewReader()` to `variable.Stream`
2. Register EventHandler to `StreamStateChanged`
3. Check in EventHandler if new data is present.
   a. if there is no event, go to the next event
      i. if there are no more events to read and the cycle is finished -> finish reading the cycle.

   b. if there is an event, but no data to read and the event is not finished -> stay in the waiting state
   c. if event exists, but there is no data to read and event is finished -> finish reading the event
   d. if yes, read data
4. If going to next event and this is not successful
   a. check for StreamBufferOverflowException and reset the StreamReader.
5. Detach EventHandler
6. Detach StreamReader

Note Variables and Context scripts

When using **Context scripts**, note that not all variables are available when the `Run()` method is called.

If variables are to be used immediately, the *VariableCreated* event of the variable should be used in the **Context script**, see Events 1862.

Read data streams

The following **using directives** are required,

```
using System;
using System.Collections.Generic;
using System.Threading.Tasks;

using imc.Data.Contract;
using imc.Data.Contract.Events;
using imc.Data.Core.Contract;
using imc.Data.Core.Contract.Streams;
using imc.Interfaces.Data;
```

To read data streams, a new `StreamReader` is added to the variable and an EventHandler is registered to the `StreamChanged` event. The data type (here: double) is needed.

```
IStreamReader<double> m_streamReader = variable.Stream.AttachNewReader<double>(); // class member
```

```
m_streamReader.StreamStateChanged += OnStreamChanged; // hook
```

A state machine is then used in the **event handler**. In the one state it waits for a new event state, in the other state it processes the data (here of type double).

```
// (1) waiting for event state
Result<bool> newEventResult = streamReader.MoveNextEvent();
if (newEventResult.IsSuccessful) {
   if (newEventResult.Value == true) {
      // => move to event state (2)
   }
} else if (newEventResult.Error is StreamBufferOverflowException) {
   // overflow handling
   Result<StreamDataCount> lostData = streamReader.Reset();
   // stay in waiting for event state (1)
}

if (streamReader.IsCycleComplete && streamReader.UnreadEventCount == 0) {
   streamReader.EndReadCycle();
   // stay in waiting for event state (1)
}
```

```
// (2) inside event state
IStreamDataEventReader<double> currentEventReader = streamReader.CurrentEvent;
// event header
DateTimeOffsetHighPrecision eventTime = currentEventReader.Time;
DateTimeOffsetHighPrecision referenceTime = currentEventReader.ReferenceTime;
TimeSpanHighPrecision offsetOfFirstSample = currentEventReader.TimeOffsetOfFirstSample;
int eventNumber = currentEventReader.EventNumber;

// current event reader state
var alreadyConsumedDuration = currentEventReader.Duration;
var unreadDuration = currentEventReader.DurationOfUnreadSamples;
long unreadSampleCount = currentEventReader.UnreadSampleCount;
bool isEventComplete = currentEventReader.IsComplete;

Result<double[]> readSamples = currentEventReader.ReadSamples(unreadSampleCount);

if (!readSamples.IsSuccessful && readSamples.Error is StreamBufferOverflowException) {
   // overflow handling
   Result<StreamDataCount> lostData = streamReader.Reset();
   // => move to waiting for event state (1)
}

if (isEventComplete && unreadSampleCount == 0) {
   currentEventReader.EndRead();
   // => move to waiting for event state (1)
}
```

At the end, the `StreamReader` must be unregistered again:

```
m_streamReader?.StreamStateChanged -= OnStreamChanged;
```

```
m_streamReader.Detach();
m_streamReader = null;
```

If a `Reader` is attached later, a `Read()` call should be made first and, if necessary, the overflow should be handled by a `Reset()`.

- If a `Reader` is attached, it must also read. Otherwise an overflow occurs (StreamBufferOverflowException). After the overflow, the `Reset()` method must be called.
- If changes are made to the variable, it is possible that after Process action the StreamReader/StreamWriter is no longer available and an AggregateException occurs.

Write data streams

To write data streams, a **StreamWriter** is added to the variable. It should be noted that not every variable (e.g. device variables) can be written to.

```
IStreamWriter<float> streamWriter = variable.Stream.OpenWriter<float>();
```

```
for (int eventNumber = 1; eventNumber <= 42; eventNumber++) {
   streamWriter.BeginNextDataPacket(
      time: new DateTimeOffsetHighPrecision(DateTimeOffset.Now),
      offset: TimeSpanHighPrecision.Zero,
      eventNumber: 1
   );

   // write data to the stream again and again
   streamWriter.CurrentDataPacket.WriteSamples(
      new[] { 13.5f, 12.1f, 45.8f }
   );

   // optional
   Task readerEndAcknowledgementReaction = streamWriter.CurrentDataPacket.EndWrite();
}
```

```
Task readerCloseAcknowledgementReaction = streamWriter.Close();
```

17.1.2.4.5 Complex variables

Complex variables are built in a structure and are accessible via the child variables (*children*).

Computer

```
var computerVariable = Variables["Computer"]
var driveVariable = computerVariable.Children["Drives"].Children["C"];
var totalSizeVariable = driveVariable.Children["TotalSize"];
var totalSize = totalSizeVariable.Value.TryGet<double>();
```

Computer\\Drives\\C		
Variable name	**Internal identifier**	**Data type**
Available free space	AvailableFreeSpace	System.Int64
Total free space	TotalFreeSpace	System.Int64
Total size	TotalSize	System.Int64

Computer\\Process		
Variable name	**Internal identifier**	**Data type**
Process: GDI handle consumption	GDIHandleConsumption	System.Double
Process: handle count	GDIHandleCount	System.Int32
Process: physical memory consumption	GDIPhysicalMemoryConsumption	System.Int64
Process: private memory consumption	GDIPrivateMemoryConsumption	System.Int64
Process: virtual memory consumption	GDIVirtualMemoryConsumption	System.Int64

SystemClock

```
var systemClockVariable = Variables["SystemClock"];
var systemTimeVariable = systemClockVariable.Children["SYSTEMTIME"];
var systemTime = systemTimeVariable.Value.Get<DateTimeOffset>();
```

SystemClock		
Variable name	**Internal identifier**	**Data type**
PC time	PCTIME	System.DateTimeOffset
System time	SYSTEMTIME	System.DateTimeOffset

Trigger

```
var triggerName = "Trigger_01"; // Trigger name, e.g. BaseTrigger
var triggerVariable = Variables[triggerName];
var triggerTime = triggerVariable.Children["TriggerTime"];
```

Trigger_NUMMER Trigger_01, BaseTrigger		
Variable name	**Internal identifier**	**Data type**
Zustand	State	System.Double
Eventnummer	EventNumber	System.Double
Trigger Zeit	TriggerTime	System.DateTimeOffset

Device System Variable

```
var variableName = "System_MyDevice_123456"; // SYSTEM_DEVICENAME
var systemDevice = Variables[variableName];
var freeMemory = systemDevice.Children["FreeSpace"];
```

SYSTEM_DEVICENAME SYSTEM_MyDevice_123456		
Variable name	**Internal identifier**	**Data type**
Connection 1: Critical channel fill level	WorstFifoLevel	System.Double
Connection 1: Critical channel name	WorstFifoLevelName	System.String
Connection 1: Data rate	DataRate	System.Double
Device: Client count	ClientCount	System.Double
Device: Disk size	DiskSize	System.Double
Device: Free memory	FreeSpace	System.String

SYSTEM_DEVICENAME\Monitor\1 SYSTEM_MyDevice_123456\Monitor\1		
Variable name	**Internal identifier**	**Data type**
Connection 2 Monitor 1: Critical channel fill level	WorstFifoLevel	System.Double
Connection 2 Monitor 1: Critical channel name	WorstFifoLevelName	System.String
Connection 2 Monitor 1: Data rate	DataRate	System.Double

Analog for more Monitors.

17.1.2.4.6 Events

To react to variable events, the EventManager is used.

Since the events must be subscribed and unsubscribed. In imc STUDIO a **Context script** is suitable for this.

```
IEventManager eventManager = Core.Base.GetAPI<API_Core_V2>().Collections["Events"] as IEventManager;
```

Libraries

```
using System.Collections.Generic;
using imc.Common.Interfaces.Logbook;
using imc.Data.Contract.Events;
using imc.Data.Core.Contract.Events;
using imc.Data.Core.Contract.Identifiers;
using imc.Studio.Interfaces.Core;
```

Subscribe event handler

```
eventManager.Subscribe(OnValueChanged, ValueEvent.VALUE_CHANGED);
eventManager.Subscribe(OnVariableCreated, VariableManagerEvent.VARIABLE_CREATED);
eventManager.Subscribe(OnVariableDeleted, VariableManagerEvent.VARIABLE_DELETED);
```

Unsubscribe event handler

```
eventManager.Unsubscribe(OnValueChanged, ValueEvent.VALUE_CHANGED);
eventManager.Unsubscribe(OnVariableCreated, VariableManagerEvent.VARIABLE_CREATED);
eventManager.Unsubscribe(OnVariableDeleted, VariableManagerEvent.VARIABLE_DELETED);
```

Interpret changed events

```
private void OnValueChanged(IEnumerable<IEventData> args)
{
    foreach (var eventArgs in args) {
         string message = Interpret(eventArgs.QueryInterface<IValueChangedEventArgs>().Value);
         Log(message);
    }
}
```

```
private void Log(string message){
    Core.Logbook.LogEntry("ListenToValueChanged", message, 0, ELogbookEntryCategory.Information);
}
```

```
private string Interpret(IValueChangedEventArgs args) {
    string message = string.Empty;
    DisplayName displayName = args.Identifier.DisplayName;

    if (displayName.ToString() == "MyVariable") // filter
    {
        double newValue = args.Value.Get<double>();
        double oldValue = args.OldValue.Get<double>();

        message = BuildMessage(displayName, oldValue, newValue);
    }

    return message;
}
```

```
private string BuildMessage(DisplayName displayName, double oldValue, double newValue){
    return "Variable '" + displayName + "' changed from '" + oldValue + "' to '" + newValue + "'.";
}
```

Interpret created/deleted events

```
private void OnVariableCreated(IEnumerable<IEventData> args)
{
    foreach (var eventArgs in args) {
       var dataObjectEvent = new DataObjectEvent(eventArgs);
       if (dataObjectEvent.DisplayName.ToString() == "Channel_001")
       {
          // do something
       }
    }
}
```


```
private void OnVariableCreated(IEnumerable<IEventData> args)
{
    foreach (var eventArgs in args) {
       var dataObjectEvent = new DataObjectEvent(eventArgs);
       if (dataObjectEvent.DisplayName.ToString() == "Channel_001")
       {
          // do something
       }
    }
}
```


```
private class DataObjectEvent : AbstractDataObjectEvent
{
   public DataObjectEvent(IEventData eventData) : base(eventData)
   {
   }
}
```

17.1.2.5 Logbook

Library

```
using imc.Common.Interfaces.Logbook;
// for ElogbookEntryCategory, ILogbookEntryBase, ...
```

Set Logbook entry

Note

For other Script types (Panel, Event, ...) the Logbook is called with `Core.Logbook`.

```
try{
   // ...
}catch (Exeption exception){
   Logbook.LogEntry("MyScript.cs", exception);
}
```

```
string errorSender = "MyScript.cs";
string errorText = "My Error";
int errorCode = 0;
ELogbookEntryCategory category = ELogbookEntryCategory.Error;
Logbook.LogEntry(errorSender,errorText,errorCode,category);
```

```
/*
ELogbookEntryCategory: Error, Fatal, Information, Warning
*/
```

```
/*
public void LogEntry(string sender, System.Exception x) ;
public void LogEntry(string sender, System.Exception x,ELogbookEntryCategory category);
public void LogEntry(string sender, string message, int code, ELogbookEntryCategory category) ;
public void LogEntry(ILogbookEntryBase entry) ;
public void LogEntry(ILogbookEntryBase entry, ELogbookEntryCategory category) ;
*/
```

Export Logbook entries since a given date

```
DateTime dateTime = new DateTime(2018, 1, 1);
Logbook.Export(@"c:\temp\logbook.xml", dateTime);
```

Show/Hide Logbook window

```
Core.Windows.ShowWindow("Logbook")
Core.Windows.HideWindow("Logbook")
```

Clear view of the logbook / delete logbook entries in the tool window

```
Logbook.ClearViews();
```

OnError/ OnWarning Event

Use a **Context script** to react to logbook changes.

```
using System.Collections.Specialized;
using System.Windows.Forms;
using imc.Common.Interfaces.Logbook;
```

```
// Run()
Core.Logbook.Base.CollectionChanged += new
NotifyCollectionChangedEventHandler(Core_Logbook_Base_CollectionChanged);
```

```
// Dispose()
Core.Logbook.Base.CollectionChanged -= Core_Logbook_Base_CollectionChanged;
```

```
void Core_Logbook_Base_CollectionChanged(object sender, NotifyCollectionChangedEventArgs e)
{
   foreach (ILogbookEntryBase entry in e.NewItems)
   {
       LogbookEntry log = LogbookEntry.GetInstance(entry);
       if (log.Category == ELogbookEntryCategory.Warning)
       {
           // do something
       }

       if (log.Category == ELogbookEntryCategory.Error)
       {
           // do something
       }
   }
}
```

17.1.2.5.1 Examples

Save Logbook entries since start of the script

In this example, all logbook entries which have occurred in the course of the script are written to a text file. For this purpose, at the beginning of the script the time is noted, which is used at the end of the script in order to query the corresponding logbook entries.

```
using System.IO;
using imc.Common.Interfaces.Logbook;

namespace SaveEntries {
   using System;

   public class SaveEntries : AbstractScript {

      public override void Run() {

         DateTime startDate = DateTime.Now; // save Time
         string filename =  "x:\myLogbook.txt";

         // ... body of the script, do something here
         Logbook.LogEntry("MyScript", "My Info", 0, ELogbookEntryCategory.Information);
         Logbook.LogEntry("MyScript", "My Warning", 0, ELogbookEntryCategory.Warning);
         Logbook.LogEntry("MyScript", "My Error", 0, ELogbookEntryCategory.Error);
         // ...

         // entries since the beginning of the script
         ILogbookEntryBase[] entries = Logbook.GetEntries(startDate);

         // save entries to text-file
         using (StreamWriter file = new StreamWriter(filename, true))
         {
            for (int k = 0; k < entries.Length; ++k){
               file.WriteLine(entries[k].ToString());
            }
         }

      } // of void run
   } // of class
} // of namespace
```

List of messages with sender "Sequencer"

In this example all messages of the logbook entries with sender "Sequencer" are listed.

In the Script options 1906 the *advanced code completion* must be enabled.

```
using System.Linq;
using System.Text;
using System.Windows.Forms;
using imc.Common.Interfaces.Logbook;


public void ShowMessages()
{
   string logbookMessages = new StringBuilder();

   foreach (LogbookEntry entry in Logbook){

      if (string.Compare(e.Sender, "Sequencer") == 0){
         logbookMessages.AppendLine(e.Message);
      }
   }

   MessageBox.Show(logbookMessages.ToString());
}

public void ShowMessagesAlternative1()
{
   // Alternative
   var query = from entry in Logbook
        where entry.Sender == "Sequencer"
        select entry.GetAPI<API_LogbookEntry_V1>();

   LogbookEntry[] entries =  query.ToArray();
}

public void ShowMessagesAlternative2()
{

   // Alternative 2
   var entries = Logbook
            .Where(logEntry => logEntry.Sender == "Sequencer")
            .ToArray();
}
```

17.1.2.6 Events

The following describes how to access events.

Methods must be registered and unregistered. In imc STUDIO a **Context script** is recommended for this.

Access to user-defined events is currently not possible.

Libraries

```
using imc.Studio.Interfaces.Core;
using imc.Studio.Tools.DevSetupGraphGenerator;
using System.Windows.Forms;
```

Event names

These must be defined within the class.

```
private const string deviceStartedEventName = "Device_Started";
private const string devicesStartedEventName = "Devices_Started";
// further names, see Sequencer
```

Register

Register in Run method

```
Events[deviceStartedEventName].Execute += OnDeviceStarted;
Events[deviceStartedEventName].Execute += OnDevicesStarted;
```

Unregister

Unregister in Dispose method

```
Events[deviceStartedEventName].Execute -= OnDeviceStarted;
Events[deviceStartedEventName].Execute -= OnDevicesStarted;
```

Event handler

```
private void OnDeviceStarted(object sender, ICoreEventEventArgsBase e)
{
   // for every single device
   var api = e.GetAPI<API_CoreEventEventArgs_V1>();
   var parameters = api.Parameters as string[];

   if (parameters != null)
   {
       var hardwareId = parameters[0];
       var device = DevSetup.Devices.GetItemByHWID(hardwareId);
       if (device == null) { return; }
       var serialNumber = device?.Parameters[EClassID.eDeviceSN].Text;
       MessageBox.Show("Device '" + serialNumber + "' started...");
   }
}
```

```
private static void OnDevicesStarted(object sender, ICoreEventEventArgsBase e)
{
   // global start
   var apiCancel = e.GetAPI<API_CancelEventArgs_V1>();
   MessageBox.Show("Devices started...");
}
```

17.1.2.7 Inline FAMOS functions

- For the functions to be available, the Inline FAMOS needs to be activated in the Product Configurator. This requires a corresponding imc STUDIO Inline FAMOS license.
- If Inline FAMOS is activated later and the functions are added to an existing script, the proxies must be regenerated first. See Regenerate proxies 1819.
- Please use the return value **IssueResult** of the methods to check, if the action was successful, because you do not get any information in the GUI of imc STUDIO, when configuring IFA-Tasks with Scripting.

Library

```
using imc.Studio.Interfaces.DataProcessing;
```

Get or create IFA-Task

```
InlineFamosTask task = InlineFamos.CreateTask(); // Display name will be generated
```

```
// If task available, get it, otherwise create
InlineFamosTask task2 = InlineFamos.GetOrCreateTask("InternalName");
InlineFamosTask task3 = InlineFamos.GetOrCreateTaskByDisplayName("DisplayName");
```

```
InlineFamosTask task4 = InlineFamos.GetTaskByName("InternalName");
InlineFamosTask task5 = InlineFamos.GetTaskByDisplayName("DiaplayName");
```

```
// Create Task with Owner
InlineFamosTask task5 = InlineFamos.GetOrCreateTask("MyPrivateTask", "Owner", ETaskAccess.Public,
"MyDisplayName");
```

A task created with an *owner* is not saved in the experiment. This allows a dynamically generation of IFA-code. For using the task, the script creating the task needs to be run first.

Create a private task

A **private** task is not shown in the GUI.

```
InlineFamosTask privateTask;
privateTask = InlineFamos.GetOrCreateTask("MyPrivateTask", "Owner", ETaskAccess.Private);
```

Get all public tasks

```
IInlineFamosTaskBase[] tasks = InlineFamos.Tasks;
```

Get all task of an owner.

```
IInlineFamosTaskBase[] tasks = InlineFamos.GetTasks("Owner");
```

Delete IFA-Task

```
IssueResult isSuccessful = InlineFamos.DeleteTask(task.Name); // delete
```

```
IssueResult isSuccessful = task.Delete();
```

Apply Task

```
IssueResult isSuccessful = task.Apply(); // check result for "IsSuccessful"
```

Task Name

```
string name = task.Name;
IssueResult isSuccessful = task.Setup.SetDisplayName("NewName");
```

Source code

The source text can be passed directly as a text variable, or a path for the associated IFA-file is specified.

```
// Get the source code
string sourceCode = task.Setup.SourceCode;
```

```
// Set source code: Path
IssueResult isSuccessful = null;
string filename = @"c:\temp\MyTask.ifa.xml";
using (StreamReader stream = new StreamReader(filename))
{
   string content = stream.ReadToEnd();
   isSuccessful = task.Setup.SetSourceCode(content);
}
```

```
// Set source code: text
task.Setup.SetSourceCode(@"Virt_Chn = Channel_001 * 2");
```

Serialization/deserialization of a task

```
task.Setup.GetXml();
task.Setup.ReadXml(stream);
task.Setup.SetXml(xml);
task.Setup.WriteXml(stream);
```

Example Create task and calculate a channel

A new public task with display name "My IFA Task" is created, which creates the double of an analog channel.

```
InlineFamosTask task = InlineFamos.GetOrCreateTask("MyTask");
task.Setup.SetDisplayName("My IFA Task");
IssueResult resultSource = task.Setup.SetSourceCode("Channel_out = 2 * Channel_001");
IssueResult resultApply = task.Apply();
```

17.1.2.8 Commands

Note

Every command can be used via `Commands.Invoke` in Scripting. While typing this command, press **CTRL+Space** and choose *Command Invoke* to open a dialog. In this dialog the according command can be chosen. The parametrization is done in the same dialog and is converted an XML-string.

Example Delete Panel page *Page 1*

```
Commands.Invoke("Delete page  Page: Page 1",
   "<AppPlotFactoryCmd version=\"1\">" +
      "<common version=\"1\" cuid=\"PanelPageDelete\" caption=\"Delete page\">" +
      "<command version=\"1\">" +
      "<properties version=\"1\">" +
      "<DeletePageCommandTemplate>" +
      "<Version>1.0</Version>" +
         "<Page1>Page 1</Page1>" +
      "<Kind>Delete</Kind>" +
      "<ShowExportDialog>False</ShowExportDialog>" +
      "<ShowPrinterDialog>False</ShowPrinterDialog>" +
      "</DeletePageCommandTemplate>" +
      "</properties>" +
      "</command>" +
      "</common>" +
   "</AppPlotFactoryCmd>");
```

Execute a menu action

```
Actions.FireAction(this, "COMMANDONAME"); // for commandos, see table below
```

React to a menu action

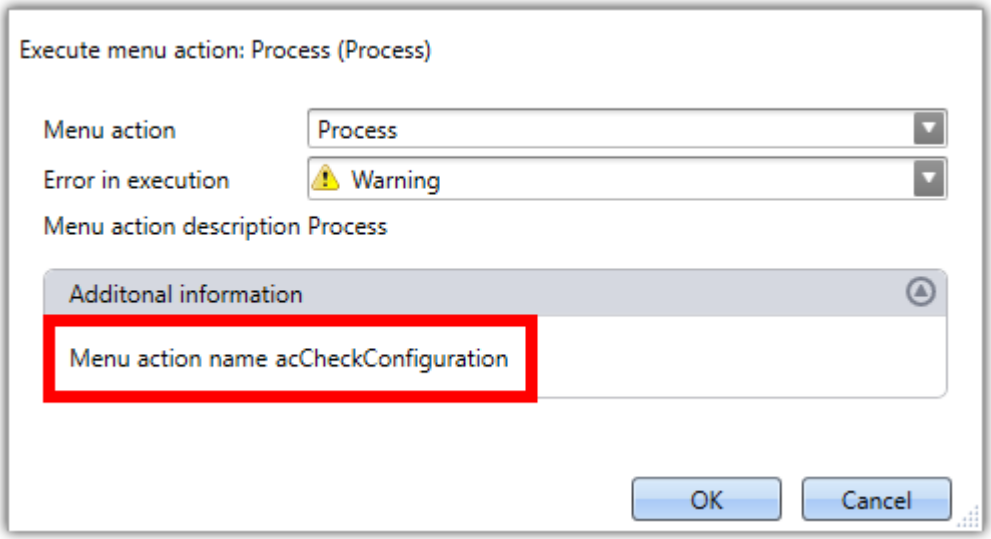
With a Context script you can react to the manually execution of a menu action.

```
using imc.Studio.Interfaces.Core;
```

```
public override void Run() {
   Actions["COMMANDAME"].Execute += new
CommonEventHandler<ICoreActionExecutionEventArgsBase>(Test_Execute);
}

void MyContextScript_Execute(object sender, ICoreActionExecutionEventArgsBase e){
   // do something
}

protected override void MyContextScriptDispose() {
   Actions["COMMANDNAME"].Execute -= MyContextScript_Execute;
}
```

Menu action name (Command name)

The commands are listed in the following table. The menu action names can be shown in the configuration dialog of the command "Execute menu action" under "Additional information".



Example of a menu action name

Sequencer

Action	Command name
start	StartSequencer
stop	StopSequencer

Device actions

Action	Command name
Device configuration (Dialog)	acImcDev30ShowDeviceConfigDlg
open imc Online FAMOS Editor	acImcDev30ShowOFADialog
open Diskstart-Dialog	acDevSetupShowLayout_Diskstart
Device search	acSearchNetwork
connect to selected devices	acDeviceConnect
start selected devices	acDeviceStart
stop selected devices	acDeviceStop
disconnect from selected devices	acDeviceDisconnect
Process	acCheckConfiguration
Prepare	acDeviceDownload
Suspend data storage	acSuspendDiscRecording
Resume data storage	acResumeDiscRecording
Reconfigure	acImcDev30ForceDownload
release all triggers	acReleaseAllTriggers

© 2023 imc Test & Measurement GmbH

Panel

Action	Command name
toggle Design mode	cmd_Panel_ToggleDesignMode
toggle Navigation bar	m_biNavBarShow
show Panel	ShowPanel
Panel page in Full-Screen mode	acPanelFullScreen
insert a Standard Panel page	cmd_Panel_InsertPageStandard
toggle lock on Panel page	cmd_Panel_Toggle_Lock
show last Panel page	m_biNavLastPage
show next Panel page	m_biNavNextPage
show previous Panel page	m_biNavPrevPage
show first Panel page	m_biNavHome

Navigation

Action	Command name
absolute time	NavigationActions.SetAbsTime
relative time	NavigationActions.SetRelTime
scroll	NavigationActions.SetXScroll
stretch	NavigationActions.SetXStretch
no scroll	NavigationActions.SetNoScroll

User administration

Action	Command name
log out user	UserLogOut
change user	ChangeUser

File actions

Action	Command name
new experiment (Dialog)	NEW
load experiment (Dialog)	LOAD
save experiment	SAVE
save experiment as	SAVEAS
Import/Export Dialog	IMPORTEXPORTDIALOG

Miscellaneous

Action	Command name
open Help	imcStudioHelpContent
save Current Measurement	imcStudioSaveMeasurement
open Options	imcStudioSharedOptionsDlg
exit imc STUDIO	EXIT
Data (PC)	acImcDev30ShowDataInDevice
Data (Device)	acSetup_ShowDataInPC

17.1.2.9 Additional functions

17.1.2.9.1 Product Information

Paths

Path are in the `Environment` class.

```
Core.Environment.ApplicationPath
Core.Environment.InstallationPath
Core.Environment.LogFilePath
Core.Environment.DefaultExperimentDirectory
```

Version information

The version information are in the `Environment` class.

```
Core.Environment.ProductVersionInformation.BuildDate
Core.Environment.ProductVersionInformation.ExtendedVersion; // 2023 R1
Core.Environment.ProductVersionInformation.ProductName
```

17.1.2.9.2 Default dialog response

In Scripting the options for the default dialog response can be set as follows:

```
using imc.Studio.Interfaces.Gui;
```

Constants for the dialog answers

```
public static class DialogConstants
{
   // Measurement running - reconnect not possible
   public const string StopAndConnect = "DevSetup_StopAndConnect";
   // Overwrite autostart configurations
   public const string OverwriteSelfStart = "Dev2x_OverwriteSelfStart";
   // Overwrite existing file
   public const string OverwriteFile = "VariablesOverwriteFileDialog";
   // Project has changed
   public const string SaveProject = "SaveProjectDocumentUserRequest";
   // Connection failed - transfer configuration
   public const string ConnectionFailed = "DevSetup_ConnectionFailed";
   // Disconnect from running measurement
   public const string DisconnectFromRunningMeasurement = "DevSetup_DisconnectFromRunningMeasurement";
   // No new devices found
   public const string NoNewDevicesFound = "DevSetup_NoNewDevicesFound";
   // Measurement running
   public const string Reconnect = "DevSetup_Reconnect";
   // Devices deselected
   public const string DeselectDevices = "DevSetup_DeselectDevices";
   // Experiment has changed
   public const string SaveExperiment = "SaveExperimentDocumentUserRequest";
   // Channel settings will be reset
   public const string ResetChannelSettingsWarning = "imcDevices2xResetSettingsWarning";
   // Changed supply settings
   public const string ModuleSupplyChangedWarning = "imcDevices2xModuleSupplyChangedWarningDialog";
   // Retrieve device configuration (Retrieve configuration / Stop and Connect)
   public const string RetrieveDeviceConfigurationBeforeReconnect =
"DevSetup_RetrieveDeviceConfigurationBeforeReconnect";
   // Hardware has changed: (Button1: Discard, Button2: Transfer, None = ShowDialog)
   public const string QuestionDeviceChanged = "imcDevices2xQuestionDeviceChanged";
}
```

Usage

```
UserRequest.SetDefaultResultValue(DialogConstants.StopAndConnect, EMessageBoxResult.OK); // Ok: Stop and
Connect - reconnect not possible

UserRequest.SetDefaultResultValue(DialogConstants.DisconnectFromRunningMeasurement,
EMessageBoxResult.Yes); // Yes: Disconnect, No: Stop and Disconnect

UserRequest.SetDefaultResultValue(DialogConstants.Reconnect, EMessageBoxResult.Yes); // Yes: Reconnect,
No: Stop and Connect

UserRequest.SetDefaultResultValue(DialogConstants.DeselectDevices, EMessageBoxResult.Yes); // Yes:
Discard, No: Transfer

UserRequest.SetDefaultResultValue(ModuleSupplyChangedWarning, EMessageBoxResult.OK);

UserRequest.SetDefaultResultValue(ResetChannelSettingsWarning, EMessageBoxResult.Yes);
```

```
// read setting for Overwrite autostart configuration
EMessageBoxResult result = UserRequest.GetDefaultResultValue(OverwriteSelfStart);
// None = Show Dialog
```

EMessageBoxResult	Description
Yes	The dialog will be automatically answered with "Yes".
No	The dialog will be automatically answered with "No".
OK	The dialog will be automatically confirmed with "OK".
Cancel	The dialog will be automatically canceled ("Cancel").
None	The dialog will be displayed
Button1, ..., Button 5	The dialog will be automatically answered with the corresponding button.

Futher information, see chapter Options: Default dialog response 129.

17.1.2.9.3 Script clipboard

If objects are to be exchanged between two scripts, the **ScriptClipboard** is available for this purpose.

Objects can be written to the Clipboard for subsequent import to a different script.

The lifetime of the clipboard is linked to the imc STUDIO process.

Copying a new object to the Clipboard (Script A)

```
string text = "My Object";
ScriptClipboard.Add("MyKey", text);
```

Querying an object from the Clipboard (Script B)

```
object o = ScriptClipboad["MyKey"];
string s = o as string;
```

Deleting an object from the Clipboard

```
ScriptClipboard.Remove("MyKey");
```

Testing for existence

```
if (ScriptClipboard.ContainsKey("MyKey")) {
  var value = ScriptClipboard["MyKey"];
}
```

If a device object is to be copied, the corresponding instance must be copied to the Clipboard.

Copying a device object to the Clipboard (Script A)

```
using imc.Studio.Interfaces.DevSetup;
```

```
EnumItem dev = DevSetup.Devices.GetItemBySerialNumber("123456");
IEnumItemBase deviceInstance = dev.Base.GetAPI<IEnumItemBase>();
ScriptClipboard.Add("MyDeviceInstance", deviceInstance);
```

Getting a device object from the Clipboard (Script B)

```
using imc.Studio.Interfaces.DevSetup;
```

```
object o = ScriptClipboard["MyDeviceInstance"] ;
IEnumItemBase item = o as IEnumItemBase;
EnumItem device = EnumItem.GetInstance(item);
```

Note

- When adding an object to the **script clipboard**, it is necessary to note that this object has a **namespace**. In the case of a Widget, this would be the script's name, e.g. *TestScript.Widget,* due to the proxy-classes.
- When retrieving the object from the script clipboard, *Casting-Exceptions* can occur, since the object type belongs to a different namespace than the script.
- To avoid this, create the script which writes the object to the script clipboard with option "**Separate proxy class library**".
- the script is created with the resource **ProxyClasses**.
- Incorporate the file **ProxyClasses.dll** from this script project into the script project which is to import the object from the clipboard, as a **Resource**.
- When using the type, be sure to specify the namespace correctly.

Reference

For details, see Using the script clipboard 1916

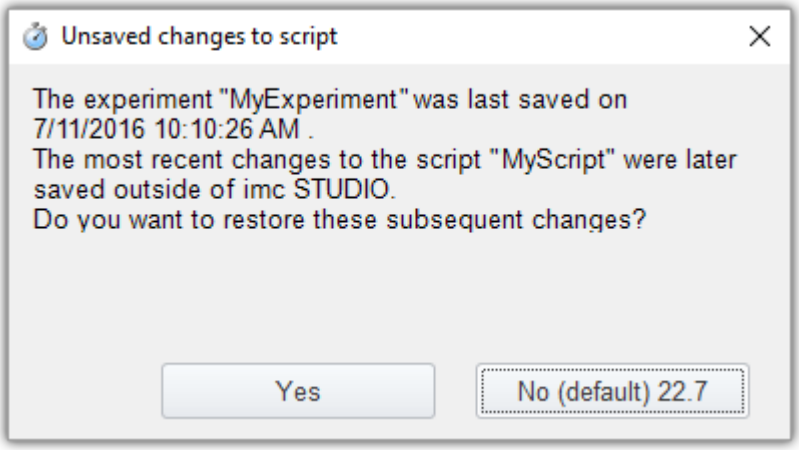
17.1.2.9.4 Restore script changes

Note

The scripts are not only saved in the **Storage scope** (Experiment, Project, ...), but also in a separate folder. Therefore the saving states can be different.

If modifications are made with the script and the corresponding **Storage scope** is not saved, these **changes** to the script are not saved with the experiment/project.

When reloading, the following dialog will open:



Dialog: Unsaved changes to Script

You have the possibility to restore the changes of the script or to load the current state of the Storage scope (*here: Experiment*).

Yes The last changes of the **Script** are used.

No The current state of the **Storage scope** (*here: Experiment*) is used.

In the Script options 1906 a default answer can be set for this dialog.

17.1.2.9.5 Placeholder

For using placeholders, you need to get the *TextPlaceholderResolver* from the SharedComponents.

With the method *ResolveTextAndPLVars()* the placeholder can be resolved.

```
using imc.Common.Interfaces.TextPlaceholder;
using imc.Studio.Interfaces.Core;
using imc.Studio.Interfaces.Core.StandardCollections;
```

```
string placeholderToResolve = "<EXPERIMENT.NAME>"
ITextPlaceholderResolverBase placeholderResolver = SharedComponents["ITextPlaceholderResolver"] as
ITextPlaceholderResolverBase;
string experimentName =
placeholderResolver.GetAPI<API_TextPlaceholderResolver_V6>().ResolveTextAndPLVars(this,
placeholderToResolve, false);
```

17.1.2.9.6 File access (imc-Format)

Read channel from imc file

```
using imc.Common.Components.DataManager;
```

```
string filename = @"c:\imc\imc FAMOS\dat\slope.dat";
DmFile file = new DmFile();
file.Open(filename, DmFileFlags.Read);
DmChannel channel = file.ReadChannel("Slope");
```

```
file.Close();
file.Dispose();
channel.Dispose();
```

Save channel in imc file

```
using imc.Common.Components.DataManager;
```

```
string channelName = "Channel";
DmChannel channel = new DmChannel();
channel.Name = channelName;
channel.Data = new double[]{1, 1, 2, 3, 5, 8, 13};

string filename = @"c:\temp\channel.dat";
DmFile file = new DmFile();
file.Open(filename, DmFileFlags.Write);
file.AddObject(channel.DmHandle);
```

```
file.Close();
file.Dispose();
channel.Dispose();
```

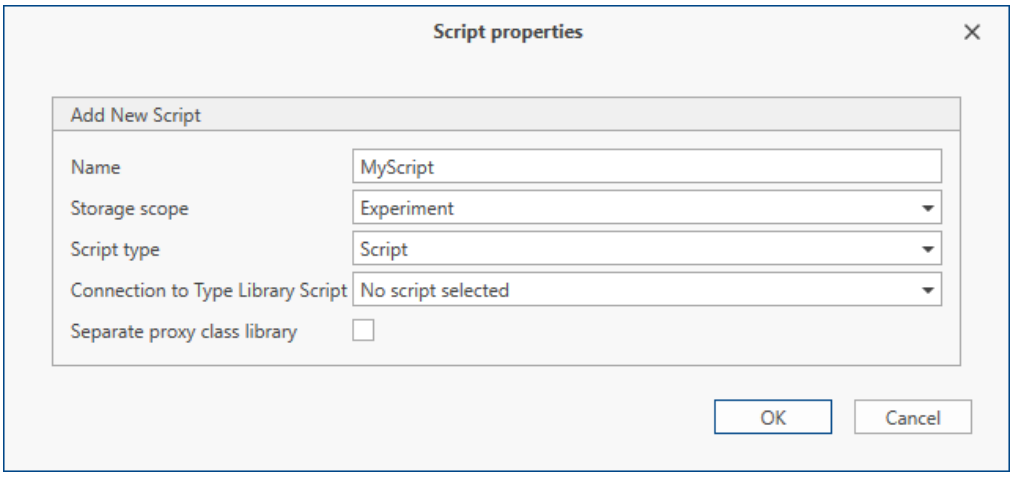
17.1.3 Script types

17.1.3.1 Script

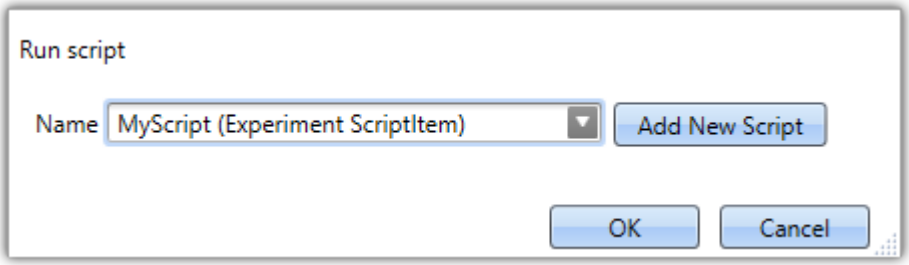
Scripts are run in response to:

- the command *Scripts > Run script* in the Sequencer,
- the command *Scripts > Run script* as button events on a Panel page,
- click on *Edit > Run* in the tool window *Scripts*.

The **Storage Scope** option can be set to Experiment, Project, Application or Sequencer in which the **script** is available.



Dialog: create Script



Command: Run script

Editor

When a new Script is added, the **ScriptEditor** will open with the following code:

```
namespace MyScript {
   using System;

   public class MyScript : AbstractScript {

      public override void Run() {
         // Start your script here
         // Access your variables by using "Variables[variableName]"
         // Access your devices by using "DevSetup.Devices[devicename]"
         // Access your channels by using "DevSetup.Channels[channelname]"
      }
   }
}
```

The code has to be inserted into the method `public override void Run()`.

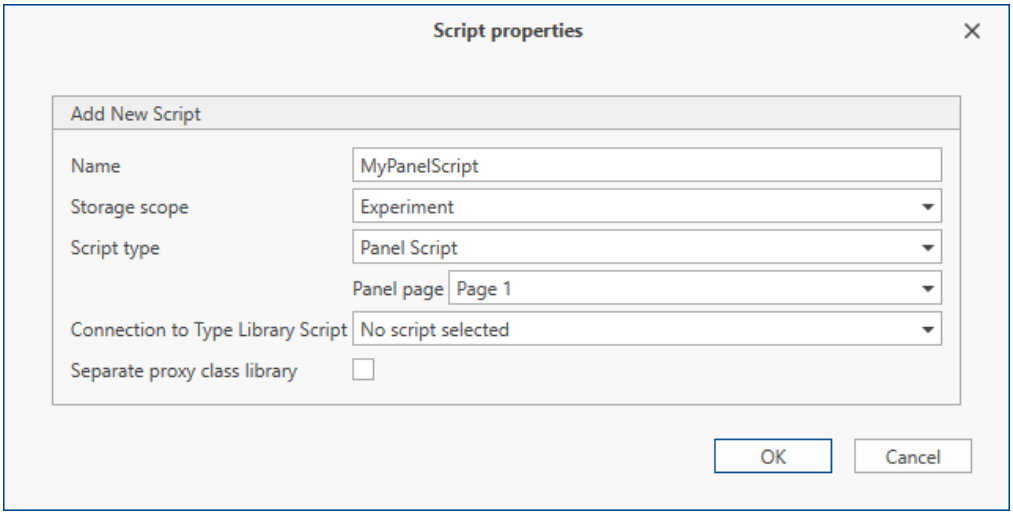
The entry points for accessing variables, devices and channels are stated in comments.
Helpful **examples** are given in the region *Examples*.

17.1.3.2 Panel script

Panel Scripts can not be run in response to commands. They are linked with the corresponding **Panel page**.

When creating a **Panel script** , the corresponding Panel page must be selected.

If no page is selected, the script can be linked to a page via its properties.



Dialog: Creating a Panel-script

If a new **Panel Script** is added, the **ScriptEditor** will open with the following code:

```
namespace myScript {
   using System;

   public class myPanel : AbstractmyPanel {

      // Custom ComboBox Example (in Region)
      // Custom ListBox Example (in Region)

      // public override bool PanelScriptInitialize() {}
      // public override bool PanelScriptDispose() {}

      /// Rocker_switch1 : ButtonWidget
      /// Standard_meter1 : PotentiometerWidget
      public override void Run() {
         // Add your initialization code here
      }

      public override void Stop() {
         // Add your cleanup code here
      }
   }
}
```

The Widgets on the Panel page are listed in comments above the function `Run()` . In this example a rocker switch (`Rocker_switch1`) and a gauge Widget (`Standard_meter1`) are on the Panel page.

The finished **Panel Script** must be compiled before being run.

The **`Run()`** and **`Stop()`** method are executed, when opening or leaving the corresponding Panel-page.

Method	is executed, when ...	Requirement / Characteristics
`Run()`	when changing to the corresponding Panel page	an other Panel page is opened
	when loading the experiment	**all** Panel scripts are executed, which are linked to a Panel page
	when leaving the Design mode	the corresponding Panel page must be opened
`Stop()`	when changing to an other Panel page	the corresponding Panel page must be opened
	when closing the experiment	the corresponding Panel page must be opened
	when entering the Design mode	the corresponding Panel page must be opened
When changing the main window, **none** of the methods are executed.		

Optionally, there are the following override methods available:

Methode	is executed, when	Application
`PanelScript Initialize()`	changing the script[1] 1: if necessary, the page must be changed or the Design mode must be toggled.	To initialize and register events on Widgets or Windows-Forms.
	loading the experiment and browsing to the Panel page.	
`PanelScript Dispose()`	closing the experiment	To dispose and unregister Events on Widgets or Windows-Forms.
	before the re-initialization.	

Note

- If events are added on a button action, these events have to be unhooked in the `Stop()` method. Otherwise this action would be performed twice/many times when starting the script again.
- With `this.PanelPage.Name` the current Panel page name can be used.

Widget

The properties of a Widget can be changed by the following line. Be sure, that the property is available for the Widget.

Using class object

```
Standard_meter1.Max = 30;
Standard_meter1.Min = 10;
Standard_meter1.Interval = 10;
```

Access via Panel page

```
Panel["Page 1"]["Standard_meter1"].Max = 30;
Panel["Page 1"]["Standard_meter1"].Min = 10;
Panel["Page 1"]["Standard_meter1"].Interval = 10;
```

Standard-Widget

```
// Change text from Standard-Widgets
Label1.StandardText = "Hello World!";
Label1.TextColor = Color.Aquamarine; // System.Drawing
Button1.StandardText = "Hello World!";
```

For futher information, please see chapter Panel and Widget functions 1840.

Add new Widgets

If new Widgets are added on the Panel page, the following steps are needed in order to use these Widgets in the script:

1. Close the **Editor**
2. Highlight the **Panel Script**
3. Regenerate the *Script Widgets* using the script action in the tool window **Scripts**
4. Edit the **Panel Script**
5. Compile

When deleting Widgets, it is necessary to ensure that the script no longer accesses the widget.

Variables and Logbook

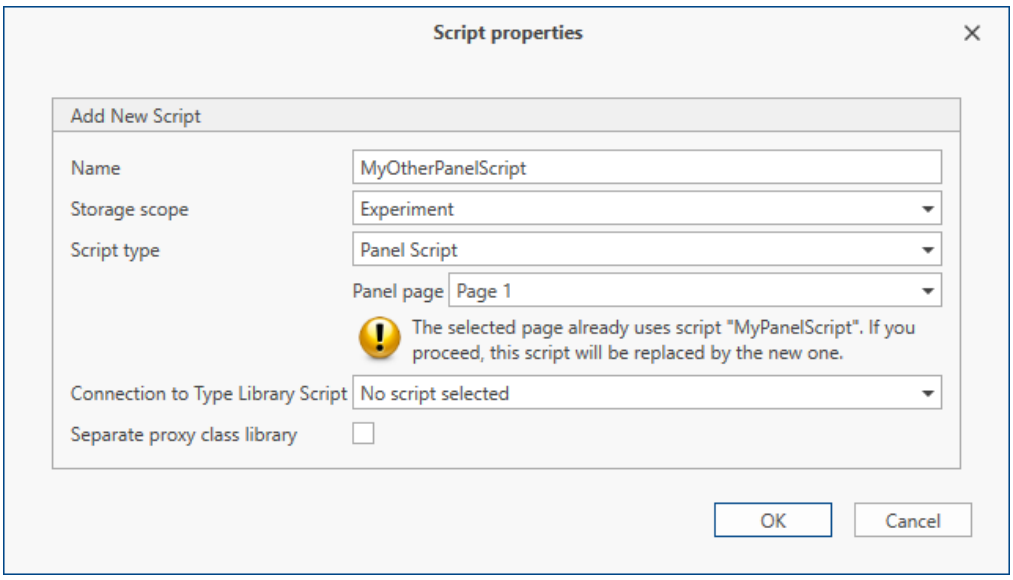
Use class `Core` to access to the variables or logbook.

```
double myVariable = Core.Variables["VariableName"].Value.Get<double>();
Core.Logbook.LogEntry("MySender", "MyMessage", 0, ELogbookEntryCategory.Warning);
```

Link to Panel page

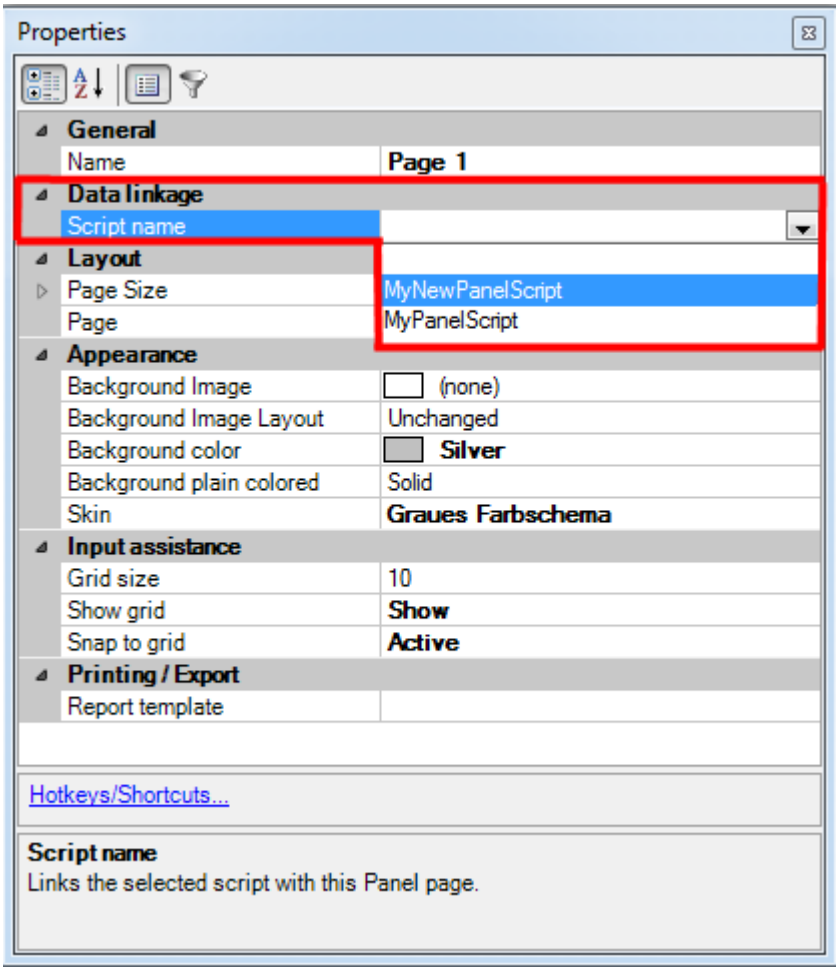
Warning

- If a Panel page is already linked with a Panel script, then the linkage is replaced with the new Panel script. The old Panel script is then no longer linked with any Panel page.
- If you import a Panel script, you have to link the script to the corresponding Panel page in the page properties.



Dialog: Panel Script - choose already linked page

In the properties of a Panel page the linked script can be changed.



Page properties: Script name

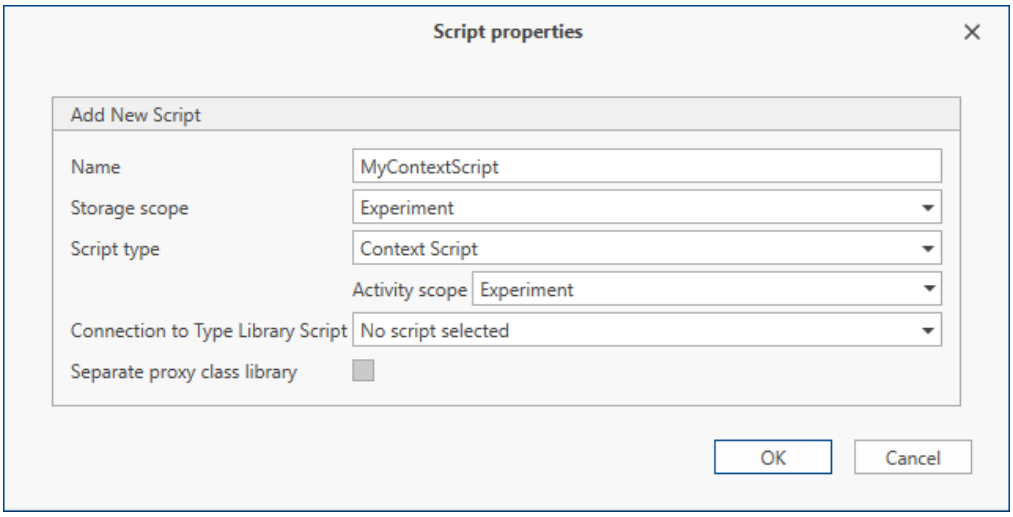
See Examples 1883

17.1.3.2.1 Examples

Switching visibility of a standard meter by a rocker switch

```
bool isHooked = false;

public override void Run() {
   if (!isHooked){
      isHooked = true; // Status -> angemeldet
      Kippschalter1.SwitchOn += new Action<object, EventArgs>(turnon);
      Kippschalter1.SwitchOff += new Action<object, EventArgs>(turnoff);

}

public void turnon(object sender, EventArgs args){
   if (isHooked){
      Zeigerinstrument1.Visible = false;
   }
}

public void turnoff(object sender, EventArgs args){
   if (isHooked){
      Zeigerinstrument1.Visible = true;
   }
}

public override void Stop() {
   if (isHooked){
      isHooked = false; // Status -> abgemeldet
      Kippschalter1.SwitchOn -= turnon;
      Kippschalter1.SwitchOff -= turnoff;
   }
}
```

17.1.3.3 Context script



Dialog: create Context Script

To add a **Context Script**, the appropriate **Activity Scope** (Experiment, Project, Application, Sequencer) has to be chosen.

The script calls the `Run()` method, when **opening** the corresponding scope. When **leaving** the scope the `Dispose()` method is called.

For example, when opening an experiment containing a **Context Script** with **Activity Scope** *Experiment*, the method `Run()` is called.

Opening other experiments or closing imc STUDIO calls the method `MyContextDispose()`. It is also called, when the script is deleted. The prefix *MyContext* is derived from the script name.

```
namespace MyContext {
   using System;

   public class MyContext : AbstractScript {

      public override void Run() {
         // Start your script here
      }

      protected override void MyContextDispose() {
         // Release all your resources and event hooks here.
      }
   }
}
```

Warning

Since these methods are called while the experiment (or project, ...) is being loaded/closed, it is not permitted to implement any **blocking** functionalities within them.

For example, the display of a message-box would interrupt the loading process. In this case, the system is waiting for the user to provide confirmation to the dialog, and only subsequently can loading of the experiment be completed.

This can cause errors and other unintended effects to occur.

Note Variables and Context scripts

When using **Context scripts**, note that not all variables are available when the `Run()` method is called.

If variables are to be used immediately, the *VariableCreated* event of the variable should be used in the **Context script**, see Events 1862.

React to events

With a Context script you can react to device events. The event names are listed on the Sequencer page.

```
using imc.Common.Interfaces.Miscellaneous;
using imc.Studio.Interfaces.Core;
```

```
public override void Run() {
   Events["Devices_Connected"].Execute += new
CommonEventHandler<ICoreEventEventArgsBase>(MyContextScript_Execute);
}

void MyContextScript_Execute(object sender, ICoreEventEventArgsBase e){
   // do something
}

protected override void MyContextScriptDispose() {
   Events["Devices_Connected"].Execute -= MyContextScript_Execute;
}
```

Text

With a Context script you can react to the manually execution of a menu action.

```
using imc.Studio.Interfaces.Core;
```

```
public override void Run() {
   Actions["COMMANDAME"].Execute += new
CommonEventHandler<ICoreActionExecutionEventArgsBase>(Test_Execute);
}

void MyContextScript_Execute(object sender, ICoreActionExecutionEventArgsBase e){
   // do something
}

protected override void MyContextScriptDispose() {
   Actions["COMMANDNAME"].Execute -= MyContextScript_Execute;
}
```

For further information, please read chapter Commands 1870.

See Examples 1886

17.1.3.3.1 Examples

In the following example, a dialog is shown, if the state of a channels has changed. In the method `Run()` the action-method `ValChanged` is assigned to the event `DevSetup.ValueChanged`. The parameters can be derived from the `DevSetupValueChangedArgs` and be processed.

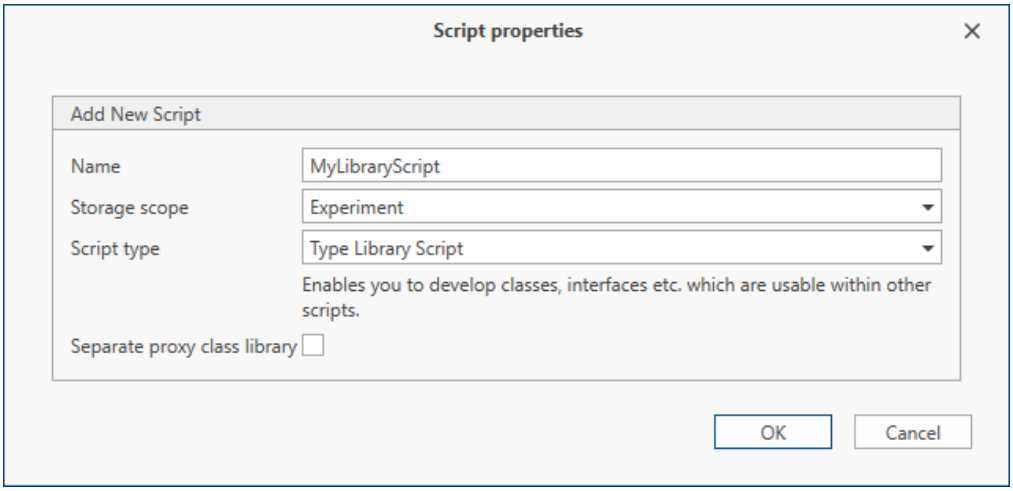
Within the method *ValChanged* a `try-finally` and a `if`-condition due to *InValueChanged* should be added.

The script starts, when the experiment is opend (Activity Scope *Experiment*).

Show Dialog, if channel parameter "Status" has changed

```
using System.Windows.Forms;
using imc.Studio.Tools.DevSetupGraphGenerator;

namespace ChannelStateChangedHandler
{
   public class ChannelStateChangedHandler
   {
      private bool isHooked = false;
      private bool InValueChanged = false;

      public override void Run()
      {
         if (!isHooked)
         {
            isHooked = true;
            DevSetup.ValueChanged += new Action<object, DevSetupValueChangedArgs>(ValChanged);
         }
      }

      private void ValChanged(object arg1, DevSetupValueChangedArgs arg2)
      {
         foreach (var item in arg2.Changes.Keys)
         {
            if (item.ClassID == EClassID.eStatus.ToString())
            {
               var channel = DevSetup.Channels.GetItemByHWID(item.HWID);
               if (channel != null && channel.Base != null)
               {
                  var channelNameItem = channel.Parameters[EClassID.eChannelName];
                  if (channelNameItem != null)
                  {
                     string strChannelName = channelNameItem.GetValue() as string;
                     MessageBox.Show(strChannelName + "=" + item.Text, "State changed");
                  }
               }
            }
         } // for each
      } // of void

      protected override void ParameterChangedDispose()
      {
         // Release all your resources and event hooks here.
         if (isHooked)
         {
            DevSetup.ValueChanged -= ValChanged;
            isHooked = false;
         }
      }
   } // of class
} // of namespace
```

17.1.3.4 Type library script



Dialog: create Type Library Script

In the **Type-Library-Script**, it is possible to enter one's own classes and methods which can then be embedded into the other script types where they are subsequently available for use. A **Type-Library-Script** can also be used in multiple other scripts. For this reason, this kind of script is especially useful when certain classes and methods are to be used repeatedly. Thus, the process of embedding these into each script separately is not necessary.

You can implement your own classes and methods in a **Type Library Script**.

A **Type Library Script** can be connected to other script types in the *Script options* dialog, when creating scripts.

```
using System;
namespace MyLibrary {
   // Insert your TypeLibrary code here
}
```

Warning

- Please make sure when using **Type Library Script**s, that the *Storage Scope* is chosen so that the **Type Library Script** is accessible from the linked scripts, even if you change the *Storage Scope*.
- After making changes to a type-library-script to which there is any link, it is necessary to restart imc STUDIO for the changes to take effect.

Note

It is recommended to set the **Assembly Version** (see file *AssemblyInfo.cs*):

```
[assembly: AssemblyVersion("1.0.1.0")]
```

See Examples 1888

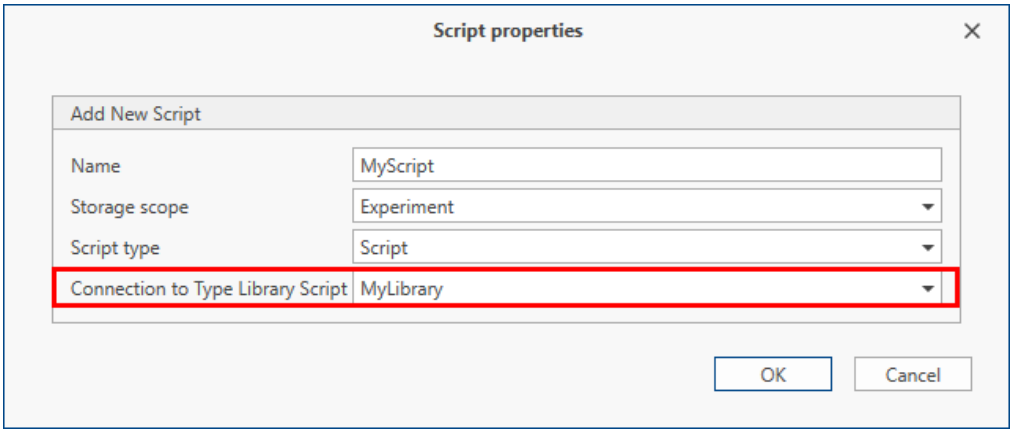
17.1.3.4.1 Examples

Type Library Script

```
using System;
namespace MyLibrary {
   // Insert your TypeLibrary code here
   public class MyClass
   {
      private double dNumber = 12;
      public string ClassName = "MyClass";

      public void SetNumber(double d){
         this.dNumber = d;
      }

      public double GetNumber(){
         return this.dNumber;
      }
   } // end of class MyClass
}
```

Script

The actual script is created with the connection to the Type Library Script *MyLibrary*.

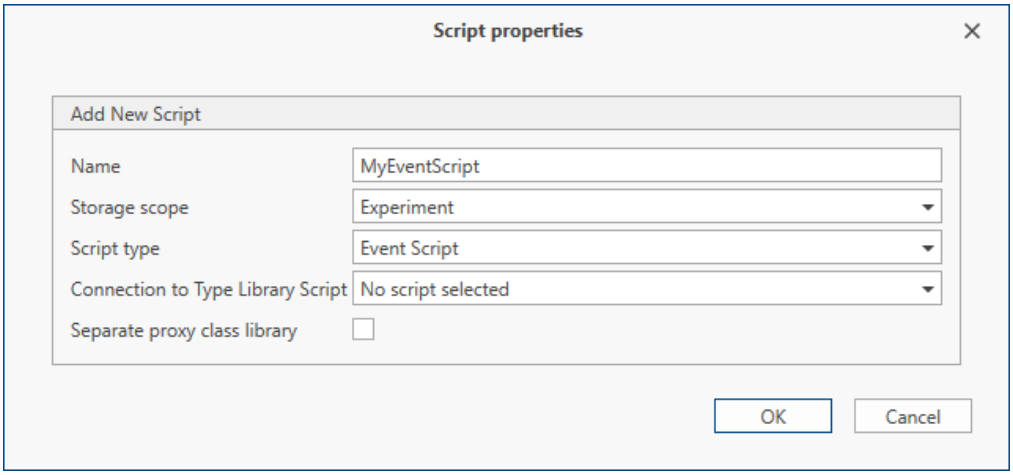


Script with Type Library Script

The class and methods of the library can be used in the script.

```
namespace MyScript {
   using System;
   using MyLibrary;


   public class MyScript : AbstractScript
   {
      public override void Run()
      {
         MyClass a = new MyClass();
         a.SetNumber(8);
         double dNum = a.GetNumber();
         string name = a.ClassName;
      }
   }
}
```

17.1.3.5 Event script



Dialog: Create Event script

To use an **Event script**, add the Run script command to an event in the Sequencer.

Within the script, the event arguments can be accessed.

```
namespace MyEventScript
{
   public class MyEventScript : AbstractScript
{
      public override void Run()
      {

      }
   }
}
```

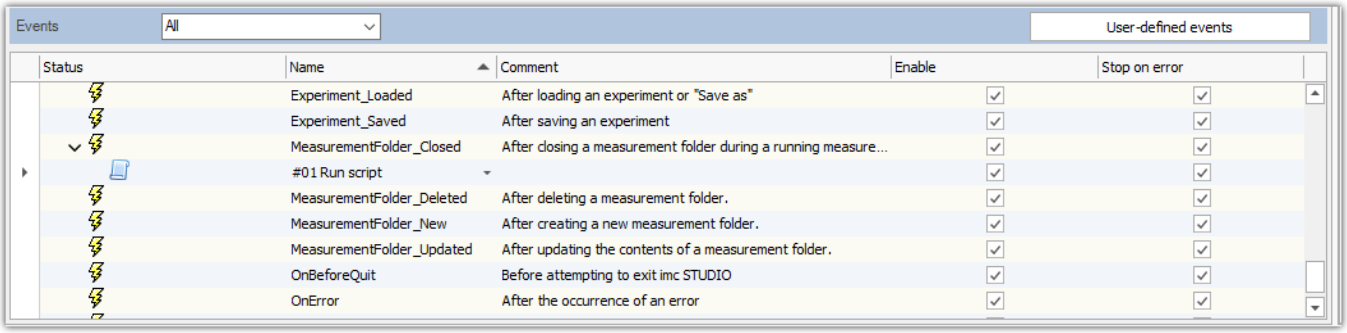
17.1.3.5.1 Measurement events

In the Sequencer the following measurement events are available:

Event name	Description: Fired when
MeasurementFolder_Closed	...when a measurement folder has been completely closed, e.g. after a measurement.
MeasurementFolder_Deleted	... when a measurement folder has been deleted
MeasurementFolder_New	... when a measurement folder is created
MeasurementFolder_Updated	... when new data has been added to the measurement.

The former event "Storage_DirectoryUpdate" is transferable to the event "MeasurementFolder_Closed".

The following EventArgs are available in the Event script: **this.MeasurementFolderEventArgs**



Sequencer: Measurement events

```
using System.IO;
using System.Windows.Forms;

namespace MeinEvent
{
   public class MeinEvent : AbstractScript
   {
     public override void Run()
     {
         var args = this.MeasurementFolderEventArgs;

         if (args != null)
         {
            MessageBox.Show(args.DisplayName.ToString());
         }
         else
         {
            MessageBox.Show("null");
         }
     } // Run
   }
}
```

17.1.3.6 Third party device script

Reference

For more information see chapter Third Party Device Script 1918.

17.1.4 EClassID - Parameters

17.1.4.1 Device parameters

Note

For setting the parameters, class *EClassID* is used. Setting values of the parameter are available in the *ParameterValues*-Class. The `EClassID` device parameters are listed below.

To use the class *EClassID* the following **library** has to be added:

```
using imc.Studio.Tools.DevSetupGraphGenerator;
```

Please click on the parameters to see the corresponding tables.

Device Control (Action)

eDeviceControlAction	
Connect	1000
Prepare	2000
Start	3000
Stop	5000
Disconnect	7000

`ExecuteAction()` has to be run after setting the value.

Diskstart (Action)

eDiskStartAction	
PC + Device HD*	2501
PC HD	2502
Device HD	2503
Removable Device	2504
Device intern	2505
Device intern RAM*	2506
All Device mem space*	2507

`ExecuteAction()` has to be run after setting the value.

* Functions not implemented.

eDiskStartOption	
immediately	1
defined time	2
next second	3
next minute	4
next hour	5
next day	6
automatically	7
next 10th minute	8

Synch Signal: Input/Output

eSynchSignalInput eSynchSignalOutput	
no signal	1
DCF	2
GPS	3
SYNC + GPS	4
Synch. signal NTP	5
Synch. Ethercat	6
PTP	7
IRIG B002W	8

Start option

eStartOption	
immediately	1
defined time	2
next second	3
next minute	4
next hour	5
next day	6
automatically	7
next 10th minute	8

Connection state

eConnectionStatus	
disconnected	0
connected	1

Measurement state

eMeasurementStatus	
stopped	1
running	2
reconfigured	3

17.1.4.2 Channel parameters

Note

For setting the parameters, class *EClassID* is used. Setting values of the parameter are available in the *ParameterValues*-Class. The `EClassID` channel parameters are listed below.

To use the class *EClassID* the following **library** has to be added:

```
using imc.Studio.Tools.DevSetupGraphGenerator;
```

Please click on the parameters to see the corresponding tables.

Status

eStatus	
active	1
passive	2
write	1001
read/write	1002

Channel Type

eChannelType	
Analog Input	1
DAC-Output	2
Digital inputs/outputs (ports)	3
Trigger	4
Virtual Channels	5
Digial inputs/outputs (bits)	6
Virtual bits	7
Net Bits	8
Counter inputs	9
Temperature reference Channel	10

eChannelType	
Monitor: Analog inputs	11
Monitor: Digital inputs / outputs (ports)	12
Monitor: Counter inputs	13
Monitor: Digital inputs / outputs (bits)	14
Display Variables	15

eChannelType	
Fieldbus: Analog inputs	17
Fieldbus: Digital input / output (ports)	18
Fieldbus: Digital input / output (bits)	19
Fieldbus: Sum event	20
Fieldbus: Sending Channels	21

eChannelType	
Synthesizer	22
Communication Object	23
Process-Vector Variables	24
Video	1001
Monitor: Video	1002
Digital inputs/outputs (ports)	4099
Digital inputs/outputs (ports)	8195

Balance (Action)

eBalanceAction	
Offset	2011
Bridge	2012
Tare	2013
Manufacture's calibration	2014

`ExecuteAction()` has to be run, after setting the value.

Balance Status

eBalanceStatus	
balance state undefined	0
not balanced	1
bridge balance run	2
bridge balance ok	3
tare run	4
tare ok	5
offset run	6
offset ok	7
manufacture run	8
manufacture ok	9

Cable Calibration (Action)

eCalibrationAction	
Two-Point Scaling	2021
Invalid	2022
Cable Compensation	2023
Reset Cable	2024
Reset Two-Point	2025

`ExecuteAction()` has to be run, after setting the value.

Calibration status

eCalibrationStatus	
undefined	0
not balanced	1
Point 1 running	10
Point 1 ok	11
Point 2 running	12
Point 2 ok	13
Point 1 and 2 ok	14
Two-Point scaling running	15
Two-Point scaling ok	16
Two-Point reset running	17
Two-Point reset ok	18
Cable compensation running	19
Cable compensation ok	20
Cable compensation reset running	21
Cable compensation reset ok	22

Filter Type

eFilterType	
Low Pass	1
High Pass	2
Band Pass	3
without	4
AAF	5
Slow	6
Fast	7
Impulse	8
Peak	9

Filter Characteristics

eFilterCharacteristic	
Butterworth	1
Bessel	2
Tschebbychew	3
Cauer	4
A-weighting	5
B-weighting	6
C-weighting	7
D-weighting	8
without weighting	9

Event

eEvent	
Signal = 1	381
Signal = 0	384
Signal change 1 -> 0	383
Signal change 0 -> 1	384
Positive slope	391
Negative slope	392
Signal > Threshold	393
Signal < Threshold	394
Signal entering range	401
Signal exiting range	402
Inside of range	403
Outside of range	404

Event-Type

eEventType	
Threshold	1
Range	2
Ratio	3

Coupling

eCoupling	
AC	1
DC	2
Half-Bridge	3
IEPE	4
Full-Bridge	5
Quarter-Bridge	6
Microphone	7
DC charge	8
AC charge	9
TTL	10

eCoupling	
Strain gauge - quarter-bridge	1001
Poisson half bridge	1002
Half bridge with 2 active strain gauges in uniaxial direction	1003
Half bridge with one active and one passive strain gauge	1004
Full bridge with Poisson strain gauges in opposed branches	1005
Full bridge with Poisson strain gauges in adjacent branches	1006
Full bridge with 4 active arms	1007
General strain gauge - full-bridge	1008
General strain gauge - half-bridge	1009

© 2023 imc Test & Measurement GmbH

Channel mode

eChannelMode	
Sampling / Distance (diff.)	1
Bit Output / Angle (diff.)	2
Bit Input / Speed	3
RPM	4
Events	5
Frequency	6
Time	7
Pulse time	8
Angle(abs)	9
Distance (abs)	10
Voltage	1000
Current	1001
Strain gauge	1002
Charge	1003

Preprocessing

eProcessing	
none	1
Arithmetic mean value	2
Minimum	3
Maximum	4
RMS	5
MinMax	6
Reduction	7

17.1.4.3 Experiment parameters

Note

The *EClassID* parameters for the **experiment** are set as *string* listed in the following table.

Example

```
DevSetup.Experiment["CompanyName"].SetValue("imc");
DevSetup.Experiment["TestEngineer"].SetValue("Average Joe");
```

Project

Description	EClassID parameter
Company name	CompanyName
Hint	Hint
Project document	ProjectDoc
Linked project document	LinkedProjectDoc
Project documentation path	ProjectDocPath
Project officer	ProjectOfficer
E-mail address project officer	EMailProjectOfficer

Test engineer, test setup, test object, test operator

Description	EClassID parameter
Test engineer	TestEngineer
Test engineer e-mail	EMailTestEngineer
Test setup description	TestObjectDesc
Test station photo	TestObjPic
Test setup document	TestObjectDoc
Test object description	TestSetupDesc
Test object picture	TestStationPhoto
Test object document	TestSetupDoc
User	User
E-mail address user	EMailUser

Annotation

Start: before measurement

End: after measurement

Description	EClassID parameter
Comment (start)	AnnotationStart
Environmental conditions (start)	ConditionsStart
Conditions parameter name (start)	ConditionsParamNameStart
Conditions parameter value (start)	ConditionsParamValueStart
Picture (start)	PicStart
Location	Location
Date	Date
Time	Time
Comment (end)	AnnotationEnd
Environmental conditions (end)end)	ConditionsEnd
Conditions parameter name (end)	ConditionsParamNameEnd
Conditions parameter value (end)	ConditionsParamValueEnd
Picture (end)	PicEnd

Result

Description	EClassID parameter
Result status (name)	ResultStatusName
Result status (value)	ResultStatusValue
Result characteristic name	ResultCharacName
Result characteristic	ResultCharac
Result description	ResultDesc
Result document	ResultDoc
Result picture	ResultPic

Misc

Description	EClassID parameter
Test part number	TestPartNo
Test object number	TestObjNo
Short Description	DescriptionShort
Description	DescriptionLong

17.1.4.4 Finding parameters and setting values

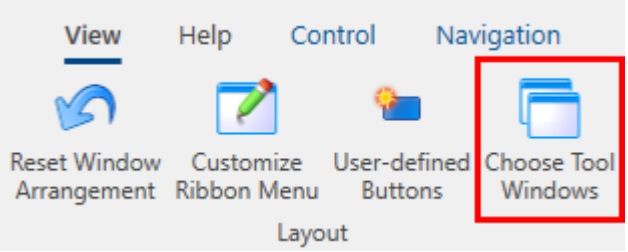
Finding parameters and their values

You can find a look-up table of the **EClassID parameters** in the document *imc_STUDIO-EClassID-Parameters.pdf* in the **installation package** under *\Products\Development\Documentation\en*

The following describes an alternatively way to find the **EClassID** parameters on the basis of the table description. Another way is described in Finding setting values 1902.

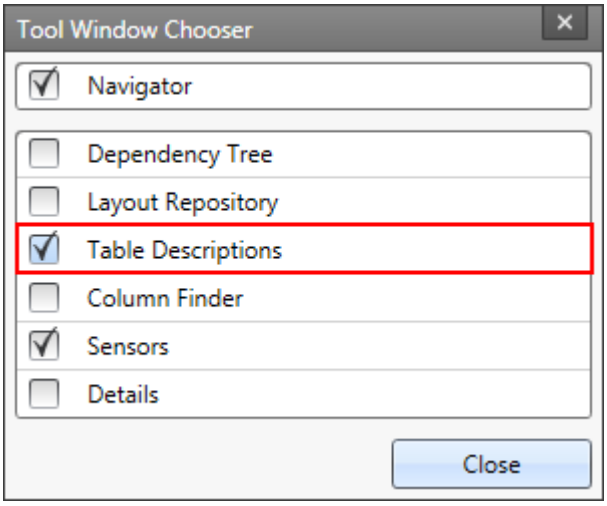
Activating the Table Description tool window

Start by activating the tool window *Table description* on the **Setup** page. Once you are on the Setup page, click in the menu ribbon on *View > Layout > Tool Windows*.



View > Layout > Tool Windows

A new dialog will open. Put a check in the box for *Table Descriptions* and close the window. A new tab is added at the right edge of the screen.

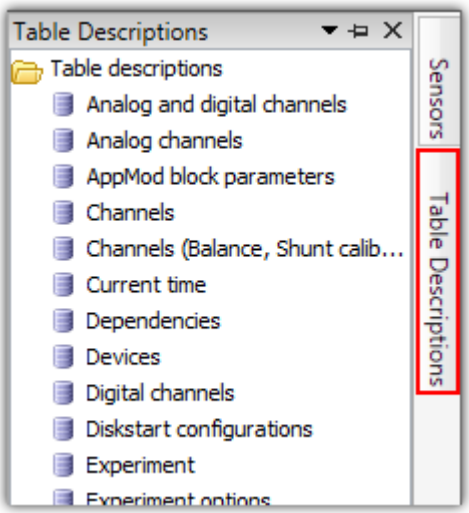


Dialog: Tool Window Chooser

The view can also be saved, so that the tool window is available once again the next time imc STUDIO is started.

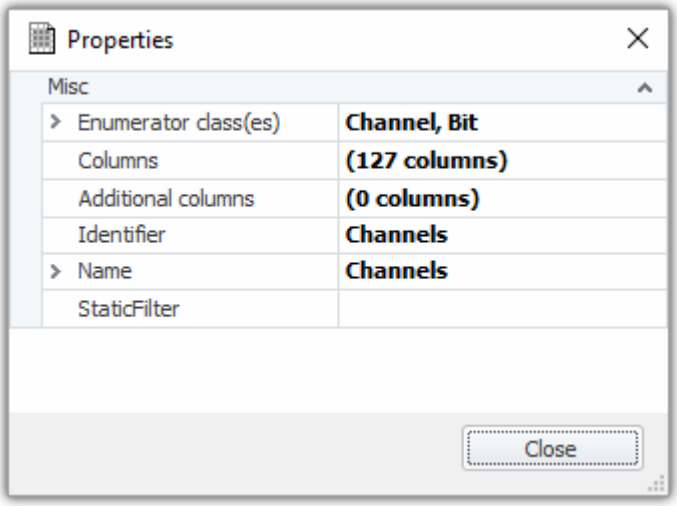
Viewing a table description

Click on the tab *Table description* on the right edge to open the description.



Tool window: Table description

Click in the context menu of the table in which the desired parameter is located, for example on the item *Properties* in the context menu for *Analog Channels*.



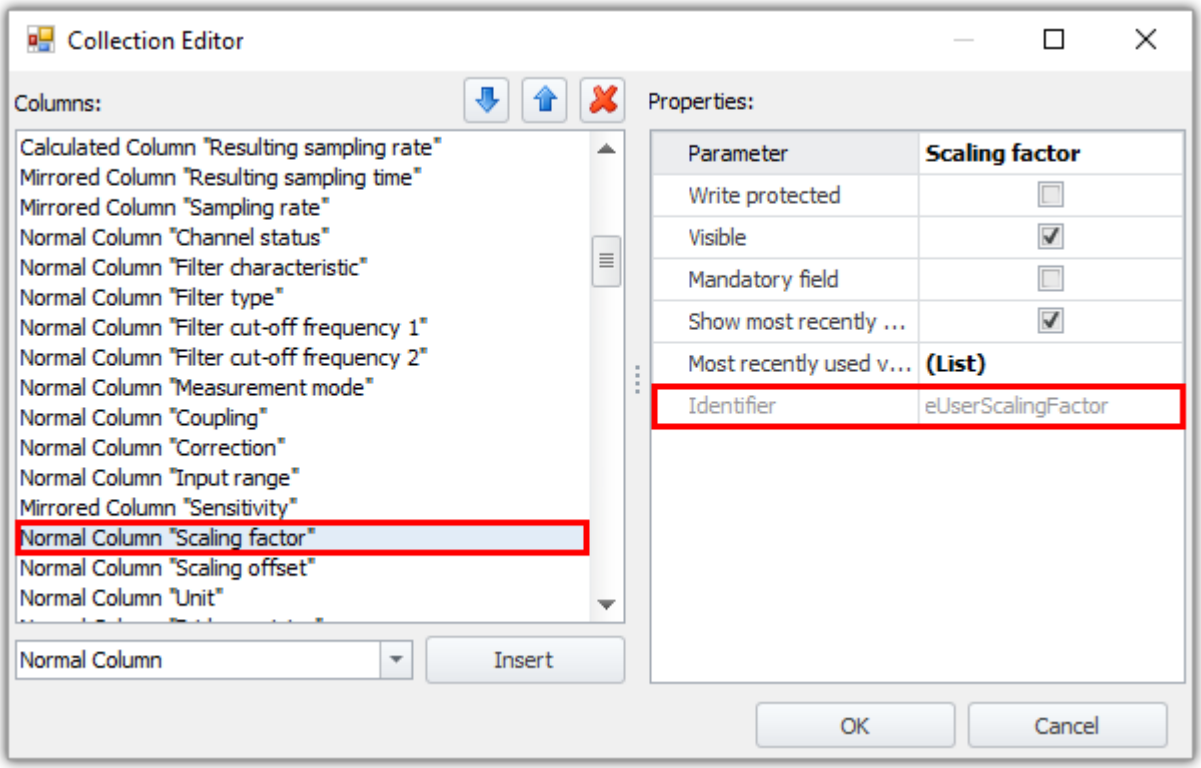
Properties: Analog channels

Click on the button with three dots "..." at the entry for Columns.

Parameter Identifier

After opening the column descriptions, select the pertinent parameter on the left side. On the right side, the associated identifier is stated.

In this example, the *Scaling factor* has been selected and is the identifier indicated is *eUserScalingFactor*.



Collection Editor

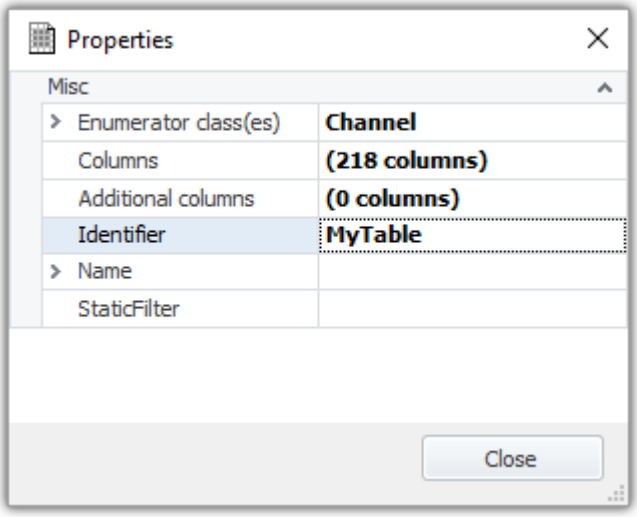
Finding setting values

The setting values can be found in the class `ParameterValues` or in `imc.Studio.Tools.DevSetupGraphGenerator.DevSetupCommonValues`.

Another way to get the parameters and the respective values to which they are set is described in the section below.

Creating one's own table description

The easiest way is to create one's own table description. To do this, right-click in the *Table Descriptions* tool window on the folder icon (also titled *Table Descriptions*) to open the context menu, where you then select the item *Add*. Enter a name for the table description.

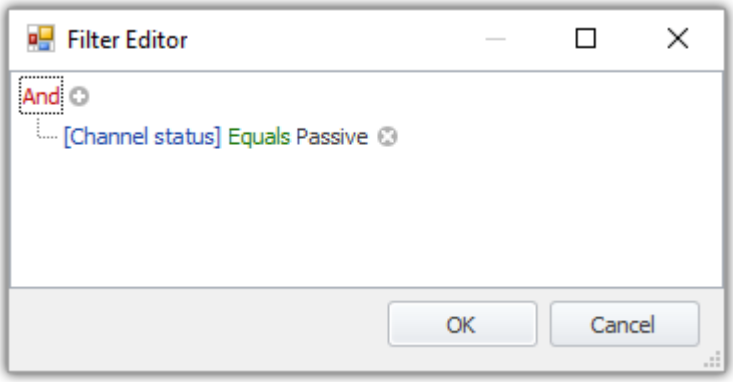


Properties of a user-defined table description

As the Enumerator class, select the desired class containing the parameter, e.g. *Channel*.

Filter Editor

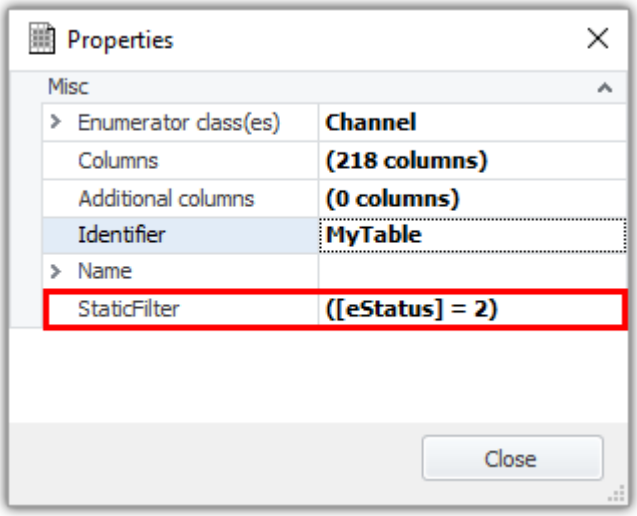
Under *StaticFilter*, select the parameter of which you wish to know what value is set for it, e.g. *Channel status*. For the value, either enter the value desired or select it from the drop-down list. Close the window.



Filter Editor

Viewing

Subsequently, you can view the **EClassID**-name and the value to which it is set under *StaticFilter* in the Properties window, e.g. ([eStatus] = 2). In this case, it means that the channel status is passive.

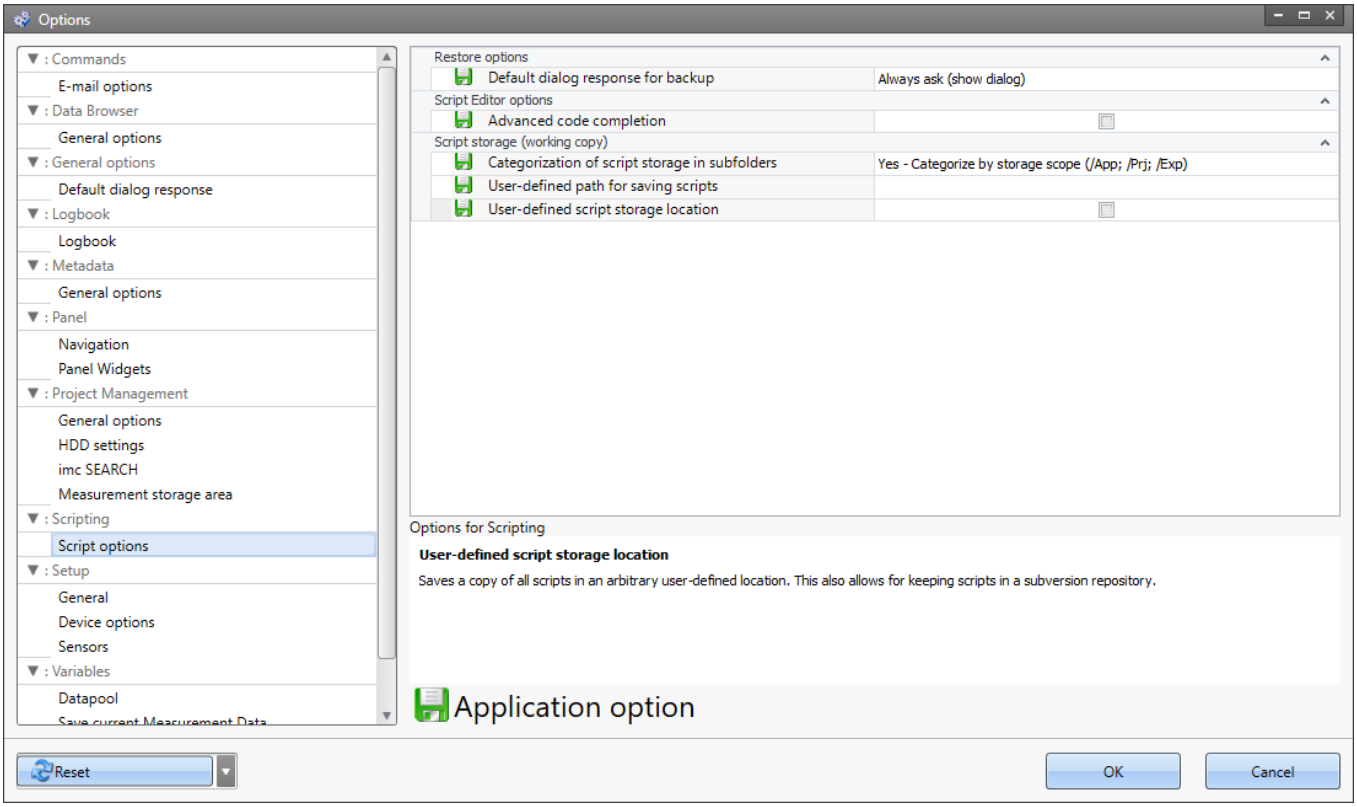


Properties: StaticFilter

17.1.5 Script options

The options can be reached by using the ribbon or by clicking on the button *Script options* in the tool window *Scripts*.

Ribbon	View
Extra > Options ( )	All



Script options

Script storage (working copy)

Option	Description
Categorization of script storage in subfolders	The categorization specifies if different subfolders depending on the Script storage should be used or one common folder.
User-defined path for saving scripts	Folder in which a working copy of all scripts is saved, if user-defined script storage location is activated.
User-defined script storage location	The scripts are stored at a different location as the experiment or project. This can be helpful when using subversion tools (SVN). Stores a copy of all scripts in an arbitrary user-defined location. All scripts can be found in the given folder.

Script Editor options

Option	Description
Advanced code completion	If the advanced code completion is enabled, internal interfaces will be shown in the code completion of the editor. These interfaces are only needed for special functions and should only be used by advanced users. When enabling/disabling this option, the editor must be reopened.

Restore options

Option	Description
Default dialog response for backup	The scripts will be stored outside of imc STUDIO. The current state of the scripts is only applied when the associated storage scope is saved. If any changes where made to the script in the Script Editor after the last saving of the storage scope, this option can be used to automatically restore scripts to the last saved state of the script. • **Always ask (show dialog):** A *dialog* appears with the options to restore the last saved state. • **Always restore (use last script saving state):** Changes are always restored. The last saved state from the *Script Editor* is used. • **Never restore (use latest backup in the respective storage scope):** Changes are never restored. The last saved state of the scripts from the *Experiment* or *Project* is used. For more information, see chapter Restore script changes 1876.

17.1.6 Tutorial

Here you will find a list of tutorials:

Tutorial	Description
Device and channel settings 1908	A device is added to the device list. After selection of the device, the analog channels are configured.
Channel calculation 1912	Different methods to calculate channels are shown.
List linked Variables 1914	Shows a list of Widgets and their connection to variables.
RS232-Interface 1915	A example how to control a serial device.
Using the script clipboard 1916	The script clipboard is explained in a little example.

17.1.6.1 Device and channel settings

Channel settings

In this tutorial, a device is to added to the device list on the basis of its serial number (here: 123456) and then selected. Furthermore, all analog channels are to be set as follows:

Parameter	Value
State	active
Save to HDD	on
Sample rate	1000 Hz
Duration:	10 s
Measurement mode	Voltage
Coupling	DC
Input Range	±5 V

The following **libraries** are needed:

```
using imc.Common.Interfaces.Logbook;
using imc.Studio.Interfaces.DevSetup;
using imc.Studio.Tools.DevSetupGraphGenerator;
```

Once imc STUDIO has been started, a scripted named *DeviceSettings* is created by means of the Sequencer-page.

In the first step, the data above are recorded.

```
string errorSender  = "DeviceSettings Script"; // Sender for Logbook
string serialNumber = "123456";

double sampleTime = 0.001; // 1 ms = 1000 Hz
double duration   = 10.0; //  10 s
double range      = 5;  //  +/- 5 V
string rangeText    = "\u00b1" +  range.ToString() + " V"; // +/- Symbol and Unit
```

If the device still does not appear in the Devices list, a device search is performed. Subsequently, the device is selected, if it is present.

```
// Get Device via serial number
EnumItem device = DevSetup.Devices.GetItemBySerialNumber(serialNumber);

// if device is not in list, do device search
if (device == null) {
   DevSetup.SearchDevices();
   device = DevSetup.Devices.GetItemBySerialNumber(serialNumber);

   // if device is not still in the list -> Warning
   if (device == null) {
      Logbook.LogEntry(errorSender, "Device with S/N " + serialNumber + " could not be found.", 0,
ELogbookEntryCategory.Warning);
      return; // end script
   } // if
} // if

// get device name
string nickName = device.Parameters[EClassID.eDeviceNickname].GetValue().ToString();
string name = device.Parameters[EClassID.eDeviceName].GetValue().ToString();

// select device for measurement
DevSetup.SelectDevice(nickName,true, EDeviceCoreType_V2.Measurement);
```

For all analog channels, the above **parameters** are to be set as follows:

```
foreach( var channel in DevSetup.Channels){
   var paraType = channel.Parameters[EClassID.eChannelType];    //
   if (paraType != null){
      int chnType = (int)paraType.GetValue();
      // for analog channels
      if ( chnType == ParameterValues.eChannelType.AnalogInput) {

         // ... next block

      } // chnType
   } // paraType
} // foreach

// Process
Action.FireAction(this, "acCheckConfiguration");
```

Setting of the **parameters** is performed as follows:

```
var paraStatus      = channel.Parameters[EClassID.eStatus];
var paraSaveHD      = channel.Parameters[EClassID.ePCTransferToHD];
var paraSampleTime  = channel.Parameters[EClassID.eSampleTime];
var paraDuration    = channel.Parameters[EClassID.eDuration];
var paraRange       = channel.Parameters[EClassID.eRange];
var paraMode        = channel.Parameters[EClassID.eChannelMode];
var paraCoupling    = channel.Parameters[EClassID.eCoupling];
```

```
// set state to active
if (paraStatus != null){
   paraStatus.SetValue(ParameterValues.eStatus.Active);
} // paraStatus
```

```
// enable saving to PC
if (paraSaveHD != null){
   paraSaveHD.SetValue(true);
} // paraSaveHD
```

```
// set sample rate
if (paraSampleTime != null){
   paraSampleTime.SetValue(sampleTime);
} // paraSampleTime
```

```
// set duration
if (paraDuration != null){
   paraDuration.SetValue(duration);
} // paraDuration
```

```
// set mode to "Voltage"
if (paraMode != null){
   paraMode.SetValue(ParameterValues.eChannelMode.DAQ_Voltage);
} // paraMode
```

```
// set coupling to DC
if (paraCoupling != null){
   paraCoupling.SetValue(ParameterValues.eCoupling.DC);
} // para Coupling
```

```
// set Range
if (paraRange != null){
   paraRange.SetValue(rangeText);
} // paraRange
```

After the script has been successfully complied, the Editor can be closed. Subsequently, the script can be run by means of a command, for example.

If the device does not yet appear in the Devices list, a device search is performed and the device is added to the list. The channel settings are performed and can be reviewed on the page *Analog Channels*. If the device is not found at all, a warning is recorded in the Logbook.

Extension: Input via Widgets

Supposing we want to expand on the example above. The goal is to be able to enter the sampling interval, the input range, and the measurement duration by means of Widgets.

First, to represent these parameters, create the user-defined variables *SampleTime*, *Range* and *Duration* in the Data-Browser. These variables are dragged from the Data Browser to a Panel page and displayed as a numerical display widget. It is also helpful the display the Setup-page *Analog channels* (Widget: Device Control Setup).

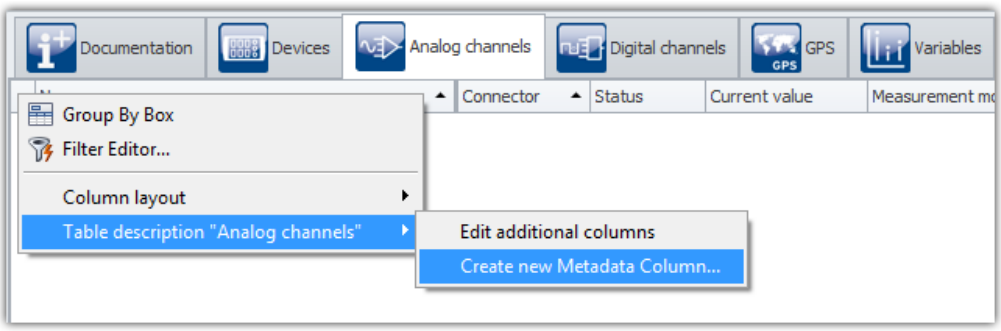
The value of each of these variables is queried in the script and assigned to the corresponding variable.

```
string errorSender  = "DeviceSettings Script"; // Sender for logbook
string serialNumber = "123456";

double sampleTime = Variables["SampleTime"].Value.Get<double>(); //
double duration   = Variables["Duration"].Value.Get<double>(); //
double range      = Variables["Range"].Value.Get<double>();  //
string rangeAsText    = "\u00b1" + range.ToString() + " V"; // +/- Symbol and Unit
```

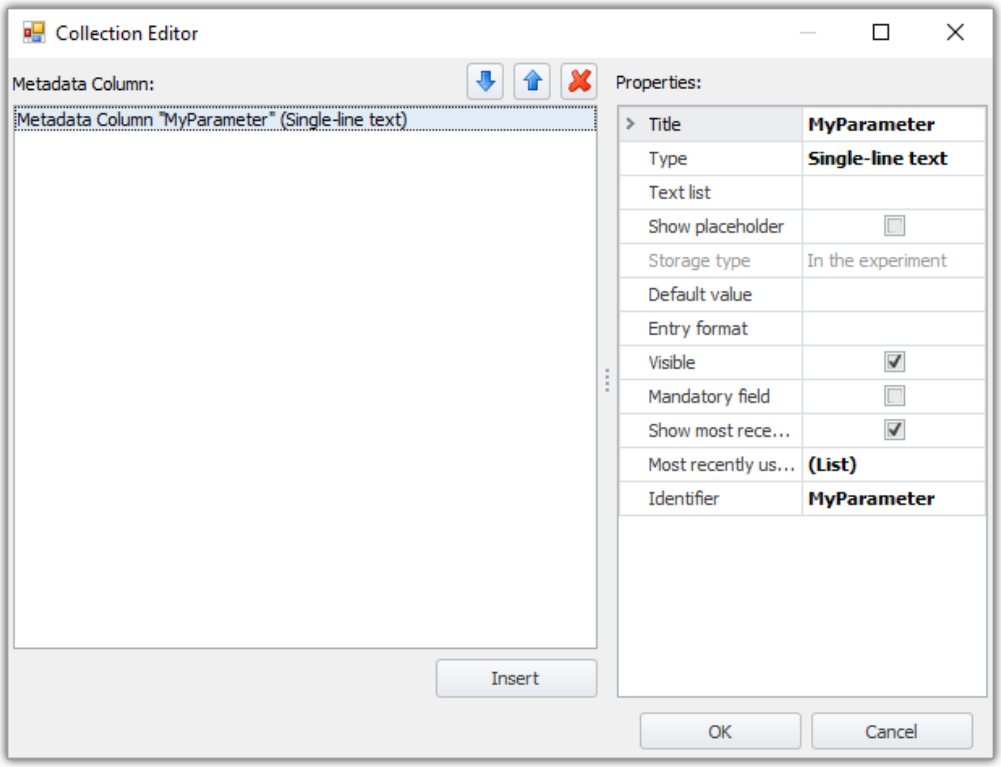
After the script has been compiled, it is possible to set the parameters by means of the widgets while the Design-mode is deactivated. A button which causes the script to run when pressed (command: *Run Script*) is added to the Panel page. When the button is activated, the values are updated to the *Analog channels* page.

Extension: Metadata columns

On the Setup-page *Analog channels*, create your own metadata column 251 *MyParameter by* opening the context menu of **the upper left corner** and clicking on *Table description "Analog channels" > Create new Metadata Column...* **.**



Analog Channels: Add Metadata column



Collection Editor: Metadata column "MyParameter"

This column is to be filled by the script with progressively higher numbers. Toward this end, a counting variable is generated. Save this view.

```
int count = 1;
foreach( var channel in DevSetup.Channels){
  // ...
} // foreach
```

The parameter setting process is supplemented with the following code:

```
var paraCustom = channel.Parameters["MyParameter"]; // Identifier
// ...
if (paraCustom != null){
   paraCustom.SetValue("#" + count.ToString());
   count++;
}
```

After running the script, the newly added table column of the analog channels fills itself.

17.1.6.2 Channel calculation

Manual calculation

In this tutorial, analog channels are calculated. The following settings are to be made for two analog channels:

Parameter	Value
State	active
Save to HDD	on
Sample rate	1000 Hz
Duration:	10 s
Measurement mode	Voltage
Coupling	DC
Input Range	±5 V

The script should be started after a measurement and add the current channels and write them to an output channel.

The following **libraries** are needed:

```
using imc Common Components DataManager Interfaces
using imc Data Contract
using imc Data Core Contract
using imc Data Core Contract Identifiers
using imc Data Storage Core Contract
using imc Studio Tools DevSetupGraphGenerator
```

Within the Run() method the sequence is implemented. Here, the variables are queried, the measurement is loaded, the channels from the variables are calculated and returned in a result channel:

```
public override void Run()
{
   const double sampleTime = 0.001; // better get from channel
   var measurementName = GetLastMeasurementName();

   var measurementVariable = GetVariable(measurementName);

   var channelVariable1 = GetChannelVariable(measurementVariable, "Cosine");
   var channelVariable2 = GetChannelVariable(measurementVariable, "DC");

   LoadMeasurement(measurementName);

   var channel1 = GetChannel(channelVariable1);
   var channel2 = GetChannel(channelVariable2);

   var resultChannelVariable = CreateChannelVariable("ResultChannel", sampleTime);

   var sumResult = AddChannels(channel1, channel2);

   WriteDataToChannelVariable(resultChannelVariable, sumResult);
}
```

The following functions are used:

```
private T EvaluateResult<T>(Result<T> result)
{
   if (result.IsSuccessful)
   {
      return result.Value;
   }
   else
   {
      // ErrorHandling
      throw result.Error;
   }
}

public string GetLastMeasurementName()
{
   var measurementManagerService = Variables.Services.Get<IMeasurementManagerService>();
   var result = measurementManagerService.FindLastMeasurement();

   if (result.IsSuccessful)
   {
      return result.Value.DisplayName.ToString();
   }

   throw result.Error;
}

public IVariable GetVariable(string variableName)
{
   var result = Variables.TryGet(new DisplayName(variableName));

   return EvaluateResult(result);
}

public double[] AddChannels(IDmChannel channel1, IDmChannel channel2)
{
   var data1 = channel1.Data;
   var data2 = channel2.Data;
   var result = new double[data1.Length];

   for (int i = 0; i < result.Length; i++)
   {
      result[i] = data1[i] + data2[i] + 42;
   }

   return result;
   // return data1.Zip(data2, (x, y) => x + y).ToArray();
}

public void WriteDataToChannelVariable(IVariable channelVariable, double[] data)
{
   channelVariable.Value.Set(data);
}

public void LoadMeasurement(string measurementName)
{
   var measurementManagerService = Variables.Services.Get<IMeasurementManagerService>();
   Result<IMeasurementService> measurementResult = measurementManagerService.TryGet(measurementName);

   if (measurementResult.IsSuccessful)
   {
      var measurement = measurementResult.Value;
      measurement.Load().ThrowExceptionIfAny();
   }
}

public IVariable CreateChannelVariable(string name, double sampleTime)
{
   var result = Variables.TryGet(new DisplayName(name));

   if (result.IsSuccessful)
   {
      Variables.Delete(new DisplayName(name));
      // return result.Value;
   }
```

The following functions are used:

```
    var variable = Variables.Create(new DisplayName(name));
    variable.Properties.Set(EClassID.eDataType, DevSetupCommonValues.eDataType.IEEEDOUBLE);
    variable.Properties.Set(EClassID.eXFormat, DevSetupCommonValues.eXFormat.Normal);
    variable.Properties.Set(EClassID.eObjectType, typeof(IDmChannel));
    variable.Properties.Set(EClassID.eSampleTime, sampleTime);

    return variable;
}

public IDmChannel GetChannel(IVariable channelVariable)
{
    var result = channelVariable.QueryInterface<IDmChannel>(); // measurement needs to be loaded,
otherwise this interface is not supported
    return EvaluateResult(result);
}

public IVariable GetChannelVariable(IVariable measurementVariable, string variableName)
{
    var result = measurementVariable.Children.TryGet(new DisplayName(variableName));

    return EvaluateResult(result);
}
```

17.1.6.3 List linked variables

In this tutorial all Widgets on all Panel pages should be listed including the variable link.

The following **Libraries** are used:

```
using System.Linq;
using System.Text;
using System.Windows.Forms;
```

In the `Run()` method you need to add the following lines.

```
StringBuilder text = new StringBuilder();

// for special Widgets
Panel.Pages.ToList()
    .ForEach(page => page.Widgets.ToList()
              .Where(widget => widget.Variable != null)
              .ToList()
              .ForEach(widget => text.Append(page.DisplayedCaption + " | " + widget.Name + " | " +
widget.Variable.Name + "\n")));

// for Standard-Widgets
Panel.Pages.ToList()
    .ForEach(page => page.Widgets.ToList()
              .Where(widget => widget.Variable == null)
            .ToList()
              .ForEach(widget => text.Append(page.DisplayedCaption + " | " + widget.Name + " | " +
widget.StandardVariable.Name + "\n")));

MessageBox.Show(text.ToString());
```

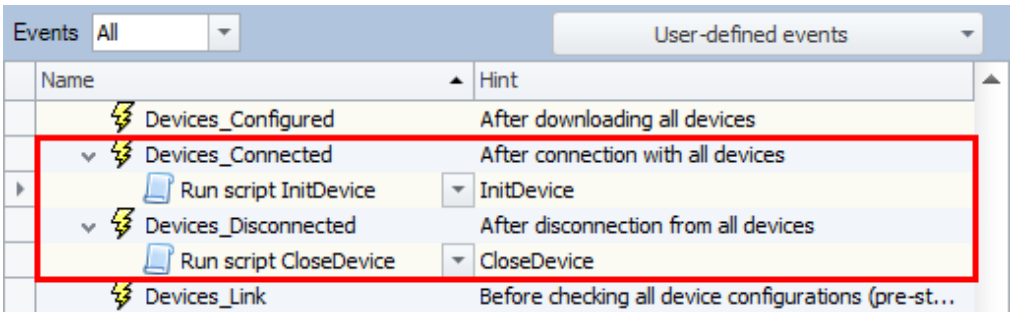
In the message box a list of all Widgets on all Panel pages and their variable connection are shown.

You can also write the results to a file.

17.1.6.4 RS232-interface

The following **libraries** are needed:

```
using System.IO.Ports;
```



Events

Initialisation / Connect

Add the command *Run script* with the following code to the event *Devices_connected*. The event is released, when connected to devices.

InitDevice.cs

```
try{
   // Create a COM port object and define the port's settings
   SerialPort serialPort = new SerialPort("COM1", 9600, Parity.None, 8, StopBits.One );
   // Open the port
   serialPort.Open();
   // Store the COM port in the clipboard
   ScriptClipboard.Add("COM1", serialPort);
}catch(Exception x){
   // error handling
}
```

Close / Disconnect

Add the command *Run script* with the following code to the event *Devices_disconnected*. The event is released, when disconnected from devices.

CloseDevice.cs

```
try{
   // Get the COM port from clipboard
   SerialPort serialPort = ScriptClipboard["COM1"] as SerialPort;
   // Close the COM port
   serialPort.Close();
   // Remove the COM port from clipboard
   ScriptClipboard.Remove("COM1");
}catch(Exception x){
   // error handling
}
```

Read data

ReadData.cs

```
try{
   // Get the COM port from clipboard
   SerialPort serialPort = ScriptClipboard["COM1"] as SerialPort;
   // Read a message from the device
   string dataAsText = serialPort.ReadLine();
}catch(Exception x){
   // error handling
}
```

Write data

WriteData.cs

```
try{
   // Get the COM port from clipboard
   SerialPort serialPort = ScriptClipboard["COM1"] as SerialPort;
   // Send a message to the device
   serialPort.WriteLine("*IDN?");
}catch(Exception x){
   // error handling
}
```

17.1.6.5 Using the script clipboard

In this tutorial, the script clipboard is to be used to make a Widget belonging to a Panel-page available to a different script.

ScriptA.cs

When adding an object into the script clipboard, it is necessary to be aware that the object has a namespace. In the case of a Widget, this would be the script's name due to the proxy-classes.

For this reason, the script must be created with the option "*Separate Proxy class library*". Then the proxy-class will be integrated into the script as a resource.

```
Widget myWidget = Panel["Page 1"]["MyWidget"]; // Type: ScriptA.Widget
ScriptClipboard.Add("MyWidget", myWidget);
```

ScriptB.cs

The script can be created without the option "Separate proxy class library".

Under *References > Add Reference,* the **ProxyClasses.dll** is added from the **ScriptA-Project**. The file is located in the **ScriptA-project's bin**-folder.

```
object o;
bool b = ScriptClipboard.TryGetValue("MyWidget", out o);

if ( (b) && (o != null) ){
   ScriptA.Widget myWidget = o as ScriptA.Widget;
}
```

17.1.7 Important information

Warning Important changes when moving from a imc STUDIO 5.x version

When switching from a 5.x version, your scripts will be converted as far as possible.

Some adjustments (among other things the variable accesses) must nevertheless be made manually.

- Change of variable accesses from "Datapool" to "Variables
- imc.Studio.Interfaces.V2 has been integrated into imc.Studio.Interfaces and is therefore obsolete.

Warnings CS1607 and MSB3270

If CS1607 and MSB3270 warnings appear when creating and building scripts, it may help to reinstall the SDK .NET Framework 4.8.

(older) Experiment import: System.IO.FileNotFoundException

When importing experiment from former versions, there might be error messages, when running the scripts.

Esp. when the scripts were imported as DLL (assembly). In this case the script cannot be compiled automatically.

If the source code is available and you have a Developer-Edition, the error can be fixed by compiling the code again and may correct the references. You might need to Regenerate proxies 1819.

In the case of a DLL-script the developer of the script needs to compile the script in the corresponding version. Please contact the developer of the script.

Close Script Editor before "Experiment > Save as"

Note

When executing *Experiment > Save As..*, the **Script editor** must be closed, otherwise it will still have the script belonging to the original experiment open, and will not automatically alter the path.

imc STUDIO GUI can not be operated

Note

While the script's `Run()` method is in progress, the imc STUDIO user interface can not be operated. For this reason, the script may not contain any blocking functions.

Renaming elements

Note

If devices, channels or Widgets which are accessed by imc STUDIO Scripting are renamed or deleted in imc STUDIO, then the script must be adapted accordingly.

Do not detach while debugging

Warning

Do not *detach* from the process while debugging. Depending on the user's programming, imc STUDIO might be in an undefined or non-operable state. Do always use *Continue Debugging* (blue "Play" symbol) to end debugging before detaching.

Event handling is asynchronous

Note

The event handling methods are run **asynchronously**.

17.2 Third Party Device Interface

Documentation on the **imc STUDIO** plug-in **imc STUDIO Third Party Device Interface**.

17.2.1 Getting started

17.2.1.1 Introduction

Note

- A **imc STUDIO ThirdPartyDevice License** (inclusive/exclusive) is needed to run a Third Party Device script.[1]
- A imc STUDIO **Developer**-Edition is needed to edit a Third Party Device script.
- The 3rd Party Device Management a available in all editions except Runtime.

[1]The Third Party Device *AudioDevice, ChannelLoader and FunctionSimulator* can be used without any 3PDI license.

With the plugin **imc STUDIO Third Party Device Interface** devices from other manufacturers (Third Party Devices) can be added to the Setup page in imc STUDIO easily.

Therefore a C#-Script has to be implemented containing properties of the Third Party Device. A template code and fundamental functions are given by imc STUDIO. In this script the interface to the Third Party Device has to be implemented. After finishing the script, the Third Party Device will appear in the device list on the Setup page. The Third Party Device can now be configured in imc STUDIO and can also be combined with devices manufactured by imc. For example an oscilloscope or a multimeter can be added.

The advantages of Third Party Device Script are:

- consistent measurement format
- synchronous visualization of the measurement data
- common data saving
- common project management
- common management of the device configuration
- no GUI Programming needed, integration of the device in imc STUDIO
- existing device and channel parameters can be used for the Third Party Device
- new parameters can be created

Needs:

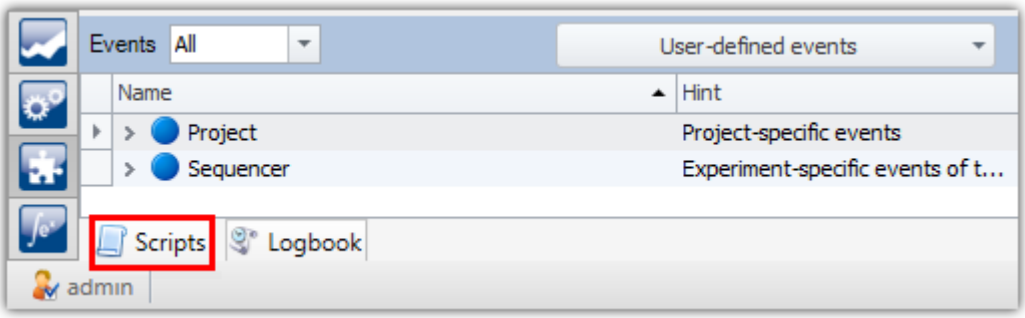
- Experience in C# programming is necessary (C++ for an *unmanaged* data transfer)
- The interface of the Third Party Device has to be known.

17.2.1.2 Add a new script

Reference

For more information about **Scripting**, see here 1812.

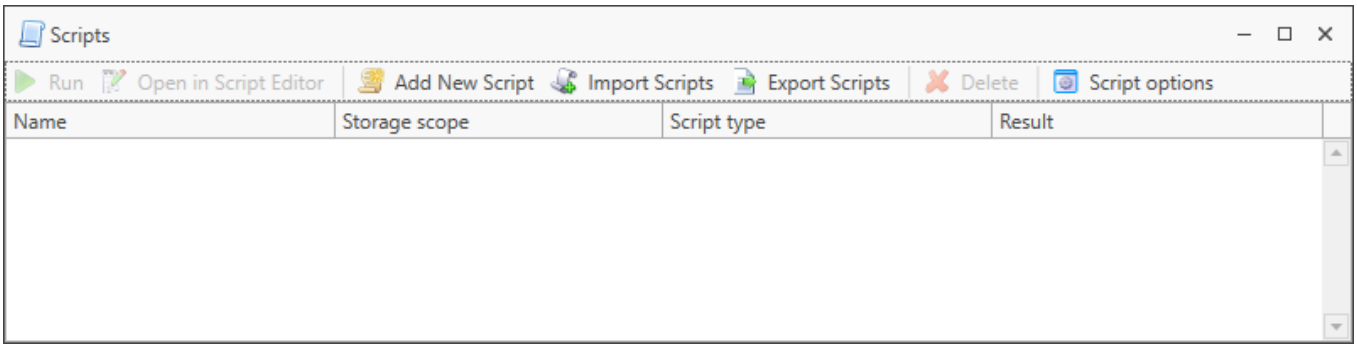
Scripts are created on the Sequencer page by clicking on the tab *Scripts* at the bottom of the page.



Sequencer: Tool window "Scripts"

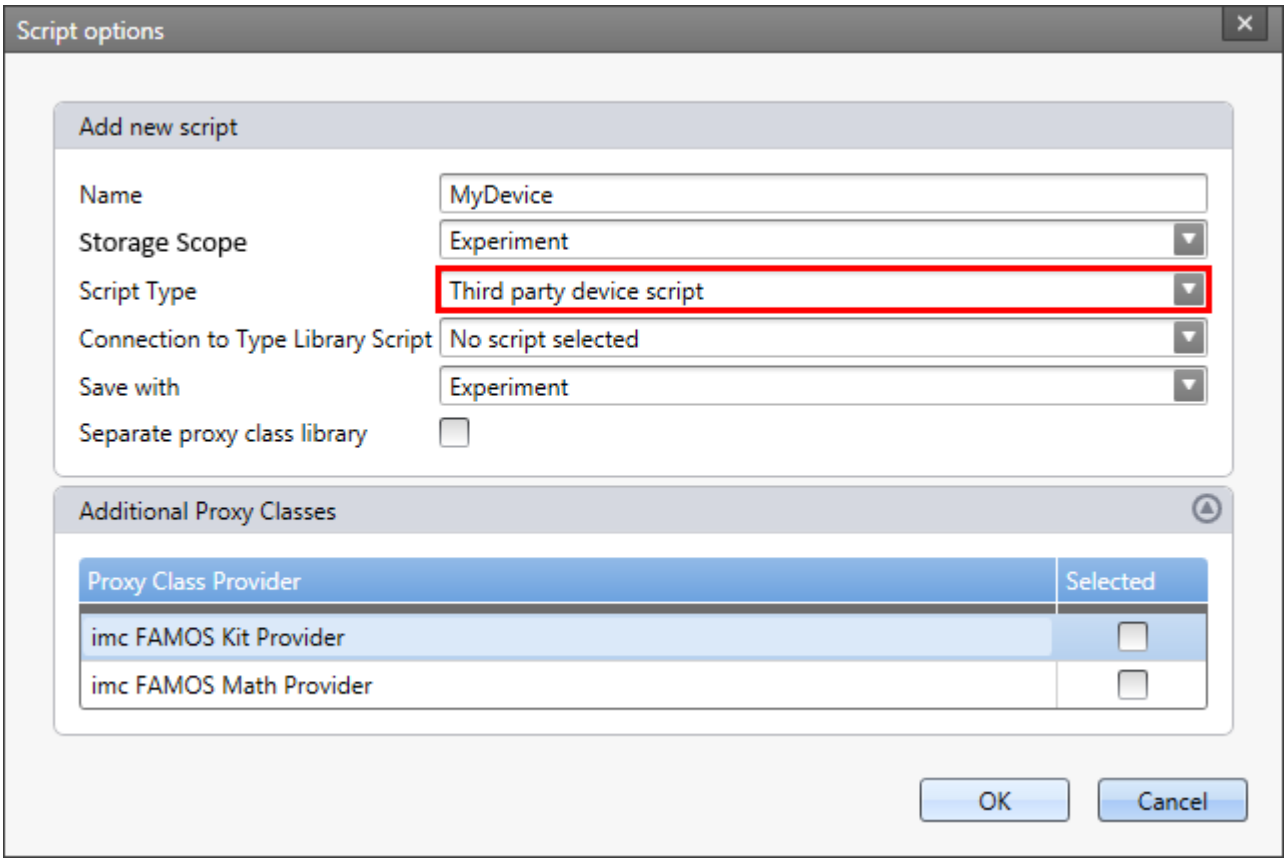
The tool window *Scripts* will open. Click on *Add new script*

to add a new Script



Sequencer: Tool window "Scripts"

The Script Type has to be chosen: Third party device script.



The editor will open automatically after adding a script containing a template of C#-Code.

Note

To use another template, please use the [3rd Party Device Management](1929).

17.2.2 Serial interface

17.2.2.1 Device control

If an integration to the setup page is not needed, and data should be *polled* from the device by an event or clicking on a button, a normal script can be used.

For example a climate cabinet controlled by RS232 which sends single values of the temperature on command.

Here scripts for opening and closing the connection and for reading and writing data are described.

These code snippets can be bound to the Sequencer events, e.g. "Device connected/disconnected".

Open COM Port/ connect to the device

```
try{
   // Create a COM port object and define the port's settings
   System.IO.Ports.SerialPort serialPort = new System.IO.Ports.SerialPort("COM1", 9600, Parity.None, 8,
StopBits.One);
   // Open the port
   serialPort.Open();
   // Store the COM port in the clipboard
   ScriptClipboard.Add("COM1", serialPort);
}catch(Exception x){
   // error handling
}
```

Close COM Port / disconnect from device

```
try{
   // Get the COM port from clipboard
   System.IO.Ports.SerialPort serialPort = ScriptClipboard["COM1"] as System.IO.Ports.SerialPort;
   // Close the COM port
   serialPort.Close();
   // Remove the COM port from clipboard
   ScriptClipboard.Remove("COM1");
}catch(Exception x){
   // error handling
}
```

Write data

```
try{
   // Get the COM port from clipboard
   SerialPort serialPort = ScriptClipboard["COM1"] as SerialPort;
   // Send a message to the device
   serialPort.WriteLine("*IDN?");
}catch(Exception x){
   // error handling
}
```

Read data

```
try{
   // Get the COM port from clipboard
   SerialPort serialPort = ScriptClipboard["COM1"] as SerialPort;
   // Read a message from the device
   string strData = serialPort.ReadLine();
}catch(Exception x){
   // error handling
}
```

17.2.3 Third party device script

17.2.3.1 Methods

void GetAvailableDevices(SetupNodeList deviceList)

Description:	Gets all the currently available devices which could be selected for measurement later on.
Is called when:	search for devcies is performed.
To implement:	Search for the 3rd-party devices, as well as for device parameters of the devices found

void GetDeviceDescription(SetupNode device)

Description:	Gets the complete definition of the device with all modules and channels.
Is called when:	the device is selected for a measurement.
To implement:	Additional parameters for the selected devices and add channels and their properties.

void DeselectDevice(SetupNode device)

Description:	Is called when the device is deselected by the user. Here it is possible to do some cleanup.
Is called when:	the device is deselected from a measurement.
To implement:	Take all actions necessary for de-selecting the device.

void PrepareMeasurement(SetupNode device)

Description:	Prepares the specified device for measurement, downloads the configuration from the PC to the device. Can throw an exception if an error occurred.
Is called when:	Performing a *Process* action.
To implement:	Actions necessary when performing a Process action. E.g. processing user inputs.

void ProcessConfiguration(SetupNode device)

Description:	Processes the device configuration and checks if the the configurations have errors. If an error is detected throws an exception.
Is called when:	Performing a *Prepare* action.
To implement:	Actions necessary when performing a Prepare action. E.g. processing user inputs.

void ConnectDevice(SetupNode device)

Description:	If necessary establishes the connection to the specified device. If the connection can not be established an exception is thrown.
Is called when:	the user connects to the devices.
To implement:	stablishment of a connection to the device.

void DisconnectDevice(SetupNode device)

Description:	If the implementation of the method ConnectDevice() established the connection, closes the connection to the specified device. If there was an error an exception is thrown.
Is called when:	the user disconnects from the device.
To implement:	Disconnection from the device.

void StartMeasurement(SetupNode device)

Description:	Starts the measurement for the specified device. If there was an error an exception is thrown.
Is called when:	Measurement is started.
To implement:	Start the device. Access to the device driver can also be done in the *DataTransferDriver*.

void StopMeasurement(SetupNode device)

Description:	Stops the measurement for the specified device.
Is called when:	Measurement is stopped.
To implement:	Stop the device. Access to the device driver can also be done in the *DataTransferDriver*.

void GetParameterConstraints(SetupNode node, SetupParameter parameter, ParameterConstraint constraint)

Description:	Gets the constraints for a parameter. The constraints can be defined with calls to the constraint instance. Constraints can be upper and/or lower value borders for numeric parameters, value lists for combo boxes, enabled or disabled states according to other parameters.
Is called when:	the device and channel configuration is loaded.
To implement:	Selection options for parameters, e.g. the sampling interval.

Please use the correct numeric format as parameter, e.g.:

```
if (parameter.Name == EClassID.eSampleTime.ToString()) {
   constraint.ValueList.Add(0.1);
   constraint.ValueList.Add(1d); // for double
// constraint.ValueList.Add(1); // for double -> error
// ...
```

void ExecuteParameter(SetupNode node, int actionStepID)

Description:	Is executed if a non standard action should be executed. If there was an error an exception is thrown.
Is called when:	Device action (e.g. balancing, eAction, ...) is invoked.
To implement:	Regarding to the ActionID further actions can be performed.

string GetParameterDisplayText(SetupNode node, SetupParameter parameter, object value, bool reciprocal, bool useRoundTripFormat)

Description:	Gets the display text of the parameter's value.
Is called when:	the device and channel configuration is loaded.
To implement:	Display texts for (numerical) parameter presets.

```
if (parameter.Name == EClassID.eAction.ToString()) {
   if (object.Equals(value, 1)) {
      return "Some action";
// ...
```

object ConvertParameterTextToValue(SetupNode node, SetupParameter parameter, string inputText, bool reciprocal)

Description:	Converts the input text of a parameter to the value type of the parameter. If the input text contains an error, throws an exception.
Is called when:	the user enters a value to a parameter.
To implement:	Conversion of inputs to parameter values.

bool? ValidateParameter(SetupNode node, SetupParameter parameter, object oldValue)

Description:	Validates the parameter. Here it is possible to change a parameter value according to the values of other parameters or to prohibit the change.
Is called when:	the user enters a value to a parameter.
To implement:	Parameter validation with respect to certain boundaries, such as limiting the measurement time to a maximum value.

void LoadDeviceConfiguration(SetupNode device, string configurationVersion)

Description:	Informs the device provider about a configuration that was loaded by the device Setup. Here it is possible to convert the device configuration to the current version, i.e. by adding new parameters.

17.2.3.2 Add a device

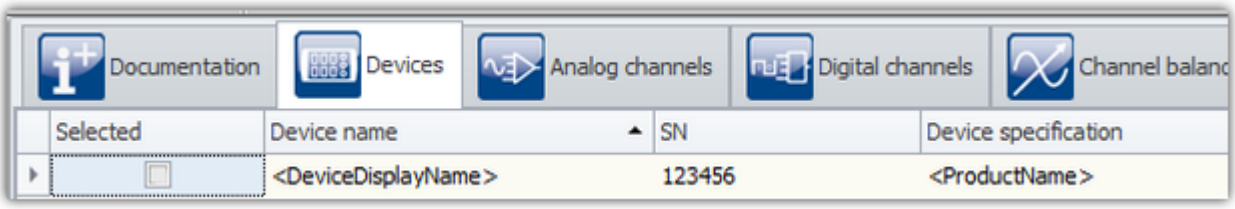
First of all the device (or the devices) has to be configured. In the method `GetAvailableDevices()` the properties of the devices is programmed.

In this example a device is added with the serial number "123456". Further device parameter can be added, too. Therefore the according **EClassID** parameters have to be added.

GetAvailableDevices

```
protected override void GetAvailableDevices(SetupNodeList deviceList) {
// Add the currently available devices here! Example:
   var device = deviceList.GetOrAdd(ESetupNodeType.Device, "<DeviceName>");
   device.Parameters.UpdateOrAdd(EClassID.eDeviceNickname, value: "<DeviceDisplayName>");
   device.Parameters.UpdateOrAdd(EClassID.eDeviceSN, value: "123456");
   device.Parameters.UpdateOrAdd(EClassID.eDeviceProduct, value: "<ProductName>");
}
```

After closing the editor, the Third Party Device is shown on the Setup in the device list.



Setup: Device list

If the device is selected, the device configuration (channels and variables) is loaded. The next step is to add channels and variables 1924 to the device.

17.2.3.3 Add a channel

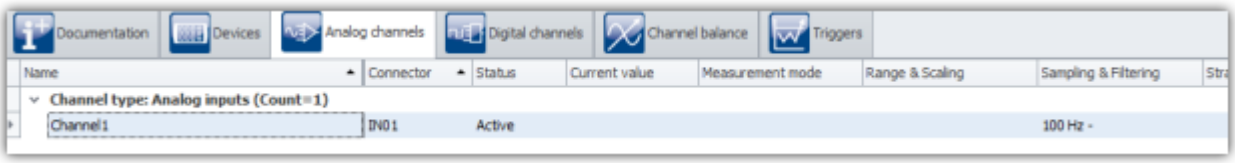
The channel configuration of the parameters is set in the following procedure. This method is run when the device is selected. In this example the configuration is made for only one channel. To configure more channels 1943, use a loop or create a method. Here, the channel type is set to analog input, but also Display variables or other types are possible. For all possibilities see `ParameterValues.eStatus`.

GetDeviceDescription

```
protected override void GetDeviceDescription(SetupNode device) {
   // Add the full device description here! Channels, modules... Example:
   var channel = device.Children.GetOrAdd(ESetupNodeType.Channel, "<DeviceName>.Channel1");
   channel.Parameters.UpdateOrAdd(EClassID.eChannelName, value: "Channel1");
   channel.Parameters.UpdateOrAdd(EClassID.ePlugInName, value: "IN01");
   channel.Parameters.UpdateOrAdd(EClassID.eStatus, value: ParameterValues.eStatus.Active);
   channel.Parameters.UpdateOrAdd(EClassID.eChannelType, value:
ParameterValues.eChannelType.AnalogInput);
       channel.Parameters.UpdateOrAdd(EClassID.eDivisionUnit, value: "V"); // The original (electrical)
measurement unit
       channel.Parameters.UpdateOrAdd(EClassID.eUserUnit, value: "V");
       channel.Parameters.UpdateOrAdd(EClassID.eDataType, value: ParameterValues.eDataType.IEEEFLOAT);
       channel.Parameters.UpdateOrAdd(EClassID.eSampleTime, value: 0.01); // 100 Hz - set also the
selectable values in the method
}
```

```
// ParameterValues.cs
public sealed class eChannelType {
    public const int AnalogInput = 1;
    public const int DAC = 2;
    public const int DIOPort = 3;
    public const int VirtChannel = 5;
    public const int DIOBit = 6;
    public const int VirtBit = 7;
    public const int NetBit = 8;
    public const int EncoderInput = 9;
    // ...
    public const int DisplayVar = 15;
    // ...
    public const int PVVar = 24;
    public const int VideoChannel = 1001;
    // ...
    public const int DIO_In = 4099;
    public const int DIO_Out = 8195;
}
```

After selecting the device, the channel is available in the analog channel list.



Setup: analog channels

To add a Display Variable 1945 to the device, the channel type has to be set to "DisplayVar".

You have to `Remove` some parameters for Display variables and pv-variables:

```
variable.Parameters.Remove(EClassID.eSampleTime);
variable.Parameters.Remove(EClassID.eDuration);
variable.Parameters.Remove(EClassID.ePCTransferToHD);
```

17.2.3.3.1 Single Value

Single value as text:

```
SetupNode singleValueText = device.Children.GetOrAdd(ESetupNodeType.Channel, "SingleVarString");
singleValueText.Parameters.UpdateOrAdd(EClassID.eChannelName, value: "SingleVariableString");
singleValueText.Parameters.UpdateOrAdd(EClassID.ePlugInName, value: "IN01");
singleValueText.Parameters.UpdateOrAdd(EClassID.eStatus, value: ParameterValues.eStatus.Active);
singleValueText.Parameters.UpdateOrAdd(EClassID.eXFormat, value: ParameterValues.eXFormat.SingleValue);
singleValueText.Parameters.UpdateOrAdd(EClassID.eChannelType, value:
ParameterValues.eChannelType.DisplayVar);
singleValueText.Parameters.UpdateOrAdd(EClassID.eObjectType, value: typeof(string).ToString());
singleValueText.Parameters.UpdateOrAdd(EClassID.eDataType, value: ParameterValues.eDataType.ASCII);

singleValueText.Parameters.Remove(EClassID.eSampleTime);
singleValueText.Parameters.Remove(EClassID.eDuration);
singleValueText.Parameters.Remove(EClassID.ePCTransferToHD);
```

Single value as number/Integer

```
SetupNode singleValueInt = device.Children.GetOrAdd(ESetupNodeType.Channel, "SingleVarInteger");
singleValueInt.Parameters.UpdateOrAdd(EClassID.eChannelName, value: "SingleVariableInteger");
singleValueInt.Parameters.UpdateOrAdd(EClassID.ePlugInName, value: "IN02");
singleValueInt.Parameters.UpdateOrAdd(EClassID.eStatus, value: ParameterValues.eStatus.Active);
singleValueInt.Parameters.UpdateOrAdd(EClassID.eXFormat, value: ParameterValues.eXFormat.SingleValue);
singleValueInt.Parameters.UpdateOrAdd(EClassID.eChannelType, value:
ParameterValues.eChannelType.DisplayVar);
singleValueInt.Parameters.UpdateOrAdd(EClassID.eObjectType, value: typeof(int).ToString());

singleValueText.Parameters.Remove(EClassID.eSampleTime);
singleValueText.Parameters.Remove(EClassID.eDuration);
singleValueText.Parameters.Remove(EClassID.ePCTransferToHD);
```

17.2.3.3.2 Ports and Bits

```
const int numberOfBitsPerPort = 8;

var port = device.Children.GetOrAdd(ESetupNodeType.Channel, "Poll_DIn_Port");
port.Parameters.UpdateOrAdd(EClassID.eChannelName, value: "Poll_DIn_Port");
port.Parameters.UpdateOrAdd(EClassID.ePlugInName, value: "DIN");
port.Parameters.UpdateOrAdd(EClassID.eChannelType, value: ParameterValues.eChannelType.DIOPort); // or
DIO_In, DIO_Out
// ... other Parameters

for (int k = 0; k < numberOfBitsPerPort; k++)
{
     string name = k.ToString("D1");
     var bit = port.Children.GetOrAdd(ESetupNodeType.Channel, "Poll_DIn_Bit" + name);
     bit.Parameters.UpdateOrAdd(EClassID.eChannelName, value: "Poll_DIn_Bit0" + name);
     bit.Parameters.UpdateOrAdd(EClassID.ePlugInName, value: "DIN_Bit0" + name);
     bit.Parameters.UpdateOrAdd(EClassID.eChannelType, value: ParameterValues.eChannelType.DIOBit);
     // ... other Parameters
}
```

17.2.3.3.3 FFT

In order to create a FFT-channel, the `eXFormat` must be set to FFT. Additionally, various segment parameters must be set:

```
var channel = device.Children.GetOrAdd(ESetupNodeType.Channel, "<DeviceName>.Channel1");
channel.Parameters.UpdateOrAdd(EClassID.eXFormat, value: ParameterValues.eXFormat.FFT);
channel.Parameters.UpdateOrAdd(EClassID.eDataType, value: ParameterValues.eDataType.IEEEFLOAT);
channel.Parameters.UpdateOrAdd(EClassID.eSamplesPerSegment, value: 512);
channel.Parameters.UpdateOrAdd(EClassID.eSegmentOffset, value: 0);
channel.Parameters.UpdateOrAdd(EClassID.eSegmentDelta, value: 5);
channel.Parameters.UpdateOrAdd(EClassID.eSegmentUnit, value: "Hz");
channel.Parameters.UpdateOrAdd(EClassID.eSampleTime, value: 0.1, editorType: EEditorType.ReadOnlyEdit);
```

The section above belongs to the source text in the collapsed region "Example for FFT channel parameters (streaming vector / segmented channel)"-

17.2.3.3.4 Time stamped

The use of time-stamped channels is explained using the *SimplePollDevice* template.

The example code is not generally valid and must be adapted according to the third-party device.

GetDeviceDescription

For a time-stamped channel, in the `GetDeviceDescription` method the channel parameter `eXFormat` is set to "Timestamp".

```
var channel = device.Children.GetOrAdd(ESetupNodeType.Channel, "Channel1");
channel.Parameters.UpdateOrAdd(EClassID.eXFormat, value: ParameterValues.eXFormat.Timestamp);
channel.Parameters.UpdateOrAdd(EClassID.eDataType, value: ParameterValues.eDataType.IEEEFLOAT);
channel.Parameters.UpdateOrAdd(EClassID.eSampleTime, value: 0.001, editorType: EEditorType.Hidden);
```

```
// for TimeStamped-ASCII (log channel) set DataType to ASCII
channel.Parameters.UpdateOrAdd(EClassID.eDataType, value: ParameterValues.eDataType.ASCII);
```

DataTransferDriver

A TimeStampedSample array must be passed to the `WriteData`-Method in the data transfer driver.

In this example, the *SimplePollDevice* template is used.

```
// OnProcess-Method
// ...
if (dataWriter != null)
{
   var timeStampedData = GetTimeStampedData(dataTransferNode); // get time stamped data from the device

   if (timeStampedData != null)
   {
      EResult result = dataWriter.WriteData(timeStampedData, 0); // write time stamped data to imc
STUDIO
      // ...
   }
}
// ...
```

```
private TimeStampedSample<double>[] GetTimeStampedData(DataTransferNode dataTransferNode)
{
   // get time stamped data from the device
   // use simulated data from the DeviceDriver
   double[] data = m_deviceDriver.ReadData(dataTransferNode.Name); // simulated data
   var sampleTime = m_deviceDriver.GetSampleTime(dataTransferNode.Name); // method was added in
DeviceDriver
   var timeStampedData = ConvertToTimeStamped(data, sampleTime); // convertion needed to get time
stamped data
   return timeStampedData;
}
```

```
private double lastArgument = 0;
private DataTransferDateTime deviceStartTime;

private TimeStampedSample<double>[] ConvertToTimeStamped(double[] data, double sampleTime)
{
   if (data == null)
   {
      return null;
   }

   var timeStampedData = new TimeStampedSample<double>[data.Length];
   var delayedTime = m_deviceDriver.CurrentTime - deviceStartTime;

   for (int i = 0; i < timeStampedData.Length; i++)
   {
      lastArgument = delayedTime.Ticks / (sampleTime * DataTransferTimeSpan.TicksPerSecond);
      var argument = lastArgument + i;
      var argumentAsLong = Convert.ToInt64(argument);

      timeStampedData[i].Offset = argumentAsLong;
      timeStampedData[i].Value = data[i];
   }

   return timeStampedData;
}
```

DeviceDriver

A method `GetSampleTime` is added to the device driver.

```
public double GetSampleTime(string channelName)
{
   var sampleTime = 0d;

   Channel channel;
   if (m_channels.TryGetValue(channelName, out channel))
   {
      sampleTime = channel.SampleTime;
   }

   return sampleTime;
}
```

17.2.3.3.5 eRange parameter

The following instructions are needed to add the eRange parameter to a channel.

GetDeviceDescription

The parameter *eRange* needs to be added to the devices description. The values can be selected in a combo box.

```
// var channel = ...
// ...
channel.Parameters.UpdateOrAdd(EClassID.eRange, value: 10, editorType: EEditorType.ComboBox);
```

GetParameterConstraints

In the methode **GetParameterConstraints** (in the region "Optional methods") the items of the selection are set.

```
if (parameter.Name == EClassID.eRange.ToString()) {
   constraint.ValueList.Add(20);
   constraint.ValueList.Add(10);
   constraint.ValueList.Add(5);
   constraint.ValueList.Add(2.5);
   constraint.ValueList.Add(1);
}
```

GetParameterDisplayText

If a different text should be shown in the list, you need to add the following code to the method **GetParameterDisplayText** (in the region "Display text handling").

```
if (parameter.Name == EClassID.eRange.ToString()) {
//   if (object.Equals(value, 20)  ) {
      return "\u00b1" + value + " V";
//   }
}
```

If you need to distinguish between the values, you can use an if- or switch-case statement.

The methods also needs to return a value, e.g. `null`.

DataTransferDrivcer.cs : Prepare()

To use the parameter, you need to iterate over the SetupNode-Children:

```
foreach (SetupNode child in configuration.Children) {
   if (child.NodeType == EClassID.eChannel.ToString()) {
      SetupParameter range = child.Parameters.GetByName(EClassID.eRange);
      // do something with      range.Value;
   }
}
```

17.2.3.4 Action column (eAction)

It is possible to add an action-column for the device/channels. This may be desirable when intending to have a configuration re-imported from the device at the push of a button. In order for this column to appear in the user interface, it must also be added to the associated table by means of the Table description tool window.

The source code already contains examples pertaining to the parameter **eAction,** which are commented-out

GetDeviceDescription

```
device.Parameters.UpdateOrAdd(EClassID.eAction, value: 1, editorType: EEditorType.ListButton);
```

Subsequently, values must be written to this selection list:

GetParameterConstraints

Add new elements with corresponding ID:

```
if (parameter.Name == EClassID.eAction.ToString()) {
   constraint.ValueList.Add(1);
   constraint.ValueList.Add(2);
   constraint.ValueList.Add(3);
}
```

GetParameterDisplayText

In the selection list, the corresponding texts are listed according to their ID.

```
if (parameter.Name == EClassID.eAction.ToString()) {
   if ((int)value == 1) {
         return "Edit configuration";
      } else if ((int)value == 2) {
         return "Import configuration";
      } else if ((int)value == 3) {
         return "Export configuration";
      }
}
```

ExecuteParameter

Here you determine which actions are to be performed in response to a particular selection.

```
switch (actionStepID) {
   case 1:
      MessageBox.Show("Edit Configuration");
      // do something
      break;
   case 2:
      MessageBox.Show("Import Configuration");
      // do something
      break;
   case 3:
      MessageBox.Show("Export Configuration");
      // do something
      break;
}
```

Once the column has been successfully added to the table in the user interface, it is operational when the corresponding device is selected. The associated action is executed upon pushing the button.

17.2.3.5 Data transfer

When creating a third-party script, you are also provided with a data transfer template in the DataTransferDriver.cs.

The next issue to address is whether the target device is polled (see SimplePollDevice 1937), or whether it causes a function call (*callback*) when it has data (see SimplePushDevice 1940).

The SimplePollDevice and the SimplePushDevice are good examples of such Third Party Device.

For further information, please read DeviceDriver and PushDeviceDriver 1932.

Access to the logbook

In the DataTransferDriver you can access to the logbook via the `Environment` class.

```
Environment.Logbook.LogEntry("MySender", "MyMessage", 0, ELogbookEntryCategory.Warning);
```

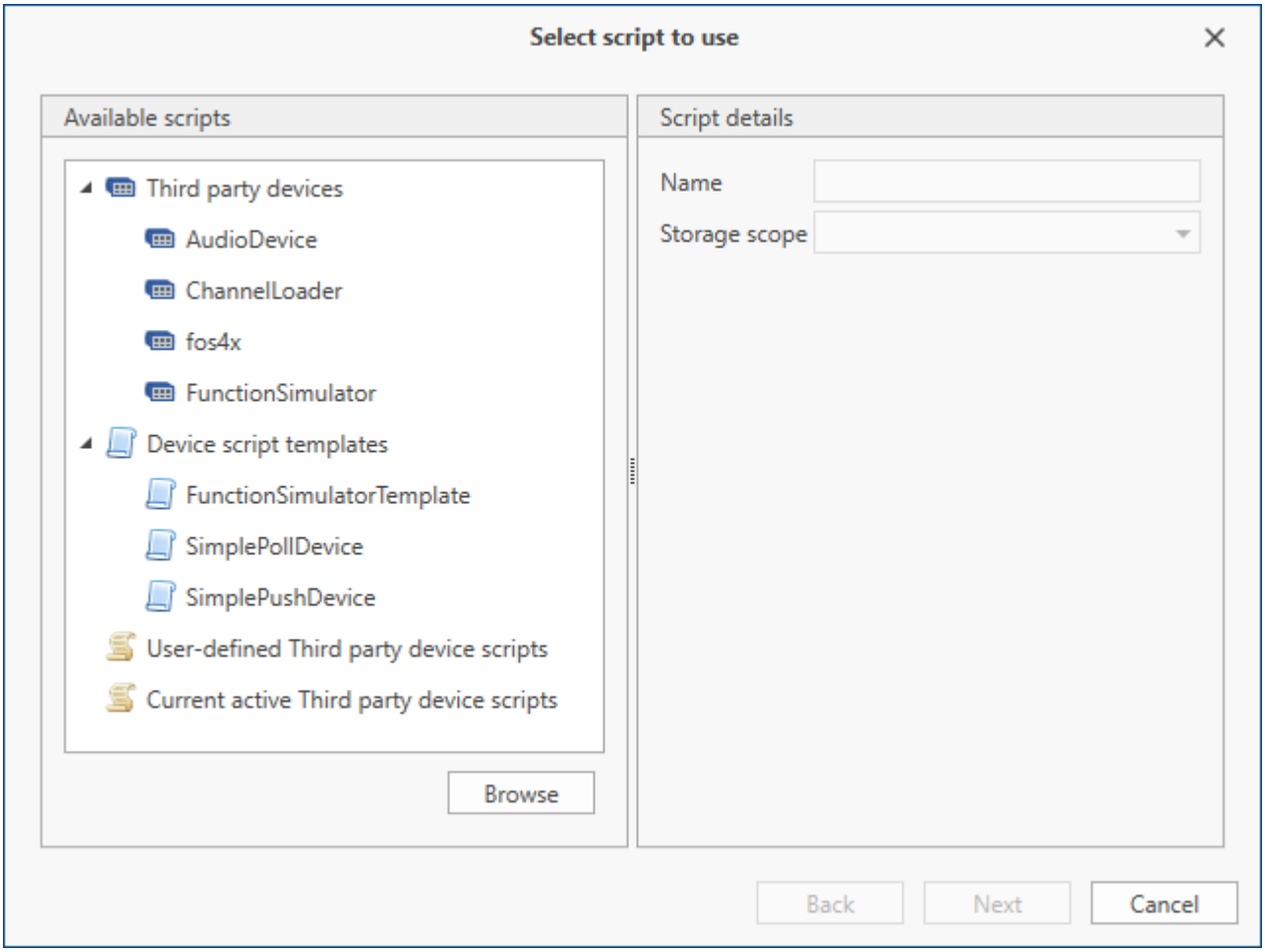
17.2.4 3rd Party Device Management

Using the **3rd-Party Device Management**, it is possible to select executable 3rd-party device script templates, e.g. the Function-Simulator, the SimplePollDevice or the SimplePushDevice. Likewise, there are already pre-made 3rd-party devices. Along with the AudioDevice and the ChannelLoader, the fos4x is also available. fos4x requires an extra license.

Note

- The 3rd-Party Device Management is available in the editions *Developer*, *Professional* and *Standard*.
- For the use of 3rd-party devices, an activated **imc STUDIO 3PDI-Inclusive** or **imc STUDIO 3PDI-Exclusive** license is required.
- The **AudioDevice**, e.g. the audio interface **2-channel IEPE/ICP-signal conditioner (FH/ICP2-USB),** the **ChannelLoader** and the **FunctionSimulator** can be used without any **3PDI license**.
- For the templates a **3PDI-license** is needed.
- For the fos4x device an **imc STUDIO 3PDI-fos4x** license is required**.**

Ribbon	View
Setup-Control > 3rd Party Device Management ( )	Complete



3rd Party Device Management Dialog

When selecting a script-template, there is an option for subsequently opening the Script-Editor.

User-defined scripts can be added by clicking on "*Browse*".

In "*Current active Third party device scripts*" all Third Party Device Scripts are listed and can be deleted by selecting and clicking on "*Next*".

AudioDevice

In order to use connected Audio-devices, select the device-script *AudioDevice* and click on *Next*. Subsequently, all Audio-devices appear in the device list. When one of the Audio-devices is selected, the computer's Audio-input channels appear on the page *Analog Channels*. In this way, when a measurement starts, the computer's Audio-inputs can be graphed and recorded.

For more information on this subject, see the chapter AudioDevice 723.

ChannelLoader

The *ChannelLoader* serves to play back data already recorded. To do this, select the *ChannelLoader* and click on *Next*. Subsequently, the device *ChannelLoader* appears in the device list. When *ChannelLoader* is selected, a file selection dialog with multi-selection appears. Here, multiple already-recorded data or imc FAMOS-data sets can be selected. On the page *Analog Channels*, the corresponding channels appear. When a measurement starts, then the selected data are played back cyclically at their respective sampling rates.

For more information on this subject, see the chapter ChannelLoader 724.

Device script templates

The devices from the script-templates simulate a sine-signal, or in the case of the *FunctionSimulator*, a number of different signals (trapezoid, sawtooth, rectangle, ...).

17.2.5 Complete third party devices

Section	Description
AudioDevice 723	Access to input channels of audio devices (sound card, microphone, ...)
ChannelLoader 724	Playback device for replay measurement data.
fos4x 725	Fos4x devices
FunctionSimulator 727	FunctionSimulator

17.2.6 Device script templates

Section	Description
DeviceDriver and PushDeviceDriver 1932	Common component for data transfer.
FunctionSimulator 1933	Device for simulating signals (sine, cosine, rectangle, triangle, ...)
SimplePollDevice 1937	Template for a device getting data via *polling*.
SimplePushDevice 1940	Template for a device which runs a *callback* function when measurement data is available.

Note Using different third party devices based on the same template

If you want to use different third party devices based on the same template, please read the following instructions:

- Select a different name, when using the template in the Third-Party-Device management
- Rename the class name via keyboard (CTRL + R)
- Change the AssemblyName in the C# project settings.

If this note is not considered, only one device will be shown in the device list.

17.2.6.1 DeviceDriver and PushDeviceDriver

Note

- Here, the virtual device classes which simulate data for the **examples** (SimplePollDevice, SimplePushDevice and FunctionSimulator) are described.
- They can be used for making one's own Third Party Device.
- To write an unmanaged Data-Transfer-Driver, you need some header files that can be provided to you on request.

DeviceDriver

For the Third Party Devices FunctionSimulator and the SimplePollDevice, there is a device class called DeviceDriver available in the Third Party Device script. It describes a virtual device which us used and governed by the respective Third Party Device Script.

In this class, there is a description of the device and its channels, as well as of methods for retrieving simulated data.

One important method here is ReadData(), by means of which you can access the virtual device's data from the DataTransferDriver. In this method, the corresponding mathematics functions (sine, cosine, trapezoid, ...) are calculated and returned in double-Array format.

PushDeviceDriver

For the device SimplePushDevice, there is a class PushDeviceDriver, which is derived from the DeviceDriver. When the object is created, a callback-method must be passed, which is called when data are available in the device.

```
internal class PushDeviceDriver : DeviceDriver {
   public PushDeviceDriver(DataTransferEnvironment environment, DataCallback dataCallback)
      : base(environment) {

      m_dataPumpTimer = new Timer(DataPumpProc, null, Timeout.Infinite, Timeout.Infinite);
      m_dataCallback = dataCallback;
   }
   // ...
}
```

In the simulation, by means of a timer, the method `DataPumpProc()` is implemented, which passes the data accordingly.

```
private DataCallback m_dataCallback = null;

private void DataPumpProc(object state) {
   if (Monitor.TryEnter(this)) {
      try {
         Synchronize();

         foreach (Channel channel in m_channels.Values) {
            double[] data = channel.ReadData(); // read data from device
            if (data != null && m_dataCallback != null) {
               m_dataCallback(channel.Name, new DataEventArgs(DataEvent.Data, data));
            }
         }
      } finally {
         Monitor.Exit(this);
      }
   }
}
```

17.2.6.2 FunctionSimulator template

The FunctionSimulator is device which simulates various kinds of mathematics functions, including sine, cosine, triangle, sawtooth and squarewave.

It is also possible to adjust the frequency, the scaling and the offset.
The analog channels and Display variables are set in the device/channel descriptions.

GetAvailableDevice

```
protected override void GetAvailableDevices(SetupNodeList deviceList) {
   // Add the currently available devices here! Example:
   var device = deviceList.GetOrAdd(ESetupNodeType.Device, "FunctionSimulator");
   device.Parameters.UpdateOrAdd(EClassID.eDeviceNickname, value: "Function Simulator");
   device.Parameters.UpdateOrAdd(EClassID.eDeviceSN, value: "");
   device.Parameters.UpdateOrAdd(EClassID.eDeviceProduct, value: "");
}
```

GetDeviceDescription

```
protected override void GetDeviceDescription(SetupNode device) {
   // Add the full device description here! Channels, modules...

   AddDisplayVariable(device, "Scale", "Scale", "V");
   AddDisplayVariable(device, "Frequency", "Frequency", "Hz");
   AddDisplayVariable(device, "Offset", "Offset", "V");

   AddChannel(device, FunctionType.Sine.ToString(), "sin");
   AddChannel(device, FunctionType.Cosine.ToString(), "cos");
   AddChannel(device, FunctionType.Rectangle.ToString(), "rect");
   AddChannel(device, FunctionType.Sawtooth.ToString(), "saw");
   AddChannel(device, FunctionType.Triangle.ToString(), "tri");
   AddChannel(device, FunctionType.Noise.ToString(), "noi");
   AddChannel(device, FunctionType.DC.ToString(), "dc");
   AddChannel(device, FunctionType.Trapezoid.ToString(), "tra");
}
```

The capabilities of the FunctionType are stated in the Data transfer 1929.

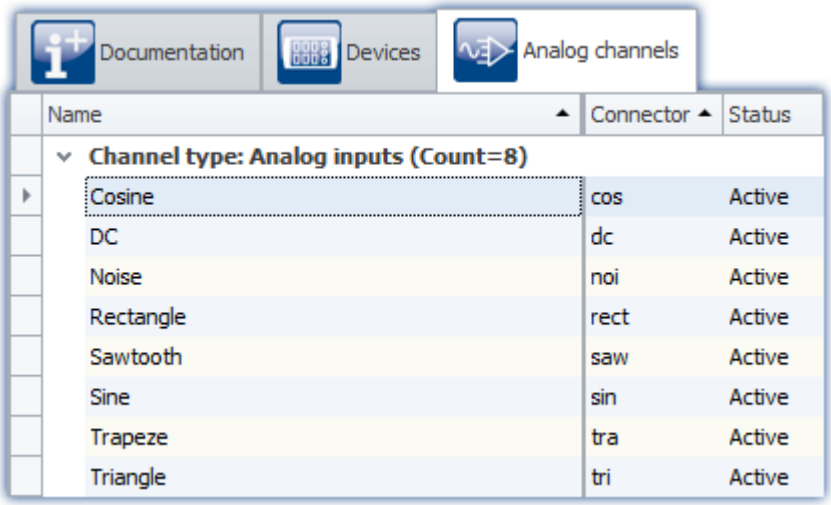
To make it easier to add analog channels, the function `AddFunctionChannelToDevices()` is provided here.

```
private void AddChannel(SetupNode device, string functionName, string functionNameShort) {
   var channel = device.Children.GetOrAdd(ESetupNodeType.Channel, functionName);
   channel.Parameters.UpdateOrAdd(EClassID.eChannelName, value: functionName);
   channel.Parameters.UpdateOrAdd(EClassID.ePlugInName, value: functionNameShort);
   channel.Parameters.UpdateOrAdd(EClassID.eStatus, value: ParameterValues.eStatus.Active);

   channel.Parameters.UpdateOrAdd(EClassID.eChannelType, value:
ParameterValues.eChannelType.AnalogInput);
   channel.Parameters.UpdateOrAdd(EClassID.eXFormat, value: ParameterValues.eXFormat.Normal);
   channel.Parameters.UpdateOrAdd(EClassID.eDataType, value: ParameterValues.eDataType.IEEEDOUBLE);

   channel.Parameters.UpdateOrAdd(EClassID.eSampleTime, value: 0.001, editorType:
EEditorType.ComboBox); // 1 kHz (default value)

   channel.Parameters.UpdateOrAdd(EClassID.eDivisionUnit, value: "V"); // The original (electrical)
measurement unit
   channel.Parameters.UpdateOrAdd(EClassID.eUserUnit, value: "V");

   channel.Parameters.UpdateOrAdd(EClassID.eExternalTriggerTarget, value: true);
}
```

The frequency, the scaling and the offset can be implemented with the help of the method `AddDisplayVariableToDevice()`.

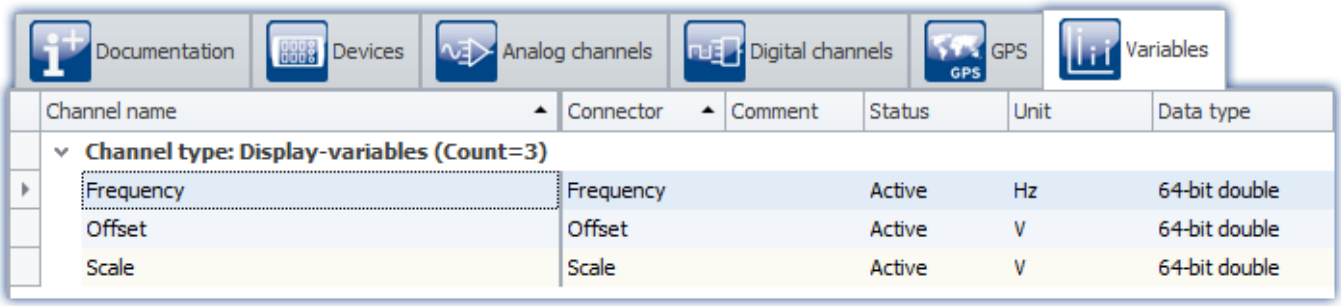
```
private void AddDisplayVariable(SetupNode device, string variableName, string variableNameShort, string
unit) {
   var channel = device.Children.GetOrAdd(ESetupNodeType.Channel, variableName);
   channel.Parameters.UpdateOrAdd(EClassID.eChannelName, value: variableName);
   channel.Parameters.UpdateOrAdd(EClassID.ePlugInName, value: variableNameShort);
   channel.Parameters.UpdateOrAdd(EClassID.eStatus, value: ParameterValues.eStatus.Active);
   channel.Parameters.UpdateOrAdd(EClassID.eChannelType, value:
ParameterValues.eChannelType.DisplayVar);
   channel.Parameters.UpdateOrAdd(EClassID.eDivisionUnit, value: unit); // The original (electrical)
measurement unit
   channel.Parameters.UpdateOrAdd(EClassID.eUserUnit, value: unit);
   channel.Parameters.UpdateOrAdd(EClassID.eDataType, value: ParameterValues.eDataType.IEEEDOUBLE);

   channel.Parameters.Remove(EClassID.eSampleTime);
   channel.Parameters.Remove(EClassID.eDuration);
   channel.Parameters.Remove(EClassID.ePCTransferToHD);
}
```

The analog channels then appear on the Setup-page under "*Analog channels*".



Setup: FunctionSimulator (Analog Channels)

It is also possible to adjust the frequency, the scaling and the offset. The Display variables appear on the Setup page under "*Variables*".



Setup: FunctionSimulator (Variables)

DataTransferDriver.cs

The DataTransferDriver handles the data transfer from Third Party Device to imc STUDIO.

For this purpose, the Third Party Device-object must be known to the DataTransferDriver. With the help of this object, it is then possible to control the Third Party Device.

In this example, upon initialization the corresponding object DeviceDriver 1932 is created.

```
public class DataTransferDriver : AbstractDataTransferDriver {
// ...
   private DeviceDriver m_deviceDriver = null; // device object

   protected override ILogbookEntryBase[] Initialize(DataTransferEnvironment environment){
      m_deviceDriver = new DeviceDriver(environment); // initializing device
      return null;
   }
// ...
}
```

Upon performing a Prepare action, the channel settings are applied and transferred to the device.

```
protected override ILogbookEntryBase[] Prepare(SetupNode configuration, DataTransferNodeList
dataTransferNodes) {
// ...
   m_deviceDriver.ConfigureChannel(child.Name, FunctionType.Sine, (double)sampleTime.Value); // setting
channels in the device
// ...
}
```

Additionally, the Display variables are set.

```
if (dataTransferNode.Name == "Frequency") {
   dataWriter.WriteData(1.0);
} else if (dataTransferNode.Name == "Offset") {
   dataWriter.WriteData(0.0);
} else if (dataTransferNode.Name == "Scale") {
   dataWriter.WriteData(1.0);
}
```

When starting and stopping, the datWriter is strted/stopped.

During measurement, the data are continually polled by the device via a timer and exported to imc STUDIO by the dataWriter. Likewise, the Display variables re imported and processed.

```
public DataTransferDriver() {
   m_processDataTimer = new Timer(OnProcessData, null, Timeout.Infinite, Timeout.Infinite);
}
```

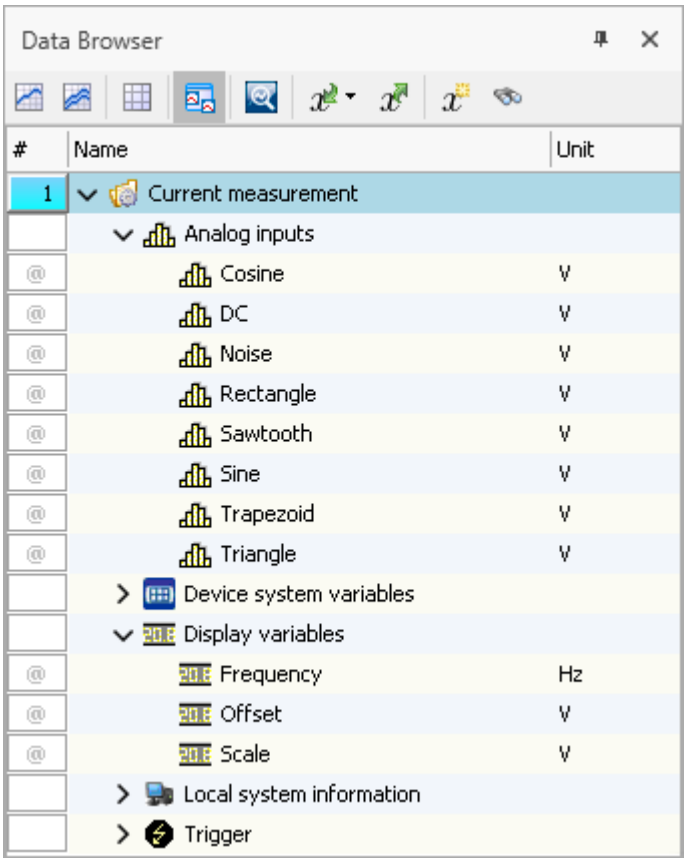
```
private void OnProcessData(object state) {
// ...
   // processing Display variables
   if (dataTransferNode.Name == "Frequency") {
      if (dataReader.ReadData(out frequency) == EResult.Success) {
         if (frequency <= 0.1) {
            frequency = 0.1;
            dataWriter.WriteData(frequency);
         } else if (frequency >= 10000.0) {
            frequency = 10000.0;
            dataWriter.WriteData(frequency);
         }
      }
   } else if (dataTransferNode.Name == "Offset") {
      if (dataReader.ReadData(out offset) == EResult.Success) {
         if (Math.Abs(offset) > 1e20) {
            offset = Math.Sign(offset) * 1e20;
            dataWriter.WriteData(offset);
         }
      }
   } else if (dataTransferNode.Name == "Scale") {
      if (dataReader.ReadData(out scale) == EResult.Success) {
         if (Math.Abs(scale) > 1e20) {
            scale = Math.Sign(scale) * 1e20;
            dataWriter.WriteData(scale);
         }
      }
   }

   // data retrieved from the device and exported to STUDIO
   double[] data = m_deviceDriver.ReadData(dataTransferNode.Name);
   EResult result = dataWriter.WriteData(data);
//...
}
```

DeviceDriver.cs

For details on this topic, see the chapter DeviceDriver and PushDeviceDriver 1932.

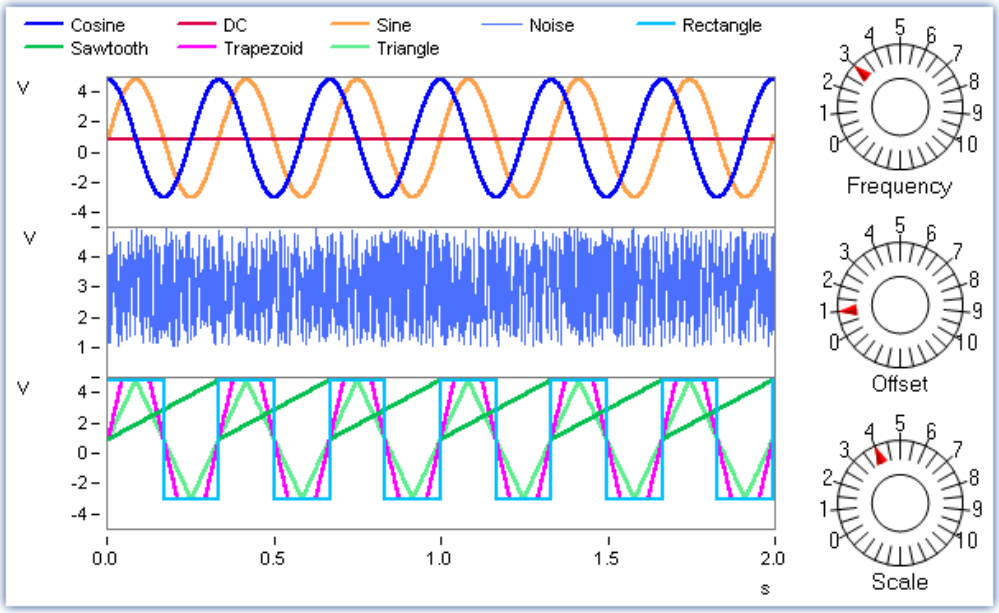
Functionality in imc STUDIO

After selection and configuration of the device, the channels and Display variable are listed in the Data Browser.



Data Browser: FunctionSimulator

The signals can be displayed in the curve window.



Curve window: FunctionSimulator

By changing the Display variables, it is possible to alter the frequency, amplitude and offset during runtime.

Note

- The FunctionSimulator can be used as trigger target action of an imc device.
- Important: The time zone in the imc device must be corresponding to the PC's time zone.

17.2.6.3 SimplePollDevice

One simple example of Third Party Device is the "SimplePollDevice", which generates a sine signal.

GetAvailableDevices

```
protected override void GetAvailableDevices(SetupNodeList deviceList) {
   // Add the currently available devices here! Example:
   var device = deviceList.GetOrAdd(ESetupNodeType.Device, "SimplePollDevice");
   device.Parameters.UpdateOrAdd(EClassID.eDeviceNickname, value: "SimplePollDevice");
   device.Parameters.UpdateOrAdd(EClassID.eDeviceSN, value: "");
   device.Parameters.UpdateOrAdd(EClassID.eDeviceProduct, value: "");
}
```

GetDeviceDescription

In the device description, the channel information is presented. In this case, the channel is an analog channel named "Poll_Input_001", having a pre-set sampling interval of 0.01 seconds.

```
protected override void GetDeviceDescription(SetupNode device) {
   // Add the full device description here! Channels, modules...
//
   // Example code (equidistant channel with a float sample every 10ms):
   var channel = device.Children.GetOrAdd(ESetupNodeType.Channel, "Poll_Input_001");
   channel.Parameters.UpdateOrAdd(EClassID.eChannelName, value: "Poll_Input_001");
   channel.Parameters.UpdateOrAdd(EClassID.ePlugInName, value: "IN01");
   channel.Parameters.UpdateOrAdd(EClassID.eStatus, value: ParameterValues.eStatus.Active);
   channel.Parameters.UpdateOrAdd(EClassID.eChannelType, value:
ParameterValues.eChannelType.AnalogInput);
   channel.Parameters.UpdateOrAdd(EClassID.eDivisionUnit, value: "V"); // The original (electrical)
measurement unit
   channel.Parameters.UpdateOrAdd(EClassID.eUserUnit, value: "V");

   channel.Parameters.UpdateOrAdd(EClassID.eXFormat, value: ParameterValues.eXFormat.Normal);
   channel.Parameters.UpdateOrAdd(EClassID.eDataType, value: ParameterValues.eDataType.IEEEDOUBLE);
   channel.Parameters.UpdateOrAdd(EClassID.eSampleTime, value: 0.01, editorType:
EEditorType.ComboBox); // 100 Hz (default value)
}
```

The data transfer is handled by a special object of a class (*managed*).

```
public override IDataTransferDriverBase CreateDataTransferDriver() {
   return new DataTransferDriver();
}
```

DataTransferDriver.cs

The DataTransferDriver handles the transfer of data from the Third Party Device to imc STUDIO.

For this purpose, the Third Party Device-object must be known to the DataTransferDriver. With the help of this object, it is then possible to control the Third Party Device.

In this example, the corresponding object DeviceDriver 1932 is created in the initialization process.

```
public class DataTransferDriver : AbstractDataTransferDriver {
// ...
   private DeviceDriver m_deviceDriver = null; // Device-Object

   protected override ILogbookEntryBase[] Initialize(DataTransferEnvironment environment){
      m_deviceDriver = new DeviceDriver(environment); // Initializing device
      return null;
   }
// ...
}
```

Upon performing a Prepare action, the channel settings are applied and transferred to the device.

```
protected override ILogbookEntryBase[] Prepare(SetupNode configuration, DataTransferNodeList
dataTransferNodes) {
// ...
   m_deviceDriver.ConfigureChannel(child.Name, FunctionType.Sine, (double)sampleTime.Value); //
Setting channels in the device
// ...
}
```

When starting and stopping, the dataWriter is started/stopped.

During measurement, the data are continually polled by the device via a timer and exported to imc STUDIO by the dataWriter.

```
public DataTransferDriver() {
   m_processDataTimer = new Timer(OnProcessData, null, Timeout.Infinite, Timeout.Infinite);
}
```

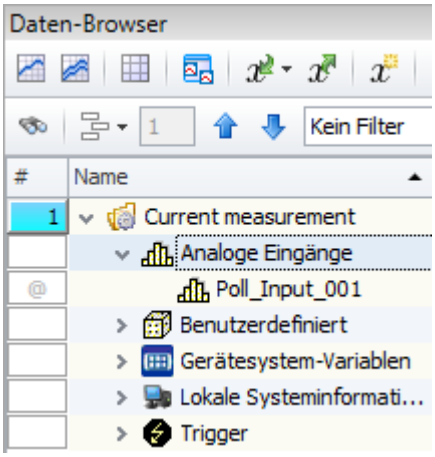
```
private void OnProcessData(object state) {
// ...
   double[] data = m_deviceDriver.ReadData(dataTransferNode.Name); // retrieving data from the device
   EResult result = dataWriter.WriteData(data);
//...
}
```

DeviceDriver.cs

For details, see the chapter DeviceDriver and PushDeviceDriver 1932.

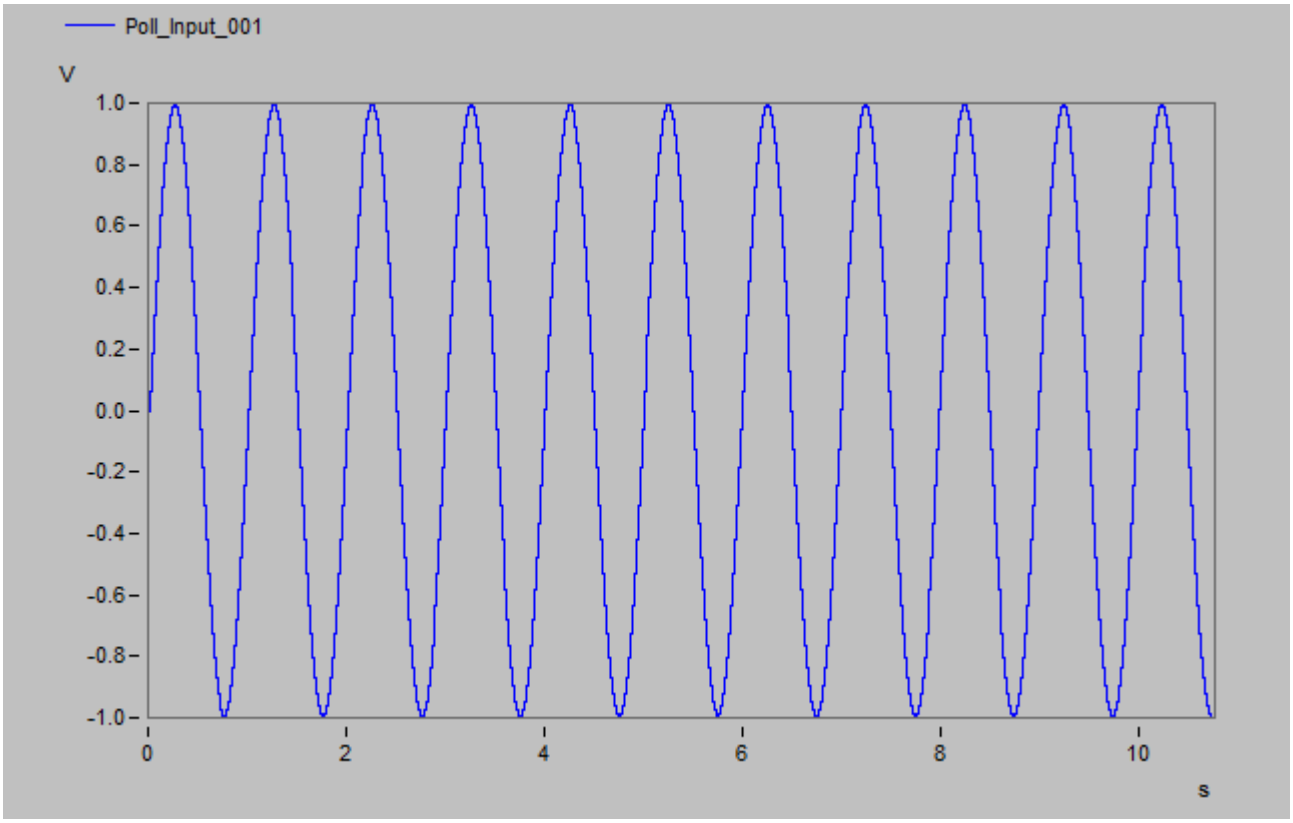
Functionality in imc STUDIO

After selection of the device and its configuration, the analog channel "Poll_Input_001" appears in the Data Browser.



Data Browser: SimplePollDevice

Once the measurement starts, the sine-signal generated can be displayed in a curve window.



Curve window: SimplePollDevice

17.2.6.4 SimplePushDevice

An additional example of Third Party Device is the "SimplePushDevice", which generates a sine-signal. In the process, the device initiates a callback.

GetAvailableDevices

```
protected override void GetAvailableDevices(SetupNodeList deviceList) {
   // Add the currently available devices here! Example:
   var device = deviceList.GetOrAdd(ESetupNodeType.Device, "SimplePushDevice");
   device.Parameters.UpdateOrAdd(EClassID.eDeviceNickname, value: "SimplePushDevice");
   device.Parameters.UpdateOrAdd(EClassID.eDeviceSN, value: "");
   device.Parameters.UpdateOrAdd(EClassID.eDeviceProduct, value: "");
}
```

GetDeviceDescription

In the device description, the channel information is presented. In this case, the channel is an analog channel named "Input_001" with a pre-set sampling interval of 0.01 seconds.

```
protected override void GetDeviceDescription(SetupNode device) {
   // Add the full device description here! Channels, modules...

   // Example code (equidistant channel with a float sample every 10ms):
   var channel = device.Children.GetOrAdd(ESetupNodeType.Channel, "Push_Input_001");
   channel.Parameters.UpdateOrAdd(EClassID.eChannelName, value: "Push_Input_001");
   channel.Parameters.UpdateOrAdd(EClassID.ePlugInName, value: "IN01");
   channel.Parameters.UpdateOrAdd(EClassID.eStatus, value: ParameterValues.eStatus.Active);
   channel.Parameters.UpdateOrAdd(EClassID.eChannelType, value:
ParameterValues.eChannelType.AnalogInput);
   channel.Parameters.UpdateOrAdd(EClassID.eDivisionUnit, value: "V"); // The original (electrical)
measurement unit
   channel.Parameters.UpdateOrAdd(EClassID.eUserUnit, value: "V");

   channel.Parameters.UpdateOrAdd(EClassID.eXFormat, value: ParameterValues.eXFormat.Normal);
   channel.Parameters.UpdateOrAdd(EClassID.eDataType, value: ParameterValues.eDataType.IEEEDOUBLE);
   channel.Parameters.UpdateOrAdd(EClassID.eSampleTime, value: 0.01); // 100 Hz (default value)
}
```

The data transfer is handled by a special object of a class (*managed*).

```
public override IDataTransferDriverBase CreateDataTransferDriver() {
   return new DataTransferDriver();
}
```

DataTransferDriver.cs

The DataTransferDriver handles the data transfer from Third Party Device to imc STUDIO.

For this purpose, the Third Party Device-object must be known to the DataTransferDriver. With the help of this objects, it is then possible to control to the Third Party Device.

In this example, upon initialization the corresponding object PushDeviceDriver 1932, which is described in the file DeviceDriver.cs, is created. The method `OnProcessData` is passed to the SimplePushDevice, and this method is called when data are finalized in the Third Party Device and can be transferred (*push*) to imc STUDIO.

```
public class DataTransferDriver : AbstractDataTransferDriver {
// ...
   private PushDeviceDriver m_deviceDriver = null;

   protected override ILogbookEntryBase[] Initialize(DataTransferEnvironment environment)    {
      m_deviceDriver = new PushDeviceDriver(environment, OnProcessData);
      return null;
   }
// ...
}
```

Upon performing a Prepare action, the channel's settings are applied and transferred to the device.

```
protected override ILogbookEntryBase[] Prepare(SetupNode configuration, DataTransferNodeList
dataTransferNodes) {
// ...
   m_deviceDriver.ConfigureChannel(child.Name, FunctionType.Sine, (double)sampleTime.Value); //
setting channels in the device
// ...
}
```

When starting and stopping, the dataWriter is started/stopped.

During the measurement, the method `OnProcessedData()` is called, if the Third Party Device itself has concluded it data package and calls this event.

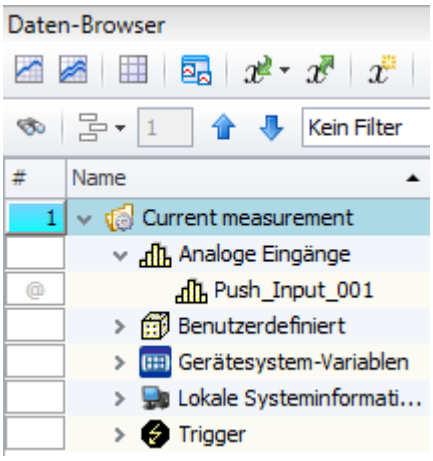
```
private void OnProcessData(string channelName, DataEventArgs e) {
// ...
   switch (e.DataEvent) {
      case DataEvent.Begin:
         dataWriter.Open();
         result = dataWriter.BeginWrite(e.Time.Value);
         // Error-Handling
         break;
      case DataEvent.Data:
         result = dataWriter.WriteData(e.Data);
         // Error-Handling
         break;
      case DataEvent.End:
         result = dataWriter.EndWrite();
         dataWriter.Close();
         // Error-Handling
         break;
      default:
         break;
   }
//...
}
```

DeviceDriver.cs

For more on this topic, see the chapter DeviceDriver and PushDeviceDriver 1932.

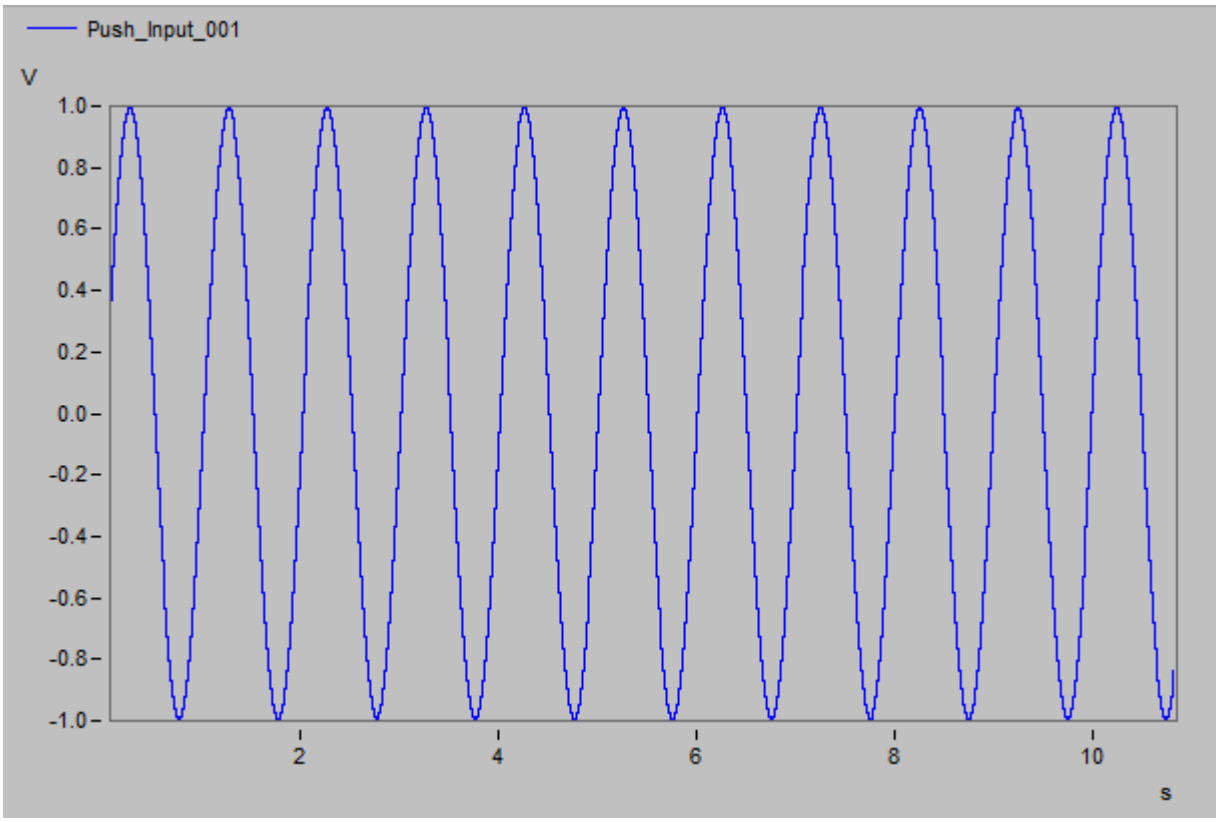
Functionality in imc STUDIO

After selection of the device and its configuration, the analog channel "Poll_Input_001" appears in the Data Browser.



Data Browser: SimplePushDevice

Once the measurement has been started, the sine-signal generated can be displayed in a curve window.



Curve window: SimplePushDevice

17.2.7 FAQ

FAQ

When importing an older experiment with a third party device, I get the error "System.IO.FileNotFoundException"

Answer: The given third party devices are imported as DLL. There are dependencies to other libraries. In the version 5.2 there are changes causing the following error message:

```
The Script "ChannelLoader" encountered: System.IO.FileNotFoundException: Could not load file or
assembly 'imc.Studio.Interfaces.V2, Version=5.0.2.0, Culture=neutral,
PublicKeyToken=68a4b1d388e6c0b6' or one of its dependencies. The system cannot find the file
specified.
File name: 'imc.Studio.Interfaces.V2, Version=5.0.2.0, Culture=neutral,
PublicKeyToken=68a4b1d388e6c0b6' [...]
```

To fix this error, please remove the corresponding third party device script (AudioDevice, ChannelLoader) with the help of the 3rd Party Device Management and add it again.

Contact the author of the third party device script if you have further questions.

17.2.8 Tutorial

Here you will find a list of tutorials:

Tutorial	Beschreibung
Add a channel 1943	A new channel is added to the SimplePollDevice.
Add a Display variable 1945	A Display variable for setting the signal frequency is added to the SimplePollDevice.

17.2.8.1 Extension of the SimplePollDevice

17.2.8.1.1 Add a channel

The task is to add an analog channel to the existing SimplePollDevice. For this purpose, a new experiment is created, and additionally the SimplePollDevice added to the device list via the 3rd Party Device Management, unless this already happened.

Note

While the changes are made, the SimplePollDevice may not be selected.

GetAvailableDeviceDevices

The device's name is changed to *SimplePollDeviceExtended*.

```
protected override void GetAvailableDevices(SetupNodeList deviceList) {
   // Add the currently available devices here! Example:
    var device = deviceList.GetOrAdd(ESetupNodeType.Device, "SimplePollDeviceExtended");
    device.Parameters.UpdateOrAdd(EClassID.eDeviceNickname, value: "SimplePollDeviceExtended");
    device.Parameters.UpdateOrAdd(EClassID.eDeviceSN, value: "");
    device.Parameters.UpdateOrAdd(EClassID.eDeviceProduct, value: "");
}
```

GetDeviceDescription

An additional channel having the same properties as the first channel is added. This segment can be copied and inserted. The new channel name is altered.

```
protected override void GetDeviceDescription(SetupNode device) {
   // Add the full device description here! Channels, modules... Example:
    var channel = device.Children.GetOrAdd(ESetupNodeType.Channel, "Poll_Input_001");
    channel.Parameters.UpdateOrAdd(EClassID.eChannelName, value: "Poll_Input_001");
    channel.Parameters.UpdateOrAdd(EClassID.ePlugInName, value: "IN01");
    // ... code is omitted
   var channel2 = device.Children.GetOrAdd(ESetupNodeType.Channel, "Poll_Input_002");
   channel2.Parameters.UpdateOrAdd(EClassID.eChannelName, value: "Poll_Input_002");
   channel2.Parameters.UpdateOrAdd(EClassID.ePlugInName, value: "IN02");
   // ... same settings as Input_001, code is omitted here
}
```

DataTransferDriver.cs

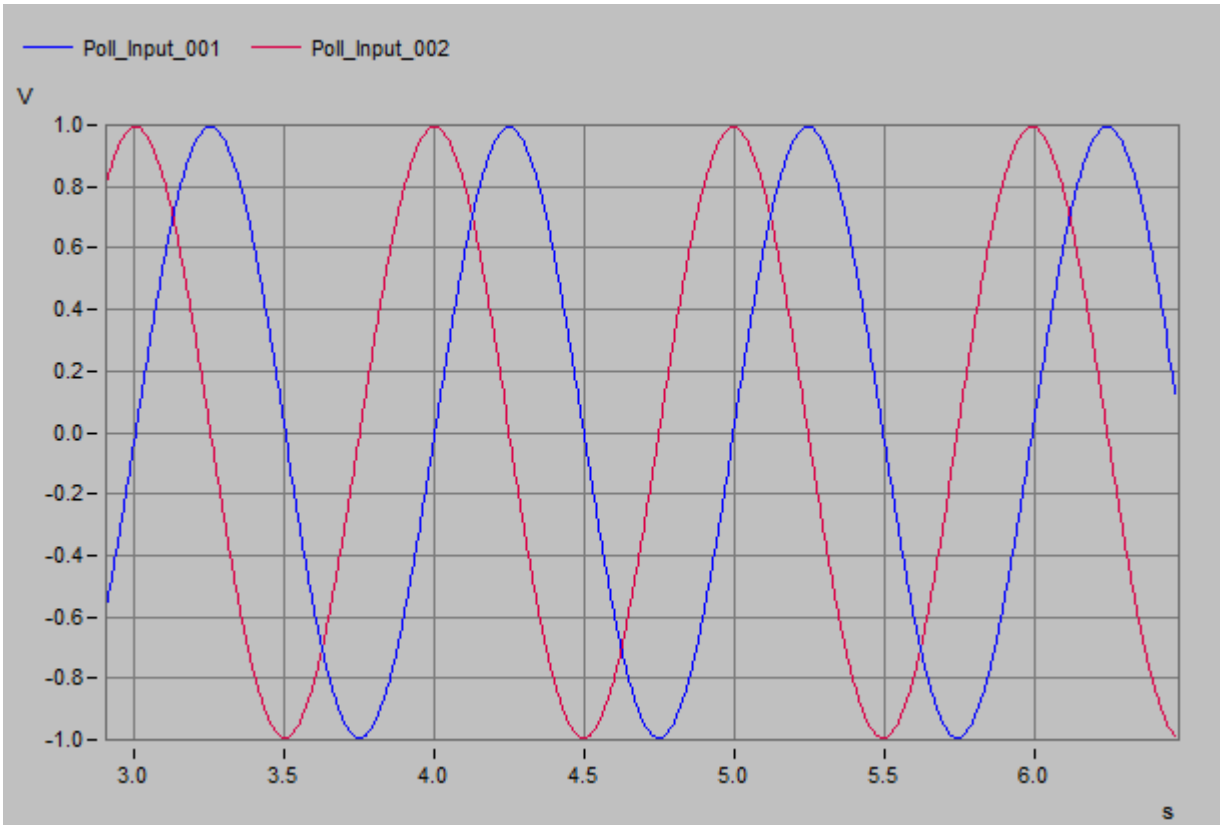
In the class DataTransferDriver, the SimplePollDevice is configured. In the method `Prepare()`, the channels are configured.

By default, sine-signals are generated for all channels. By polling the channel name, it is possible to set a different signal, e.g. cosine.

```
protected override ILogbookEntryBase[] Prepare(SetupNode configuration, DataTransferNodeList
dataTransferNodes) {
// ...
   if (sampleTime != null && sampleTime.Value is double) {

      // Channel 1 : Sine
      if (child.Name == "Poll_Input_001"){
         m_deviceDriver.ConfigureChannel(child.Name, FunctionType.Sine, (double)sampleTime.Value);
      }

      // Channel 2 : Cosine
      if (child.Name == "Poll_Input_002"){
         m_deviceDriver.ConfigureChannel(child.Name, FunctionType.Cosine, (double)sampleTime.Value);
      }

   }
//...

}
```

After selection of the device, the channels *Poll_Input_001* and *Poll_Input_002* appear in the list of analog channels, as well as in the Data Browser.

When a measurement is started, the channels can be displayed in the curve window.



Curve window: Extended SimplePollDevice

17.2.8.1.2 Add a display variable

A Display variable is to be added, which adjusts the frequency for the analog signal.

GetDeviceDescription

The following segment is added to the `GetDeviceDescription`-method. The Display variable "Frequency" is thus added to the device.

```
var displayVar = device.Children.GetOrAdd(ESetupNodeType.Channel, "Frequency");
displayVar.Parameters.UpdateOrAdd(EClassID.eChannelName, value: "Frequency");
displayVar.Parameters.UpdateOrAdd(EClassID.ePlugInName, value: "Freq");
displayVar.Parameters.UpdateOrAdd(EClassID.eStatus, value: ParameterValues.eStatus.Active);
displayVar.Parameters.UpdateOrAdd(EClassID.eChannelType, value:
ParameterValues.eChannelType.DisplayVar);
displayVar.Parameters.UpdateOrAdd(EClassID.eDivisionUnit, value: "Hz"); // The original (electrical)
measurement unit
displayVar.Parameters.UpdateOrAdd(EClassID.eUserUnit, value: "Hz");
displayVar.Parameters.UpdateOrAdd(EClassID.eDataType, value: ParameterValues.eDataType.IEEEDOUBLE);

displayVar.Parameters.Remove(EClassID.eSampleTime);
displayVar.Parameters.Remove(EClassID.eDuration);
displayVar.Parameters.Remove(EClassID.ePCTransferToHD);
```

DataTransferDriver.cs

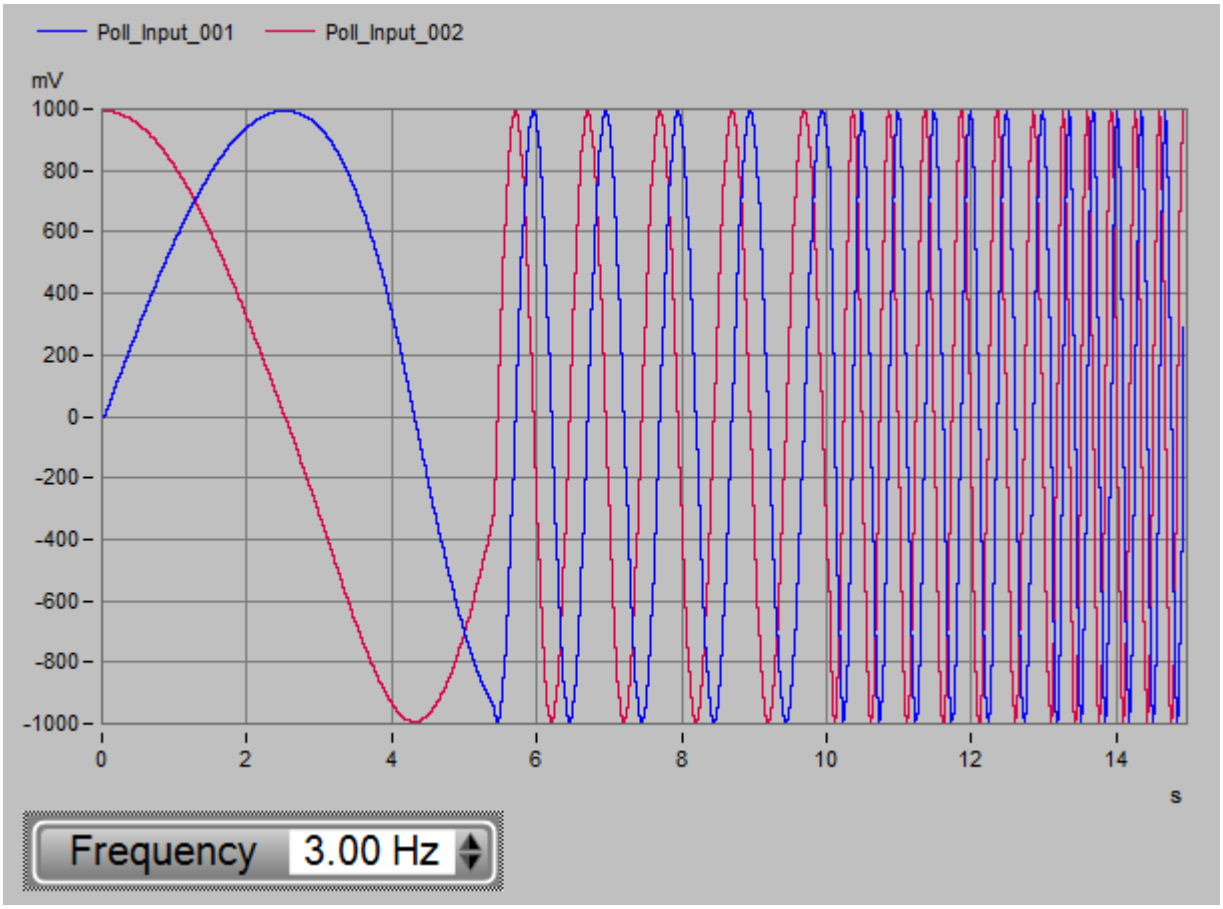
In the class DataTransferDriver, the following code must be amended/edited in the method `OnProcessData()`.

```
if (dataTransferNodes != null) {
   // defining parameters (inserting into existing framework)
   double frequency = 1.0;
   double offset = 0.0;
   double scale = 1.0;
// ...
}
```

```
if (dataTransferNodes != null) {
// ...
// importing Display variable (foreach-block to be inserted before other foreach)
   foreach (DataTransferNode dataTransferNode in dataTransferNodes) {
      DataReader dataReader = dataTransferNode.Reader;
      if (dataReader != null){
         DataWriter dataWriter = dataTransferNode.Writer;
         if (dataWriter != null) {
            if (dataTransferNode.Name == "Frequency") {
               if (dataReader.ReadData(out frequency) == EResult.Success) {
                  if (frequency <= 0.1) {
                     frequency = 0.1;
                     dataWriter.WriteData(frequency);
                  } else if (frequency >= 10000.0) {
                     frequency = 10000.0;
                     dataWriter.WriteData(frequency);
                  } // frequency
               } // ReadData
            } // Name = Frequency
         } // dataWriter
      } // dataReader
   } // foreach
/ ...
}
```

```
// the existing foreach-block must be edited
foreach (DataTransferNode dataTransferNode in dataTransferNodes) {

   DataWriter dataWriter = dataTransferNode.Writer;
   if (dataWriter != null){
      // this line must be inserted for setting the frequency
      m_deviceDriver.SetTunableParameters(dataTransferNode.Name, frequency, offset, scale);
      double[] data = m_deviceDriver.ReadData(dataTransferNode.Name);
      // ...
   }
}
```

After closing the Editor and selecting the third-party device, the Display variable is displayed in the Data Browser. The signal's frequency can now be altered by means of the Display variable.



Curve window: SimpleDevice extended - Display Variable "Frequency"

Adjustment of the amplitude and offset can be accomplished by adding more Display variables. On this subject, see also the FunctionSimulator 1933.

17.3 API

The **imc STUDIO API** (ApplicationInterface) is a programming interface which provides both the ability to freely develop programs and also **fundamental functions** belonging to imc STUDIO, such as:

- reading and writing of device and channel settings (Setup) ,
- access to the Panel and the Widgets,
- access to the Data Browser: creating, reading and writing variables,
- applying of imc FAMOS-functions to channels,
- executing menu actions in the ribbon,
- executing Sequencer commands,
- reaction to events.

This manual explains certain introductory points and describes the interfaces with imc STUDIO.

Note

- The **imc STUDIO API** is provided as part of the installation of imc STUDIO *Developer*.
- The programming language is a .NET programming language (e.g. C#, Visual Basic .NET) .
- You cannot use .NET-Core for the API

17.3.1 Licensing

The **imc STUDIO API** is marketed in conjunction with the **imc DATA API** as **imc API.**

To develop a project, the *imc API Developer* license is needed. To run the programs an *imc API Runtime* license is needed.

With **imc DATA API**, you obtain access to the functionalities of imc FAMOS, including of its Kits, the curve window and the data objects.

The *Base-Project* of the**imc STUDIO API** is installed on the computer as part of installation of the *imc STUDIO Developer*.

If functions from the *imc STUDIO Standard* Edition or higher are needed by **API**, a corresponding license must be activated via the License Manager.

Reference

For more information, see:

- Installation 1947
- Base Interface 1949

17.3.2 Installation

In the process of installing the **imc STUDIO Developer**, the primary project as well as extra sample projects are installed on this path:

"**C:\Users\Public\Documents\imc\imc STUDIO\API 2023**" (or corresponding version)

The subfolder "Shared" contains the basic project which is required for making one's own projects.

The subfolder "Projects" contains executable sample projects.

Prerequisites for development based on the imc STUDIO API

- imc STUDIO Installation (Developer Edition)
- imc DATA API Installation (Developer Edition)
- Development environment, including compiler for the .NET programming language (e.g. Visual Studio 2019 or higher)
- .NET Framework (Version depends on the imc STUDIO Version)
- Use of an WPF or Windows-Forms project

Prerequisites for using the imc STUDIO API as runtime

- imc STUDIO Installation (corresponding to the functionality)
- imc DATA API Installation (Runtime Edition)

Note

It is recommended that you have only one imc STUDIO version installed at a time in order to avoid version conflicts with the assemblies.

Note .NET Framework Version

The calling application (*.exe) must be the same or higher .NET Framework version of the imc STUDIO API component.
In imc STUDIO 2023, **.NET Framework 4.8** is used.

Reference imc DATA API

You can find the Setup file of the **imc DATA API** in the installation package under \Products\Development\Setup\imc_DATA_API

- Setup_imcDataApiDeveloper.exe
- Setup_imcDataApiRuntime.exe

17.3.3 Collaboration with the imc DATA API

Note License

To use the imc DATA API, also the **imc API Developer** or **imc API Runtime** license is required.

Reference

You will find more information in the imc API manual.

17.3.4 Specifying the imc STUDIO version

Warning

In the development process it is recommended that there is only one imc STUDIO version installed to avoid conflicts with the assemblies.

When using the completed application also parallel installation is possible.

The interface attempts to automatically detect the version of imc STUDIO Version and enters it into the configuration file default.ispc 1948..

The following steps are necessary in order to use a particular version of imc STUDIO in the imc STUDIO API:

In the **default.ispc**-file used, for each entry in the assembly, the corresponding version number must be entered. This number can be obtained from the application path of the corresponding imc STUDIO version.

In the Initialization 1949 the parameter correctVersionInISPC must be set to `false`. Otherwise the default.ispc is overwritten.

Product config file: default.ispc

By default, the **default.ispc** for the API is located in the following folder: C:\Users\Public\Documents\imc\imc STUDIO\API\Projects\AppDir

Notes on default.ispc

This file is loaded when an instance of imc STUDIO is started. The content indicates which components are to be loaded.

Along with the information whether a component is active, the file also contains the version number. This number must match the version number of the imc STUDIO installed; non-matching versions will cause the start to be canceled.

Therefore it is necessary to perform manual verification and, if applicable, alteration of the version:

1. Version of imc STUDIO installed:
   a. see file properties of **imc.Studio.Version.dll** in the imc STUDIO installation folder
2. Version specification in the file **default.ispc**

   a. To edit, the file can be opened in an XML- or text editor.
   b. The installed version of imc STUDIO must be indicated in the XML-tag **Version**: e.g. the plug-in **imc.Studio.PlugIns.Panel.dll**
3. If the versions don't match or if there is no version indicated:
   a. Change the text of the version within the tag <Version>**xxx**</Version>
   b. save **default.ispc**
   c. restart the program

Example of a plug-in component in default.ispc

```
<PlugIn>
      <PlugInType>DotNet</PlugInType>
      <StartUpClass>imc.Studio.PlugIns.Panel._ThePlugIn</StartUpClass>
      <Culture>neutral</Culture>
      <PublicKeyToken>68a4b1d388e6c0b6</PublicKeyToken>
      <AssemblyName>imc.Studio.PlugIns.Panel</AssemblyName>
      <Active>True</Active>
      <Version>23.0.1.6502</Version>
      <LoadPositionNumber>13</LoadPositionNumber>
 </PlugIn>
```

17.3.5 Base interface

The project **imc.Studio.ApplicationInterface** is considered the Basis-project.

imc STUDIO has a core (imc STUDIO Core) having a framework onto which all functionalities sign on.

These functionalities only become operable in consequence of referencing various assemblies.

In order to make it easier to gain familiarity, the basis-project contains the most important references and provides simplification of these functionalities.

Note

- imc makes the source code of the basis-project available. It is highly recommended **not** to alter it. In future versions, the scope of functionalities will be expanded. Any changes would be lost upon updating and/or would need to be reconstructed.
- What is recommended is to create **one's own** imc STUDIO-class which is derived from `SimplifiedImcStudioComponent`. Here, it is possible to make any enhancements desired.

17.3.5.1 Initialization

Parameter

For the **initialization** of the imc STUDIO components, the following **parameters** must be provided:

```
imcStudioObject = new SimplifiedImcStudioComponent(<application folder>, <parent window>, <installation folder> )
```

Parameter	Description
Application folder	• In the standard case, imc STUDIO is started from its application folder, e.g. "C:\ProgramData\imc\imc STUDIO\Applications_1". This is where the configuration is saved in **default.ispc** and loaded. • When setting a new folder for your application, the folder "AppDir" (application settings) from the Demo-projects belonging to imc STUDIO can be used. • For the purpose of modifying the components used, the file **default.ispc** can be copied from the STUDIO-applications folder (once all all components to be used have been activated in the imc STUDIO user interface). • Alternatively, for testing purposes a complete copy of the folder ".._1\" can be used. • To check which application folder is used by the current installation of imc STUDIO, have the properties of imc STUDIO's desktop link displayed.
Parent window	• Necessary for displaying dialogs from the imc STUDIO component • Necessary for controlling the multi-threading within imc STUDIO. • A *Windows.Forms* or *WPF*-window can be transferred.
Installation folder	• To start imc STUDIO, the installation folder must be provided as a parameter. • By default, imc STUDIO installs in "C:\Program Files\imc\imc STUDIO 2023\" (or according to the respective version). • The path can be determined automatically by means of the following function: `SimplifiedImcStudioComponent.GetImcStudioInstallPath();`

Optional parameters

Parameter	Description
initialize	Initializes the imc STUDIO core during the constructor. If `false` is specified here, this will have to be rectified manually using the `Init()`-method.
applicationLoaded	Triggers the "imc STUDIO loaded" event.
correctVersionInISPC	Overwrites the imc STUDIO configuration file (default.ispc), so that all plug-ins specified have the same STUDIO core.
wpfApplicationThemeName	Name of the WPF theme. For the default imc STUDIO theme, use an empty string. For the default WPF theme, use "None".
setImcDevExpressWinFormSkin	Aktiviert bzw. deaktiviert den DevExpress-Skin für WinForms-Anwendungen

Note Initialize

If you want to initialize the component later, the applicationLoaded parameter must also be set to false in the constructor.

```
ImcStudioComponent = new SimplifiedImcStudioComponent(appDir, this, strImcStudioInstallPath, false,
false);
ImcStudioComponent.Init();
ImcStudioComponent.ApplicationLoaded(this);
```

17.3.5.2 Overview

Here, the main functions are described.

All objects themselves come with notes on their respective functionality and call parameters.

Basis class: ImcStudioComponent	
Actions	Triggers the actions registered by the plugins; see *imc.Studio.ApplicationInterface.Constants.ActionNames*
Commands	for running commands
Documents	Access to applications-, project- and experiment settings and supply of supply of events for loading and saving
Environment	Gets global imc STUDIO variables (e.g. installation path)
Events	Registration of EventHandlers, e.g. for the event "device started", "device stopped"; see *imc.Studio.ApplicationInterface.Constants.EventNames*
InlineFamos	Configuration of an InlineFamos task (requires appropriate license).
Logbook	Access to logbook entries
Notifications	Internal communication between imc STUDIO components
Panel	Access to Panel pages
Scripts	Access to imc STUDIO scripts
Variables	Access to measured data, or all data from a variety of sources. Generating one's own variables. Access to the IVariablesManager by using "Variables.Base"

Class: SimplifiedImcStudioComponent : ImcStudioComponent	
DevSetup	*DevSetupExtended* Functionality for, among other things, configuration of devices and channels (enhancement of the Basis-class's DevSetup-functionality)
Devices	*DevicesActions* Devices actions such as device search, connect, disconnect
Experiment	*ExperimentActions* Actions affecting the experiment such as Load, Save
Measurement	*MeasurementActions* Actions associated with the measurement such as Start, Stop
Windows	*WindowsExtended* Calling of dialogs and views present in imc STUDIO

Reference

Usage and functionality of these classes are identical as in imc STUDIO Scripting. For details and example use cases, please see the documentation on Scripting 1808.

Note

The above classes provide access to frequently used functionalities by encapsulating interfaces. Access to further functionalities is partially possible by manual encapsulation. The approach is based on the following naming pattern using *Core* as an example:

`Core.Base` is of type `ICoreBase`. Therefore you need to get an `API_Core_Vx` interface via `GetAPI()`. The letter **x** stands for the interface's version.

Example code:

```
API_Core_V2 coreApi = Core.Base.GetAPI<API_Core_V2>();
```

17.3.5.3 Actions

The following actions are available in the Basis-project:

Actions in imc.Studio.ApplicationInterface.Constants	
ActionNames	various menu actions
DeviceActions	Connect, Disconnect ProcessConfiguration, Download, Reconfigure ReleaseAllTriggers ResumeDataStorage, SuspendDataStorage Start, Stop

17.3.5.4 Misc functions

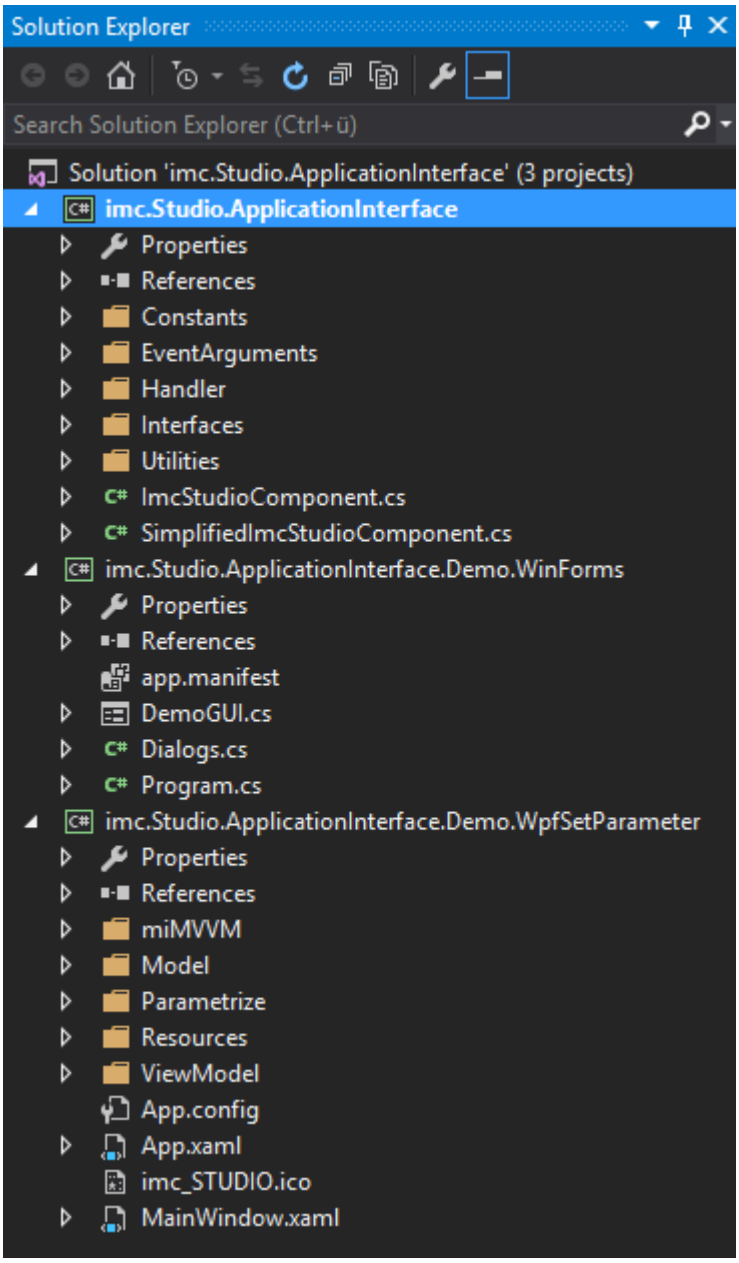
imc.Studio.ApplicationInterface.Constants	
DocumentNames	Collection of constants for accessing documents.
EventNames	Collection of event names.
SharedComponentNames	Collection of shared component names.
WindowNames	Collection of window names.

17.3.6 Sample projects

The example projects are installed in the following folder:

C:\Users\Public\Documents\imc\imc STUDIO\API 2023\Projects

Projects	Description
WinForms 1953	Simple GUI (Windows Forms) for using a device and configuring channels as well as displaying the channels in the curve window. Solution: imc.Studio.ApplicationInterface
WpfSetParameter 1955	Simple GUI (WPF) for using a device. Solution: imc.Studio.ApplicationInterface
WpfCalculate 1955	Simple GUI (WPF) for using a device and calculation of channels Solution: imc.Studio.ApplicationInterface.Demo.WpfCalculate



Solution Explorer (Visual Studio 2015)

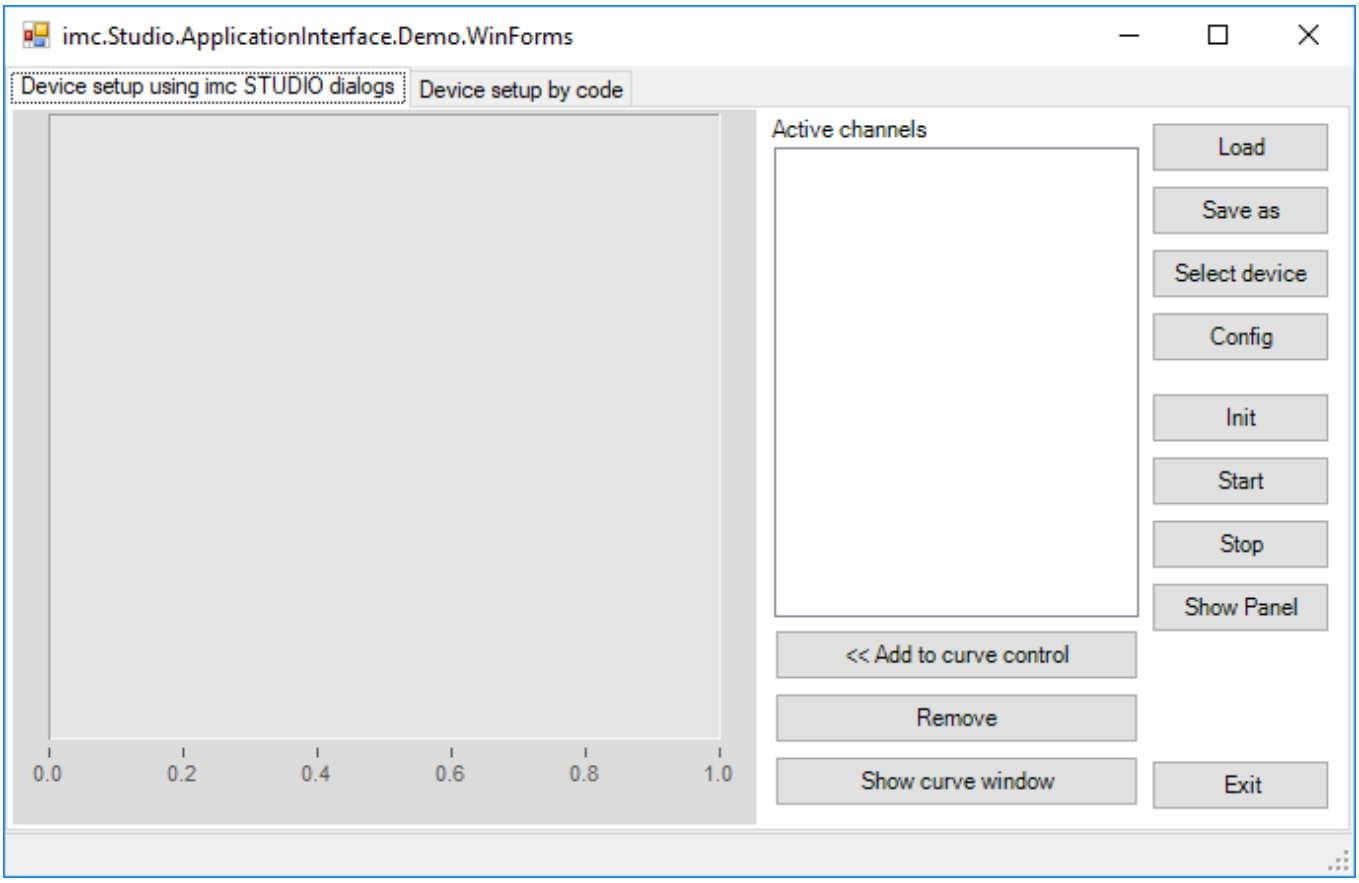
17.3.6.1 WinForms

Warning License required

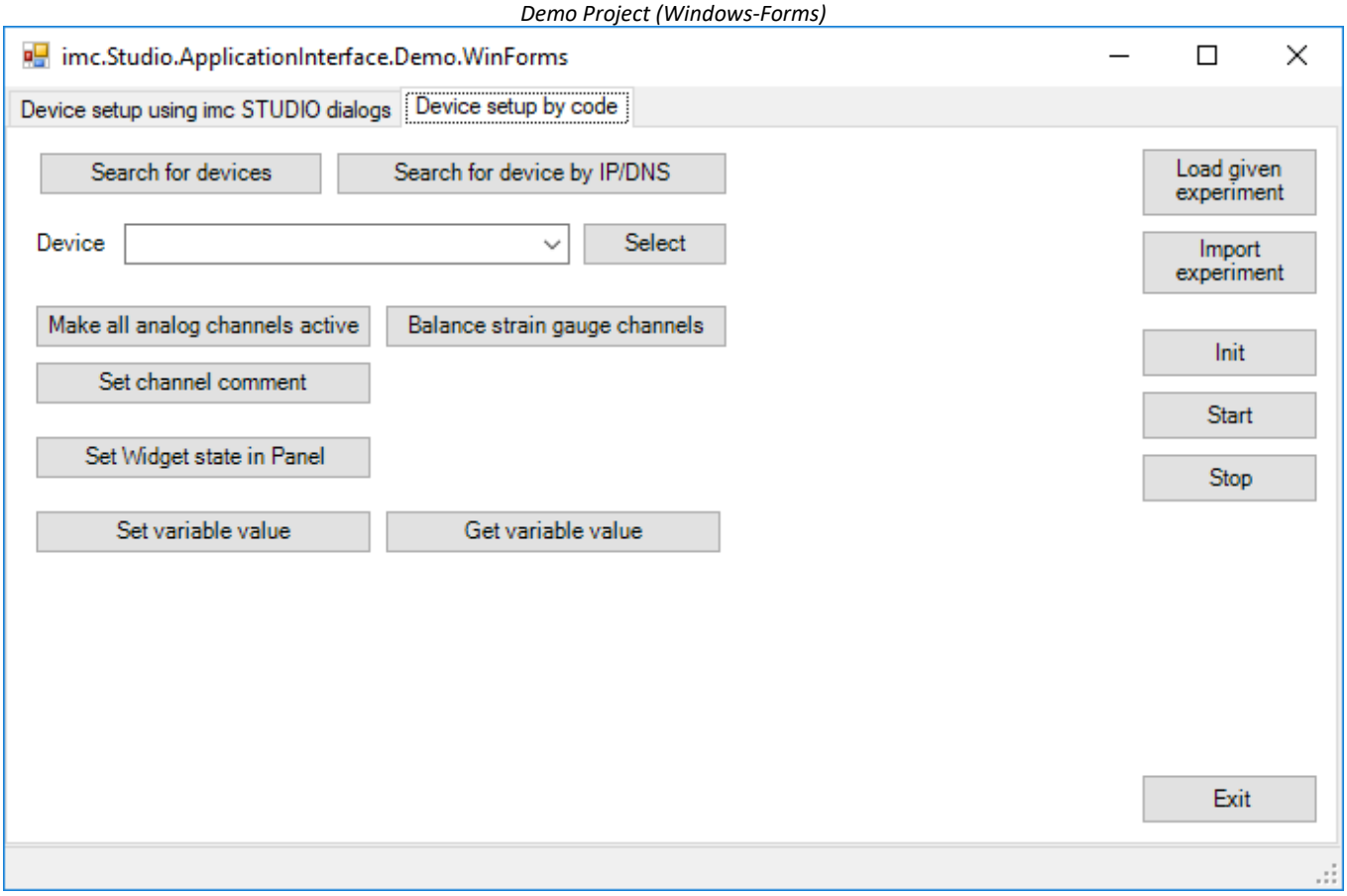
To use the project in its entirety, a **imc API Developer** or **imc API Runtime** license is required.

imc.Studio.ApplicationInterface.Demo.WinForms

- Implementation in C# and WinForms
- Example of small-scope application for controlling a measurement.
- Calling of various dialogs for device searching and selection of devices, as well as the Setup-dialog for setting parameters.
- Start/Stop of measurement and visual output of data.



Demo Project (Windows-Forms)



Demo Project (Windows-Forms)

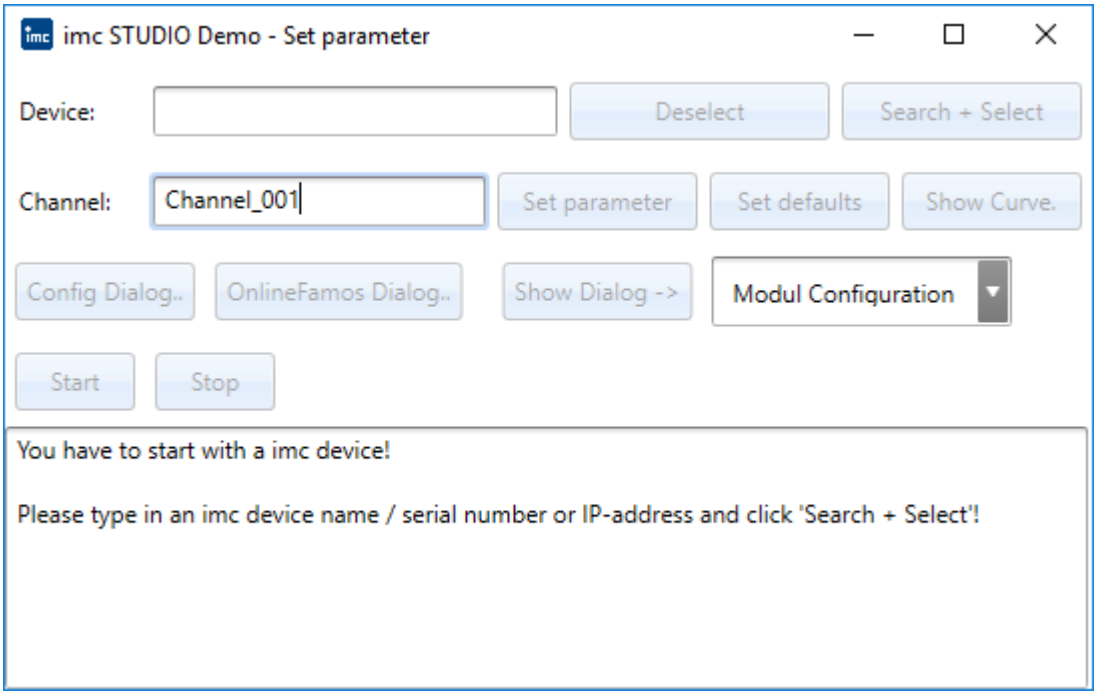
17.3.6.2 WpfSetParameter

Warning License required

To use the project in its entirety, a **imc API Developer** or **imc API Runtime** license is required.

imc.Studio.ApplicationInterface.Demo.WpfSetParameter

- Implementation in C# and WPF
- Example of parameterization on command, without manual parameter entry.



Demo Project (WPF)

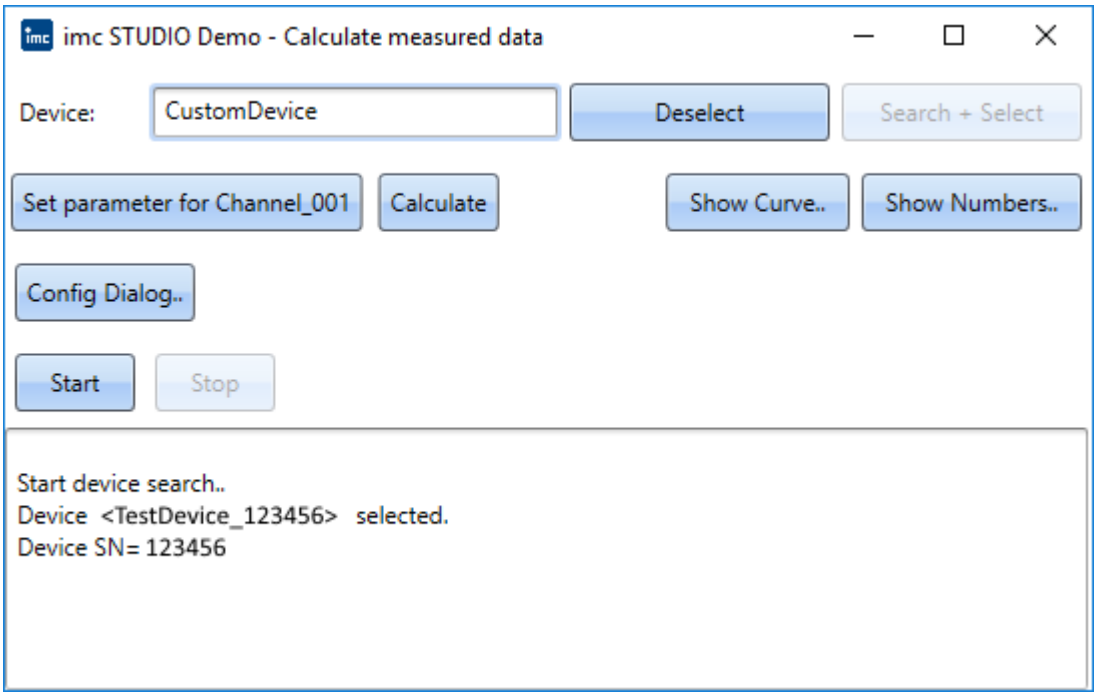
17.3.6.3 WpfCalculate

Warning License required

To use the project in its entirety, a **imc API Developer** or **imc API Runtime** license is required.

imc.Studio.ApplicationInterface.Demo.WpfCalculate

- Implementation in C# and WPF
- Calculation of minimum, maximum and mean value.



Demo-Project (WpfCalculate)

17.3.7 User-created application

When creating one's own personal project (Windows.Forms or WPF), the following **references** must be incorporated.

The references are in the Global Assembly Cache (**GAC**), also see *C:\Windows\Microsoft.NET\assembly\GAC_MSIL.*

You need to compile the **Shared-Project** in *C:\Users\Public\Documents\imc\imc STUDIO\API\Shared* for getting the imc.Studio.ApplicationInterface.dll.

References	Content
imc.Studio.ApplicationInterface	Entry point for imc STUDIO (*in the Shared-Project*)
imc.Common.Interfaces	General imc components (Logbook,..)
imc.Interfaces	EClassID parameters for configuring device and channel parameters in DevSetup class.
imc.Studio.Interfaces	General imc STUDIO components
imc.Common.Components.DataObjects	Working with data
imc.Common.Controls.ImcCurves	Display of data

References	Content
PresentationFramework	Required by the SimplifiedImcStudioComponent-class
System.Windows.Forms	For use with WinForms

Namespaces	Content
imc.Studio.Tools.DevSetupGraphGenerator	EClassID (for parameterizing devices)
imc.Common.Components.DataManager imc.Common.Components.DataManager. Interfaces	Data objects such as DmChannel, DmFile, IDmChannel, IDmText.
imc.Studio.ApplicationInterface.Constants	Various actions and constants for parameterizing, e.g. ParameterValues

An example of Windows-Forms or alternatively WPF-project is presented here 1957.

Reference

For additional information also see the documentation on Scripting 1808.

17.3.7.1 Example

Initialization example of the imc STUDIO components

In the example code (C#), only the main functions are displayed without any inspection or error handling.
There are differences when calling imc STUDIO windows depending on whether the calling program is a WinForms- or a WPF-application.
The main difference is the basis class of the view (System.Windows.Forms or System.Windows.Window).

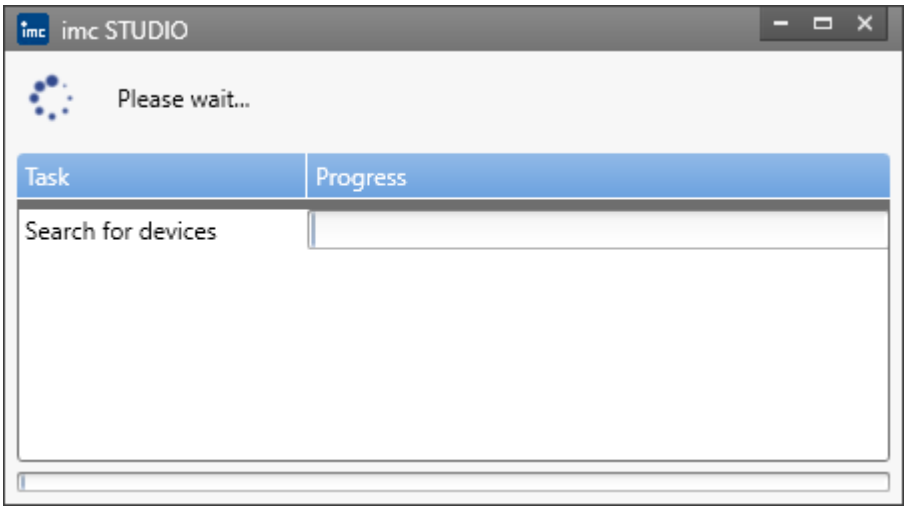
WPF application

```
using imc.Studio.ApplicationInterface;

public partial class WPFMainWindow : Window
{
   private SimplifiedImcStudioComponent _component;

   public WPFMainWindow ()
   {
      // Application path for this imc STUDIO application
      // imc Studio user interface has its own completely separate path for this!
      string appPath = @"C:\ProgramData\imc\imc STUDIO\Applications\AppDir";

      // Complete path for imc STUDIO installation which is to be used,
      // or zero if imc STUDIO and the main application are located in same folder
      // Example:
      string imcStudioInstallPath = @"C:\Program Files\imc\imc STUDIO 2023";

      // Generating the imc STUDIO components instance
      _component = new SimplifiedImcStudioComponent(appPath, this, imcStudioInstallPath);
   }
   protected override void OnClosing(CancelEventArgs e)
   {
      // Prompt whether to save before closing the application
      _component.ClosingImcComponent(this, e);
      if (!e.Cancel)
      {
         base.OnClosing(e);
      }
   }
 protected override void OnClosed(EventArgs e)
   {
      // Cleanup upon closing application
      base.OnClosed(e);
      _component.Dispose();
      _component = null;
   }
}
```

WinForm-application

```
using imc.Studio.ApplicationInterface;

public partial class WinFormsMainWindow : Form
{
   private SimplifiedImcStudioComponent _component;

   internal WinFormsMainWindow()
   {
      InitializeComponent();

      // Application path for this imc STUDIO application
      // imc Studio user interface has its own completely separate path for this!
      string appPath = @"C:\ProgramData\imc\imc STUDIO\Applications\AppDir";

      // Complete path for imc STUDIO installation which is to be used,
      // or zero if imc STUDIO and the main application are located in same folder
      // Example:
      string imcStudioInstallPath = @"C:\Program Files\imc\imc STUDIO 2023";

      // Generating the imc STUDIO components instance
      _component = new SimplifiedImcStudioComponent(appPath, this, imcStudioInstallPath);
   }
   protected override void OnClosing(CancelEventArgs e)
   {
      // Prompt whether to save before closing the application
      _component.ClosingImcComponent(this, e);
      if (!e.Cancel)
      {
         base.OnClosing(e);
      }
   }
  protected override void OnClosed(EventArgs e)
   {
      // Cleanup upon closing the application
      base.OnClosed(e);
      _component.Dispose();
      _component = null;
   }
}
```

17.3.7.2 ProgressVisualizer



Progress-Dialog (ProgressVisualizer)

When you execute actions that exceed a certain duration, a progress dialog called ProgressVisualizer opens.

In your own applications, it is therefore useful to hide this and transfer the information to your own status element.

```
using imc.Common.Interfaces.Logbook;
using imc.Studio.ApplicationInterface;
using imc.Studio.Interfaces.Core.Services.ProgressVisualizer;
```

```
private SimplifiedImcStudioComponent _component;
```

```
private void HideProgressVisualizer()
{
   var services = _component.Core.Services.GetAPI<API_CoreServices_V2>();
   var progressVisualizer = services.ProgressVisualizer.GetAPI<API_CoreServicesProgressVisualizer_V3>();
   progressVisualizer.GUIEnableHideTemporary();

   progressVisualizer.TaskProgress += OnProgressVisualizerTaskProgress;
}
```

```
private void OnProgressVisualizerTaskProgress(object sender, ICoreServicesTaskEventArgsBase e)
{
   var taskEventArgs = e.GetAPI<API_CoreServicesTaskEventArgs_V1>();

   ProgressTask(taskEventArgs);
}
```

```
private void ProgressTask(API_CoreServicesTaskEventArgs_V1 taskEventArgs)
{
   _component.Logbook.LogEntry(ErrorSender, taskEventArgs.Text, 0, ELogbookEntryCategory.Information);
}
```

17.3.7.3 Experiments

Activated project management

Experiment Name

```
string experimentName = _component.ProjectManagement.CurrentExperiment.Name;
```

Load experiment

When loading experiments, the path relative to the database must be specified. This includes the project name.

```
var experimentPathInDatabase = @"StandardProject\Experiment_0001";
var result = studioComponents.Experiment.Load(experimentPathInDatabase);
```

Import experiment

When importing an experiment, the full path of the experiment file as well as the project name must be specified.

```
var projectName = "StandardProject";
var experimentFileName = @"c:\temp\Test.imcStudio";
var result = _components.Experiment.Import(projectName, experimentFileName);
```

Save experiment

```
_components.Experiment.Save()
```

Save experiment as

Variant 1:

```
var experimentFileName = @"c:
\Users\Public\Documents\DB\StandardProject\Experiment_0001\Experiment_0001.imcStudio;
_component.Experiment.Save(experimentFileName);
```

Variant 2:

To use the ProjectManagement class, the following references must be added to the C# project.

This can also be done in Visual Studio via the tool-tip and "Show potential fixes".

- imc.Common.Interfaces (für IExperiment_V1)
- imc.Studio.Interfaces (für IProjectManagementBase, API_ProjectManagement_V7)

```
var currentExperiment = _component.ProjectManagement.CurrentExperiment;
currentExperiment.Name = "NewExperimentName";
component.ProjectManagement.ExperimentSaveAs(null, currentExperiment, false);
```

Save as dialog

```
_component.Experiment.SaveAs()
```

Without project management

Load experiment

```
var experimentFileName = @"c:\temp\Test.imcExp";
var result = _component.Experiment.Load(experimentFileName);
```

Save experiment

```
_components.Experiment.Save()
```

Save experiment as

```
var experimentFileName = @"c:\temp\Test.imcExp;
_component.Experiment.Save(experimentFileName);
```

17.3.7.4 Curve window

Note License activation

For using the curve window control in your development environment you need to activate a **imc API Developer** license.

If you wish to use the curve window on one of your user interfaces (Windows.Forms, WPF, ...), perform the following necessary steps.

Adding the curve window to the toolbox

The instructions below demonstrate how to add the curve window to the toolbox on the basis of Visual Studio as an example:

1. Open the toolbox tool window (while your GUI designer is open)
2. From the context menu (opened by right-clicking), select "Choose Items…"
3. Select the tab ".NET Framework Components"
4. Click on the Browse button
5. Select the DLL from one of the following two paths:
   C:\ProgramData\imc\imc DATA API**7.3\Rx**\Assemblies\Release\x86\ imc.Common.Controls.ImcCurves
   C:\ProgramData\imc\imc DATA API**7.3****Rx**\Assemblies\Release\x64\ imc.Common.Controls.ImcCurves
   Rx is to be replaced with the revision number, e.g. "R1".
6. Two new entries will appear in the list: CimcCurveControl and CimcCurveGlobalSettings.
7. Leave the check mark for both of these and close the dialog with OK.
8. Subsequently, the curve window object will be available in the toolbox.

Usage in Windows.Forms projects

In Windows.Forms-projects, it is possible to use Drag&Drop to move the curve window-object in the Designer directly into the GUI.

Usage in WPF projects

1. In WPF-projects, the following references must be added to the project:

- System.Windows.Forms
- WindowsFormsIntegration
- imc.Common.Controls.ImcCurves

2. In the Designer, the following namespace is used in the xaml-code:

```
xmlns:imc="clr-namespace:imc.Common.Controls.ImcCurves;assembly=imc.Common.Controls.ImcCurves"
```

3. Within the grid object, the WindowsFormsHost object containing the curve window is added:

```
<WindowsFormsHost>
   <imc:CimcCurveControl x:Name="mainCurveWindow" />
</WindowsFormsHost>
```

4. The curve window-object is not displayed visually, but it can be addressed in the code by its associated name.

Show data in a curve window

To display channels in a curve window (`mainCurveWindow`), the data object of the channel is needed.

The following shows how all Analog Channels are displayed in the curve window:

```
using imc.Studio.ApplicationInterface;
using imc.Common.Components.DataManager.Interfaces;
using imc.Common.Controls.ImcCurves;
using imc.Common.Controls.ImcCurves.Interfaces;
using imc.Common.Controls.ImcCurves;
using imc.Data.Core.Contract;
```

```
private SimplifiedImcStudioComponent _component;
```

```
public void ShowVariableInCurve(IVariable variable)
{
   var result = variable.QueryInterface<IDmChannel>();

   if (result.IsSuccessful)
   {
      var channel = result.Value;

      mainCurveWindow.Clear();
      mainCurveWindow.AppendChannel(channel, CwAppendConstants.AppendNewLine, 0);
   }
}
```

17.3.7.5 Commands

Import an OFA file

To import an imc Online FAMOS (OFA) file, the command "Import supplemental file" is used.

```
using imc.Studio.ApplicationInterface;
```

```
private SimplifiedImcStudioComponent _component;
```

```
/// <summary>
/// Imports an *.ofa file to a given device.
/// </summary>
/// <param name="fileName">OFA filename.</param>
/// <param name="deviceName">Device identifier (eDeviceName); e.g. imcDev__18123456.</param>
private void ImportOfaFile(string fileName, string deviceName)
{
   _component.Commands.Invoke("Die Zusatzdatei 'C:\\\\Training\\\\API_2018-11\\\\Misc\\\\TestCalc.ofa'
wird importiert",
          "<AppPlotFactoryCmd version=\"1\">" +
          "<common version=\"1\" cuid=\"ImportImcDevicesSatelliteFile\" caption=\"Zusatzdatei
importieren\">" +
          "<command version=\"1\"><properties version=\"1\">" +
          "<cmd owner=\"imc.Studio.PlugIns.AppPlot.Engine.AppPlotImportSatelliteFile\" version=\"2\"
cuid=\"ImportImcDevicesSatelliteFile\">" +
          "<filename>" + fileName + "</filename>" +
          "<overwrite>eOverwriteWithoutPrompt</overwrite><targetname />" +
          "<map><devsatimportmap>  " +
          "<devicename>" + deviceName + "</devicename>  <used>True</used></devsatimportmap>" +
          "</map></cmd></properties></command></common></AppPlotFactoryCmd>");
}
```

17.3.7.6 Windows (WinForms)

General

The desired window/control *windowControl* is added to the WinForm *form*:

```
using (var form = new Form())
{
    form.Width = windowUi.Width;
    form.Height = windowUi.Height;
    form.Text = "WindowName";
    form.Controls.Add(windowControl); // whereas windowControl is the desired control
    form.Dock = DockStyle.Fill;
    form.ShowDialog();
}
```

Logbook

```
private void ShowLogbook()
{
   var logbookUi = _component.Windows.GetWindowControl("Logbook").AsWinFormsControl();
   logbookUi.Width = 1200;
   logbookUi.Height = 500;
   logbookUi.Dock = DockStyle.Fill;

   using (var logbookForm = new Form())
   {
       logbookForm.Width = logbookUi.Width;
       logbookForm.Height = logbookUi.Height;
       logbookForm.Text = "Logbook";
       logbookForm.Controls.Add(logbookUi);
       logbookForm.Dock = DockStyle.Fill;
       logbookForm.ShowDialog();
   }
}
```

Setup pages: Devices

```
private void ShowDevicesPage()
{
   var devicesPageName = "Devices";
   var setupControl = _component.Windows.CreateSetupControl(devicesPageName);

   using (var devicesForm = new Form())
   {
       devicesForm.Width = 1000;
       devicesForm.Height = 800;
       devicesForm.Text = devicesPageName;
       devicesForm.Controls.Add(setupControl);
       setupControl.Dock = DockStyle.Fill;
       devicesForm.ShowDialog();
   }
}
```

Panel

```
private void ShowPanel()
{
   var panelControl = _component.Windows.Panel.AsWinFormsControl();

   if (panelControl == null)
   {
       MessageBox.Show(@"Panel was not found. Make sure that the Panel plug-in is properly loaded!");
       return;
   }

   using (var panelForm = new Form())
   {
       devicesForm.Width = 1000;
       devicesForm.Height = 800;
       devicesForm.Text = @"Panel";
       devicesForm.Controls.Add(panelControl);
       setupControl.Dock = DockStyle.Fill;
}
```

17.3.7.7 Windows (WPF)

Logbook

```
private void AddLogbookToWindow()
{
   var logbookElement = CreateStudioLogbook(_studioComponent);
   Logbooks.Children.Add(logbookElement); // Logbooks is a StackPanel element
}
```

```
private static FrameworkElement CreateStudioLogbook(SimplifiedImcStudioComponent studioComponent)
{
   var logbookAsWpf = _cComponent.Windows.GetWindowControl("Logbook").AsWPFControl();

   var logbookElement = (logbookAsWpf as FrameworkElement);
   logbookElement.LayoutTransform = new ScaleTransform(0.8, 0.8);
   logbookElement.MaxWidth = 1024;
   logbookElement.MaxHeight = 800;
   return logbookElement;
}
```

Setup Seiten: Devices

```
private void ShowSetupPage()
{
   var devicesPageName = "Devices";
   var deviceUi = _component.Windows.CreateSetupControl(devicesPageName).AsWPFControl();
   deviceUi.Width = 1400;
   deviceUi.Height = 700;

   var window = new Window()
   {
       Width = deviceUi.Width,
       Height = deviceUi.Height,
       HorizontalContentAlignment = System.Windows.HorizontalAlignment.Stretch,
       VerticalContentAlignment = VerticalAlignment.Stretch,
       Content = deviceUi
   };

   window.ShowDialog();
}
```

Panel

```
private void ShowPanelAsWpf()
{
    var panelControl = _component.Windows.Panel.AsWPFControl();

    if (panelControl == null)
    {
        MessageBox.Show(@"Panel was not found. Make sure that the Panel plug-in is properly loaded!");
        return;
    }

    Window w = new Window
    {
        Width = 1000,
        Height = 800,
        Content = panelControl
    };

    w.ShowDialog();
}
```

17.3.8 Distribution

In order to transfer the project you have developed, the following prerequisites must be met by the target computer:

- imc STUDIO is installed
- depending on the functional scope, activation of the appropriate licenses
- Activation of imc API Runtime license
- Deployment of the desired project
- Application folder (including default.ispc)

17.3.9 Notes

Tips

It is recommended to use imc STUDIO to make the project settings (such as views, Setup-pages, ...) by using the API-database path.

Curve window

For using the imc curve window, an activated license for **imc API Developer** is required.

FileNotFoundException: imc.Common.Components.DataObjects.dll

If you start the examples of imc API directly after the installation of imc STUDIO, a FileNotFoundException may occur.

In this case please restart the PC and try again.

18 Miscellaneous

18.1 Last Changes

18.1.1 in Doc. Rev. 6.4

Setting Up - Software

Section	Amendments
Select the products / Installation type	imc SENSORS is no longer pre-selected in any installation variety.
Command line parameters Recommended virus scanner settings	The exe-file of any program (imc STUDIO, imc WAVE, imc STUDIO Monitor) is now located in the respective installation folder and no longer in the folder "*ProgramData*". Furthermore, the file is provided with a signature.

Setting Up - Connect the device

Section	Amendments
Associated firmware (imc DEVICES)	The term "Firmware group" had been used in two meanings.
The network	The example of a complex network with devices belonging to firmware group B has been expanded.

imc STUDIO (general)

Section	Amendments
Data management	"*.Prepare*"-files in the folder "*config*" now included in the description.
Logbook 144	New function: Export logbook-files Function eliminated: Send logbook via E-Mail
Options > Multi-lingual text input 130	The description text was missing.

Setup - Device (general)

Section	Amendments
Transferring an experiment to other devices Connect and disconnect 190	Note added concerning devices of Firmware-group B, indicating what constraints to observe when transferring a configuration from one device to another.
Disconnect/Connect - with a measurement running 195 Creating a Diskstart configuration 208	No longer applicable text deleted: When re-connecting with a running autostart experiment located on a removable hard drive, the connection is only possible if the experiment is present on the PC. The text contained a note stating that in this case it would work even without any experiment.
Disconnect/Connect - with a measurement running 192	More information about connecting to a running measurement from another PC.
Device properties 221	imc REMOTE SecureAccess has been removed from the device properties. The obsolete TLS 1.0 technology is no longer utilized.
Metadata	Saving metadata with the channel Abgleich-Informationen können nun auch als Metadaten im Kanal gespeichert werden. Über die Option "*Setup*" > "*Traceability of data*" steht die neue Auswahl "*Balance information*" zur Verfügung.

Setup pages - Configuring device

Section	Amendments
Sampling rate and the Aggregate sampling rate	Info added: Active monitor channels are included in the calculation of the aggregate sampling rate at the reduced rate.
Trigger and events - Hysteresis 395	Info added: Automatic adjustment of the hysteresis to the measurement range
Sampling & Preprocessing	New reduction method for imc ARGUS*fit*: Monitor channel as envelope type
Trigger and events - Sources and events 389	Note added: Channels having the preprocessing "*Envelope*" can not serve as a trigger source.
Synchronization 292	imc ARGUS*fit*-devices can now be synchronized via GPS by means of a connected GPS-mouse.
Device hard disk, removable drive - Formatting 445	Tool for formatting removable data volumes for imc ARGUS*fit*-devices described.

Data analysis and signal processing

Section	Amendments
Bus Decoder - Decoding fieldbus log channels	Firmware groups added to "Supported fieldbus types" table.
Inline Analysis - Monitoring the overmodulation and under range 1087	New: Monitoring the under range

imc Online FAMOS and imc Inline FAMOS

Section	Amendments
Comparison operators	The more detailed example from the function reference is used.

imc Online/Inline FAMOS function reference

Section	Amendments
DelayBuffer	New example: Delay measuring channel
OnInitAll	New note: OnInitAll is not necessarily performed when the user clicks "*Prepare*".

Channel types, variable types, data types and data storage formats

Section	Amendments
Process vector variables (pv-variables) 1141	Info added: Deactivation of pv-variables for data recording channels is not possible for devices belonging to the firmware group A.
Data types	Data type: Text arrays from imc FAMOS
Envelope type monitor channel (min-/max-channel)	The functioning of the data reduction in envelope-type monitor channels for imc ARGUS*fit* is described.

Parameter set > Mapping
Import parameters > Mapping
	• "*Table description*" from the mapping declaration "*by connector*" has been removed: The table description is not relevant to the mapping.
	• The mappings have been renamed:
	  • "*Import of channels according to connector and device serial number*" → "*Multiple-device operation (Firmware group A)*"
	  • "*Import of channels according to connector*" → "*Single-device operation (Firmware group A)*"
	• New mapping: "*Uniformly per module type*"

Panel

Section	Amendments
Page printout or print to PDF	More detailed description of the "*Report template*"
Fullscreen	Note added: Moving the fullscreen view to a different monitor:
Load/Refill variable 1222	Note revised: Previously, there was a mention that with "*Load Variable*", only user-defined variables having the scope "*temporary*" can be overwritten. However, all other user-defined variables can also be overwritten.
Graphical switch	The description of the widget "*Graphic switch*" is now available.
Preview in Windows Explorer 1520	The preview in the Microsoft Explorer is now independent of the default program linked. For this reason, RAW-data are now displayed correctly in the preview, even if RAW is linked with an image processing program.
Multi-lingual text input	The option is now deactivated by default and can be activated as needed.

Automation

Section	Amendments
General event imc FAMOS automation (data cutting) Page through Panel	Note added regarding interactions in the "*Terminate*" block.
Repository	Saving the repository in the file "imcCommonUserRepository.asr"

Sequencer, events and commands

Command Reference

Section	Amendments
Execute device action	Note added, for the action "*Start*": No calculations or analyses performed on the PC side, such as imc FAMOS-calculations, are started. Only the devices are started.
Set variables	• Example of toggling variables added. • The Unequal-operator has been included in the description: !=
Load/Refill variable	Note revised: Previously, there was a mention that with "*Load Variable*", only user-defined variables having the scope "*temporary*" can be overwritten. However, all other user-defined variables can also be overwritten.

imc STUDIO GoPro

Section	Amendments
Running the measurement	Not on maximum video file size added.
Startup operations	• Note added on the following problem: When connecting via Bluetooth, the error 20000332 would repeatedly appear. • Note added indicating that the device must be visible in the WLAN.
Running the measurement	Note added indicating that one measurement channel must be governed by an immediate trigger.

18.1.2 in Doc. Rev. 6.3

Setting Up - Software

Section	Amendments
Device Connection / Network / Firewall	The firmware programs are automatically allowed in the "*Windows Defender Firewall*" upon installation. For this reason, popup dialogs about the firewall no longer appear upon first connecting to the device.
Installation - Step by step	Description of how to download and start the Installer-file.

imc STUDIO (general)

Section	Amendments
Setting access rights for actions	Note on the right: "Save experiment
Logbook 144	Copying entries, including supplemental information
Options 134	New option described: Traceability of data

Setup - Device (general)

Section	Amendments
imc CANSAS 210	In some cases, the unit of imc CANSAS channels is not displayed correctly.
Metadata	New function described: Saving metadata with the channel

Setup pages - Configuring device

Section	Amendments
Events and circular buffer 422	New section summarizing the settings for Events and Circular buffer.
Trigger and events Hysteresis 395	Effect of the hysteresis on the measurement start.
Exchanging values among the devices	For exchanging or processing values across devices, there are a variety of approaches and methods. The collection of these is now presented here.
imc SIMPLEX - RFID tag - link/release with the imc SIMPLEX app 749	New
imc SIMPLEX - Opening the Web- and App-Assistant 745	QR codes that link to the app stores.
imc SIMPLEX - Administration 743	Initial setup of the database and of the administrator using the license card.
Fieldbusses 474	The description of the fieldbuses has been moved to the section "*Setup pages - Configuring device*".
CAN-bus interface 514	Description of the new CAN FD fieldbus assistant for imc ARGUS*fit*.

Data analysis and signal processing

imc Online FAMOS and imc Inline FAMOS

Section	Amendments
User interface	In the imc Online FAMOS Editor's toolbar there is now a button "*Undo*".
Menu	In the menu, the item "*Undo all*" has been added.

Channel types, variable types, data types and data storage formats

Section	Amendments
Parameter set	The selection options for the export of the configuration for the parameter sets have been made more precise.

Panel

Section	Amendments
Preview in Windows Explorer 15520	Preview function in Microsoft Windows Explorer This function is now also available without having to install imc FAMOS. Activation of these functions for *.raw without imc FAMOS is not possible at this time.

Sequencer, events and commands

Command Reference

Section	Amendments
Run Python code file	New command with which to perform Python-analyses

imc STUDIO GoPro

Section	Amendments
The Livestream-Widget	GoPro HERO8 enables display of the livestream in the widget during running measurement.

Scripting

Section	Amendments
Channels	Code-Snippet updated
Inline FAMOS functions	New result type and notes added.
Placeholder	Chapter added
Default dialog response	Retrieve device configuration added.

Third Party Device Interface

Section	Amendments
Single Value	Chapter added.
Ports and Bits	Chapter added.

API

Section	Amendments
Experiments	Current experiment name added.

18.1.3 in Doc. Rev. 6.2

Setting Up - Software

Section	Amendments
Installation variant "User-defined"	Installation steps for imc STUDIO, imc DEVICES and imc DEVICEcore have been merged.
Notes and troubleshooting	Updated "*User Account Control*" images.
Unattended installation - Silent installation	In the step-by-step instructions, the information has been added which indicates that the data carrier's content must first be unpacked by the installer.

Setup - Device (general)

Section	Amendments
Transferring an experiment to other devices	In the "*Open Experiment*"-dialog, the selected can now not only be opened but also transferred to a different device.

Setup pages - Configuring device

Section	Amendments
Sampling & Preprocessing	Detailed description of the "*Preprocessing*".
imc SIMPLEX 742	New product description: imc SIMPLEX enables extremely fast configuration of imc ARGUS*fit*-channels by means of validated sensor settings prepared in a cloud. This massively reduces the risk of errors in configurating a measurement system.
Channel table	Included: Maximum channel count per device in firmware group B (imc DEVICEcore)
Device hard disk, removable drive 443	Separation of the description for firmware groups A and B.
Storage media - Formatting 452	Information on cluster sizes updated.
Filter settings 360	The filter description was extended.

Setup - Advanced Device Functions (imc DEVICES)

imc REMOTE WebServer

Section	Amendments
Activation	Note added that imc REMOTE WebServer permanently occupies a connection.

Channel types, variable types, data types and data storage formats

Section	Amendments
Monitor channels	New: Monitor channels for imc ARGUS*fit* devices
imc Format Converter 1191	Description of the Format converter included in this chapter.

Sequencer, events and commands

Command Reference

Section	Amendments
Transfer device settings	The command "Transfer device settings" is no longer supported.

18.1.4 in Doc. Rev. 6.1

Data analysis and signal processing

imc Online/Inline FAMOS function reference

Section	Amendments
FFTAverage	New example: Peak-Hold spectrum from the beginning of the measurement.

imc STUDIO GoPro

Section	Amendments
All	Newly added

API

Section	Amendments
User-created application	imc.Interfaces added

18.1.5 in Doc. Rev. 6.0

Setting Up - Software

Section	Amendments
Training - Guided first steps	New Assistant described: What steps are necessary in order to obtain measured data?
System requirements	The system requirements have been adjusted.

Setting Up - Connect the device

Section	Amendments
Device connection via LAN	By means of the device interface's configurations dialog, you can directly open the PC's adapter settings.

imc STUDIO (general)

Section	Amendments
Exporting and importing experiments and projects 117	Note: If the export file contains only one experiment, the selection of experiments to import is skipped.

Setup - Device (general)

Section	Amendments
Balance - All active channels 197	New menu item described

Setup pages - Configuring device

Section	Amendments
Storage	Note added, indicating that when the option "*Save trigger events in individual files*" is activated, the column "*Event time*" must be added manually in the Data Browser.
Sampling & Preprocessing	Note added, indicating that results of the function "*TransRec*" and DI-port-compressed channels are blocked from transfer to the PC.
Curve properties	• Parameter renaming implemented. • The description of the parameter "*Range*" has been revised - The choice "*Auto*" now also transfers a range to scale-Widgets. • Description supplemented: Setting the expected signal value range in addition to the measurement range
Range & Scaling	New notes on rounded measurement ranges.

Data analysis and signal processing

Section	Amendments
Inline Analysis - imc WAVE Vibration 1078	The function "*Human vibrations*" has been renamed to "*Filter rms*" and has more functionality.

imc Online/Inline FAMOS function reference

Section	Amendments
TransRec	• The function "*TransRec*" is no longer available imc Inline FAMOS. • Not added, indicating that results of the function "*TransRec*" and DI-port-compressed channels are blocked from transfer to the PC.
TSAGate	New functions for imc Inline FAMOS have been described: TSAGate

Panel

Section	Amendments
Curve window colors on the Panel page 1518	Description adapted: When the colors are used for the display and when for the printout.
Properties - Widget	Range: New behavior described: Channels having a measurement range now also always transfer a display range.

Sequencer, events and commands

Section	Amendments
Order of events in response to actions	Description of some device actions and of the events they prompt.

Scripting

Section	Amendments
Supplemental Files	New chapter
Variablen	• IsPersistent is explained. • Process many variables.
Channels	• Use of channels supplemented. • Note due to channel names added. • Creating channels
Streams	New chapter
Finding parameters and setting values	Hint for finding parameters added
Logbook	New LogbookEntry class is used.
Script options	Example for getting an interface via GetAPI added.
Events, Events	• Hint for register events added. • Event for creating variables added.

API

Section	Amendments
Installation	Reference for imc DATA API added

18.1.6 in Doc. Rev. 5.4

Section	Amendments
Miscellaneous	Some missing components and functions were removed from the documentation: Metadata assistant.

Setting Up - Software

Section	Amendments
Select the products / Installation type	• When you select the installation variant "*Demo*", imc WAVE is also selected and installed. • New overview table of all installation variants and their respective product contents.
Product configuration / Licensing	Removal of the note indicating that the required licenses are shown summarized at the bottom, and that the Automation requires a separate license.
Error causes upon starting	New: Corrective actions for cases where the logbook appears with error messages directly upon starting imc STUDIO.
System requirements	The system requirements have been expanded to include the required and installed OS-components.

Setting Up - Connect the device

Section	Amendments
Connecting via LAN in three steps 46	The dialog for configuring the device's IP-address, also returns the IP-address of the PC. Thus the first step has been eliminated, which presented various ways to find the IP-address in Windows.
Firmware update 64	Description eliminated: "*Calling an update manually*". It is no longer possible to call the Update-dialog for imc DEVICES-devices if no update is needed.

imc STUDIO (general)

Section	Amendments
What is saved where?	Table redesigned for greater clarity.
Experiments, projects and the database 113	The folder "*Meta*" is now created along with the experiment and no longer needs to be created manually.
Data management	File structure updated.

Setup - Device (general)

Section	Amendments
Assistants 240	The dropdown-box for device selection has been eliminated. Instead, all device tabs are displayed from the very beginning.

Setup pages - Configuring device

Section	Amendments
Additional parameters	New: Display of instantaneous values for 3rd-pary device channels, incremental counter inputs, device-variables, etc.
Transfer and storage on the PC 421 Storage in the device 415	Information gathered: Data transfer - Access via Windows Explorer
Access via Windows Explorer 444	Description added: Access to the saved measurement data in the device for devices of the firmware group B (imc DEVICEcore).
Trigger and events - Hysteresis 395	New: The event hysteresis of the devices belonging to the firmware group B (imc DEVICEcore).
Special modules and third party devices	Combining the documentation under the device configuration
Third party device: ChannelLoader 724	Notes on the system's treatment of missing files and relative paths.
Third party device: FunctionSimulator 727	The 3rd-party device "*FunctionSimulator*" is now available as a complete 3rd-party device which can be activated as desired.

Data analysis and signal processing

Section	Amendments
Inline Analysis -Monitoring the overmodulation 1087	Monitoring the overmodulation of "analog channels" (violations of the specified measurement range) as per the standard: "*DIN EN 61672-1*".

imc Online FAMOS and imc Inline FAMOS

Section	Amendments
What happens when imc Online FAMOS overflows?	Error in the example corrected. In the OnSyncTask it is not possible to subject a channel to calculations → adapted to a pv-variable.

imc Online/Inline FAMOS function reference

Section	Amendments
ReduceDataRate	New functions for imc Online FAMOS have been described: ReduceDataRate
FFTAmplitudePhase	Note added: The function FFTAmplitudePhase calculates the peak values when applied in imc Online FAMOS, but in imc Inline FAMOS it calculates the RMS-values.
NOT	Notes adopted from the function description, which indicate that applying the function to such expressions as "If NOT(pv.x > 0) = 0" does not return correct results.

Channel types, variable types, data types and data storage formats

Section	Amendments
imc Data format	Notes on imc2 and imc3 file formats.
Data file format - Saving measured data during the measurement	Note: With which version and which device were the measured data generated?

Panel

Section	Amendments
Operation - Docking	Widget docking is now only possible by means of the Properties and no longer via the context menu.

Scripting

Section	Amendments
Allgemein	EClassID-Parameters are now used directly instead of the string parameters.
Important information	Notes about warnings on compiling added

Third Party Device Interface

Section	Amendments
FunctionSimulator template	The chapter for the FunctionSimulator template was renamed and a chapter for the DLL-Script was added.
Complete third party devices	The description of the complete third-party devices has been moved to the description of the device configuration.

API

Section	Amendments
Installation	• Chapter was renamed • Installation path has changed.
Misc functions	Chapter added

18.1.7 in Doc. Rev. 5.3

Section	Amendments
Miscellaneous	Some missing components and functions were removed from the documentation: video, report channel and navigation bar, among others.

Setting Up - Software

Section	Amendments
Important settings	Another important settings: "*Checking and modifying the time zone*"
System requirements	The list of operating systems supported has been revised: New: Windows 11

Setup - Device (general)

Section	Amendments
Report of the Setup-configuration	The following function is no longer available: Report creation via the Windows Explorer

Setup pages - Configuring device

Section	Amendments
Create a new sensor characteristic curve 736	Chapter on generating a characteristic curve supplemented, including how to generate curves using the command: "*Measure a user-defined characteristic curve*".
Data recording with sampling or with time stamp	The description of data recording has been moved from the section on the Fieldbus to the general information portion.
Setup page: Summary	A new Setup page named "*Summary*" has been added.
Measurement types	In the chapter, information on the different types of measurements has been brought together: • Temperature measurement 454 • Bridge measurements with wire strain gauges 457 • LVDT 464 • Incremental counters channels 465

Section	Bug-fix
List of sensor supported (TEDS/imc SENSORS) 740	Title of the table was wrong: "*NOT supported sensor information (TEDS or imc SENSORS)*"

Data analysis and signal processing

Section	Amendments
Inline Analysis - imc WAVE Vibration	imc WAVE Vibration 1080 has been supplemented with the function: "*Machine diagnosis*" (in accordance with ISO 10816/20816).

Channel types, variable types, data types and data storage formats

Section	Amendments
All	Re-arrangement: the information had previously been distributed among various sections.
Monitor channels	New: Description of the Envelope-type monitor channels
Data types	New: Description of the various data types and the properties: "*Segmented*" and "*Event-based*"

Panel

Section	Amendments
Special Widgets	The following Widget is described: • DIO
Zone dialog > Other options and settings	Note added: "*Zone-texts are ignored when ...*"
Properties - Widget	The description for the option "Decimal Places" has been extended.

Sequencer, events and commands

Command Reference

Section	Amendments
Measure a user-defined characteristic curve	Description added
Information and Tips 1098	Info about breakpoints and bookmarks in sequences
Export 1167/Import 1172 parameters	Description merged with other format descriptions in the chapter "*Channel types, variable types, data types and data storage formats* 1130"

API

Section	Amendments
Initialization	Additional parameters of the constructor added.
First steps	System requirements have changed.

Index

(

( ) Opening Closing Bracket 900
(%) Modulus 898
(-) Subtraction 897
(*) Multiplication 897
(/) Division 897
(;) Semicolon 900
(^) Power 897
(+) Addition 897
(<) Less than operator 899
(<=) Less than or equal operator 899
(<>) Inequality operator 899
(=) Equal Operator 898
(=) Equal Sign 898
(>) Greater than operator 900
(>=) Greater than or equal operator 900

.

.Count 1600

?

?C
  Unit not correct 210
?m/m
  Unit not correct 210

<

<auto> (marker) 1454

1

1/3-octave, octave labeling 1267
16-bit integer 364

2

24-bit mode 364
2-point calibration 382
2-point controller 1060
2-point scaling 351

3

32 bit integer
  pv-variable 1141
3D 1267
  Color palette 1408
  Display 1315
  Extras 1412
  Isolines 1411
  options 1404
  Perspective 1410
  Rotate 1413
  surface 1405
3D-bars 1405
3PDI
  AudioDevice 723
  ChannelLoader 724
  fos4x 725
  FunctionSimulator 727
3rd Party Device Management 189, 200, 1929

4

4D
  3D + color 1398

A

A2L 557, 602
A2L import 555
A2L-file XCPoE
  creating 711
  loading 713
AAF 360
  Preprocessing 366
AAF-filter 360
ABCRating 901
Abs 901
Absolute date, time (curve window) 1358
Absolute time
  Display for synchronized start 327
Access codes 805
Access rights of imc user group 154
Accessing the data carrier 444, 446
AccuLength 902
AccuMax 903
AccuMean 903
AccuMin 904
AccuRMS 904
AccuStDev 905
Acos 905
Acquired
  Point 351
Acquisition time
  UPS 222
Acquisition time after power fail (s) 222
Action 388
  Setting bits 397
  Starting data recording 397
  Stop recording of data 397
Actions and targets
  Trigger 397
activating map mode retroactively 1390
Adapt

Adapt
Menu (curve window) 1505
toolbar (curve window) 1505
Adapt Panel page to window size 1541
Adapt Panel page to window size (keep aspect ratio) 1541
Adapt Panel page to window size (keep Widget size) 1541
Adapt Widgets to the size of the Panel page 1541
adapting maps 1388
Adaption channel selection
More Channels in Curve Window 1348
Add 746
Add device 48
Add device interface 449
Add file
Command Load/Refill variable 1762
Add user 743
Add variable
Command Load/Refill variable 1762
Adding a device (Add) 449
Addition 897
Additional Columns 251
Combined column 254
export 157
import 157
Metadata column 260
Parameter set column 262
Parameter set import column 267
save 155
Additional Documents 138
Adjusting variable 736
adjustment 9
Admin 743
Administration 743
Adopting sensor information in the sensor database 741
Advanced Configuration
Interface configuration 55
Advanced Diagnosis 1117
Advanced display
Zone 1255, 1259
AFDX-Assistant 502
AFDX-Bus
Assistant 502, 503
channel parameter 512
configuration 512
Configuration file list 509
configuration list 512
copy channel 510
cut channel 510
defining search filters 512
delete channel 510
edit channel 510
find channel 511
paste channel 510
Starting the Assistant 503
AFDX-Bus Channel
Byte position 508
create 505
data word 508
destination 507
function state 508
IP address 507
IP address mask 507
LSB value 509
MAC address 507
number format 508
number of Bits 508
Port 507
Scaling 509
settings 507
Source 507
start bit 508
start byte 508
Unit 509
value 508
virtual link 507
Y-Offset 509
AFDX-Bus Configuration file 504
create 504
load 504
save 505
AFDX-Bus Interface 502
Aggregate sampling rate 369
aliasing 360
Aliasing effects 360
Aligning Widgets 1242
All available 1218
Allow a change of direction 1528
Prevent a change 1528
Allow data export 1739
Allow discarding of all measurements 1739
Allow login of anonymous users 147
Allow selection of the export folder 1739
Alphabetical 345
alternative IP and FW-Update 52
Amount 400
Amplifiers
Address 226
Module address 226
Modules 226
Permanently installed amplifiers 226

Amplitude-Shift 1378
Analog channels 340
Analog inputs 340
Analog outputs 340, 868
AND 906
  Trigger event 397
Angle (differential, abs, sum) 471
Angle measurement
  Incremental counters 471
Anonymous users
  Logging allow 147
  Logging prohibit 147
antialiasing filter 360
Anti-aliasing filter 361
  Low pass 361
anti-aliasing filter (AAF) 360
API
  Actions 1952
  Aktionen 1952
  Basis Interface 1949
  Beispiele 1957
  Classes 1951
  Curve window 1948, 1964
  default.ispc 1949
  Demo 1964
  DLL 1956
  Einstieg 1947
  Examples 1952, 1953, 1955
  Functionality 1951
  Hints 1964
  imc Managed Class Libraries 1948
  imc DATA API 1948
  imc FAMOS 1948
  Initialisierung 1949
  Logbook 1962, 1963
  PresentationFramework 1956
  Problems 1964
  References 1956
  Version 1948
  Version festlegen 1948
  Voraussetzungen 1947
  Windows.Forms 1953
  WPF 1955
Appearance
  Property - Widget 1247
Application module 409, 636
Arangement
  Widgets 1242, 1243
ARINC-Bus
  Assistant 476
  BCD data word 483, 486
  BCD parameter 500
  BNR data word 480, 490
  BNR parameter 499
  Bus speed 497
  Channel parameters 497
  Configuration 496
  Configuration file list 494
  Configuration list 497
  copying label definition 495
  cutting label definition 495
  deleting label definition 495
  DSC data word 487
  DSC parameter 501
  editing label definition 495
  Equipment-ID 479, 489
  Label (Rx) 479
  Label (Tx) 489
  Label-number 479, 489
  Log channel 497
  Mixed configurations 489
  pasting label definition 495
  Receiver Channel (Rx) 479
  Rx 497
  Searching label definition 496
  Starting the Assistant 477
  Transmitter Channel (Tx) 489
ARINC-Bus Assistant 476
ARINC-Bus Bus speed
  All High 497
  All Low 497
  High 497
  Low 497
ARINC-Bus configuration file
  Creating 478
  idb 478
  idb2 478
  Loading 478
  Save 478
ARINC-Bus Interface 476
Arith. Mean 369
Arithmetic mean
  Preprocessing 366
ARON measuring circuit 884
Arrangement
  Axes 1360
  Configuration (curve window) 1422
Arrangement (Widgets)
  Align Bottom 1242
  Align left 1242
  Align Right 1242
  Align top 1242
  Cascade 1242

Arrangement (Widgets)
To Background 1241
To Foreground 1241
ARXML 557
As presettings
Curve Window 1423
ASCII
imc Format Converter 1193, 1716
imc Format Converter Command 1716
ASCII display (curve window) 1288
Asin 906
Aspect ratio for RGB image 1361
Assignment 365, 561, 576
Assistant EtherCAT-IF 611
At defined time 336
Atan2 906
AudioDevice 723
auto
Buffer time 428, 429
Auto filename 413
Auto-IP 55
Auto-IP (DHCP + APIPA) 52
Automatic table length 1528
Automatic timed start 336
Automation
Functions 1611
Automation elements 1569, 1628, 1648
Automation Task
Widget 1625
Autorepeat mode 206
Autosize
Property - Widget 1248
Autostart 206
Available Channels
Context menu 1344
More Channels in Curve Window 1344
Available events 370, 414, 422
Display and calculations 422
Axes
Arrangement 1360
labeling 1267
Scaling 1350
Text 1362
Axes List 1341
Symbols 1341
Axes Navigation Bar (curve window) 1516

B

Back (curve window) 1472
Background
Property - Widget 1247
Background color
Property - Pages 1539
Background image
Property - Pages 1539
Background image layout
Property - Pages 1539
Background plain colored
Property - Pages 1539
backround picture (curve window) 1385
Backups of experiments 163
Balance 189
All active channels 197
Balance action
execute via Scripting 1893
Balance at device startup 374
Balance at startup 374
Balance by Display 783
Balance values import 1175
Balancing values
balance value after change of device 377
ex-/import 377
Balancing with setted offset 353, 378
Bar meter (curve window) 1267, 1294
Barcode/QR code 749
Base Diagnosis 1110
Base path for measurement data storage 129, 435
BaseTrigger 387
BCD data word 483, 486
Beeper 868
Overflow 854
Begin action 397
Begin=0 397
Begin=1 397
End=0 397
End=1 397
start 397
stop 397
Binary 1532
BitAnd 907
Bitmask
Property - Widget 1248
BitNot 907
BitOr 907
Bits 1130
LIN 654
BitXor 908
Blink duration 1628
Blinking duration 1636
Blinking LED 1628, 1636
Blob
CAN 539

Blob
FlexRay 632
Block Comment 871
Block Editor 1586
BNR data word 480, 490
Bookmarks
imc FAMOS 1098
boolean variables
from files 877
IF conditions (OFA/IFA) 877
Border
Background 1247
boundaries (curve window) 1439
boundary intervals (curve window) 1439
Boxplot (curve window) 1329
Branching 1628, 1636, 1642, 1648
Break condition 1596
Breakpoint
Scripting 1820
Breakpoints
imc FAMOS 1098
Bridge 377
Bridge by Display 783
Bridge factor N 354
bridge measurement 457
Bridge mode 354
Bridge resistor 349, 354
Bridge supply 375
Bridge supply (On/Off) 375
Bridge unit 355
Browse in workspace
Layout 1750
Page 1750
Parameters 1750
Bubble Chart 1331
Bubble Plot
change edge 1331
mit Farbkanal 1336
Standard 1331
Buffer time 427
auto 428, 429
increase 755
Bus Decoder 530, 551, 566, 703, 1101
CAN 539
FlexRay 629, 632
MVB 666
Opening the Editor 1102
sampling rate 632
User interface 1102
Which fieldbus 1102
Button 1529
create 126

C

Cable compensation without sense 384
Calibrate IEPE sensors 202
Calibrating the amplifiers 344
Calibration 9, 746
Calibration date 344
Calibration interval 746
Call by Call Internet access
GSM 806
Land lines 805
CAN
Blob 539
Bus Decoder 539
Decode all channels 539
duplicate values 604
CAN analyzer 583
CAN bus
extended-ID and TSA evaluation 540
receive a message 540
send a message 540
Send standard message in extended node 540
CAN channel
Data formats 524
CAN channel not appearing in the imc Online FAMOS 896
CAN database import
enable 224, 225
CAN FD 518
Activate 527
CAN FD Node (ARGUS)
Terminator 564
CAN Monitor channel 536
CAN protocol
CANAlyser-MDF format 1154
CAN send with imc FAMOS 543
CAN-1 Protocol 529
CANALYSER 540, 568
CANAlyser-MDF format 1154
CAN-Botschafts-Protokoll 568
CAN-Bus 514
A2L 602
A2L import 555
Analog CAN channel 546
Analoger CAN Kanal 570
Assignment 561
Assistant 525
assistant (ARGUS) 562
Balancing a bridge module via CAN 577
Botschaft Definition 568

CAN-Bus 514
Botschaft empfangen 568
Botschaft senden 569
cabling 515
CAN protocol 520
CBA import, export 555
CCP 599
Channel - Error Handling 553
Channel - State 551
Channel - Validity 551, 552
Channel configuration 561
CURVE 602
DBC import, export 555
Diagnostic On CAN 595
DiagOnCAN 595
Digital CAN channel 550
D-SUB - pin configuration 515
ECU configuring 585
Editing a CAN configuration 526
Error handling 520
Expert settings 528
Format 520
GMLAN 601
ID 520
Initialization 538
Intel format 546
Intel-Format 570
J1939 544
KWP2000 590
KWP2000 TP2.0 595
log message 540
Log-channel 551, 562
MDF 540, 568
Message log 540
Messages 519
Motorola format 546
Motorola-Format 570
Number of CAN nodes 519
Number of samples per bit 528
OBD-2 596
Operating (ARGUS) 563
Options 560
protokolliere Botschaft 568
Quick Tutorial 526
Receiving a message 540
Sample location in bit time 528
Sampling interval 561
Sampling time 561
Send with imc Online FAMOS 875
Sending messages 542
Sensor initialization 538
Shunt calibration 579
Slope detection 528
Specifications 519
Terminator 516
Time stamp 520, 561
Transfer rate 517
UDS 602
VAL_BLK 602
Validity 529
XCP 601
Y-adaptor 515
CAN-Bus (ARGUS)
Assignment 576
Channel settings 575
Sampling interval 576
Sampling time 576
Time stamp 576
CAN-Bus (ARGUS) - Channel
Channel - Error handling 574
Intel 570
Motorola 570
number format 570
replacement value 574
Scaling 570
Start byte 570
Unit 570
Validity 573
CAN-Bus (ARGUS) - Message
extended-ID and TSA evaluation 568
receive a message 568
send a message 568
Send standard message in extended node 568
CAN-Bus (ARGUS) - Node
Activate (ARGUS) 563
Baud rate 563
Clear Node settings 567
Connection 563
Export Node settings 567
Extended format 563
Extended+ format 563
format 563
Import Node settings 567
Initialization 564
Log channel 566
Node context menu 567
node definition 563
Sensor initialization 564
Standard format 563
Validity 566
CAN-Bus (Menu)
Edit 555
Extras 560
File 555

CAN-Bus (Menu)
Insert 556
Transform messages 555
transform receiving- in send-messages 555
XCP 555
XCPplus 555
CAN-Bus Definition
Baud rate 527
CAN FD 527
Connection 527
Format 527
Nodes 527
Terminator 528
CAN-Bus Gültigkeit
Acknowledge 566
CAN-Bus Synchronization
CAN-1 Protocol 529
CAN-Bus Validity
Acknowledge 530
Deactivate device if no measured value x s 534
Designation 536
Diagnostic Message (DM) 531
Errorframes 534
J1939 531
Monitor channel 536
Support of Wake On CAN 534
Wake on CAN 534
CAN-Bus-Assistant 562
import formats 557
CAN-Bus-Data (curve window) 1288
Cancel Button
Show message box 1754
Text (Button caption) 1754
Visible (Hide) 1754
Visible (Show) 1754
CAN-database 555, 560
CAN-FD (ISO 11898-1:2015) 514
CAN-message
Sensor delay time 545
CanMsg 1055, 1056
CanMsg.GetData 1057
CanMsg.SetData 1058
CanMsg.Transmit1_S1 1058
CANopen object folder EtherCAT-IF- 607
CANopen over EtherCAT-IF 605
Capabilities FlexRay 620
Capabilities MVB-Bus 658
Caption in tables 1529
Cascade Widgets 1243
CBA 557
CBA import, export 555
CCP 522, 599
ccv file
loading 1479
saving 1481
CE Certification 9
Cell title 1529
Centered
Background image 1539
Certain parts from data (curve window) 1415
Certificates 9
CF card (Compact Flash) 447
CFast Storage medium 447
Change requests 9
changing the scale 1496
Channel 1575
CAN-Bus 556
time stamped 1926
Channel action
execute via Scripting 1893
Channel balance 377
Channel comment 344
Channel count 342
Channel data type 364
Channel definition (Setup page) 344
Channel file name 413
Channel info
More Channels in Curve Window 1349
Channel name 344, 871
Channel name assistant 345
Channel name for the log 646
Channel names
Fieldbus 475
Channel number 375
Channel numbering 226
Channel parameters
get/set via Scripting 1832, 1833, 1893
Channel playback 724
Channel properties
get/set via Scripting 1893
Channel status 344
Channel table 340
Channel wiring 350, 718
ChannelLoader 724
Channels 340
add to SimplePollDevice 1943
add to Third Party Device 1924
Parameters (Scripting) 1832, 1833
Properties (Scripting) 1832, 1833
Saving (Scripting) 1832, 1833
Channels with invalid properties 560

Charact 909
Characteristic curve 729
creating 736
Characteristic curves
export 157
Charakteristic (filter) 360
Check 746
Check - Syntax 861
Check Box 1529
Choose Tool Windows 125
CiA Draft Standard 514
Circular buffer memory 755
Circular buffer operation 426
Circular buffer time 370, 414, 423
Display and calculations 423
Class Learning Time 1111
Class-counting 891
ClHistogram 910
Clipboard 1489
curves 1467
clipboard format (curve window) 1500
Clipboard settings 1489
Font 1489
Line style 1489
Line width 1489
plot 1489
Print 1489
printout 1489
Ticks 1489
ClLevelCrossing 911
ClMarkov 912
Clock 1522
Diskstart 206
Clock types
DCF77 298
GPS 298
IRIG-B 298
NTP 298
PTP 298
SyncRTC 298
VRTC 298
Close folder on command 441
CloseSaveInterval 441, 913
ClRainFlow 914
ClRainFlowRes 915
ClRainFlowTM 916
ClRainFlowTMRes 917
ClRangePairCount 918
ClRevolutionsHistogram 918
ClRevolutionsMatrix 919
ClTrueMax 919
ClTwoChannelHistogram 919
Cluster FlexRay
Properties 625
Cluster measuring FlexRay 635
Cluster size 1156
CodeRange 920
CoE 605
coldjunction compensation 456
Coldstart FlexRay 635
Collection Editor 251
Color 344, 373
Title 1246
Color components 1493
Color grandient
Background color 1539
color map (curve window) 1267, 1398
Display 1303
general 1306
ISO-lines 1312
measure mode 1434
Waveforms with associated x-,y-,z-variables 1314
color of Y-axis 1364
Color palette
3D 1408
color palette (curve window) 1398
Color spectrum 1303
color-coding thermocouples 455
Coloring
Property - Widget 1247
Colors (curve window)
Copy 1493
Curves 1493
Screen/Printer 1493
Column captions 269, 1529
Column description
save 155
Column descriptions
export 157, 273
import 157, 273
Column ID
CurrentValueDisplay 376
eAllocation 365, 561, 576
eBalanceAtDeviceStart 374
eBridgeEModule 355
eBridgeEps 355
eBridgeFactor 354
eBridgeMode 354
eBridgeN 354
eBridgeResistor 349, 354
eBridgeSupplyOnOffAction 375
eBridgeUnit 355

Column ID
eChannelComment 344
eChannelMode 349, 356, 364, 717
eChannelName 344
eCorrection 350
eCoupling 349, 354, 717
eCouplingIsDC 349
eCurveColor 344, 373
eCurveYAxisMax 373
eCurveYAxisMin 373
eCurveYAxisOption 373
eDataType 364
eDuration 363
eEnumeratedChannelNumber 375
eHistogramSaveInterval 372
eHistogramUpdateInterval 372
eHysteresis 357
eInputRange 357
eIsolatedThermoCouple 351
eLevel 357
eMaximum 357
eNullImpuls 356
ePlugInName 374
ePolarization 351
eProcessing 368
eProcessingPoints 368
eRange 350, 351, 355, 718
eSampleTime 363
eSignal 356
eSignalform 357
eStartEdge 357
eStatus 344
eStopEdge 357
eSupply 350
eUserOffset 352, 357
eUserScalingFactor 352
eUserScalingFactorENC 356
eUserUnit 352, 357
eWiring 350, 718
eXFormatVariable 365, 561, 576
PreprocessedSampleRate 368
PreprocessedSampleTime 368
SampleCount 363
SampleRate 363
UserScalingFactor_Reciproce 352
Column title 1529
Column value 345
Columns
Showing and moving 248
Combination mode 467
Combinations of events 397
Combined column 254
Combined event 389
Combined Parameters 254, 264
Combo Box 1529
Command
Automation 1604
Delete supplemental file 1750
Delete Variable 1764
Execute device action 1741
Execute file 1713
Execute imc FAMOS sequence 1708
Export Panel page 1735
Export supplemental file 1748
Export Variable 1758
If 1722
imc FAMOS project 1707
imc FAMOS sequence 1708
Import imc Inline FAMOS source code 1728
Import supplemental file 1749
Load variable 1760
Measure a user-defined characteristic curve 1740
Open Logbook viewer 1755
Panel page - Export 1735
Panel page as dialog 1732
Panel page entfernen 1736
Refill variable 1760
Set measurement number 1729
Set variables 1764
Show Panel page as dialog 1732
Stop Sequencer 1724
Supplemental file - Delete 1750
Supplemental file - Export 1748
Supplemental file - Import 1749
Command line parameters 35
Command name 1870
Command Reference 1706
Commands
execute via Scripting 1870
Run script 1712
Tool window 1683
Comment 344, 736
Property 881
Title - Default Source 1246
Comment 1, 2, 3 746
Comment a line 871
Comment several lines 871
Communicator
create scroll link 1473
Transfer to imc FAMOS 1482
Compact Flash 447
comparator conditioning
Incremental counter 469

Comparison operators 856, 874
Compatible measurement modules & systems 746
Compatible modules and systems 750
Compiler
  1607 1916
  3270 1916
  CS0029 1916
  CS0234 1916
  CS0525 1916
  CS1607 1916
  CS1729 1916
  Error CS0029 1916
  Error CS0234 1916
  Error CS0525 1916
  Error CS1729 1916
  MSB3270 1916
  Warning CS1607 1916
  Warning MSB3270 1916
Complete layout 274
Complete layouts 230
  export 157
  import 157
Components 33
  Product configuration 29
COM-Port 1915, 1919
Computation uncertainty in summations 894
Conditional AND
  Trigger event 397
Configuration
  e-mail 791
Configuration (curve window)
  Arrangement 1422
  loading 1479
  Menu 1488
  saving 1481
Configuration EtherCAT-IF 607, 615
Configuration file 478
Configuration transfer
  Curve window 1515
Configuration with mapping EtherCAT-IF 614
Configuration without mapping EtherCAT-IF 612
Configure via FTP - Syntax 814
Connect 189, 190
  running measurement 192
Connect (curve window) 1472, 1473
  cross-section with 3D 1474, 1475
connect device 46
Connect XY with 2nd curve window 1473
Connected sensors 230, 729, 734
Connecting via LAN 46, 55
Connecting via WLAN 59
Connection 291
  to Device 33
Connection failed - transfer configuration 129
Connection status 291
  get/set via Scripting 1891
  Variables 1139
Connection via a direct address 58
Connector 374
Consider defined message length 537
Constraints FlexRay 620
Constraints MVB-Bus 658
Context menu
  Available Channels 1344
  Curve window 1515
Context Script
  Activity Scope 1883
  create 1812, 1813
  Parameter changed (Event) 1883
  Status changed (Event) 1883
  Value changed (Event) 1883
Control 168, 1642
Control commands 855, 856, 872
  activate 855
  Array 873
  Comparison operators 874
  Create a process vector variable 873
  Create a varialbe 873
  Data array 873
  deactivate 855
  Single value 873
  SingleValueChannel 873
  Trigger 405
Control functions 856
Control of synchronization 331
Control sync state - PC 332
Control units 584
Control units (ECU) 522
Control units in imc Online FAMOS 589
Controller functions 1060
Conversion 112
  Database 19
Coordinate system size 1267
Coordinates (map) 1397
Copy
  Logbook 144
Copy & Paste
  Curve window 1515
Copy data to the PC 444, 448
Copy with details
  Logbook 144
Correction 350

Cos 920
Counter 465, 1636, 1723
Counter hysteresis 357
Counter input range 357
Counter scaling factor 356
Counter signal 356
Counter signal shape 357
Counter start edge 357
Counter stop edge 357
Counter switching level 357
Counting loop 1599
Coupling 349, 354, 717
Coupling is DC 349
Create a new characteristic curve 736
Create a process vector variable 873
CreateVChannel 920
CreateVChannelInt 921
Creating the columns with the Collection Editor 251
CRFX
  Module address 226
CRFX Modules 343
Critical channel fill level 755
Cross sections
  Lines 1380
CrossCorrelation 921
Cross-section with 3D
  connect 1474, 1475
CRXT
  Module address 226
CRXT Modules 343
CtPID 1061
CtPID.Calc 1062
CtTwoPos 1063
CtTwoPos.Calc 1064
cumulative measurements 468
Current device configuration
  Interface configuration 55
Current measurement 1550
Current PC configuration
  Interface configuration 55
Current value 376
Currently not reachable
  Interface configuration 55
CurrentValue 922
Cursor
  Measure window 1431
CURVE 602
Curve color 344, 373
Curve configuration
  loading 1479
  saving 1481
Curve properties (Setup page) 372
Curve window 1960
  Add channel 1847
  Axes 1847
  changing the scale 1496
  clipboard 1467
  clipboard format 1500
  construction 1339
  context menu 1515
  Date/time absolute 1267
  Days/hours/minutes relative 1267
  Display of synchronized start 327
  Double-click on empty areas 1500
  Enable horizontal measurement cursor 1500
  Fill mode 1414
  Font 1496
  Global Handling 1500
  Graphic export 1500
  graphic export as Vector graphic 1496
  hiding data 1339
  history 1496
  import a picture or map as backround 1387
  Introduction 1265
  legend 1382
  Line-Shift 1496
  link 1496
  Max. delay [ms] 1496
  Measure 1429
  Measure window 1429
  More Channels 1339
  Navigation in x-direction 1500
  New coordinate systems 1341
  New x-axis 1341
  New y-axis 1341
  Oscilloscope mode 1414
  PDF orientation 1500
  Rezoom 1500
  Scripting 1847
  scroll links to tables 1473
  Scroll mode 1414
  selection mode 1496
  settings 1496
  Size of measurement point 1500
  sound output 1496
  Start delay [ms] 1496
  Stretch mode 1414
  Time-Shift in the ccv 1496
  touchscreen operation 1368
  Transfer to imc FAMOS 1500
  Undo 1496
  Updating 1554

Curve window 1960
XY plots 1346
y-Axis 1847
Zoom 1500
Custom base path for measurement data storage 129, 435
Custom folder naming for measurement data storage 129, 435
Customer support
Tech support 8
Customize Ribbon Menu 125, 159
Cut sound
sound output (curve window) 1510
Cut-off frequency 1 (filter) 360
Cut-off frequency 2 (filter) 360
Cycle time
FlexRay 632
Cyclical calculations 1596

D

DAC 868
Scaling 868
DAC does not output every value 896
Data format
imc2 1190
imc3 1190
Data formats
CAN-Bus 524
Data loss 423
Data overflow 417, 427, 755
Data rate 756
reduce 755
Data recording with sampling 752
Data recording with time stamp 752
Data saving 411
Data Saving Assistant 1738
Allow data export 1739
Allow discarding of all measurements 1739
Allow selection of the export folder 1739
Default button 1740
Default path 1739
Delete original files 1739
Deleting a measurement 1739
Discard measurement 1739
Export measurement settings 1739
imc Format Converter 1739
Measurement comments 1738
Parameter set 1738
Prompt to confirm discard 1739
Setup pages 1738
Data security mechanism for experiments 163
Data storage 433
Resume 198, 440
Suspend 198, 440
Data stream 411
Data transfer 413, 444, 448, 1929
FTP access 449
LIN-Bus 639
Password 444
Storage medium 444, 448
Data transfer (Setup page) 370
Data type 364
16-bit integer 364
24-bit mode 364
Float (24-bit mode) 364
Data types 1130
Memory requirement 1156
Data Browser 1214
All available 1218
current measurement in the Data Browser 1216
Event time 1218
Load variable 1222
Measurement number 1220
measurements in the Data Browser 1216
Refill variable 1222
renaming measurements 1216
saved measurements 1216
Data Browser columns
Category 1215
Comment 1215
Event time 1215
Metadata columns (optional) 1215
Name 1215
Numbering column (#) 1215
Unit 1215
Database
conversion 112
Database conversion 19
Database folder 112
DATABASE/DATABASES 169
Date (format)
imc Messaging 800
UDP Status Monitoring 800
UDPNoise 800
Date and time 1522
Daylight saving time 297
Days/hours/minutes relative (curve window) 1358
dB 923
DBC 557
J1939 544
DBC import, export 555
DC (Coupling is DC) 349

DCB 560
DCF
Synchronization 312, 313
DCF77
External clocks 300
D-component 887
D-Controller
imc Online FAMOS 886
Deactivating the pv-variables 1141
Debug
Breakpoint 1820
Scripting 1820
Decimal 1532
Decimal places
Property - Widget 1248
Zone 1263
Default 1053
Default button
Data Saving Assistant 1740
Default configuration
FTP 796
Default path
for export 1739
Default printer (curve window) 1484
Default Source
Title 1246
Default Value 261, 1580
Default values 214
Applied condition 214
Default value 214
Element type 214
Enumerator class 214
Group name 214
Parameter 214
Parameter identifier 214
Row scope 214
Definitions FlexRay 618
Definitions MVB-Bus 657
Definitions XCP
CCP 709
XCP 709
XCPoE 709
Delay 736
DelayBuffer 923
Fill 925
Next 926
SetSize 927
DelayBuffer.Fill 925
DelayBuffer.Next 926
DelayBuffer.SetSize 927
DelayLine 928
Delete
experiment 115
project 115
Set variables 1764
Supplemental file 1750
views 156
Delete original files 1739
Delete sample 1528
Delete supplemental file 1750
Delete Variable 1764
Show dialog 1764
Treat error as warning 1764
Variable name 1764
Variables list 1764
Deleting a measurement
Data Saving Assistant 1739
Deleting sensor information from channel 733
Deleting the process vector backup file 1144
Delta-connection 885
Delta-x 881
Delta-z 881
Description 289
Design Mode 1234
Designation data folders 433
Design-Mode
activate via Scripting 1870
deactivate via Scripting 1870
Destination 801, 802
DestinationIP 801, 802
DEV001.MSG 790
Deviation of system time from NTP 1141
Device
add 48
connect 46
connection 33
Properties 221
Device actions
Events: Sequencer 1702
execute via Scripting 1870, 1891
Device configuration 221
Device connection via LAN 55
Device control 291
Starting measurement 197, 291
Device description 1921
Device group 187
Device information
Interface configuration 55
Device Interfaces 53, 55
Device is known 241
Device is selected for measurement 290
Device name 291

Device overview 187
Device parameters
get/set via Scripting 1891
Device properties 221
get/set via Scripting 1891
Device selection 241
Device SN 291
Device specification 291
Device system variables 1138
Device Table 290
Device_AfterCheckConfiguration 1704
Device_AfterRequestConnect 1704
Device_AfterRequestDisconnect 1704
Device_BeforeCheckConfiguration 1704
Device_BeforeCreateDiskStart 1704
Device_BeforeDownload 1704
Device_BeforeReconnect 1704
Device_BeforeRequestConnect 1704
Device_BeforeRequestDisconnect 1704
Device_BeforeRequestStart 1704
Device_BeforeRequestStop 1704
Device_Compile 1704
Device_Configured 1704
Device_Connected 1704
Device_Disconnected 1704
Device_Link 1704
Device_PreCompile1 1704
Device_PreCompile2 1704
Device_Reconnected 1704
Device_Started 1704
Device_Stopped 1704
Devices
add to Third Party Devices 1923
Parameter (Scripting) 1824, 1825
Parameters (Scripting) 1891
Properties (Scripting) 1824, 1825, 1891
Devices - Documentation 70
Devices deselected 129
Devices_AfterCheckConfiguration 1704
Devices_AfterRequestConnect 1704
Devices_AfterRequestDisconnect 1704
Devices_BeforeCheckConfiguration 1704
Devices_BeforeCreateDiskStart 1704
Devices_BeforeDownload 1704
Devices_BeforeReconnect 1704
Devices_BeforeRequestConnect 1704
Devices_BeforeRequestDisconnect 1704
Devices_BeforeRequestStart 1704
Devices_BeforeRequestStop 1704
Devices_Compile 1704
Devices_Configured 1704
Devices_Connected 1704
Devices_Disconnected 1704
Devices_Link 1704
Devices_PreCompile1 1704
Devices_PreCompile2 1704
Devices_Reconnected 1704
Devices_Started 1704
Devices_Stopped 1704
Device-specific ESI
EtherCAT-IF 616
DFilt 928
DHCP 50, 55
DHCP ClientID 49, 52
DHCP Hostname 52
Diagnostic Message (DM) 531
Diagnostic On CAN 595
DiagOnCAN 595
Dialog response 1874
Diff 929
differential measurement procedures 468
Digital bits
Trigger target 397
Digital channels 340
Digital events
Signal = 0 393
Signal = 1 393
Signal change: 0 -> 1 393
Signal change: 1 -> 0 393
Digital inputs / outputs (bits and ports) 341
Digital output
Trigger target 397
Digitization 746
Dimension Lines 1448
DIN-EN-ISO-9001 9
DIO
Widget 1532
DirClosed 417, 442
Direction 1528
Directory structure 433
Trigger 434
Disable if this column is empty 254
Discard measurement
Data Saving Assistant 1739
Disconnect 189, 190
running measurement 192
Disconnect from running measurement 129
DiskFreeSpace 929
DiskFreeTime 930
DiskRunDir 931

Diskstart 206
creating 208
execute via Scripting 1824, 1828
selection via Display 773
Display 769
3D 1315
backlight 772
Balance 783
bar meter 776, 1294
Browse 781
color map 1303
Combobox 776
Connector 222
Curve window 779
Diskstart selection 773
Display Demo 772
Display editor 770
display mode 772
Display variables 770
exit 772
FAQ 786
Firmware update 771, 786
Function keys 780
GPS connector 222
Input control 775
Input/ Output control 775
Keyboard 772
Line chart 779
Needle meters 778
Number (curve window) 1283
Output control 775
Overview 770
page timer 772
Push button 782
Recognize the Display type 770
Refresh rate 770
Set values 782
Shunt calibration 785
Static text 774
System menu 772
Text selection 776
Time 774
Toggle button 782
Type of display 222
waterfall diagram 1277
Display (curve window) 1267
Display before trigger events 405
Display editor 770
Add page 771
Delete page 771
Export configuration 771
Import configuration 771
Display editor menu
Device 771
Edit 771
Extras 771
File 771
Object 771
Display for synchronized start
Absolute time 327
Curve window 327
Display format 254
Display format parameter 254
display options
third-octave axes 1321
Display Setup dialog 1745
Display text
Zone 1530
Display type
recognize 770
display uncertainty 1377
Display update interval
Histogram 372
Display variables 341, 766
LIN 654
Display, calculations of measured data 370
Display-GPS connector 222
DisplaySetButton 931
DisplaySetPage 931
Distance (differential, abs, sum) 471
Distance measurement
Incremental counters 471
Division 897
DM-services 531
DNS-server-address 49
Dock
Property - Widget 1247
Dock (Widgets)
Below 1244
Fill 1244
Free 1244
Left 1244
Right 1244
Top 1244
Documentation 70
Documentation (Setup page) 289
Documentation of devices 70
Domain 49
DotNET (.NET) 1721
DoWhile 1599
drag to WORD/PPT 1469
Drag&Drop
curve window 1515

Drive Element 1125
Drive elements
Belt Gearing 1118
Clutch 1118
Drive Shaft 1118
Gear Drive 1118
Planetary Gear 1118
Rolling Bearing 1118
Shaft 1118
Steady Frequency 1118
Steady Order 1118
Three-Phase Motor 1118
DSC data word 487
DST 297
DSUB-9
FlexRay 621
LIN 637
dual-track encoder 470
duplicate values
CAN 604
Duration 363, 1575
Duration of events 394
Dwell time 388
for Trigger events 394
max 394

E

ECAT-IF 605
ECU
CAN-Bus 556
Clear Diag Information 933
Cmd Return 933
Properties in common 603
Read Trouble Codes 932
Start Session 932
Stop Session 932
ECU (A2L) 555, 557
ECU configuration 584
ECU configuring 585
ECU Control units 522
ECU Monitor channel 536
ECU requirements 522
ECUClearDiagInformation 933
ECU-protocol
enable 224, 225
ECU-protocols 522
ECUReadTroubleCodes 932
ECUStartSession 932
ECUStopSession 932
edge (incremental counter) 472
Edit 746
Menu (curve window) 1487
signals (curve window) 1435
Edit box color
Property - Widget 1248
Edit EtherCAT-IF 615
Edit in Designer 276
Edit menu 124
Editor 743, 1529
Elastic modulus 463
Else 1050
e-mail 808
Configuration 791
Message Types 788
Embedded 254
Enable
ECU-Protokolle 224, 225
imc Online FAMOS 224, 225
imc Online FAMOS Professional 224, 225
imc STUDIO Monitor 224, 225
Online-Klassierung 224, 225
Online-Ordnungsanalyse 224, 225
Enable horizontal measurement cursor 1500
Encoder (Setup page) 356
Encoder w/o zero impulse 356
End online mode (curve window)
Scroll mode 1414
Entering range 392
Entry format 261
Enumerated channel number 375
Enumeration classes 246
Envelope
Preprocessing 366
Envelope type monitor channel 1132
Envelope type monitor channels 1136
Environment variables of the OS 185
Equal 934
Equal Operator 898
Equal Sign 898
eRange 1928
Error (Logbook) 142
Error handler 1591
Error handling 1596
Error Message 103 286
Error/Exception 1575
ESI
EtherCAT-IF 616
ESI device-specific
EtherCAT-IF 616
ESI EtherCAT-IF 616
ESI save
EtherCAT-IF 616

EtherCAT Slave 605
EtherCAT-IF 605
CANopen object folder 607
Configuration with mapping 614
Configuration without mapping 612
Device-specific ESI 616
ESI 616
ESI device-specific 616
ESI save 616
Inputs 615, 616
Mapping 615
Outputs 616, 617
PDO 616
PDO add 618
PDO Create automatically 615
PDO delete 615
Prerequisites 606
RxPDO 616
TxPDO 616
Xml 616
EtherCAT-IF Assistant 611
configuration 612
EtherCAT-IF configuration 607
EtherCAT-IF general 605
EtherCAT-IF Menu Configuration 615
Accept changes 615
Export 615
Import 615
New 615
EtherCAT-IF Menu Edit 615
Add PDO 615
Create PDOs automatically 615
Delete PDOs 615
Inputs 615
Mapping 615
Outputs 615
EtherCAT-IF Menu ESI 616
Save device-specific ESI 616
Save general ESI 616
EtherCAT-IF Menu Show 616
Inputs 616
Inputs/Outputs 616
Outputs 616
PDO 616
RxPDO 616
TxPDO 616
Xml 616
Ethernet bits 341, 765, 1137
Ethernet for Control Automation Technology 605
Evaluating the last dialog's answer
If 1727
Switch 1727
While Loop 1727
Event 388, 400
Event (curve window) 1415
Event history (imc REMOTE) 842
Event Script
create 1812, 1813
Measurement storage area 1889
Event time
All available 1218
Data Browser 1218
Event type
Digital 393
Level 392
Range 392
Event-based data set 422, 1131
Events 370, 422
Available events 414, 422
Open Experiment/Start measurement 1731
Property - Widget 1248
Trigger 389
Tutorial 1780
Events counting 466
Incremental counters 471
Events from digital signals 393
Events from virtual bits 393
Events: Sequencer 1694
device actions 1702
EventTyp 1575
EXCEL
imc Format Converter 1193, 1716
imc Format Converter Command 1716
Excel-Display (curve window) 1288
Exception handler 1591
Exception handlers 1648
Exception handling 1596
Exception/Error events 1596
Exchange display variables via network 338, 766
Exchange real-time value 767
Execute device action 1741
Starting measurement 197
Execute file 1713
Execute imc FAMOS sequence 1708
Execute menu action
Command 1756
Widget 1526
Execute program 1713
Exit fullscreen mode 1546
Exiting range 392
Experiment 110, 115
copy 116
delete 115

Experiment 110, 115
export 115
import 115
new 114, 116, 121
open 121
save (as) 116, 121
Experiment file
Open Experiment/Start measurement 1731
Experiment from template 116
Experiment has changed 129
Experiment parameters
get/set via Scripting 1899
Experiment template 112, 115, 119
export 115
import 115
new 114
preferred 120
Experiment, open
Automatically upon starting imc STUDIO 35
With a specific imc STUDIO installation 35
EXPERIMENT/EXPERIMENTS 170
Experiment/Project
Difference 162
What is saved where? 162
Experiment_Loaded 1704
Experiment_Saved 1704
Expert mode 1170
ExpoRMS 935
Export
Column descriptions 273
experiment 115, 117
experiment template 115
Interface-configuration 53, 57
Logbook 144
Measure window 1431
project 115, 117
Scripting 1816
Table description 273
Export all channels
Mapping 1181
Export all channels and settings
Mapping 1181
Export current data 439
Options 440
Export measurement settings 1739
Export Panel page 1735
Export parameters 1167
Export supplemental file 1748
Export Variable 1222, 1758
Add variables 1758
Always overwrite existing files 1759
File format 1759
File options 1758
Filename 1759
Folder 1758
Mapping instruction 1758
Measurement 1758
Measurement name 1759
Save all in a single file 1758
Save to one file per file format 1758
Show file options 1759
Show variables options 1759
Treat error as warning 1759
Variable name 1759
Variables 1758
Variables list 1759
Export views, meta data columns, sensors, ... 157
ExpressCard Storage medium 447
Extended Format
CAN 527
Extended ID (CAN)
TSA evaluation 540
Extended+ Format
CAN 527
Extension for Windows-Explorer 448
Extra menu
Administration 127
Login 127
Logout 127
Metadata Assistant 127
Options 127
Start (Sequencer) 127
Startup behavior 127
Stop (Sequencer) 127
User Access Rights 127
Extras (curve window)
3D 1412
Lines 1377
Menu 1503
Menu and Toolbar 1504

F

Factor 202, 352
Property - Widget 1248
Factor-offset scaling 351
Factory calibration 377
FAMOS-ZIP format
Limitations 1153
FAQ
Compiler error 1916
imc Messaging 804
Scripting 1916
FAQ synchronize devices 327

Farbkarte
Wirkung (Extras) 1378
Fast loading
Property - Pages 1540
FAT16 1156
FAT32 1156
Fatal (Logbook) 142
FFT 936, 1925
FFTAmplitudePhase 937
FFTAverage 938
FFTInverse 939
FFTRealImaginary 939
FIBEX FlexRay 633, 634
FIBEX-file FlexRay 635
FIBEX-Plus FlexRay 633
fieldbus
Analog inputs 340, 365
Channel names 475
data recording 752
Digital inputs / outputs 365
Digital inputs / outputs (bits and ports) 341
General notes 474
Operating methodology 474
File
Menu (curve window) 1478
File access
Scripting 1877
File comment 1759
File format 414, 1152, 1153
CANAlyser-MDF 1154
Header 1155
idb 478
idb2 478
imc FAMOS 1153
imc FAMOS ZIP 1153
Key 1155
File header 1155
File name for channel 413
File system 756
File System FAT16/FAT32 450
FileNotFoundException 1942
Filesize (maximum) 450
Fill mode
Curve window 1414
FiltBP 940
FiltBS 940
Filter
instable state 940
Setup table 236
Filter concept 361
Filter digital
Limits 895
Filter types
AAF 360
band pass 360
high pass 360
low pass 360
without 360
filtering (curve window) 1437
Filtering (Setup page) 360
FiltHP 941
FiltLP 941
Firewall 50
Security alert 33
Firmware update 64
blocking 66
Display 771, 786
Logbook 66
password 66
Firmware version 63
Fix Axes 1472
Fixed-point (curve window) 1283
Flags 35
FlexRay 618
Blob 632
Bus Decoder 629, 632
capabilities 620
Cluster 625
Cluster measuring 635
Cluster with missing Coldstart-node 635
Coldstart 635
constraints 620
cycle time 632
Decode all channels 632
Definitions 618
DSUB-9 621
FIBEX 633, 634
FIBEX-file 635
Fibex-Plus 633
Frame Trigger 627
Frames 626
Frame-Triggerings 627
Import of FIBEX-Plus files 633
KeySlotID 634
Loading FIBEX-files 633, 634
Menu 623
Multiplexer 631
Node mode 634
Overview window 622
PDU 633
Pin configuration 621
Properties of Clusters 625
Properties of Frames 626

FlexRay 618
Properties of Frame-Triggerings 627
Properties of Signals 628
Properties Window 622
Rules for the KeySlotID and node mode 634
sampling rate 632
Settings 623
Signals 628
UpdateBits 627, 628
FlexRay-Assistant
Introduction 621
Float
pv-variable 1141
Float (24-bit mode) 364
Floating point 1580
floating-point notation 1283
Floor 941
Flush right (curve window) 1283
Folder naming for measurement data storage 129, 435
Font
Property - Widget 1247
Title 1246
Font (curve window) 1496
Last value as number 1283
For 1052
For changed message clock rate 560
For-loop
.Count 1600
Count of elements in a variable 1600
Variable for counter 1600
Format item 345
format marker (curve window) 1454
Format of numerical variables
imc Messaging 798
UDP Status Monitoring 798
UDPNoise 798
Formatkonverter
Export formats 1713
Formatting 445, 452, 756
Formatting (x-axis absolute)
Format line 1, 2 1351
Forward (curve window) 1472
fos4x 725
Balance 726
Balancing 726
Boundary 726
Characteristic curve 726
Compensation 726
Connetor 726
License 725
lower Wavelenght 726
nSens 726
Reference temperature 726
Reference wavelength 726
Sensitivity 726
Sensor 726
Signal quality 726
Tare set point 726
Time 726
Unit 726
upper Wavelength 726
Wavelength 726
Frame Trigger FlexRay
Properties 627
Frames (MVB) 667
Frames FlexRay
Properties 626
Frame-Triggerings FlexRay
Properties 627
Frequency 474
frequency bands
third-octave display 1321
Frequency error
Synchronization 322
From default settings 119
From existing experiment 119
From present settings 119
From Variable
Range 1248
FTP
Configure 809
Configure via FTP - Syntax 814
Default configuration 796
FTP access
Data transfer 449
full bridge: 4 active strain gauges 462
full bridge: general 461
full bridge: Poisson full bridge (strain gauges adjacent branches) 462
full bridge: Poisson full bridge (strain gauges opposed branches) 461
full bridge: WSG 461
Full functionality for 30-day demo 23
full-bridge
LVDT 464
Fullscreen 1544
Automatically upon starting imc STUDIO 35
Exit fullscreen mode 1546
Launch 1544
Panel 1206, 1210
Function 368
Function keys display 780

Function Reference
imc Inline FAMOS 896
imc Online FAMOS 896
Functions 1628, 1648
Automation 1611
Functions Assistant
Automation 1611
FunctionSimulator 727, 1933
FW-Update and alternative IP 52

G

Gauge factor 354
GearRatio 942
General event 1604
General terms and conditions 9
Generate default values from selection 214
Generate invitation link 743
GetDateTime 942
GetDuration 944
GetHistoValue 944
GetHistoValue2 945
GetLastError 946
GetSampleCount 949
GetSamplingTime 949
GetSoftwareVersion 819
Global Handling (curve window) 1500
Global RAM 895
GMLAN 601
GMW 3110 Version 1.5 601
Go to 1591
GPS
connector 222
External clocks 300
process vector variables 1146
RS232 settings 1148
Synchronization 314
GPS + DCF Master/Slave
Synchronization 315
GPS data as XY-plot 1393
GPS-Channels 341
GPS-Display connector 222
Graphic export
Bitmap 1496
Vectorgraphic 1496
Graphic export (curve window)
Bitmap 1500
Vektorgraphic 1500
Graphical switch
Widget 1524
Greater 949
Greater than operator 900
Greater than or equal operator 900
GreaterEqual 949
Greek characters (curve window) 1512
Grid
Curve Window 1404
Grid (curve window) 1267
small x-ticks (-1 = auto) 1267
small y-ticks (-1 = auto) 1267
Grid (Panel)
Realign to Grid 1543
Snap to Grid 1543
Grid size
Property - Pages 1540
Group
Device overview 187
Variables 345
Group (Widgets)
Create Group 1245
Dissolve Group 1245
Enter Group 1245
Leave Group 1245
Group By Box
Setup table 235
Grouping
Setup table 235
GSD files 684
GSD folder 684
Guarantee 9
Guardian 165
GuardianService 165
GuardianWDService 165

H

half bridge: 1 active and 1 passive starin gauge 460
half bridge: 2 sctive strain gauges 460
half bridge: general 459
half bridge: Poisson 459
half bridge: strain gauge 459
half-bridge
LVDT 464
Hand-Terminal
Typ 222
Hard drive 415, 443
harmonic cursor 1462
Header (file header) 1155
Headline
Show message box 1754
Heavy burdening of the processor 1554
Height
Property - Pages 1539
Help and Documentation 70

Help menu
Additional documents 138
imc website 138
Product configuration 138
Hexadecimal 1532
Hide disabled columns from caption 254
Hide Passive Channels 189
hiding data
curve window 1339
HighLowRatio 950
high-pass (curve window) 1437
Histogram / Rainflow (Setup page) 372
Histogram display update interval 372
Histogram storage interval 372
history (curve window) 1496
Homepage 32
Hookup and Circuiting
LIN-Bus 638
Hostname (DHCP) 52
Hotline
Tech support 8
Hot-Plug
FTP access 449
Storage medium 443, 446
Hyst 950
Hysteresis 357, 388
for trigger events 395

I

I-component 887
I-Controller
imc Online FAMOS 886
idb 478
idb2 478
Identifier 253
Identity 147
iDiv 951
IEEE FLoat
pv-variable 1141
If 1050
Command 1722
Condition 1722
Else 1722
IF conditions
boolean variables 875
If-instruction 1603
Image 746
Widget 1524
imc CRONOSflex
Module address 226
imc CRONOSflex Modules 343
imc CRONOS-XT
Module address 226
imc CRONOS-XT Modules 343
imc Data format
imc2 1190
imc3 1190
imc ECAT-Slave-IF 605
imc Help and Documentation 70
imc ID 746
imc Language Selector 40
imc REMOTE
Event history 842
imc REMOTE WebServer 822
Activation 824
Adapting the design 831
Bar meter 839
Certificate 841
Close a session 827
Current values 830
Curve diagram 840
Designer 828
display language 841
Experiment change 829
Files 833
Grid 834
Language Selection 827
LED 838
Monitor 828
Numerical input/output 840
OEM identifier 841
Operator 828
Panel 834
Pointer instrument 838
Press key 837
Shapes 837
Standard - Text area 836
Starting measurement 830
Stopping measurement 830
System requirements 823
Toggle switch 837
User Role 828
imc REMOTE WebServer settings
default settings 832
im-/export 832
save as 832
imc REMOTE WebServer ume.zip 829
imc SIMPLEX Tag I, II, III 746
imc Software License Agreement 10
imc Systems 448, 449
formatting 452
imc CANSAS 210

imc FAMOS
Always overwrite existing files 1088
Bookmarks 1098
Breakpoints 1098
congestion 1097
Current measurement 1091
Debug Mode 1088
From imc FAMOS 1091
Last concluded measurement 1091
many data cutting interaction 1097
Measurement 1091
Measurement number 1091
Open 1088
Options 1091
parallel data cutting interaction 1097
parallel processing of sequences 1097
Synchronous event 1088
Timeout 1088
To imc FAMOS 1091
Transfer table 1091
imc FAMOS automation (data cutting) 1607
imc FAMOS project 1707
imc FAMOS sequence 1708
imc Format Converter
ASCII 1193, 1716
Command line parameters 1200
Data Saving Assistant 1204, 1739
EXCEL 1193, 1716
Export formats 1191, 1713
Installation 1192
Save in one file 1193, 1716
Sequencer Command 1202, 1714
Settings 1193, 1716
Stand alone (Operation) 1198
Storage location 1193, 1716
Windows-Explorer 1201
imc Format Converter Command
ASCII 1716
EXCEL 1716
Save in one file 1716
Sequencer Command 1714
Settings 1716
Storage location 1716
imc Inline FAMOS
Import source code 1728
Ribbon 1100
System requirements 850
imc LICENSE Manager 29
imc Messaging 786
Basics 786
Configuration 790
Configuration file 790
Error sources 804
FAQ 804
Format date/time 800
Format of numerical variables 798
Recognizing events 788
Syntax 798
System requirements 788
Technical Specifications 805
Templates 805
UDP Status Monitoring 801
UDPNoise 801
Variables 796
imc Messaging Configuration
e-mail 791
imc Messaging Message Types
e-mail 788
imc Online FAMOS
Control units 589
Digital Inputs 868
enable 224, 225
FAQ 896
Filter digital - Limits 895
Global RAM 895
Local channels 867
Local single value variables 867
Local variable 867
Single value variables 867
System requirements 849
imc Online FAMOS and imc Inline FAMOS 851
imc Online FAMOS Professional 856
Controller 886
D-Controller 886
enable 224, 225
I-Controller 886
P-Controller 886
PI-Controller 886
PID-Controller 886
imc Online FAMOS program is not saved 896
imc SENSORS
Adopting sensor information in the sensor database 741
Importing sensor data 741
Reset channel's sensor information 733
supported sensors 740
imc SIMPLEX 742
imc STUDIO Automation 1557
imc STUDIO DataProcessing 1098
Ribbon 1100
imc STUDIO Monitor
enable 224, 225
imc WAVE Noise 1068
imc WAVE Rotation 1074

imc WAVE Structure 1084
imc WAVE Vibration 1078
imc2
Data format 1190
imc3
Data format 1190
imcSyslog 417
Immediate
Trigger event 397
immediate trigger 387
Immediately 336
Import
Column descriptions 273
experiment 115, 118
experiment template 115
project 115, 118
Scripting 1816
Source code 1728
Table description 273
Import / Export
menu 121
import a picture or map as backround (curve window) 1387
Import imc Inline FAMOS source code 1728
Import mapping by name
Mapping 1181
Import of channels according to connector
Mapping 1181
Import of channels according to connector and device serial number
Mapping 1181
Import Panel page 1736
Import parameters 1172
Import supplemental file
Mapping table 1749
Import views, meta data columns, ... 157
Importing sensor data 741
ImportRoot 269
In experiment
Metadata 261
Increment
Property - Widget 1248
Incremental counter
combined measurement 474
comparator conditioning 469
edge 472
max. number of pulses per rev. 471
maximum input range 468
sampling rate 465
scaling 468
start edge 472
stop edge 472
time measurement 472
Incremental counter (Setup page) 356
Incremental counter does not go to zero 474
Incremental counter inputs 341
index-channel 470
Individualized colors for this window 1493
Inequality operator 899
Information (Logbook) 142
Init 1590
Init / Terminate 1590, 1628, 1636, 1642, 1648
Initialization
CAN-Bus 538
CAN-Bus (ARGUS) 564
Sensor 538, 564
Initializing a sensor 581
InitTimer 1617
InitTimout 1617
Inline Analysis 1064
Inline Function 1065
Inline FAMOS
private Task 1868
Scripting 1868
Input range 350, 351, 355, 357, 718, 759
Inputs
EtherCAT-IF 615, 616
Insert sample after 1528
Insert sample before 1528
Insert Widget 1235
Inside of range 392
instable filter functions 940
Installation 17
imc DEVICEcore 25
imc DEVICES 25
imc FAMOS 27
imc SENSORS 26
imc Shared Components 25
imc STUDIO 26
PROFIBUS Configurator 684
Projects 42
Security oftware 21
Silent 43
Step by step 21
Unattended 43
Update 17
User account control 20
Views 42
Virus scanner 21
Your own personal programs 42
Installation type
Full functionality for 30-day demo 23

Installation type
Typical functionality incl. imc STUDIO Professional 23
User-defined 23, 25
Integer 1580
Integral 951
Integral2 951
IntegralFFT 951
IntegralP 952
IntegralP2 952
Interface configuration
Device Interfaces 53, 55
Interface-Configuration 53, 57
Export current settings 53, 57
Internal storage media 443
Internet Settings for maps 1397
Interval
Count 423
Property - Widget 1251
Saving 423
Interval (Timeout)
Show message box 1754
Interval funktions 952
IntervalFrom1Level 954
IntervalFromLevels 955
IntervalFromPulse 956
IntervalMax 956
IntervalMean 957
IntervalMin 957
IntervalMult 958
IntervalResample 959
IntervalRMS 959
IP address 49
configure 46
of the devices 46
of the PCs 46
IRIG Format 1522
IRIG-B
External clocks 299
Synchronization 312, 313
ISO / DIS 11898 514
ISO 11519-2 514
ISO-9001 9
isolated synchronization 334
Isolated thermo couple 351
Isolines
3D 1411
Issues
Storage medium 445, 454
IsSyncronized 960

J

J1939 544
CAN 531
DBC 544
Jiggling display 41, 1554
JKFlipFlop 960

K

K-factor 463
KiRoad 695
Kurvenfenster
freie Texte 1402
Text 1402
KWP2000 522, 590
KWP2000 TP2.0 522, 595

L

Label for raw data 1367
Labeling
Lines 1375
LAN 49
Device connection 55
Language change 40
Language installing 41
Last calibration date 746
Last measurement 441
Last value as number (curve window) 1267
Display 1283
Font 1283
Launching in Fullscreen mode 1544
Layout 125
Browse in workspace 1750
Layout Repository 230, 275
Layout-Designer 276
Layout-Repository
export 157
import 157
LED 868
LED6 197, 893
Overflow 854
LED-flashing during measurement 893
legend
curve window 1382
legends (line) 1376
LEQ 961
Less 961
Less than operator 899
Less than or equal operator 899
LessEqual 961
Level - Trigger-Event type

Level - Trigger-Event type
Negative slope 392
Positive slope 392
Signal < Level 392
Signal > Level 392
Level Indicator 1518
Liability restrictions 9
Licensable components 29
License 29, 743, 1960
Limit monitoring 1648
Limit value monitoring
Channel 1575
Disable 1575
Duration 1575
Error/Exception 1575
Event Type 1575
Lower Limit 1575
Rank 1575
Upper Limit 1575
Limitations
Storage medium 445, 454
Limited Warranty 9
LIN
Bits 654
Display variables 654
Prozess vector variables 654
sending variables 654
LIN Sleep 655
LIN WakeUp 655
LIN-Assistant 641
Limitations 642
Properties 641
LIN-Bus
Assistant 641
Baud rate 646
Data transfer 639
DSUB-9 637
Frame (message) properties 648
Frames 645
Hookup and Circuiting 638
Logging 640
Master 640
Master-Breaks (Duration) 646
Master-Breaks Delimiters (Duration) 646
Monitoring 640
Overview window 643
Pin configuration 637
Polling cycle time 646
Properties window 643, 646
Signal 645
Slave 640
Sleep-WakeUp 646
Testing the configuration 645
Wakeup-Breaks (Duration) 646
LIN-Bus Interface 637
line color
depend on magnitude 1398
Lines (curve window) 1369
Color 1370
Cross sections 1380
Extras 1377
Labeling 1375
Line structure 1370
Line thickness 1370
Line type 1370
Lines 1370
Printer / Screen: 1370
Symbol 1370
Symbol size 1370
Line-Shift 1378, 1472, 1476
in the ccv 1496
Reset 1476
Link
Metadata 261
link (curve window) 1496
Link + in experiment
Metadata 261
Link with several curve windows 1474, 1475
Link XY with 2nd curve window 1473
linking a map with time data 1392
Lissajous figures
curve window 1328
List 345
literature (CAN-Bus) 514
Little-Endian 680
Ln 961
Load
views 156
Load CAN configuration CBA 555
Load dialog (curve window) 1479
Load measurement settings 1227
Load variable 1222
Add file 1762
Add variable 1762
Category 1225, 1762
Command 1760
Data Browser 1222
Load all 1763
Show dialog 1763
Target measurement 1225, 1763
Target variable name 1224, 1762
Treat error as warning 1225, 1763

Load variable 1222
Variable name 1224, 1762
Variables list 1224, 1762
Local channels 867
Local single value variables 867
Local system information
Variables 1138
Local variable 867
Localtime 297
Lock page 1543
log channel
CAN Log channel 530
with/without sended channels 529, 538
Log off
Logging a user off 151
Log on
anonymous user 151
registered user 151
LogAnd 962
Logbook 142
Access via Scripting 1864, 1866
Autoscroll 144
Category 142
Copy 144
Copy with details 144
Deleting 144
Duplicates 144
Entries 1864
Export 144, 1864
Filtering 143
Firmware update 66
Location 142
Logbook-Viewer 144
OnError 1864
OnWarning 1864
Options for the Logbook 144
Search 143
Zip file 144
Logbook (curve window) 1288
Logbook category
Error 142
Fatal 142
Information 142
Warning 142
Logbook viewer
Opening by command 1755
logging on
imc SIMPLEX 745
LogNot 962
LogOr 962
LogXor 962
Long date 1522
Long name
Title - Default Source 1246
Long time 1522
Long-term measurement 755
Loop 1628
For Loop 1723
While Loop 1723
Loop count
Persistent 1600
Variable 1600
Loops 1599
Loops: Counting loop 1599
Loops: DoWhile 1599
Loops: While 1599
Lower 962
Lower Limit 1575
low-pass (curve window) 1437
LVDT 464

M

MAC-address 61
Magnitude (curve window) 1283
Mail servers
imc Messaging 807
Main window 33
maintenance 9
Making a device known 241
Manage Projects 114, 121
Managed DLL 1721
Mandatory box 261, 264, 269
Map 1387
cache 1397
curve window 1391
from Internet 1393, 1394
link 1392
Map provider, Adding- 1395
mode 1388
provider 1394
settings 1397
streched 1396
map coordinates adjust 1387
Map provider, Adding-
Map in curve window 1395
Mapping 1181
EtherCAT-IF 615
Export all channels 1181
Export all channels and settings 1181
Import mapping by name 1181
Import of channels according to connector 1181

Mapping 1181
Import of channels according to connector and device serial number 1181
Multiple-device operation (Firmware group A) 1181
Single-device operation (Firmware group A) 1181
Mapping table
Export supplemental file 1748
Import supplemental file 1749
Marker
Voreinstellungen 1443
Marker (curve window) 1443, 1454
Delete all markers 1460
extra functions 1460
Markers at Min/Max 1460
Markers on all lines 1460
move 1452
new line 1446
Select al lines 1460
set 1445
marker precision 1454
Master-Send-List
Aktivate 705
Deaktivate 705
Matrix (curve window) 1420
Max 373, 963
max dwell time 394
max. delay (sound) 1511
Max. delay [ms] (curve window) 1496
Maximaum stack
OFA/IFA 892
Maximum 357
Preprocessing 366
Range 1248
Maximum amount of variables in imc Inline FAMOS 850
Maximum amount of variables in imc Online FAMOS 849
maximum input range
INC-channels 468
MDF 568
MDF format 1154
Mean 369, 964
Measure (curve window)
1/dx (frequency) 1429
dx 1429
Slope 1429
Slope per decade 1429
xl 1429
xr 1429
xr/xl 1429
yl 1429
yr 1429
yr/yl 1429
yr-yl 1429
Measure a user-defined characteristic curve 1740
Measure context menu
Channels List 1431
Clipboard 1431
Expand List 1431
Export Curve Segment 1431
Place marker with left click 1431
Send Curve Segment... 1431
Measure mode
Curve 1427
Measure window (curve window) 1427, 1429
color maps 1434
context menu 1431, 1435
cursor positioning 1431
Date/ time display 1429
Days/ hours/ minutes display 1429
Measured level 202
Measured value 761
Measurement 291, 746
loading 1219
Start 197
start/stop via Scripting 1824, 1825, 1870
Stop 197
unloading 1219
Measurement comments
about Data Saving Assistant 1738
Measurement cursor (curve window) 1427
Measurement data 110
Memory requirements 1155
Measurement folder structure 435
Measurement mode 349, 356, 364, 717
Encoder 358
temperature 454
Measurement mode (Setup page) 349
Measurement modes for incremental counter inputs 466
Measurement number 1550
Assigning 1220
Locking 1220
Measurement options
Setup page 338
Measurement procedure for incremental counter channels 358
Measurement process states 1043
Measurement running 129
Measurement running - reconnect not possible 129
Measurement status 291
get via Scripting 1891
Variables 1139
Measurement storage area 435

Measurement variable 736
MEASUREMENT.SQL 173
MeasurementFolder_Closed 1704
MeasurementFolder_Deleted 1704
MeasurementFolder_New 1704
MeasurementFolder_Updated 1704
Median3 964
Median5 964
Memory cards 187
Memory requirements 1155
Memory volumes 447
Menu
  Edit (curve window) 1487
  Extras (curve window) 1503
  File (curve window) 1478
  FlexRay 623
  Options (curve window) 1489
Menu (curve window)
  Adapt 1505
  Configuration 1488
  Toolbar 1505
Menu actions
  execute via Scripting 1870
Menu and Toolbar (curve window) 1504
Mercator-projection 1396
Message
  CAN-Bus 556
  multiple with different length 537
  Synchronization 546
Message Types
  e-mail 788
Messunsicherheit 1380
Meta 113
Metadata 115
Metadata column 260
Metadata readout 244
Metadata save in the channel 244
Metadata template 275
Metadata type
  Date 260
  Directory 260
  Document 260
  Logical value 260
  Multiline text 260
  Picture 260
  Single-line text 260
  Text from list 260
  Time 260
Metadata-column
  export 157
  import 157
Metafile
  transferring curve windows 1467
microSD Storage medium 444
Min 373, 965
Minimum
  Preprocessing 366
  Range 1248
Minimum accuracy [ms] 135
Minimum length of
  Password 147
  User name 147
Min-Max
  Preprocessing 366
Modbus Assistant 672
  connections 673
  Mapping 676
  menu 672
  Receive-dialog 674
Modbus Interface 671
  system requirement 671
Mode 349, 354, 356, 364, 717
Modify values (curve window) 1472
Modubs
  Preparation 671
Module 343
  Address 226
  exchange 226
  Properties 226
  removing 226
Module address
  Assign manually 226
  Reassign 226
Module numbering 226, 343
Modulus 898
Modulus of elasticity E 355
Monitor 1544
  Analog inputs 340
  Digital inputs / outputs (bits and ports) 341
  Incremental counter inputs 341
  PROFIBUS 690
Monitor channel
  anti-aliasing filter 360
  CAN 536
  ECU 536
Monitor channels 340, 405, 1135
Monoflop 965
MonoflopRT 966
More Channels in Curve Window 1339
  Adaption channel selection 1348
  Available Channels 1344
  Channel info 1349

More Channels in Curve Window 1339
Waveform Preselection 1345
Waveforms in axes list 1346
More than one sequence
Tutorial 1780
Most recently use values 261
mouse wheel
curve window 1515
move area in curve window 1515
Move markers
curve window 1452
Moving and showing columns 248
ms-files 161
msg 790
Multi-Event
Curve window 1554
Multifunction Vehicle Bus (MVB) 658
Multiline comment 871
Multi-lingual text input 130, 1555
Multilingual title
Title 1246
Multimaster Synchronization 321
Multimeter (curve window) 1283
Multiple triggering 400
Memory requirements 1157
Multiple-device operation (Firmware group A)
Mapping 1181
Multiplexer
FlexRay 631
Multiplication 897
MVB
Bus Decoder 666
Frames 667
Protocol- and signal-channel name 665
Signals 668
MVB-Bus
Activation 660
capabilities 658
Connection 660
constraints 658
Definitions 657
General description 658
Overview window 662
Password 657
Prerequisites 657
Properties Window 662
Starting the assistant 661
Topology 660
MVB-Bus Interface 657

N

Name 291, 344, 736
Property 881
Property - Pages 1539
Property - Widget 1246
Show message box 1754
Title - Default Source 1246
User-defined variables 1580
Names (curve window) 1283
Navigation bar (curve window)
Axes 1516
Navigation in x-direction (curve window) 1500
Navigation Pane 33, 139
Necessary system component
Installation 25
Negative slope 392
Nesting depth
OFA/IFA 892
Net bits 341
Network
Firewall 33
Optimize 62
Network drive 415, 419
Network drive storage settings 419
network name (device hostname) 52
New
experiment 116, 121
experiment from template 116
experiment template 114
project 114, 121
New device configuration
Interface configuration 55
New experiment 114, 116
from template 116
Next 1589, 1590
Next-Branching 1628, 1636, 1642
Next-skips 1590
NFC/RFID 749
NMEA 1146
NMEA Talker IDs
GA, GB, GI, GL 1148
GN, GP, GQ 1148
No branching 1589
No new devices found 129, 200
Nodes
CAN-Bus 556
Noise 1068
Trigger 395
noise (curve window) 1437
Normal data set 1130

NorthCorrection 967
NOT 968
NTP
External clocks 300
Synchronization 316
Number (curve window)
Display 1283
Number of folders (max.)
Storage medium 757
Number of symbols (0 = every sample) 1267
Number of trigger releases 400
Number trimming (useful when scrolling) 1267
NumberOfPulses 968
Numeric 345
Numeric Zone
Zone 1262
Numerical zones 1530
Nyquist frequency 360

O

OBD-2 596
Assistant 597
PIDs 597
user defined PID 597
Octal 1532
octave labeling
display options 1321
OFA
Scripting 1829
OFA/IFA
Apply 861
Check - Syntax 861
Class-counting 891
Comparison 845
Context Menu 865
Data flow schematic 845
maximaum stack 892
Menu 863
Nesting depth 892
pv-variable 1141
Schematic 852
Starting 858
Supplemental Files 891
Syntax check 861
Time Base for Channels 871
Offline cache (map) 1397
Offset 352, 377, 1528
Offset (curve window) 1437
OK Button
Show message box 1754
Text (Button caption) 1754
Visible 1754
Visible (Hide) 1754
Visible (Show) 1754
OLE 1469
OnAboveLevel 1619
OnAfterStartup 1704
OnAlways 1045
OnBeforeQuit 1704
OnBelowLevel 1619
OnCanMessageReceive1_S1 1060
OnECUCmdReturn 933
OnECUCmdReturn_ECU_001 969
OnError 1704
OnInitAll 875, 1044
OnInsideRange 1619
Online classification
enable 224, 225
Online FAMOS
Scripting 1829
Online-Order tracking
enable 224, 225
Online-Trigger 405
OnMeasureEnd 1046
OnMeasureStart 1045
OnOutsideRange 1619
OnPowerOff 1049
OnRampEnd 1613
OnStable 1619
OnSyncTask 1047
OnTimeout 1617, 1628, 1642, 1648
OnTimer 1049, 1617
OnTriggerEnd 405, 875, 1048
OnTriggerMeasure 405, 1048
OnTriggerStart 405, 875, 1047
Open 114
experiment 121
imc FAMOS 1088
Open Experiment/Start measurement
Events 1731
Experiment file 1731
Overwrite settings 1731
Start measurement 1731
Warning in case of infinite measurement duration 1731
Open for different device 281
Open source software licenses 39
Opening Closing Bracket 900
Opening the Bus Decoder Editor 1102
Operable
Property - Widget 1247
Operation of the Display editor 770

Optional components 29
Options 128
3D 1404
colors 1493
Dialog response 1874
Export current Data 440
imc SIMPLEX 745
Menu (curve window) 1489
Presettings (curve window) 1496
Save current Data 440
OR 969
Trigger event 397
Orientation
Property - Widget 1247
Oscilloscope mode
Curve window 1414
OtrAngleAdd 970
OtrEncoderPulsesToRpm 970
OtrFrequLine 973
OtrFrequLine2 974
OtrFrequLine3 974
OtrOrderSpectrum 975
OtrOrderSpectrumP 977
OtrPulseDuration 979
OtrResample 980
OtrResampleAngle 982
OtrResampleFromRpm 983
OtrRpmComplexOrder 984
OtrRpmOrder 987
OtrRpmPresentation 989
OtrRpmPresentVector 990
OtrRpmSpectrum 991
OtrSynthSin 991
OtrTrackingLowPass 992
Outputs
EtherCAT-IF 615, 616, 617
Outputting a waveform to a DAC module 580
Outside of range 392
Overflow
Beeper 854
imc Inline FAMOS 854
imc Online FAMOS 854
LED6 854
overload 1087
overmodulation 1087
Overview Drive Elements 1125
Overview window 1485
Overwrite autostart configurations 129
Overwrite existing file 129
Overwrite settings
Open Experiment/Start measurement 1731
Overwrite this multilingual text? 1555

P

P3 time out 590
Page 176
Browse in workspace 1750
Property - Pages 1539
Repository 1230
Page setup 1536
Page size
Property - Pages 1539
Page through Panel 1610
PAGE.NUMBER
Formatting 182
Panel 176
Access via Scripting 1840, 1841, 1843, 1845
Add Windows.Forms 1840, 1843
Repository 1230
Tables 1840, 1845
Panel Fullscreen 1206, 1210
Panel page
Adapt Page Size 1541
Deleting by command 1736
Dialog 1732
Export 1735
Repository 1230
Zoom 1544
Panel page as dialog 1732
Panel Script 1879
create 1812
Data Browser 1879
New Widgets 1879
Variables 1879
Widgets 1879
PANEL.PAGECOUNT
Formatting 182
Paper size 1536
Parallel elements 1587
Parameter set 1159, 1177
Mapping 1181
Parameter set column 262
Parameter set columns
export 157
import 157
Parameter set import column 267
Parameterizing bridges 759
Parameters
Browse in workspace 1750
Channels (Scripting) 1893
Devices (Scripting) 1891
Experiment (Scripting) 1899

Parameters
Showing and moving 248
partition 445, 452
Passive
Trigger event 397
Password 147, 151
Data transfer 444
Minimum length 147
MVB-Bus 657
Password locking
Firmware update 66
PauseSignal 1613
PC time 1522
P-component 887
P-Controller
imc Online FAMOS 886
PDF orientation (curve window) 1500
PDO
EtherCAT-IF 616
PDO add
EtherCAT-IF 615, 618
PDO Create automatically
EtherCAT-IF 615
PDO delete
EtherCAT-IF 615
PDU FlexRay 633
Peakhold (curve window) 1283
Peaks 754
Peaks (curve window) 1435
Percentage
Zone 1260
Period comparison (curve window) 1267, 1415, 1420
Periodic (curve window) 1420
Periodic calculations 1636
Persistent 1580
Perspective
3D 1410
Phase offset correction
Synchronization 322
Physical unit 736
PI-Controller
imc Online FAMOS 886
Pictogram
Property - Widget 1247
PIDcontroller 1060
PID-Controller
imc Online FAMOS 886
Pin configuration
CAN 515
FlexRay 621
LIN 637
Place (map) 1397
Placeholder
CONTROL 168
DATABASE/DATABASES 169
EXPERIMENT/EXPERIMENTS 170
PAGE 176
PANEL 176
PROJECT/PROJECTS 176
SELCONTROL 178
SETUP 179
SYSTEM 180
VARS 181
Placeholder Formatting
PAGE.NUMBER 182
PANEL.PAGECOUNT 182
SYSTEM.DATE 183
SYSTEM.TIME 183
VALUE 184
Placeholders 166
Placeholders for folder naming
SETUP.SQL 436
STORAGE.FOLDERNAME 436
STORAGE.MEASUREMENT 436
VARS.VALUE 436
Places left of point (curve window) 1283
Places right of point (curve window) 1283
Platzhalter
MEASUREMENT 173
Plug-in
Info 39
Version information 39
Points 368
Poisson full bridge 461
Poisson half bridge 459
Poisson's ratio 463
polar plot 1315
Polarization 351
Poll 993
Polygon needle
Property - Widget 1247
Ports 50
Position
Property - Widget 1247
Title 1246
Positive slope 392
post checked loop 1599
Potential difference with synchronized devices 334
Power 897
Power fail 222
Power measurements 884
Power1 884, 993

Power2 884, 994
Power3 885, 995
Powertrain Monitoring 1106
PPT 1469
Preferred experiment template 120
Prepare 189, 195
Preprocessing 366
Prerequisites
EtherCAT-IF 606
Prerequisites MVB-Bus 657
Prerequisites XCP 709
Presettings
Curve Window 1423
Presettings (curve window) 1267
options 1496
Pretigger 388
Pretrigger 402
Preview in Windows-Explorer 1520
Print 1536
Curve Window 1482
Print Panel page 1735
Print preview
Widget 1549
Printer
Setup (curve window) 1484
Printer preview 1503
printer settings 1489
Problems
Storage medium 453
Procedure 1628, 1636, 1642, 1648
Procedures 1590
Process configuration 189, 190
Process vector
backup file 1144
pv-variable 1141
Process vector backup file 1144
Process vector variable 1141
Create 873
process vector variables 341
GPS 1146
Process vector variables for data acquisition channels 338
Processing function 368
Processing points 368
Product
Configuration 29
Edition 29
Info 39
Licensing 29
Version information 39
Product configuration 29
Changing 29
Product improvement 9
PROFIBUS
Baudrate 690
Highest Address 690
Loading the configuration 691
Monitor 690
Name 690
Transferring configuration 691
PROFIBUS Assistant 692
Channel settings 693
Connection 692
Menu 692
Message settings 693
PROFIBUS Configurator
GSD files 684
GSD folder 684
Installation 684
Main window 685
Master 686
Menu 685
Slave 686
PROFIBUS Interface 684
Profinet
Preparation 677
Profinet Interface 676
PROFINET-IRT Assistant 678
Export as PDF 683
Mapping 682
Menu 678
Receive Dialog 680
Send Dialog 679
Program
execute 1713
Program add
Installation 42
Progress dialog 1958
ProgressVisualizer 1958
Project 111, 115
delete 115
export 115
import 115
new 114
Project has changed 129
Project Menu
Import / Export 121
Manage Projects 121
New experiment 121
Open experiment 121
Save / Export current Data 121
Save as template 121

Project Menu
Save experiment (as) 121
Save project 121
Project/Experiment
Difference 162
What is saved where? 162
PROJECT/PROJECTS 176
Project_Loaded 1704
Project-internal imc STUDIO user 147
Prompt to confirm discard 1739
Properties
Device 221
log file 221
Properties of moldules 226
Properties of virtual channels
Comment 881
Delta-x 881
Delta-z 881
Name 881
x-offset 881
x-unit 881
y-unit 881
z-offset 881
z-unit 881
PROPS
Metadata readout 244
Protection against changing 1546
Protocol- and signal-channel name (MVB) 665
Protocol display (curve window) 1288
Protokollkanal 568
CAN Protokollkanal 566
Prozess vector variables
LIN 654
Pt100 455
PTP
External clocks 301
PTP-Master only 319
Synchronization 318
Pulse spikes in the signal 754
pulse time 473
PulseDuration 995
PulseFrequency 996
PulsePhase 996
pulses number max. 471
Push button 1642
Push-button
Switching behavior 1248
PushDeviceDriver 1932
pv.State.ExternalPower 1141
pv.State.SyncTimeDeviviation 1141
pv-variable 1141
As monitor channels 1146
Create 873
Deactivating 1141
Name 1141
Process vector 1141
restore 1144
Restoring 1144
Update rate 1146
pv-variables 341
PWM mode (INC4) 473

Q

QR code/Barcode 749
Quality Management 9
quarter bridge
120 Ohm WSG 458
WSG 458
Quick access toolbar 139

R

Radius
Scale position 1251
RAM buffer time 427
RampSlope 1613
RampTime 1613
Range 373, 1928
Property - Widget 1248
Range - Trigger-Event type
Entering range 392
Exiting range 392
Inside of range 392
Outside of range 392
Range & Scaling (Setup page) 351
RangeMax 997
RangeMin 997
Rank 1575
raw 1153
Reaction Time 1111
Read sensor information 344, 732
Reader 743
Ready for measurement
Interface configuration 55
ReadyForPowerOff 998
Real time 1636
Realign to Grid 1543
RealTimeClock 297
Receiver
GPS 1146
Reconfigure 189, 195
RecordEvent 999
RecordText 1000

Red 1001
Reduce logbook entries 133, 1706
ReduceDataRate 1001
Reduction
Preprocessing 366
Reference level 202
Reference Literature and Standards 620
Refill variable 1222
Add file 1762
Add variable 1762
Command 1760
Data Browser 1222
Import all 1763
Show dialog 1763
Target measurement 1225, 1763
Target variable name 1224, 1762
Treat error as warning 1225, 1763
Variable name 1224, 1762
Variables list 1224, 1762
Refresh rate
Property - Widget 1246
Released 397
Remote Frames (RTR) 514
Removable data carrier 415
repair 9
ReplaceFirstValues0 1002
ReplaceFirstValuesN 1003
Report
Automation 1625
Setup 285
Report Generator 1482
Report template 1549
Property - Pages 1540
Report-page 1536
Repository 1571, 1648
Page 1230
Panel 1230
Panel page 1230
Widget 1230
Reprocessing signals (curve window) 1435
ReSample 1003
Reset
Set variables 1764
Slave Pointer 1294
Reset channel's sensor information 733
Reset Slave Pointer... 1472
Reset Window Arrangement 125
Resistor 349, 354
Restart measurement automatically 333
RestartTimer 1617
restore 1004
pv-variable 1144
views 156
Restoring pv-variables 1144
Resulting sampling rate 368
Resulting sampling time 368
Resume 1591
Resume data storage 189, 198, 440
Resume-Branch 1648
ResumeSignal 1613
Rezoom (curve window) 1500
RFID Tag
Link with RFID tag 749
release 749
Removing the Tag-assignment 749
RFID/NFC 749
RFID/QR 746
RGB image
Aspect ratio 1361
Resolution 1361
RGB-image 1338
Ribbon
Customize 159
Rights
denied 154
disabled 154
enabled 154
full access 154
hidden 154
write protected 154
Ring (circular) buffer storage 414, 423
Ring buffer memory 755
RMS 882, 1004
Preprocessing 366
RoaDyn 695
all channels 697
assistant 696
CLK 696
prerequisites 696
TRG 696
Rosette1 1005
Rosette2 1006
Rotate
3D 1413
Rotation 1074
Round 1006
Router 49, 58
RPM 474
RS232 1915, 1919
RS232 settings
GPS 1148
RSFlipflop 1007

RTC 297
RunAutoBalance 1008
RunAutoShuntCalibration 1008
RVDT 464
RxPDO
EtherCAT-IF 616

S

Sample count 363
Sample delay 754
SamplesGate 1009
sampling
data recording 752
Sampling & Preprocessing (Setup page) 363
sampling rate 363
FlexRay 632
Incremental counter 465
sampling theorem 360
Sampling time 363
Save
experiment 121
project 121
views 156
Save / Export current Data 121
Save as 114
experiment 116, 121
Save as template 119
Save CAN configuration CBA 555
Save current data 439
Options 440
Save data 370
Save dialog (curve window) 1481
Save measurement data
Saving subsequent to measurement 439
Targeted saving 439
Save trigger events in individual files 434
Save/load view settings
additional columns 155
column descriptions 155
metadata columns 155
parameter set columns 155
table descriptions 155
window arrangements 155
Saving 411
Cluster size 1156
Data types 1156
to the internal drive 415
Saving interval 423, 426
Saving pv-variables 1144
Saving subsequent to measurement 439
Sawtooth 1009
Scale angle
Property - Widget 1251
Scale center
Range 1248
Scale decimal places
Property - Widget 1251
Scale position
Property - Widget 1251
Scaled
Point 351
scaling 759
Incremental counter 468
lines 1377
x-axis 1351
Scaling factor 352, 356
scaling for strain analysis 463
Scaling offset 352, 357
Scaling with offset 760
scaling: strain gauges 463
schematic
imc thermo plug 456
T4 456
Script 1812, 1813, 1878
create 1812, 1813
execute 1878
run 1878
Script Editor 1812, 1813
Language 1812, 1813
Script name
Property - Pages 1539
Script options 1906
Script Scope 1812, 1813
Scripting 1808
1607 1916
3270 1916
abstract base class 1819
AbstractScript.cs 1916
Activity Scope 1814, 1883
Advanced code completion 1906
Axes 1847
Balance 1832, 1833
base class 1819
Breakpoint 1820
Changes 1876
Channel calculation 1912
Channel number 1832, 1833
Channel parameters 1832, 1833, 1908
Channel properties 1832, 1833
Channel state 1832, 1833
Classes 1823
Clipboard 1875

Scripting 1808
Clock state 1824, 1830
Clock type 1824, 1830
Commands via Scripting 1870
COM-Port 1919
Connector 1832, 1833
Context menu 1810
Context Script 1813, 1883
Coupling 1832, 1833
CS0029 1916
CS0234 1916
CS0525 1916
CS1607 1916
CS1729 1916
Curve window 1847
Datapool 1823
Debug 1820
Device actions 1824, 1825
Device parameters 1824, 1825
Device properties 1824, 1825
Device selection 1824, 1825
DevSetup 1823
Dialog response 1874
Diskstart 1824, 1828, 1891
Duration 1832, 1833
Edit 1815
Editor 1820
Enumerated channel number 1832, 1833
Error 1864, 1916
Error CS0029 1916
Error CS0234 1916
Error CS0525 1916
Error CS1729 1916
Event Script 1813, 1889
Events 1867
Execute 1816
Export 1816
FAQ 1916
File access 1877
Filtering 1832, 1833
get/set Device properties 1824, 1825
imc FAMOS Kit 1819
imc FAMOS Math 1819
Import 1816
Import 4.0R1 1916
Inline FAMOS 1868
Input range 1832, 1833
IntelliSense 1906
Introduction 1812, 1813
Logbook 1864, 1866
Measurement storage area 1889
Menu 1810
Menu actions via Scripting 1870
Meta data 1832, 1833
Metadata 1908
Module type 1832, 1833
MSB3270 1916
NTP 1830
OFA 1829
Offset 1832, 1833
OnError 1864
Online FAMOS 1829
OnWarning 1864
Open 1815
Options 1814, 1906
Panel 1823, 1840, 1841, 1843, 1845
Panel Script 1812, 1813
Parallel-Istallation 1916
Parameter changed (Event) 1883
Parameters 1902
ParameterValues 1824, 1825, 1832, 1833
Preprocessing 1832, 1833
private Task (IFA) 1868
Provider 1819
Proxy class 1819, 1916
Range 1832, 1833
Regenerate proxies 1819
Restore 1876, 1906
RS232 1919
Run 1816
Run script 1712
Sample time 1832, 1833
Scale 1832, 1833
Scaling 1832, 1833
Script 1812, 1813, 1878
Script clipboard 1916
Script Editor 1815, 1820
Script Scope 1812, 1813
Script storage location 1906
Sensitivity 1832, 1833
Separate proxy class library 1814, 1875
SerialPort 1919
Setting values 1902
start/stop Measurement 1824, 1825
Status changed (Event) 1883
Storage 1827
Storage scope 1814
Strain gauge 1832, 1833
Subversion 1906
Supplemental files 1829
SVN 1906
Synchronization 1830
Tables 1840, 1845
TED 1832, 1833

Scripting 1808
Third Party Device Script 1918
Timed start 1830
Tool window 1810
Tutorial 1908, 1912
Type Library Script 1813, 1887
Unit 1832, 1833
Value changed (Event) 1883
Warning CS1607 1916
Warning MSB3270 1916
Widget 1840, 1843, 1845
Working copy 1906
y-Axis 1847
Scroll mode (curve window) 1414
Curve window 1414
End online mode 1414
Scroll-mode 755
Search CAN channel, message or node 555
Search for devices 189, 199
No new devices found 200
Search for devices by IP/DNS 58, 189, 200
Security alert 33
Security oftware 21, 41
Seed/Key 590
Seed-and-Key 587
Seed-Value 590
Segment (curve window) 1415
Segmented data set 1131
SELCONTROL 178
Select device with explorer 444, 448
Select mode (curve window) 1441
Selected 241, 290
Selected x-unit (curve window) 1267
Semicolon 900
Send channels with imc FAMOS 543
Send channels with process vector variables 544
Send Curve Segment to FAMOS!:context menu 1431
Sendable message
CAN-Bus 556
sended channel
not sended by log channel 529, 538
Sending an e-mail 808
sending variables
LIN 654
SendMessage 1009
Sensitivity 352
Sensor 344, 729, 746
Link with RFID tag 749
release 749
Removing the Tag-assignment 749
Write sensor information 733
Sensor Database
Importing sensor data 741
Sensor delay time (CAN message) 545
Sensor information
TEDS 732
writing to channel 732
Sensor information - reset 733
Sensor initialization 538, 564
Sensor name 746
Sensor order no. 746
Sensor range lower limit 746
Sensor range upper limit 746
Sensor serial no. 746
Sensor supplier 746
Sensor-PROM 731
Sensors 230
export 157
Reset channel's sensor information 733
Sensors (Menu)
<< (level higher) 735
>> (indentation) 735
Add to imc SENSORS 735
Create a new characteristic curve 735
Delete characteristic curve 735
Delete Filter 735
Edit characteristic curve 735
Filter list hidden 735
Find 735
Import Filter 735
Open imc SENSORS 735
Standard layout 735
Sensor-TEDS 729, 731
Sequence table 1684
Sequencer 1681
Commands via Scripting 1870
Conditional branches 1783
create scripts 1812, 1813
Else 1786
Event handling 1694
Events 1780
For 1783
If 1786
imc FAMOS 1777
More than one sequence 1780
Output box 1769
Show dialog 1772
Show message box 1769
Start 1688
start/stop via Scripting 1870
Stop 1688
Switch 1780

Sequencer 1681
Switch-Case 1786
User-defined events 1696
While 1785
Sequencer stop
about commands 1724
Sequencer: Context menu 1692
Sequencer: Tool window Commands 1683
Serial Interface 1915
SerialPort 1919
Service
Tech support 8
service and maintenance 9
service check 9
Set as default
Curve Window 1423
Set as preferred experiment template 120
Set markers (curve window) 1445
Set measurement number 1729
Set the root directory 1230
Set value
Zone 1530
Set values
Display 782
Set variables
Delete 1764
Reset 1764
SetPoint 887
Settings
FlexRay 623
Settings (curve window) 1496
Setup
Printer (curve window) 1484
Setup data in Panel 179
Setup data in Sequencer 179
SETUP.SQL 179
Shell extension 448
Short date 1522
Short time 1522
Show EtherCAT-IF 616
Show Grid 1543
Property - Pages 1540
Show in Dialog as menu action 275
Show message box
Cancel 1754
Headline 1754
Interval (Timeout) 1754
Name 1754
OK 1754
Size 1754
Text 1754
Timeout-action 1754
Voice 1754
Show most recently use values 261
Show Panel page as dialog 1732
Show placeholder 261
Show Start Trigger 189
Show Tabs 1544
Show x=0 (trigger) 1267
Showing and moving columns 248
Shunt calibration 381
Shunt calibration by Display 785
Signal 356
Signal < Level 392
Signal = 0 393
Signal = 1 393
Signal > Level 392
Signal change: 0 -> 1 393
Signal change: 1 -> 0 393
Signal name 746
Signal shape 357
Signals (MVB) 668
Signals FlexRay
Properties 628
UpdateBits 628
Silent installation 43
SimplePollDevice 1937
add channel 1943
add Display variable 1945
Extend 1943, 1945
SimplePushDevice 1940
Simultaneous start of all devices 337
Sin 1010
single value (curve window)
window 1287
Single value variables 867
Single values 1130
Single-device operation (Firmware group A)
Mapping 1181
Single-phase power measurement 884
single-track encoder 470
SingleValueChannel 1010
Size
Show message box 1754
Size automatic (curve window) 1283
Size of measurement point (curve window) 1500
skb files 590
Skin
Property - Pages 1539
SkipFirstValues 1013
Slave Pointer

Slave Pointer
Reset 1294
Sleep (LIN) 655
Slider 1529
SlopeClip 1013
Small Ticks (curve window) 1351
Smo3 1013
Smo5 1014
smoothing (curve window) 1437
SN 291
Snap to Grid 1543
Property - Pages 1540
so files 590
Software
Installation 17
Uninstall 17
Update 17
Solid
Background color 1539
Sorting
Setup table 235
sound output (curve window) 1496, 1507
cut sound 1510
directly 1511
Toolbar 1508
volume 1508
SoundPressureLevel 1014
Source code
Import 1728
space curve 1405
Special characters 871
Specific Values 1125
Specification 291
SpecThirds 1014
Spectra 1125
Speed 474
SPI Assistant 704
16-Bit Word Structure 705
Chipselect 708
error handling 708
Master-Send-List 705
Menu 704
SPI-Bus Interface 702
Hardware prerequisites 703
Software prerequisites 703
Spikes 754
Spinner 1529
Spline (curve window) 1437
Split up pretrigger data into intervals 136
Spread angle
Scale angle 1251
Sqrt 1016
SSD Storage medium 444, 447
Standard display (curve window) 1267
Standard Format
CAN 527
Standard time 297
Start 197
Measurement 197
Simultaneous 337
Start angle
Scale angle 1251
Start button 387
Start by Trigger 397
start delay (sound) 1511
Start delay [ms] (curve window) 1496
Start edge 357
start edge (incremental counter) 472
Start measurement 189
Automatically upon starting imc STUDIO 35
Open Experiment/Start measurement 1731
Start option 336
At defined time 336
Automatic timed start 336
Diskstart 206
Immediately 336
Start page 32
Start Sequencer
Automatically upon starting imc STUDIO 35
Start with shortcut 35
Start/Stop behavior 761
Start: Sequencer 1688
Starting
imc STUDIO 31
OFA/IFA 858
StartTimerPeriodic 1054
StartTimerSingle 1055
Startup behavior 150
Startup parameters 35
State 1628, 1636, 1642, 1648
State description 1596, 1628, 1636, 1642, 1648
State of the external power 1141
States: If-instruction 1603
Static text 345
Static vector 1580
Status 344, 1684
Property - Widget 1246
STD 297
StDev 1016
Step 1628, 1636, 1642, 1648
Stop 197

Stop 197
Measurement 197
Stop by Trigger 397
Stop edge 357
stop edge (incremental counter) 472
Stop measurement 189
Stop on error 1684, 1689
Stop Sequencer
Command 1724
Stop: Sequencer 1688
StopSignal 1613
StopTimer 1055, 1617
Storage 1824, 1827
Memory requirements 1155
Setup page 292
storage device
formatting 445, 452
partition 445, 452
Storage interval 423
Histogram 372
Storage interval count 423
Storage location (PC) 421
Storage location measurement data 129, 435
Storage media 443
Storage medium
Age 757
CF card 447
CFast 447
Compact Flash 447
Data transfer 444, 448
ExpressCard 447
FAT16/FAT32 450
File system 450
Filesize (maximum) 450
Hot-Plug 443, 446
Issues 445, 454
Limitations 445, 454
Manufacturer 757
Memory volumes 447
microSD 444
Number of folders (max.) 757
Possible problems 453
Problems 453
SSD 444, 447
USB 447
Storage on a network drive 419
Storage options 413
Storage type 261
strain gauge 457
Strain Gauge (Setup page) 354
strain gauge: scaling 463
strain gauges 457
Stretch
Background image 1539
Stretch mode
Curve window 1414
STri 1017
Structure 1084
StudioWDService 165
Subnet 51
Subnet mask 49
Subtraction 897
Sum 1017
Sum sampling rate 369
Sum2 1018
Summary 408
Summation
Calculation uncertainty 894
Supplemental file
Delete 1750
Export 1748
Import 1749
Supplemental files
Characteristic curves 212
Filter data 212
imc Online FAMOS source code 212
Messaging configurations 212
Scripting 1829
Synthesizer directory structures 212
Supply 350
Surface
3D 1405
Suspend data storage 189, 198, 440
Swapping the storage medium 443, 446
Switch 1051, 1529
Switching behavior 1248
Tutorial 1780
Switch - Case 1725
Switching behavior
Property - Widget 1248
Switching level 357
Switching off the device 286
Switching on the device 286
Symbols
Fixed count at line 1377
for lines in curve window 1370
SYNC 312, 313
synchron (sound) 1511
Synchronisation
get/set via Scripting 1891
synchronity re-establish - automatical 333
Synchronization 292, 294, 297, 754, 1604

Synchronization 292, 294, 297, 754, 1604
CAN-1 Protocol 529
CAN-Bus 529
Constraints 335
Control 331
Fieldbus channels 474
Frequency error 322
GPS 314
GPS + DCF Master/Slave 315
imc CANSAS 529
IRIG-B 312, 313
No external timer 312
NTP 316
Phase offset correction 322
Precision 334
PTP 318, 319
Synchronization varieties 311
Synchronization signal PTP (Master-Only) 319
Synchronization status
Variables 1139
synchronize devices FAQ 327
Synchronized recording 546
Synchronous start 297
Diskstart 206
Simultaneous 337
Synchronous Tasks 857
SyncOverload 1019
SyncRTC 297
Internal clocks 299
Syntax
UDP Status Monitoring 802
UDPNoise 802
Syntax-Check 861
Syslog 417
SYSTEM 180
System information
Variables 1138
System requirements
Hard drive 16
imc Inline FAMOS 850
imc Online FAMOS 849
imc Online FAMOS Professional 849
Operating System 16
Storage 16
Windows 16
System time 1522
SYSTEM.DATE
Formatting 183
System.IO.FileNotFoundException 1942
SYSTEM.TIME
Formatting 183
SystemClock 1522

T

Table
Direction - Prevent a change 1528
Widget 1527
Table (curve window) 1267
Table description 274
export 273
import 273
Table descriptions
export 157
import 157
save 155
Table display (curve window) 1288
Tables
Title 1529
Tables (curve window) 1288
Tag
Link with RFID tag 749
release 749
Removing the Tag-assignment 749
Tan 1020
Tare by Display 783
Target measurement 1225, 1763
Targeted saving 439
Targets
Trigger 397
Taring 377
Task 1628, 1636, 1642, 1648
Task Editor 1584
Task Management 1573
Tasks 1573
TCP/IP 55
TC-TEDS 731
Tech support 8
Technical Specifications
imc Messaging 805
TEDS 729, 731
Read sensor information 732
Reset channel's sensor information 733
supported sensors 740
Write sensor information 733
TEDS ID 746
Telephone numbers
Tech support 8
temperatur characteristic curve
How to select? 454
temperature measurement 454
temperature table 454
Temperature unit °F instead of °C 338

Terminate 1590
Terminate condition 1628, 1636, 1642, 1648
Terminator
CAN 516
Terminator in device
CAN 528
Terminator in device (ARGUS)
CAN 564
Test specimen 746
Testing the configuration
LIN-Bus 645
Text 1529
Axes 1362
Show message box 1754
Zone 1530
Text array 1130
Text as set value
Zone 1262, 1530
Text display (curve window) 1288
Text format
Property - Widget 1248
Text input for report channel 1521
Text list 261
Text Off/On
Property - Widget 1247
TextAdd 1020
TextFormatE 1021
TextFormatF 1022
TextFormatH 1023
TextFormatI 1024
Textual Zone
Zone 1262
Textual zones 1530
Text-variables 1130
The configuration is saved in XML format. However, the loading dialog is preset to "CSV". 692
The device has been shut down inaccurate in the past 195
The measurement process
Trigger 404
thermo plug 456
schematic 456
thermocouples 454
thermocouples color-coding 455
Third Party Device 1932
add Channel 1924
Data transfer 1929
FunctionSimulator 1933
Logbook 1929
SimplePollDevice 1937
SimplePushDevice 1940
Simulator 1932
virtual 1932
Third Party Device Script 1890
create Script 1918
Third Party Devices
add Device 1923
eRange 1928
Range 1928
Third Party Device Interface 1917
Third/ octave labeling (curve window) 1358
third-octave axes
display options 1321
Three-phase power measurement 885
Tick distance
Property - Widget 1251
Ticks (curve window)
small 1351
TI-float
pv-variable 1141
Tile
Background image 1539
Time Base
OFA/IFA 871
Time Base for Channels in a Formula 871
Time counter
GPS 1146
Time measurement 467, 472
Time offset 336
Time resolution between fieldbus and analog channels 474
time stamp
data recording 752
Time Stamp ASCII (curve window) 1288, 1290
time stamped 1926
Time to suspend 534
Time tracks 1587
Time zone 34, 297, 1522
Time zones 322
Timed start 297
Setup page 335
Start option 336
Synchronous start 337
Timeout-action
Show message box 1754
Timer function 1628, 1648
Timer functions 1054
Timerfunction 1642
Time-Shift 1378, 1476
in the ccv 1496
TimeStamp-ASCII (TSA) 1130
time-stamped numeric channels 1132

time-stamped text channels 1132
Title 253
Property - Pages 1539
Property - Widget 1246
Title column 1529
Title in tables 1529
Tolerance 1024
Tool window 33, 140
docking 142
freely positioning 142
hiding 141
operation 140
pinning 142
showing 141
Tool window chooser 141
Toolbar
Sound output (curve window) 1507, 1508
Toolbar (curve window) 1504
touchscreen operation
in curve window 1368
Traceability of data 134, 244
Tracks 1587
Traffic light 1642
Transfer experiment to other devices 281
Transfer Options
Transfer to imc FAMOS 1502
Transfer rate
CAN 517
Transfer to imc FAMOS 1500
from curve window 1482
Transfer Options 1502
Transfer to PC 413
Transferring curves
using clipboard 1467
TransRec 1024
Transverse strain coeff. ν 355
Treat error as warning
Load variable 1225
Refill variable 1225
Trigger 384, 388
Event-controlled digital outputs 406
Hysteresis 395
Multiple triggering 400
Multiple-triggered signals 1554
Noise suppression 395
Variables 405, 1138
Virtual channels from imc Online FAMOS 404
Trigger events 389
in individual files 434
Trigger machine 387
Trigger time 1522
Triggering behavior 761
Troubleshooting 286
TSA 1130, 1132
extended ID 540
TSAGate 1029
TSA-Kit 1290
Twin Window (curve window) 1486
Two point calibration 382
Two-phase power measurement (ARON) 884
Two-Point controller 1060
Two-position controller 890
TxPDO
EtherCAT-IF 616
Type 1580
Type (filter) 360
Type Library Script 1887
create 1812, 1813
Types of Variables 879
Typical functionality incl. imc STUDIO Professional 23

U

UAC 20
UDP Status Monitoring 801
Format date/time 800
Format of numerical variables 798
Syntax 798, 802
Variables 796
UDP Status Monitoring test 803
UDP Status Monitoring.exe 811
UDPNoise 801
Format date/time 800
Format of numerical variables 798
Syntax 798, 802
Variables 796
UDPNoise test 803
UDS
Dynamic lists 602
ume.zip (WebServer) 829
Unable to find any new devices 200
Unable to find device 50
Unattended installation 43
uncertainty (display) 1377
Unchanged
Background image 1539
Unconfirmed template variables 1571
Undo (curve window) 1435, 1496
UnEqual 1030
Uninstall 17
Unit 352, 355, 357
Property - Widget 1246
Unit not correct 210

Unit not correct 210
?C 210
?m/m 210
Unlock page 1543
Untriggered Measurements 387
Unzoom (curve window) 1426
Update
Backup View settings 19
Database 19
Restore View settings 19
Update rate
pv-variable 1146
UpdateBits
FlexRay 627
Signal 628
Updating
Curve window 1554
Upper 1030
Upper Limit 1575
UPS
Acquisition time 222
USB
supply of storage medium 447
USB storage medium 447
Use DHCP 52
User
adding 147
deleting 147
log off 151
log on 151
removeing 147
switch 151
User account control 20
User administration 146, 147
activating 147
deactivating 147
log on 151
Startup behavior 150
User access rights 154
User group 146, 147, 154
imc Administrators 146, 147
imc Advanced Users 146, 147
imc Developers 146, 147
imc Standard Users 146, 147
User name
Minimum length 147
User Ticks 1364
User type
Active Directory domain accounts or groups 147
Local computer accounts or groups 147
Project-internal imc STUDIO user 147
User-defined
Installation type 23, 25
Title - Default Source 1246
User-defined buttons 125, 126
User-defined characteristic curve 230
creating 736
User-defined device 58
User-defined events 1696
User-defined variables 1149, 1636
User-defined variables (Automation)
Default Value 1580
Name 1580
Persistent 1580
Type 1580
UTC 297
UTC (IRIG Format) 1522

V

VAL_BLK 602
VALUE
Formatting 184
Value display 254
Value representation
Property - Widget 1247
Variable
loading 1219
Property - Widget 1246
unloading 1219
Variable display format 254
Variable linkage 1550
Variables
Access via Scripting 1879
imc Messaging 796
Types 879
UDP Status Monitoring 796
UDPNoise 796
VARS 181
Vector from datapool 1580
VectorChannel 1032
VectorChannelSet 1033
VectorFromFile 1033
VectorizeAndSkip 1033
VectorizeOverlapped 1034
Vectors 1130
VectorStatic 1034
Version
Get version with GetSoftwareVersion 819
Version information 39
Version of the firmware 63
Vibration 1078
VibrationFilter 1036

View menu
Choose Tool Windows 125
Customize Ribbon Menu 125
delete view 125
load 125
Reset Window Arrangement 125
restore 125
ribbon 125
save view (as) 125
User-defined buttons 125
View settings export/import 157
Views 125, 155
delete 156
load 156
restore 156
save (as) 156
Views, metacolumns, sensors, ... 157
Virtual bits
Trigger target 397
Virtual channel from single value 1580
Virtual channels 340, 867
Properties 881
Trigger 891
trigger machine 404
Virtual channels and the trigger machine 891
virtual clock 307
virtual device 1932
virtual device clock 307
Minimum accuracy [ms] 135
Virus scanner 21, 41, 1554
VIsAnyGreater 1038
Visible 253
Property - Widget 1247
Visible in printout
Property - Widget 1246
VMax 1039
VMaxV 1039
VMean 1039
VMeanV 1039
VMin 1040
VMinV 1040
Voice
Show message box 1754
volume sound output 1508
VRedV 1040
VRMS 1040
VRTC 307
VSum 1040
VValueAtXValue 1041
VXValueOfMax 1041
VXValueOfMin 1041
VXValueWithYValue 1041

W

Wait
Command 1726
Wake on CAN 534
WakeOnLAN 803
WakeUp (LIN) 655
Warning (Logbook) 142
Warning in case of infinite measurement duration
Open Experiment/Start measurement 1731
Warranty 9
WatchDog 165
Waterfall diagram (3D) 1267
display 1277
Waterfalls 1267
WAVE_OverloadVariable 1087
Waveform Preselection
More Channels in Curve Window 1345
Waveforms in axes list
More Channels in Curve Window 1346
WebServer
Login 825
While 1053, 1599
While-Loop 1628
Widget
Access via Scripting 1840, 1843, 1845, 1879
DIO 1532
Repository 1230
Table 1527
Widget linkage 1550
Widgets 1213, 1521
Aligning 1242
Arrangement 1241, 1242
Cascade 1243
Dock 1244
Grouping 1245
insert 1235
new 1235
Realign to Grid 1543
Snap to Grid 1543
Width
Property - Pages 1539
Window arrangements
export 157
import 157
load 155
save 155
Window size
Panel 1541
Windows

Windows
Firewall 33
Security Alert 33
Shell extension 448
User account control 20
Windows-Explorer
Preview 1520
WindRoseCorr 1042
WinForms 1960
Wiring 350, 718
WLAN
Access Point 60
Accesspoint (Server) 59
Ad-Hoc 59
Integrated WLAN 60
Managed (Client) 59
WPA2-PSK 59
WORD 1469
WPF 1960
Write protected
Property - Widget 1246
Write-protected 253
Writing
sensor information 733

X

X-axis 365, 561, 576
automatic 1351
changes 1500
fixed range 1351
lin/log 1351
markings 1351
rounding 1351
scaling 1351
Scaling the Axes 1350
XCP 601
XCP on Ethernet 709
XCP on UDP/IP 709
XCP onTCP/IP 709
XCPoE
A2L-file creating 711
A2L-file loading 713
Channel settings 715
Definitions 709
ECU-settings 715
Master 712
Master assistant 712
Node settings 714
on Ethernet 709
on UDP/IP 709
onTCP/IP 709
Prerequisites 709
Slave 710
triggert channels 711
Variables 710
XCPoE Assistant
Slave 711
X-link with curve windows 1473
Xml
EtherCAT-IF 616
XmlRpcCmd 809
x-offset 881
x-unit 881
XY data
complex data 1328
different sampling rates 1328
Lissajous figures 1328
measurement cursors 1328
polar plots 1328
XY-data sets 1130, 1132

Y

Y-adaptor
CAN-Bus 515
Y-axes stacked 1267
Y-axis
color 1364
line thickness 1364
small ticks 1364
y-axis max 373
y-axis min 373
y-axis range 373
Y-connection 885
y-unit 881

Z

Z-axis 1277, 1315
zero pulse 470
Zip - Logbook 144
ZIP format
Limitations 1153
z-offset 881
Zone
Advanced display 1259
Zone dialog 1252
Zone representation
Property - Widget 1248
Zone ring
Zone 1263
Zones
Property - Widget 1248
Zoom

Zoom
Background image 1539
Zoom (curve window) 1425, 1500
Zooming Panel pages 1544
z-unit 881

Contact imc

Address

imc Test & Measurement GmbH
Voltastr. 5
13355 Berlin

Phone: (Germany): +49 30 467090-0

E-Mail: info@imc-tm.de

Internet: https://www.imc-tm.com

Tech support

If you have problems or questions, please contact our tech support:

Phone: (Germany): +49 30 467090-26

E-Mail: hotline@imc-tm.de

Internet: https://www.imc-tm.com/service-training/

imc ACADEMY - Training center

The safe handling of measurement devices requires a good knowledge of the system. At our training center, experienced specialists are here to share their knowledge.

E-Mail: schulung@imc-tm.de

Internet: https://www.imc-tm.com/service-training/imc-academy

International partners

You will find the contact person responsible for you in our overview list of imc partners:

Internet: https://www.imc-tm.com/imc-worldwide/

imc @ Social Media

https://www.facebook.com/imcTestMeasurement
https://www.youtube.com/c/imcTestMeasurementGmbH
https://twitter.com/imc_de
https://www.linkedin.com/company/imc-test-&-measurement-gmbh